Aprendizaje Máquina y Minería de Datos

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

Alejandra Ramos Porras

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 147, No. 5,** mayi 2018. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 147**, **No. 5**, May 2018. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Aprendizaje Máquina y Minería de Datos

María de Lourdes Martínez Villaseñor (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2018

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2018

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

ISSN 1870-4069

Editorial

Este volumen de la revista "Research in Computing Science" contiene 27 artículos seleccionados cuidadosamente al menos por dos miembros del comité revisor tomando en cuenta la originalidad y contribución al área de aprendizaje máquina y minería de datos.

Se pueden encontrar dentro de estos artículos análisis del comportamiento de diferentes técnicas de aprendizaje máquina aplicados a dominios y tipos de datos muy variados para desarrollar soluciones tales como:

Aplicaciones para analizar el desempeño de estudiantes:

- Clasificación de estudiantes que abandonan o terminan una carrera universitaria e identificación de factores de riesgo de fracaso. De manera similar se puede determinar el perfil del estudiante con técnicas de aprendizaje máquina. Análisis de impacto de un curso de matemáticas.

Aplicaciones para analizar diferentes problemas urbanos:

- Catalogar los delitos en zonas metropolitanas.
- Predicción de la generación de residuos sólidos urbanos en la Ciudad de México e identificación de aspectos relevantes con su generación.
- Segmentación de placas vehiculares.
- Implementación de un sistema híbrido para buscar relaciones entre la irradiación solar total y el calentamiento global.

Aplicaciones para analizar problemas agrícolas:

- Clasificación de vegetación polinífera usando imágenes multiespectrales y redes neuronales.
- Control de riego inteligente usando instrumentación y análisis de imágenes para mejorar el desarrollo cultivos de Albahaca en un micro-invernadero.
- Predicción de condiciones limitantes en cultivos por lote en bioreactor de las tasas de incremento de biomasa en la levadura.
- Clasificación de clorosis en hojas de árboles de naranja.
- Predicción de la velocidad del viento.

Aplicaciones para problemas diversos:

- Implementar redes neuronales dispositivos tales como la matriz de puertas programables en una tarjeta SoC Zynq
- Adición de características de alto nivel a imágenes de restaurantes de Yelp para mejora el desempeño de su clasificación usando redes convolucionadas de aprendizaje profundo. Así como el desarrollo de un sistema recomendador basado en datos restauranteros.
- Clasificación de géneros musicales y reconocimiento de dígitos manuscritos.
- Reconocimiento de actividades infantiles utilizando sonido ambiental.
- Detección de comunidades de redes sociales con un enfoque evolutivo.
- Clasificación de jugadores de futbol soccer de acuerdo a sus habilidades.

– Análisis de influencia de eventos expresados en Twitter en eventos financieros.

Estudios para mejorar métodos y técnicas de aprendizaje máquina.

– Aprendizaje para tratar el problema de desbalance de múltiples clases.
– Estudio del desbalance de clases en bases de datos de expresión genética usando Deep Learning.
– Estudio comparativo de entornos de trabajo para procesamiento de datos masivos y aprendizaje automático.
– Uso de funciones base adaptables en redes neuronales con wavelets para minimizar el número de neuronas para aproximar una función
– Método de compresión de electrocardiogramas basado en Transformada Wavelet Discreta.

Los procesosq de envío, revisión y selección de artículos así como la preparación de memorias del congreso fueron realizados de manera gratuita por el sistema EasyChair (www.easychair.org).

María de Lourdes Martínez Villaseñor
Editor invitado

Mayo 2018

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Análisis de funcionamiento de una red neuronal implementada sobre una tarjeta SoC Zynq

Alberto Martínez Contreras, Juan Iván Díaz Reyes,
Alam Armando Minor González, David Tinoco Varela

Universidad Nacional Autónoma de México, FESC, Departamento de Ingeniería,
Ingeniería en Telecomunicaciones Sistemas y Electrónica (ITSE),
México

{phama_contra26,elec_st,dativa19}@hotmail.com, teeniisgool@gmail.com

Resumen. Hoy en día el uso de dispositivos tales como las FPGA, ha tomado fuerza para el diseño de diferentes proyectos tecnológicos y computacionales, estos dispositivos tienen características que las hacen atractivas para la ejecución de diferentes esquemas de desarrollo. En este trabajo se presenta la implementación y análisis de una red neuronal tipo Backpropagation sobre una tarjeta de desarrollo SoC Zynq®-7000. La placa de trabajo consta de un elemento FPGA y una unidad de procesamiento, los cuales pueden emplearse de manera autónoma. Se generaron los modelos neuronales basados en el lenguaje de descripción VHDL. Cuando se ha tenido la red neuronal completamente descrita dentro de la FPGA, se han realizado experimentos de clasificación de imágenes ejecutándose en ambos dispositivos (FPGA y unidad de procesamiento), comparando y analizando la eficiencia y velocidad de ejecución en cada uno de ellos. Se utiliza la tarjeta de desarrollo PYNQ-Z1 de Digilent, ya que proporciona los recursos necesarios para cumplir los objetivos del proyecto.

Palabras clave: Redes neuronales, FPGA, tarjetas de desarrollo.

Functional Analysis of a Neural Network Implemented using a Zynq SoC Board

Abstract. Nowadays the use of devices such as FPGA, have had relevance for the design of different technological and computational projects, these devices have characteristics that make them attractive for the execution of different development schemes. In this paper, the implementation and analysis of a Backpropagation-type neural network on a Zynq®-7000 SoC development board is presented. The development board consists of an FPGA element, and a processing unit, both can be used autonomously. Neural models based on the description language VHDL were generated. When the neuronal network has been completely described within the FPGA, image classification experiments have been carried out on both devices (FPGA and processing unit), comparing and analyzing the efficiency and the execution time in each one of them. The Digilent PYNQ-Z1 development board is used, as it provides the necessary re-courses to meet the project's objectives.

Keywords: Neural network, FPGA, development boards.

1. Introducción

Las redes neuronales artificiales, o clasificadores conexionistas, son sistemas de computación masivamente paralelos que se basan en modelos simplificados del cerebro humano. Sus complejas capacidades de clasificación, combinadas con propiedades tales como generalización, tolerancia a fallas y aprendizaje, las hacen atractivas para una gama de aplicaciones en las que las computadoras convencionales encuentran dificultades de cálculo. Ejemplos de estos incluyen detección de rostros [1], reconocimiento de caracteres escritos a mano [2], clasificación de patrones [3], y tareas de control [4].

Las redes neuronales actuales son utilizadas cada vez más en diferentes áreas del conocimiento [5], lo que implica la necesidad de sistemas computacionales cada vez más potentes tanto en procesamiento como en almacenamiento. Como es de imaginarse, no es barato conseguir un equipo de cómputo con los requerimientos necesarios para ejecutar una red neuronal de gran potencia, por lo que una alternativa económica es de suma importancia para el desarrollo e investigación de este campo.

En este proyecto se buscó una alternativa para poder diseñar una red neuronal en paralelo sin equipos muy costosos, esta opción es usar un SoC que combina una unidad de procesamiento y un FPGA (*Field-Programmable Gate Array*) de alta gama que nos permite hacer diseños propios usando un lenguaje de descripción de *hardware*, se empleara un SoC Zynq®-7000 de *Xilinx*, con la tarjeta de desarrollo PYNQ-Z1. En la figura 1, podemos observar la tarjeta sobre la cual se han realizado los experimentos.

Se utilizó la FPGA de la placa para implementar una red neuronal en hardware y verificar las características que presenta este tipo de redes con respecto a una red neuronal implementada sobre una unidad de procesamiento, también embebida en la tarjeta. La red neuronal implementada, fue probada con diferentes imágenes para poder medir la velocidad de procesamiento tanto de la FPGA como del microprocesador de la placa PYNQ-Z1.

Fig. 1. Placa PYNQ-Z1 de *Digilent*, las características, especificaciones y una descripción completa de la tarjeta puede verificarse en el manual [21].

2. Estado del arte

Las redes neuronales se comenzaron a desarrollar desde el siglo pasado cuando *Warren McCulloch* y *Walter Pitts* [6] crearon un modelo computacional en 1943 para redes neuronales, basado en matemáticas y algoritmos, llamado lógica de umbral.

En 1958, el concepto del *perceptrón* fue mostrado por *Rosenblatt* [7], este fue uno de los más importantes conceptos en este campo, ya que representaba un modelo matemático de una neurona biológica que podía ser o no activada de acuerdo a los valores de sus entradas. Este modelo podía resolver problemas que tuvieran un comportamiento linealmente dependiente, sin embargo problemas como la resolución de una función de tipo *Or-Exclusive*, no puede ser resuelta por este modelo. La representación básica del perceptrón puede verse en la figura 2.

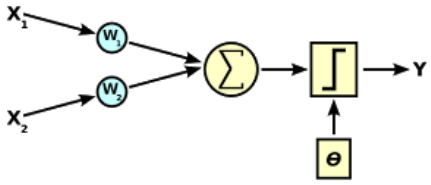


Fig. 2. Representación de un perceptrón con diferentes entradas con función de activación de tipo escalón.

Cuando los primeros conceptos relacionados a las redes neuronales se publicaron, se tuvieron dificultades con su implementación, principalmente por el hecho de que en ese momento no existían ordenadores lo suficientemente potentes en procesamiento y memoria para poder ejecutar una red neuronal compleja. Estas dificultades mantuvieron detenidos los avances relacionados a este campo de conocimiento, sin embargo, un puno clave para un renovado interés en las redes neuronales y el aprendizaje fue el algoritmo *Backpropagation* de *Werbos* [8] que podía resolver eficazmente el problema de la or exclusiva , y en términos más generales, logró acelerar el entrenamiento de redes neuronales multicapa.

Otro gran factor para el desarrollo de las *Redes Neuronales Artificiales* (RNA) fue que a mediados de la década de 1980, el procesamiento distribuido paralelo se hizo popular bajo el nombre de conexionismo. *Rumelhart* y *McClelland* describieron el uso del conexionismo para simular procesos neuronales [9].

A partir de 2011, las técnicas de redes de aprendizaje progresivo alternaron capas convolucionales y capas de máximo aprovechamiento [10, 11] encabezadas por varias capas totalmente, o escasamente conectadas, seguidas por una capa de clasificación final.

Por otro lado, se ha tenido un gran avance tecnológico en el desarrollo de microcomputadoras y distintos dispositivos embebidos, entre ellos los FPGA, que cuentan con suficiente poder para ejecutar estos algoritmos de formas distribuidas y aceleradas.

Recientemente, las RNA se han implementado con FPGA reconfigurables. Estos dispositivos combinan la facilidad de ser programados, con una mayor velocidad de operación asociada al paralelismo de hardware.

Se han generado varias herramientas de diseño de FPGA en la última década, entre las que se encuentran la *Intel FPGA SDK* para *OpenCL* [12] y Xilinx SDSoC [13], lo que ha provocado una reducción en los costos de producción y adquisición de este tipo de dispositivos.

Existen gran cantidad de experimentos y aplicaciones montadas sobre FPGAs, sin embargo, debido a la línea principal de este artículo, solamente se mencionan algunas de ellas, relacionadas al desarrollo de aceleradores y de implementación neuronal.

Recientemente, en el año 2017 [14], *Zhao* y otros autores implementaron un clasificador de tipo red neuronal binaria (BNN, por sus siglas en inglés) en una placa de desarrollo FPGA de bajo costo (*ZedBoard*) y mostraron, según sus propias palabras, mejoras en comparación con las líneas de base de CPU (*Central Processing Unit*) y GPU (*Graphics Processing Unit*) incorporadas, así como con los aceleradores de FPGA existentes.

En su trabajo, *Zhao* et al. [14] evaluaron el diseño en un ZedBoard, que utiliza un Xilinx Zynq-7000 SoC de bajo costo que contiene un FPGA XC7Z020 junto con un procesador integrado ARM Cortex-A9. Compararon su diseño con dos plataformas informáticas de servidor: un procesador multinúcleo (CPU) Intel Xeon E5-2640 y una GPU NVIDIA Tesla K40 (GPU). Según los autores, ellos fueron los primeros en implementar un acelerador para redes neuronales binarias en FPGA.

Por otro lado, *Jin, Gokhale* y otros autores [15] presentaron una implementación en tiempo real de redes neuronales convolucionales profundas (DCNN, por sus siglas en inglés) acelerada por *hardware*. Su sistema fue implementado en una plataforma Xilinx Zynq-7000.

En [16], *Lysaght, Stockwood* y otros autores describen la implementación de una RNA en una matriz de compuerta programable de campo *Atmel AT6005* (FPGA). En propias palabras de los autores, "el trabajo se llevó a cabo como un experimento en el mapeo de una aplicación lógicamente intensiva a nivel de bit en los recursos lógicos específicos de un FPGA de grano fino. Al explotar las capacidades de reconfiguración del FPGA de Atmel, las capas individuales de la red se multiplexan en el tiempo en la matriz lógica. Esto permite implementar una RNA más grande en una sola FPGA a expensas de una operación más lenta del sistema en general".

Gadea, Cerdá y otros autores [17] describen la implementación de una matriz sistólica para un perceptrón multicapa en una FPGA Virtex XCV400 con un algoritmo de aprendizaje amigable con el *hardware*. Ellos muestran una adaptación segmentada del algoritmo de tipo *Backpropagation* en línea. Según los autores, el paralelismo se explota mejor porque las fases hacia adelante y hacia atrás se pueden realizar simultáneamente.

Según este grupo de científicos, las simulaciones de *software* son útiles para investigar las capacidades de los modelos de redes neuronales y crear nuevos algoritmos; pero las implementaciones de *hardware* siguen siendo esenciales para aprovechar al máximo el paralelismo inherente de las redes neuronales.

Venieris, y Bouganis [18] presentaron *fpgaConvNet*, un marco para mapear redes neuronales convolucionales en FPGA.

Según su evaluación experimental, fpgaConvNet ofrece predicciones de rendimiento bastante precisas y logra mejoras en la densidad y la eficiencia del rendimiento en comparación con los trabajos existentes de FPGA y GPU incrustado.

En [19] se presentaron tres aceleradores de hardware para RNN en Zynq SoC FPGA de Xilinx para mostrar cómo superar los desafíos involucrados en el desarrollo de aceleradores RNN. En sus experimentos, el hardware produjo con éxito texto de *Shakespeare* utilizando un modelo de nivel de personaje.

Una aplicación interesante de la mezcla de RNA y FPGA se da en [20], en este trabajo se propone un sistema de clasificación de gas en tiempo real de baja latencia, que utiliza una red neuronal artificial de perceptrón multicapa (MLP) para detectar y clasificar los datos del sensor de gas por medio de un MLP (*Multi Layer Perceptron*) paralelo, implementado en una plataforma de sistema sobre chip SoC de Xilinx.

Para un resumen más amplio relacionado a las implementaciones dentro de sistemas embebidos y FPGAs, pueden visualizarse las referencias [23-26].

Como se puede ver a través de esta sección, se han generado una gran cantidad de tarjetas de desarrollo, entre ellas, las tarjetas FPGA. Tales dispositivos embebidos, han sido de utilidad para la implementación de diferentes modelos de redes neuronales, con la intención de mejorar el rendimiento y/o acelerar los procesos de cálculo. En el presente trabajo, se ha tomado como punto de partida la tarjeta de desarrollo PYNQ-Z1 de *Digilent* que incluye una FPGA y una unidad de procesamiento. Se ha buscado implementar un mismo modelo neuronal tanto en la unidad de procesamiento como en la FPGA de la placa, con la finalidad de poder determinar cuál de los dos dispositivos presenta una mejor eficiencia temporal, y definir las principales diferencias de una implementación en *hardware* con respecto a la implementación en *software*.

3. Conceptos básicos

3.1. FPGA

Un FPGA (*Field Programable Gate Array*) es un dispositivo semiconductor que contiene componentes lógicos programables (CLP) e interconexiones programables entre ellos. Los CLP pueden ser programados para duplicar la funcionalidad de puertas lógicas básicas tales como AND, OR, XOR, NOT o funciones combinacionales más complejas tales como decodificadores o simples funciones matemáticas.

En muchos FPGA, los CLP (o bloques lógicos, según el lenguaje comúnmente usado) también incluyen elementos de memoria, los cuales pueden ser simples *flip-flops* o bloques de memoria más complejos. La figura 3, muestra un esquema básico de un FPGA. Estos dispositivos son programados por medio de lenguaje VHDL.

3.2. VHDL

VHDL es un lenguaje de especificación definido por el IEEE (*Institute of Electrical and Electronics Engineers*), bajo el esquema ANSI/IEEE 1076-1993, utilizado para describir circuitos digitales y para la automatización de diseño electrónico. VHDL es acrónimo proveniente de la combinación de dos acrónimos: VHSIC (*Very High Speed Integrated Circuit*) y HDL (*Hardware Description Language*). Aunque puede ser usado de forma general para describir cualquier circuito digital, se usa principalmente para programar PLD (*Programmable Logic Device*), FPGA (*Field Programmable Gate Array*), ASIC (*Application-specific Integrated Circuit)* y similares.

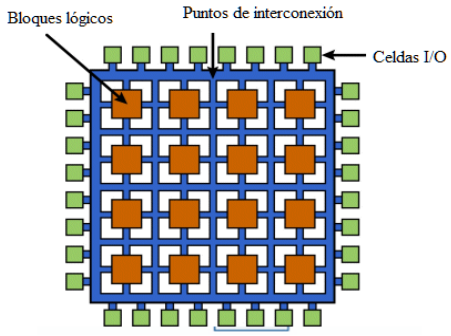


Fig. 3. Estructura interna de un FPGA: donde se muestran los bloques lógicos que pueden ser configurables para generar la función a realizar.

Originalmente, el lenguaje VHDL fue desarrollado por el departamento de defensa de los Estados Unidos a inicios de los años 80 basado en el lenguaje de programación *Ada* con el fin de simular circuitos eléctricos digitales.

Posteriormente se desarrollaron herramientas de síntesis e implementación en hardware a partir de los archivos VHD. En la siguiente sección se presentan algunos detalles extra de este tipo de lenguaje de programación, haciendo hincapié en su importancia dentro del proyecto.

3.3. Neuronas artificiales con VHDL

Antes de crear una red en VHDL, se deben analizar las características y ventajas que nos proporciona este lenguaje de descripción. La primera característica, y la más importante, es que a diferencia de los lenguajes de programación estándar que necesitan compilarse o interpretarse en instrucciones que un microprocesador pueda ejecutar, una descripción con VHDL se convierte directamente en hardware, este dinamismo nos permite crear módulos que trabajen intrínsecamente en forma paralela.

Un ejemplo sencillo para ver las ventajas que nos proporciona VHDL sería hacer dos contadores, que cada tiempo t_x incrementen su valor en uno, usando un lenguaje de programación clásico como C, primero se aumentaría el valor de un contador, y posteriormente del otro, pero nunca los dos a la vez, pues las instrucciones en C están siendo ejecutadas por un microprocesador que solamente puede hacer una tarea a la vez, pero usando VHDL, nosotros podemos diseñar dos circuitos independientes que hagan la tarea, en pocas palabras, los dos contadores pueden incrementar su valor al mismo tiempo pues diseñamos dos circuitos iguales que trabajan en paralelo y que no dependen uno del otro para poder ejecutarse.

Ahora que sabemos las ventajas que nos puede proporcionar VHDL, se realiza un modelo de red neuronal que pueda aprovechar estas ventajas, para hacer este diseño se debe pensar en las neuronas que componen la red como pequeños procesadores que tienen todos los recursos necesarios para trabajar en su arquitectura y no los comparten.

El proceso que hace una neurona para obtener un valor en su salida es multiplicar cada entrada por el valor de su peso y sumarlos todos, después sumar el offset y a esta sumatoria aplicar una función sigmoidea, escalón, rampa o algún otra, en este caso se

ha utilizado la función rampa, por ser más fácil implementarla en VHDL, pero puede utilizarse cualquier función deseada.

En este esquema tenemos que p_i son las entradas de la neurona, estas entradas pueden estar conectadas a un valor específico o la salida de otra neurona, es importante que una entrada esté conectada únicamente a un elemento, de lo contrario se podría producir un comportamiento inesperado, w_i son los pesos y cada entrada tiene el propio, estos valores son los que se irán modificando para hacer que la red neuronal converja, b es el offset, este valor también puede ser modificado por la neurona para ajustar su salida, y le permite tener una salida diferente de cero, aun cuando todas sus entradas sean cero, esta descripción está dada en la ecuación (1):

$$a = f(n) = f\left(\sum_{i=0}^{i=q}(p_i * w_i) + b\right). \tag{1}$$

Cabe mencionar que los pesos y el *offset* son valores aleatorios generados dentro de un rango y asignados a los enlaces la primera vez que se ejecuta una red.

Ahora veremos el proceso de corrección de los pesos, para poder acercarse al resultado que esperamos. Donde W_{ni} son los nuevos pesos, W_{ai} son los pesos anteriores, I_i es el valor de la respectiva entrada, E es el valor del error y f el factor de aprendizaje, lo podemos ver en la ecuación (2):

$$W_{ni} = W_{ai} + (I_i * E * f). \tag{2}$$

Esta fórmula de corrección es aplicada a cada uno de los pesos en los enlaces donde la neurona tenga una entrada valida. Para corregir el offset se utiliza la ecuación (3). Donde O_n es el nuevo offset calculado, O_a es el offset actual, E es el valor del error y f el factor de aprendizaje:

$$O_n = O_a - (E * f). \tag{3}$$

La corrección de pesos y offset se hace tantas veces como sea necesario, hasta obtener en la salida un resultado aceptable, pero este proceso de corrección únicamente funciona para una neurona que tenga conectadas las entradas y salidas directamente, pero no será efectivo si se tiene una red neuronal multicapa con varias neuronas conectadas entre sí.

3.4. Redes multicapa

En una red multicapa es importante entender que las neuronas de salida, son a las que se les indica el valor esperado y pueden compararlo con el valor actual y corregir sus pesos, pero a las neuronas de capas anteriores no les podemos indicar cuál es el valor esperado, este valor se lo deben dar las neuronas que tengan conectadas a la salida.

Las neuronas de la última capa, después de corregir sus pesos y offset, calculan que valor para cada una de sus entradas podría beneficiales, después la proporcionan según el peso que le dan a esta entrada, y este valor lo pasan como valor ideal a la neurona que tienen conectada en esa entrada.

Cuando una neurona de capa intermedia recibe todos los valores de corrección de las neuronas de la capa de salida, los pondera y decide a que valor deberá aproximarse, calcula su error, corrige sus pesos y nuevamente hace el proceso anterior para poder pasar a la capa anterior un valor de corrección, y así sucesivamente hasta que el valor de corrección llegue a la capa que tiene conectadas las entradas.

4. Implementación de red

Tomando en cuenta las ecuaciones y procesos explicados anteriormente, se implementó el siguiente algoritmo dentro de la FPGA considerando cada neurona como una unidad de procesamiento individual, capaz de generar la salida y corrección de errores por sí misma, los cambios en la salida o en el valor de error de la neurona se consideran como eventos que se ejecutan de forma concurrente al percibir un cambio y no como funciones que se ejecutan cada vez que se las llama.

Evento 0: Creación de la neurona. Este proceso lo lleva a cabo únicamente una vez cuando se comienza la ejecución.

1. Poner con valor de cero la salida y la señal de error.
2. Inicializa con un valor aleatorio o precargado los pesos y el offset.
3. Inicializa con un valor aleatorio o precargado el factor de aprendizaje.
4. Queda a la espera de percibir un cambio en las entradas.

Evento 1: Cambio en la entrada. Este proceso se ejecuta de forma concurrente, cada vez que la neurona detecta un cambio en sus entradas.

1. Multiplica cada entrada por su respectivo peso.
2. Realiza la sumatoria de todas las entradas y suma el valor del offset.
3. Aplica la función programada, puede ser rampa, sigmoidea o escalón.
4. Cambia el valor de la salida.

Evento 2: Cambio en la señal de error. Este proceso se ejecuta de forma concurrente cada vez que la neurona detecta un cambio en la señal de error.

1. Calcula el valor esperado considerando todas las neuronas que tiene a su salida.
2. Calcula el valor de error comparando el valor esperado contra el obtenido.
3. Hace la corrección de los pesos, ya sea aumentándolos o disminuyéndolos en proporción al error que aporto cada entrada y el factor de aprendizaje.
4. Hace la corrección del offset.
5. En base al error obtenido, hace el cálculo del valor ideal en cada entrada y lo pasa a la neurona en esa entrada.

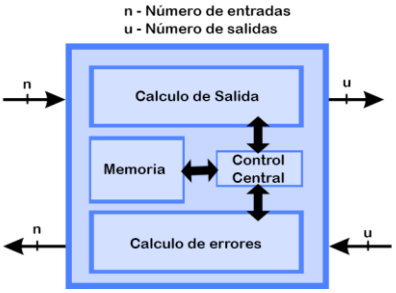


Fig. 4. Modelo de una neurona en una FPGA.

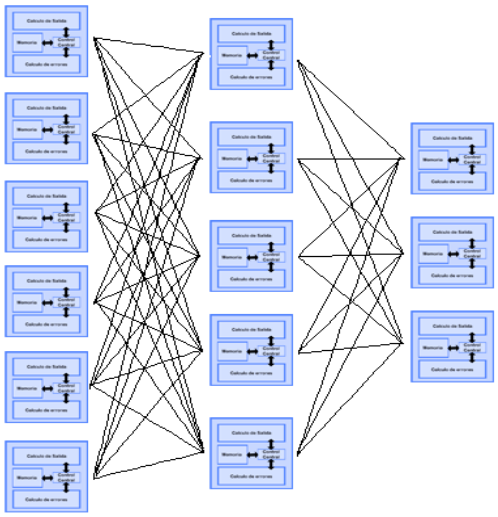


Fig. 5. Modelo de la red dentro de la FPGA.

En base a este algoritmo, el diseño de la neurona seria parecido al mostrado en la figura 4.

Ahora que se tiene el diseño de cada neurona individual, se procede a conectarlas entre sí, en VHDL se vería cada neurona como un módulo o procesador que comparte entradas y salidas con otros semejantes comunicándose entre sí, parecido al diseño del tejido neuronal de un ser vivo, pero con muchas menos conexiones y capacidad.

Antes de implementar el algoritmo para conectar las neuronas entre sí, se debe tomar en cuenta la imposibilidad de ejecutar en paralelo todas las neuronas en la red, sólo las de la misma capa, pues la capa dos depende del resultado obtenido de la capa uno, la capa tres depende del resultado de la capa dos y así sucesivamente. Una vez que todas las capas se ejecuten y obtengamos el error, tendremos el mismo problema, pero a la inversa pues el error lo calcula la última capa, y lo pasa a la capa anterior, está a la anterior y así sucesivamente hasta que todas las capas conozcan su parte de error y puedan corregirlo. El algoritmo de la red neuronal completa se describe a continuación:

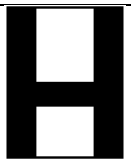
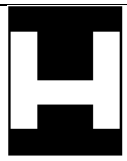
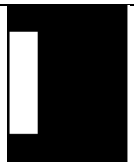
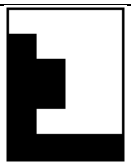
	Patrón	Salida 1	Salida 2	Salida 3
Prueba1		85%	-	-
Prueba2		75%	-	-
Prueba3		30%	-	-
Prueba4		97%	-	-
Prueba1		80%	-	-
Prueba2		73%	-	-
Prueba3		26%	-	-
Prueba4		45%	-	-
Prueba1		-	70%	-
Prueba2		-	54%	-
Prueba3		-	76%	-
Prueba4		-	69%	-
Prueba1		-	45%	-
Prueba2		-	32%	-
Prueba3		-	58%	-
Prueba4		-	47%	-
Prueba1		-	-	60%
Prueba2		-	-	76%
Prueba3		-	-	54%
Prueba4		-	-	65%
Prueba1		-	-	90%
Prueba2		-	-	87%
Prueba3		-	-	92%
Prueba4		-	-	86%

Fig. 6. Patrones de prueba que se ejecutaron en la implementación dentro de la FPGA, donde se muestran tres columnas de salida, cada una de ellas representando una salida distinta. Cada una de estas salidas de la red expresa el porcentaje que representa el parecido del patrón de entrenamiento por salida.

1. Se ejecuta en paralelo la capa uno, que está conectada a las entradas de la red.
2. Se ejecuta en paralelo la capa dos de la red con los valores provenientes de la capa uno.
3. Se ejecuta en paralelo la capa tres de la red, con los valores obtenidos de la capa dos.
4. Se compara la salida obtenida en la capa tres con la salida que se espera obtener.

5. La capa tres ejecuta en paralelo el cálculo del error entre la salida esperada y la obtenida, corrige sus pesos y pasa el error a la capa dos.

6. La capa dos recibe el error de la capa tres, corrige sus pesos y pasa el valor de error a la capa uno.

7. La capa uno recibe el error de la capa dos y corrige sus pesos de ser necesario.

8. Comienza nuevamente el flujo de ejecución hasta que el error sea mínimo.

El modelo implementado se puede observar en la figura 5.

4.1. Pruebas realizadas en la red neuronal

La primera prueba realizada, es una prueba muy simple que solamente consiste en la identificación de caracteres especiales. Con una red que está conectada a una imagen representada por una matriz de 24 posiciones, cada posición puede contener un 1 que se representa con color negro, o un -1 que se representa con color blanco, la capa de salida tuvo tres neuronas, por lo que puede reconocer tres patrones. La red neuronal fue entrenada con tres patrones base: A, I, E. Con estos patrones se obtuvo una respuesta satisfactoria de su ejecución, ya que lograba clasificar correctamente la mayoría de los casos de prueba. En la figura 6 podemos observar el comportamiento de esta red neuronal cuando ingresamos diferentes imágenes matriciales para poder reconocer o identificar el matriz origen. Este primer experimento solamente fue realizado para verificar el funcionamiento de la red en su implementación en FPGA.

Un segundo experimento fue realizado para la verificación y clasificación de diferentes imágenes, y por medio de este experimento, analizar el funcionamiento y ejecución de una implementación neuronal en un FPGA y en una unidad de procesamiento.

Para comprobar las ventajas que nos proporciona un modelo implementado en hardware contra uno implementado en software, compararemos ambos modelos en la tarjeta de desarrollo PYNQ-Z1 que tiene un dispositivo SoC Zynq®-7000, será utilizado el modelo FINN que implementa una red neuronal binarizada programable en Phyton para clasificación de imágenes.

El modelo que usaremos esta previamente entrenado para clasificar imágenes. Sera implementado en el procesador del SoC, que es un Dual-Core ARM® Cortex®-A9, y también en el FPGA Artix-7 de la misma comparando ambos resultados.

De acuerdo a las observaciones relacionadas a la tabla 1, vemos que los tiempos de clasificación de las distintas imágenes son diferentes, debido principalmente a los tamaños de cada una de ellas, sin embargo, a pesar de las diferencias en tamaños, podemos notar que el promedio de tiempo de clasificación del software sobre el hardware es 1300 veces más rápido en todos los casos.

Tabla 1. Tiempos de clasificación de diferentes imágenes, ejecutadas dentro del FPGA y dentro del procesador.

Muestra	Tiempo total	Tiempo por imagen
1 (hardware)	.0063 seg	.00063 seg
1 (software)	8.32 seg	.83245 seg
2 (hardware)	.0062 seg	.00062 seg
2 (software)	8.34 seg	.83459 seg
3 (hardware)	.0061 seg	.00061 seg
3 (software)	8.35 seg	.83562 seg
4 (hardware)	.0064 seg	.00064 seg
4 (software)	8.34 seg	.83412 seg
5 (hardware)	.0067 seg	.00067 seg
5 (software)	8.28 seg	.82894 seg

Tabla 2. Clasificaciones correctas y erróneas de las imágenes base, modificadas con características diferentes tales como colores, fondos, iluminación, y direccionamientos aleatorios.

Muestra	Número de imágenes	Muestras correctamente clasificadas.	Muestras erróneamente clasificadas
1	10	8	2
2	10	9	1
3	10	8	2
4	10	7	3
5	10	8	2

Tabla 3. Características de las imágenes analizadas con la red neuronal para verificar su nivel de exactitud.

Características de las imágenes probadas.	Resultado
Colores estandar.	Correctamente clasificada.
Imágenes con elementos visuales agregados.	Incorrectamente clasificada.
Cambio de color en las imágenes.	Incorrectamente clasificada.
Imágenes con pequeñas variacion de colores.	Correctamente clasificada.
Imágenes en mabientes cargados de elementos extras.	Incorrectamente clasificada.

Es importante mencionar que en el modelo implementado en la PYNQ no se hace corrección de errores, simplemente se cargan los pesos pre programados.

Esta red fue sometida a ejemplos distorsionados y con características diferentes (como colores y fondos aleatorios) de cada una de las imágenes base. En la tabla 2, podemos observar el número de muestras correctamente clasificadas y el número de muestras no clasificadas correctamente por medio de la RNA.

En basa a la tabla 2, los principales factores que generan una clasificación errónea son el cambio de color de la imagen base, la orientación de la imagen (dependiendo del grado de orientación o inclinación, la red puede identificarla o no identificarla), y cuando hay muchos elementos extra, tales como fondos aleatorios o caracteres adicionales sobre la muestra. La iluminación también ha jugado un papel interesante, ya que si se cambia la iluminación de la imagen, la red confunde los colores y los identifica como colores diferentes a la muestra, generando una mala clasificación. La tabla 3, describe algunos ejemplos de imágenes correctamente clasificadas y erróneamente clasificadas.

5. Conclusiones

En este trabajo se ha presentado la implementación de una red neuronal dentro de una FPGA, con esta implementación se han obtenido las métricas relacionadas al tiempo y a la eficiencia de clasificación de imágenes de la red neuronal. Las pruebas realizadas dentro de la FPGA, han sido realizadas también dentro de una unidad de procesamiento (Ambas, FPGA y unidad de procesamiento, embebidas dentro de una tarjeta de trabajo Soc Zynq de Xilinx) con la intención de comparar sus velocidad y eficiencia, logrando observar que la implementación en la FPGA es, por mucho, más rápida que la ejecución cobre la unidad de procesamiento.

A pesar de que las diferencias en tiempo de ejecución entre la unidad de procesamiento y la FPGA, la clasificación de las imágenes llevada a cabo dentro de la unidad de procesamiento, tiene la misma eficiencia que la llevada a cabo dentro de la FPGA.

Las FPGA son dispositivos que permiten una ejecución de algoritmos intrínsecamente paralela, esto permite que las redes neuronales artificiales, puedan potenciar su tiempo de ejecución cuando son adecuadamente implementadas, sumado a esta característica, está el hecho de su bajo coste y gran accesibilidad, logrando posicionar a las FPGA como dispositivos bien capacitados para la investigación relacionada a inteligencia artificial y redes neuronales.

6. Trabajo futuro

Como trabajo futuro, se buscará implementar una misma red en diferentes tipos de sistemas embebidos y diferentes tipos de tarjetas de desarrollo de *Digilent*, para poder definir en cual, de toda esta gama de posibilidades existentes, se tiene una mejor respuesta tanto temporal como de cálculo, sumando al estudio una comparativa en relaciones tiempo de ejecución-precio.

Se plantea la generación de un *Cluster* realizado con diferentes FPGA, para verificar su funcionamiento en este esquema.

Agradecimientos. Este trabajo fue en parte financiado por el proyecto PAPIIT IN 113316 y el proyecto PIAPI 1634, de la UNAM.

Referencias

1. Curran, K.; Li, X.; McCaughley, N.: Neural network face detection. The Imaging Science Journal, 53(2), pp. 105–115 (2005)
2. Patil, V.; Shimpi, S.: Handwritten English character recognition using neural network. Elixir Comput Sci Eng, 41, pp. 5587–5591 (2011)
3. Bartlett, P. L.: The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE transactions on Information Theory, 44(2), pp. 525–536 (1998)
4. Åkesson, B. M.; Toivonen, H. T.: A neural network model predictive controller. Journal of Process Control, 16(9), pp. 937–946 (2006)
5. Timotheou, S.: The random neural network: a survey. The computer journal, 53(3), pp. 251–267 (2010)
6. McCulloch, W.; Walter, P.: A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. 5(4), pp. 115–133 (1943)
7. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65(6), pp. 386–408 (1958)
8. Werbos, P. J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University (1975)
9. Rumelhart, D. E; McClelland, J.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press (1986)
10. Ciresan, D. C.; Meier, U.; Masci, J.; Gambardella, L. M.; Schmidhuber, J.: Flexible, High Performance Convolutional Neural Networks for Image Classification. International Joint Conference on Artificial Intelligence (2011)
11. Martines, H.; Bengio, Y.; Yannakakis, G. N.: Learning Deep Physiological Models of Affect. IEEE Computational Intelligence, 8(2), pp. 20–33 (2013)
12. Czajkowski, T. S.; Aydonat, U.; Denisenko, D.; Freeman, J.: Kinsner, M; Neto, D.; Wong, J.; Yiannacouras, P.; Singh D. P.: From OpenCL to High-Performance Hardware on FPGAs. In: Int'l Conf. on Field Programmable Logic and Applications (FPL), pp. 531–534 (2012)
13. Kathail, V.; Hwang, J.; Sun, W.; Chobe, Y.; Shui, T.; Carrillo, J.: SDSoC: A Higher-level Programming Environment for Zynq SoC and Ultrascale+ MPSoC. In: Int'l Symp. On Field-Programmable Gate Arrays (FPGA), pp. 4–4 (2016)
14. Zhao, R.; Song, W.; Zhang, W.; Xing, T.; Lin, J. H.; Srivastava, M.; Zhang, Z.: Accelerating binarized convolutional neural networks with software-programmable FPGAs. In: Proceedings of the 2017 (ACM/SIGDA), International Symposium on Field-Programmable Gate Arrays, pp. 15–24 (2017)
15. Jin, J.; Gokhale, V.; Dundar, A.; Krishnamurthy, B.; Martini, B.; Culurciello, E.: An efficient implementation of deep convolutional neural networks on a mobile coprocessor. In: Circuits and Systems (MWSCAS), IEEE 57th International Midwest Symposium on, pp. 133–136 (2014)
16. Lysaght, P.; Stockwood, J.; Law, J.; Girma, D.: Artificial neural network implementation on a fine-grained FPGA. In: International Workshop on Field Programmable Logic and Applications, pp. 421–431, Springer (1994)
17. Gadea, R.; Cerdá, J.; Ballester, F.; Mocholí, A.: Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation. In: Proceedings of the 13th international symposium on System synthesis, IEEE Computer Society, pp. 225–230 (2000)
18. Venieris, S. I.; Bouganis, C. S.: A framework for mapping convolutional neural networks on FPGAs. In: Field-Programmable Custom Computing Machines (FCCM), IEEE 24th Annual International Symposium on, pp. 40–47 (2016)

19. Chang, A. X. M., Culurciello, E.: Hardware accelerators for recurrent neural networks on FPGA. In: Circuits and Systems (ISCAS), IEEE International Symposium on, pp. 1–4 (2017)
20. Zhai, X.; Ali, A.; Amira, A., Bensaali, F.: MLP neural network based gas classification system on Zynq SoC. IEEE Access, 4, pp. 8138–8146 (2016)
21. Digilent PYNQ-Z1: Board Reference Manual. https://reference.digilentinc.com /_media/reference/programmable-logic/pynq-z1/pynq-rm.pdf
22. PYNQ: Python Productivity for ZYNQ. Available: http://www.pynq.io/
23. Guo, K.; Zeng, S.; Yu, J.; Wang, Y.; Yang, H.: A Survey of FPGA Based Neural Network Accelerator. arXiv preprint arXiv:1712.08934 (2017)
24. Zhu, J.; Sutton, P.: FPGA implementations of neural networks a survey of a decade of progress. In: International Conference on Field Programmable Logic and Applications, pp. 1062–1066 Springer (2003)
25. Misra, J.; Saha, I.: Artificial neural networks in hardware: A survey of two decades of progress. Neurocomputing, 74(1-3), pp. 239–255 (2010)
26. Liu, J.; Liang, D.: A survey of FPGA-based hardware implementation of ANNs. In: Neural Networks and Brain, (ICNN&B'05) IEEE International Conference on, 2, pp. 915–918 (2005)

ISSN 1870-4069

Uso de datos abiertos y técnicas de minería de datos para la clasificación de estudiantes de instituciones privadas de educación superior de Belem-PA

Matheus Ferreira Vasconcelos[1], Bruno Kenji Hosoda Mineshita[1], Edian Franklin Franco de Los Santos[2], Alan Marcel Fernandes Souza[1]

[1] Universidade da Amazônia, Centro de Ciências Exatas, Belém, Pará, Brasil

[2] Universidade Federal do Pará, Centro Genômica e Biologia de Sistemas, Belém, Pará, Brasil

vasconcelosmf@outlook.com, hosoda_kenji1997@hotmail.com, edianfranklin@gmail.com, alan.souza@unama.br

Resumen. Las universidades privadas enfrentan el desafío de reducir la tasa de evasión de estudiantes de los cursos de grados. Si esta tasa es muy alta, el beneficio de estas universidades puede reducir drásticamente, alejando a los accionistas e inversores. Este trabajo tiene como objetivo aplicar técnicas de minería de datos clasificar entre estudiantes que abandonan y los que terminan los cursos de grados, para facilitar la identificación de los posibles factores de riesgos que influyen en la evasión universitaria. Para el análisis fueron utilizados datos abiertos extraídos del portal del gobierno y diversos algoritmos de minería de datos. En los experimentos realizados, fue posible seleccionar la técnica más eficiente de clasificación para este problema, mediante el análisis de métricas de confianza, consistencia y calidad. Además, fue realizado un levantamiento de los factores que tiene mayor incidencia en la evasión y abandono de los estudiantes de las universidades privadas de la ciudad de Belem do Pará (Brasil).

Palabras clave: Evasión de estudiantes, minería de datos, datos abiertos, análisis de métricas de calidad.

Usage of Open Data and Data Mining Techniques for the Classification of Students from Private High Education Institutions of Belém-PA

Abstract. The private universities face the challenge to reduce the evasion rate of the undergraduate students. If this rate is too high, the profit of these universities can drastically reduce, driving away shareholders and investors. This paper aims to apply data mining techniques to classify students who completed the course and those who dropped out, in order to facilitate the identification of possible risk factors that influence university dropout. The analysis used open data extracted from the government portal and various data mining algorithms.

In the experiments carried out, it was possible to select the most efficient classification technique for this problem, through the analysis of confidence, consistency and quality metrics. In addition, a survey was carried out of the factors that have a greater incidence in the evasion and abandonment of the students of the private universities of the city of Belem do Pará (Brazil).

Keywords: Student evasion, data mining, open data, analysis of quality metrics.

1. Introducción

La cantidad de estudiantes que ingresaron en una Institución de Enseñanza Superior (IES) y no llegan a graduarse es alarmante. Según el Ministerio de Educación [7], en la educación superior, hay más estudiantes de nuevo ingreso que concluyentes. En el año 2012 se registraron 7.037.688 alumnos matriculados, pero el total de alumnos que ingresó a las IES fue 2.747.089 y el número de concluyentes fue 1.050.413.

El número de matrículas en cursos de grado continuó creciendo en 2016, sin embargo, si se compara con años anteriores hubo una desaceleración. Según los estudios estadísticos, entre los años 2006 a 2016 hubo un aumento del 62,8% con un promedio anual del 5% en el número de matrículas, prevaleciendo el 75,3% en IES privadas. En 2016 se ofrecieron más de 10,6 millones de vacantes para estudios superiores, siendo el 73,8% nuevas y el 26% ya existentes, de las cuales solo el 33,5% de las nuevas vacantes y el 12% de las existentes fueron ocupadas por nuevos estudiantes. A partir de este estudio es posible diagnosticar que el número de vacantes ofertadas no implica la permanencia de los alumnos hasta la conclusión, lo que podemos considerar como evasión [8].

La evasión puede ser causada por diversos factores y variables que afectan la vida de un estudiante, basados en el trabajo de Couto y Santana [2], es evidente que las IES aún no cuentan con las informaciones o conocimientos de los motivos y factores que llevan a la evasión en la educación superior. Martins [6], en su trabajo estudió los diversos factores que pueden llevan a la evasión y clasificó la evasión en tres tipos: i. Evasión de carrera, la interrupción de la carrera por diversas situaciones (abandono, desistimiento, transferencia, re-opción, bloqueos, exclusión por normas institucionales), ii. Evasión de la institución (abandono o desistimiento de la IES en la cual está matriculado) y iii. Evasión del sistema (abandono de forma definitiva o temporaria de la educación superior). En este estudio serán considerados como estudiantes evasores, todos aquellos que abandonaron la institución o carrera de forma temporal o definitiva.

Este artículo utiliza datos abiertos, haciendo uso de cuatro técnicas de minería de datos para identificar las variables que más impactan en la evasión de los alumnos de IES privadas de Belém do Pará:

- Árboles de decisión (CART);
- Naïve Bayes;
- K-NN;
- Support Vector Machine (SVM).

Para la evaluación del desempeño de los modelos fueron utilizadas las siguientes metricas: precisión, sensibilidad, especificidad, valor predictivo positivo (PPV), valor predictivo negativo (VPN) y Matthew Coefficient Calculator (MCC).

Este trabajo está organizado de la siguiente manera: la sección 2 presenta los trabajos relacionados. En la sección 3, se muestran las etapas del proceso de descubrimiento de conocimiento en la base de datos. En la sección 4, los experimentos desarrollados utilizando las técnicas de minería de datos, siguiendo el análisis de los resultados. Por último, en la sección 5, se presentará la conclusión y los trabajos futuros.

2. Trabajos relacionados

Los trabajos correlacionados en esta sección se refieren a enfoques de métodos de minería de datos con el uso de diversos algoritmos de clasificación, buscando analizar el desempeño a través de métricas, para la identificación del mejor clasificador para el problema en cuestión: evasión de estudiantes de las IES.

En [3] realizaron un estudio de caso sobre la predicción de evasión de alumnos, usando datos del Electrical Engineering department of Eindhoven University of Technology junto a técnicas de Minería de datos. Para identificar a los estudiantes con riesgos de evasión en el estado inicial, se realizaron pruebas con 8 algoritmos, (CART (simple Cart), C4.5 (J48), redes bayesianas, modelos de regresión logísticas (Simple Logistic), reglas basadas en aprendizaje (JRip), Random Forest y OneR), en las pruebas todos mantuvieron una exactitud entre 75% - 80%.

En [14] presentaron un estudio utilizando registros de alumnos registrados en el Online Information Technologies Certificate Program en el periodo 2007-2009, con el objetivo de prevenir la evasión de estudiantes, a través de la utilización de métodos de minería de datos. Se realizaron pruebas con cuatro algoritmos de clasificación (K-NN (k = 3), Árboles de decisión, Naïve Bayes y Redes Neuronales). Las sensibilidades de detección de los algoritmos fueron, respectivamente, el 87%; 79,7%; el 76,8% y el 73,9%. Los modelos fueron entrenados, probados y validados con utilizando validación cruzada (10-fold).

El trabajo de Couto e Santana [2] presenta una serie de pruebas realizadas con datos proporcionados por el SIGAA (Sistema de Gestión de Actividades Académicas) referentes a los estudiantes que cursos de grado que ingresaron hasta el año 2016, con el objetivo de creación de subsidios para auxiliar las IES e identificar a los alumnos con riesgo de evasión utilizando Knowledge Discovery Databases (KDD).

Se realizaron pruebas con nueve algoritmos en la etapa de minería de datos, con la utilización de métricas para análisis de desempeño de los clasificadores, con el objetivo de identificar el mejor modelo para cumplir al objetivo del trabajo. Fue seleccionado un modelo basado en redes Bayesianas, con el consiguieron obtener una precisión global del 85%.

3. Etapas de la Minería de Datos

La minería de datos puede ser definida como un conjunto de técnicas que permiten la asociación y correlación de información de forma automatizada en grandes bases de datos. Esto hace posible el descubrimiento de patrones de manera más eficiente. [12]

La aplicación de técnicas de minería de datos forma parte del proceso de KDD. Sin embargo, para ello, se hace necesario pre-procesar, organizar y filtrar los datos brutos para que sea posible obtener un patrón, que permita el análisis e interpretación de los resultados obtenidos. De esta forma, es necesario seguir algunas etapas del proceso de KDD que se describen a continuación.

3.1. Selección de los datos

La etapa de selección de datos nos permite agrupar todas las fuentes de datos en una base de datos. Para Weiss [13], entender los datos que se utilizarán es crucial para el éxito en el desarrollo de las aplicaciones.

Los datos proporcionados por el Instituto de Estudios e Investigaciones Educativas Anisio Teixeira (INEP), fueron obtenidos de forma estructurada en el formato CSV (Comma-separated values) junto con sus metadatos, lo que facilito el entendimiento de los atributos y sus valores. La herramienta computacional utilizada para procesar los datos fue el RStudio versión 1.0.143, a través del lenguaje de programación R versión 3.4.1 (30/06/2017). Para el agrupamiento de datos, se creó un banco SQLite conteniendo los datos de las Instituciones de Educación Superior, los cursos y los estudiantes, obteniendo una base de datos con 265 atributos y 11.449.222 registros.

3.2. Preprocesamiento

Según Han e Kamber [5], con los datos organizados en una base de datos es necesaria la realización de etapas como: limpieza, integración, transformación y reducción del conjunto de datos.

La limpieza de los datos consiste en eliminar las inconsistencias encontradas: atributos incompletos, valores con errores, valores nulos, etc. En este caso, se optó por eliminar los registros con atributos incompletos, también fueron eliminados de varios conjuntos de datos, los atributos con pocas cantidad de instancia.

La integración de los datos consiste en un análisis en profundidad de los datos observando redundancia, categorías diferentes para los mismos atributos, atributos divergentes, datos repetidos, entre otros.

En el conjunto de datos utilizado gran parte de los atributos no fueron considerados por ser redundantes, debido a que presentaban el mismo valor o la misma información. Por ejemplo, el conjunto de datos utilizados contenía los atributos “sexo” y “cod_sexo”, los dos con la misma información, en este conjunto de datos solo fue preservado el primer atributo referente al sexo y el segundo fue eliminado por ser redundante.

La transformación de los datos se realiza de manera diferente de acuerdo con la técnica a ser aplicada, pues algunos algoritmos sólo funcionan con valores numéricos y otros con valores categóricos. Por lo tanto, se hace necesario la utilización de técnicas de: suavización, agrupamientos, generalización, normalización o la generación de

nuevos atributos (generados a partir de otros ya existentes). En R, las bibliotecas utilizadas para la creación de los modelos ya aplican algunas técnicas de transformación de los datos de forma intrínseca, de acuerdo con el algoritmo a ser utilizado. Sin embargo, se hicieron algunas conversiones de atributos con valores numéricos (ejemplo: 0 o 1) en datos cualitativos y la creación de nuevos atributos (por ejemplo: se sumaron las cantidades de empleados con maestría por sexo (femenina y masculino) y se creó un solo atributo: cantidad de empleados con maestría).

La reducción del conjunto de datos es importante, pues en algunos casos el volumen de datos utilizado en el proceso de minería de datos se considera muy grande hasta el punto de hacer inviable la aplicación de las técnicas minería de datos e incluso el análisis de los datos [6]. Este proceso de reducción de los datos permitió reducir el conjunto de datos original que tenía 256 atributos y 11,449,222 registros, para un conjunto de datos más manejable con 41 atributos y 25,468 registros, conteniendo solo las informaciones de las IES privadas de la ciudad de Belém-PA

La etapa de preprocesamiento es crucial para obtener un buen resultado y es la que demanda más tiempo. Según Olson e Delen [10], dicha fase puede comprender más del 50% del tiempo en proyectos de minería de datos.

3.3. Algoritmos de clasificación

Los algoritmos de clasificación crean un modelo que consigue determinar a cuál categoría pertenece una instancia específica. La clasificación generalmente es un proceso de aprendizaje supervisado, donde los datos se dividen en dos conjuntos: entrenamiento y pruebas [1]. En la fase de entrenamiento, los modelos son construidos utilizando el conjunto de datos de entrenamiento. Para evaluar el desempeño los modelos son testados utilizando del conjunto de prueba y diferentes métricas específicas.

Los algoritmos de clasificación obtienen resultados diferentes dependiendo del conjunto de datos utilizado, existen algoritmos que se adaptan mejor a un conjunto de datos que otros.

En este trabajo fueron utilizados los algoritmos de: arboles de decisión (CART), Naïve Bayes, K-NN y Support Vector Machine (SVM), debido a que presentaron un mejor desempeño en las pruebas realizadas con los conjuntos de datos en la fase de pre-procesamiento, además fueron seleccionadas algoritmos con distintas metodologías de clasificación, lo que permitió comparar y seleccionar el mejor modelo para los datos.

4. Clasificación de los alumnos

Esta sección presenta los experimentos desarrollados con los microdatos de censo de Educación Superior. Los experimentos comparan las métricas de los algoritmos utilizados, identificando si un estudiante concluyó o evadió la carrera. El objetivo del estudio es generar resultados para análisis y llegar a las variables que más impactan en la permanencia o no del alumno en una IES.

4.1. Experimentos

Para los experimentos se utilizaron más de 25 mil instancias con dos clases objetivo - Graduado y Evasor - siendo 17.921 alumnos graduados y 7.547 evasores.

Durante los experimentos, los datos fueron sometidos a la ejecución de los algoritmos de clasificación (Árboles de decisión (CART), Naïve Bayes, K-NN y Support Vector Machine (SVM)) con la finalidad de identificar cuál genera un mejor modelo de clasificación para los dos tipos de alumnos.

Los datos fueron divididos en un 70% para entrenamiento y un 30% para la prueba manteniendo la proporción de cantidades de registros en cada clase (Graduados y Evasores).

Para evitar la pérdida de patrones y tendencias importantes que pudieran aumentar el margen de error en la validación del modelo [4], fue utilizada la técnica de validación cruzada, que divide el conjunto en k partes iguales, donde k-1 partes se utilizan para entrenamiento y la parte restante se utiliza para la prueba, repitiendo hasta que todas las k partes se hayan utilizado para la prueba. En cada prueba se calcula la exactitud para obtener el mejor modelo que será utilizado con los datos de prueba. En los experimentos se utilizó diez para el valor de k.

4.2. Resultados

Los resultados presentados a continuación son referentes a la aplicación de los cuatro algoritmos de clasificación, utilizando los datos abiertos obtenidos del INEP. Fueron calculados los resultados de las pruebas realizadas, utilizando métricas distintas obtenidas a partir de la matriz de confusión resultante de cada modelo. A partir de las comparaciones, el modelo que obtuvo mayor desempeño fue utilizado para identificar los factores con mayor influencian o peso en la evasión de un alumno de grado, estos factores fueron clasificado en una escala 0 a 100.

Se aplicaron los cuatro algoritmos de clasificación y se generaron sus respectivas matrices de confusión a partir del conjunto de datos de prueba. Los modelos fueron evaluados a de la siguiente forma: en primer lugar, se generaron tres métricas: Exactitud (total de aciertos / total de datos en el conjunto), Sensibilidad (aciertos positivos / total de positivos) y Especificidad (aciertos negativos / total de negativos). La Fig. 1 muestra los resultados en porcentaje.

CA partir del experimento los mejores resultados fueron: para exactitud y sensibilidad - Árbol de Decisión con 82,65% y 92,87%, respectivamente, y para la especificidad el K-NN (k = 7) con 66,67%.

Después de las comparaciones con las métricas que calculan la tasa de confiabilidad del algoritmo en relación a su aplicación en la minería de datos. La Fig. 2 presenta los valores predictivos positivos (VPP) y los valores predictivos negativos (VPN). En el caso de que el VPP es representado por las fórmulas: VPP = aciertos positivos / total de predicción positivas y VPN = aciertos negativos / totales de aciertos negativos. [9, 11].

Según Oliveira e Kaestner [9] cuanto más similares los valores de VPP y VPN más consistentes son los resultados de la clasificación.

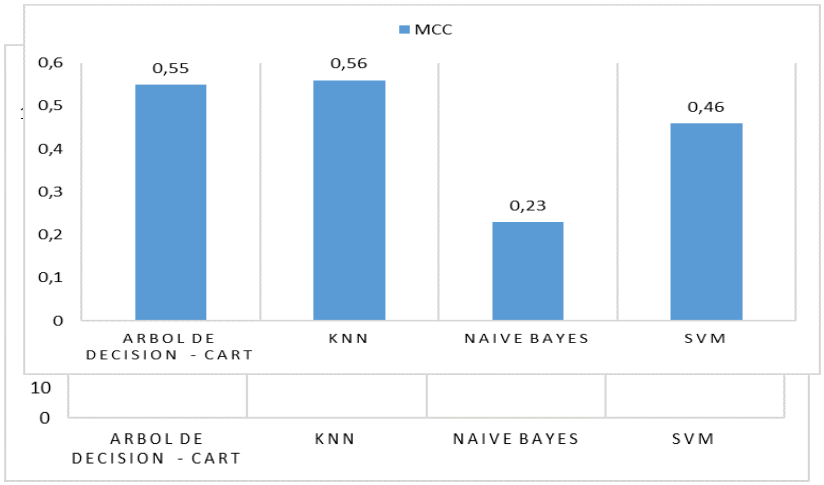


Fig. 1. Exactitud, Sensibilidad, Especificidad de los resultados.

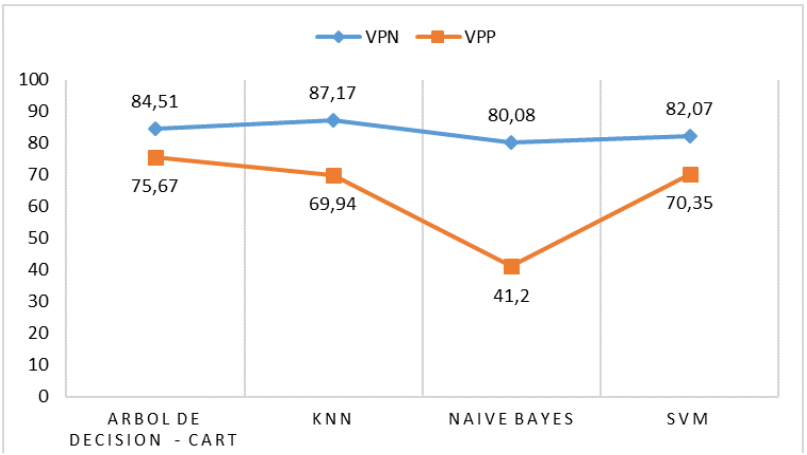


Fig. 2. VPP x VPN de cada clasificador.

Al analizar las dos métricas, que también se basan en la matriz de confusión, la que obtuvo los resultados más discrepantes fue Naïve Bayes, teniendo la mayor variación entre los cuatro modelos probados. El Árbol de Decisión (CART) y SVM fueron los que presentaron los mejores desempeños, obteniendo valores menos discrepantes.

Después del análisis de la consistencia, se calculó el Matthews Correlation Coefficient (MCC), también conocido como coeficiente de PHI, para cada modelo. El MCC es responsable de clasificar la calidad de los modelos. La métrica devuelve valores entre -1 y +1, siendo -1 para clasificación inapropiada, 0 para una clasificación aleatoria y +1 para clasificador correcto [11]. La Fig. 3 resume los valores de MCC encontrados para cada modelo.

El K-NN (k = 7) obtuvo el mejor resultado (0,56) seguido del árbol de decisión (0,55), siendo los dos mejores resultados comparado al SVM y NB. Este último no obtuvo un resultado muy satisfactorio.

En la comparación de los resultados presentados a través de las métricas que fueron utilizadas para clasificar el mejor algoritmo para solución del problema, fue identificado que dentro de los experimentos realizados dos algoritmos, obtuvieron los mejores resultados con valores similares, quedando por cuenta del Árbol de Decisión y K-NN, que obtuvieron valores superiores en cuestión de confiabilidad. Sin embargo, las matrices de confusión que presentaron los mejores resultados en términos de consistencia fueron el Árbol de Decisión y el SVM, con valores similares entre VPP y VPN. La diferencia de las métricas ocurre en el gráfico (Fig. 3) referente al MCC, donde el K-NN y el Árbol de Decisión obtuvieron los mejores resultados, 0,56 y 0,55 respectivamente.

Una mejor forma de observar el desempeño fue trazar una línea referente a la media de cada métrica y los algoritmos que posee la mayor cantidad de valores por encima de la media, presentan los mejores resultados, como muestra la Fig. 4. A través de estos análisis y pruebas se concluyó que el mejor algoritmo para la solución del problema es Árbol de Decisión, basados en las seis métricas utilizadas, este presentó un desempeño superior con relación a los demás algoritmos en cinco de las métricas.

Como el árbol de decisión fue el modelo que obtuvo los mejores resultados, se utilizó su modelo para identificar las variables que más impactan en la evasión de los estudiantes de las IES privadas. Para ello, se utilizó el valor de importancia que el algoritmo CART, que es calculado durante la construcción del Árbol de decisión. Esto permitió la construcción de un ranking de las diez variables que más impactan en la clasificación del alumno como evasor o graduado (Fig. 5).

A partir de análisis de la figura 5, donde son presentadas las 10 variables más influyentes en la evasión del estudiante de una IES, objetivo principal de esta investigación, fue posible identificar que la edad es la variable con más peso. Esta situación que puede ser comprobada cuando se hace un análisis de las reglas obtenida en el modelo de árbol de decisión, en el cual fueron obtenidas las siguientes reglas:

```
22) NU_EDAD_ALUNO < 21.5 Evadió (0.74 0.26) 1%

7573) NU_EDAD_ALUNO >= 34.5 Graduado (0.29 0.71) 1%
```

En estas es posible percibir que cuando la edad de un estudiante es inferior a 22 años, la probabilidad de evasión es un 74%, en cambio cuando un estudiante posee una edad superior a 35 años la probabilidad de graduarse es de un 71%, basados en estas informaciones podemos inferir que estudiantes con edades más elevadas presentan una mayor probabilidad de graduarse en los cursos de grados.

Es importante esclarecer que a pesar de esta variable poseer el mayor peso, no podemos llegar a la conclusión que solamente esta variable es decisiva para predecir la evasión o conclusión de los estudios superiores de un estudiante, pues, existen otras variables que tienes incidencia en el resultado del modelo. El hecho de poseer mayor

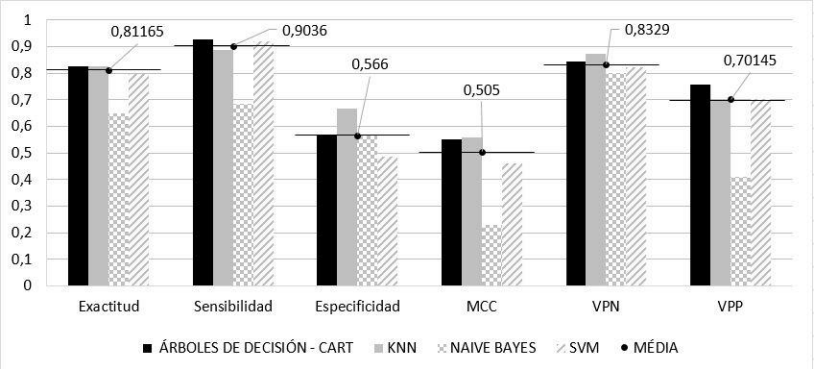


Fig. 4. Análisis de desempeño con todas las métricas calculadas.

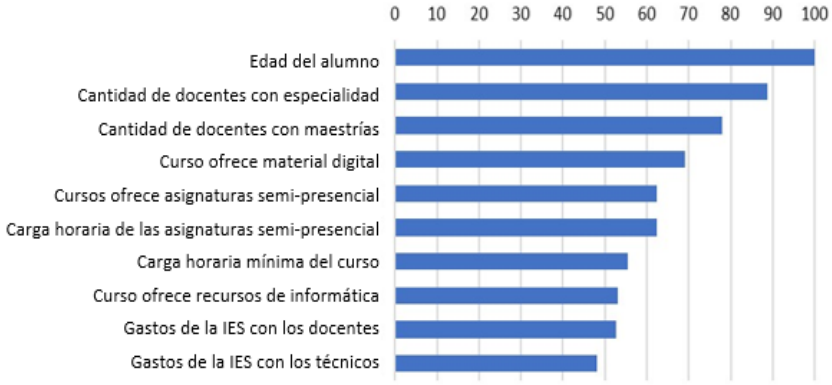


Fig. 5. Análisis de desempeño con todas las métricas calculadas.

peso significa que esa variable tuvo un mayor poder decisión con relación a las demás variables en el modelo.

5. Conclusiones y trabajo a futuro

Los experimentos realizados en esta investigación buscaron identificar al mejor clasificador, a partir de los resultados obtenidos por la minería de datos, referente a los datos abiertos del gobierno sobre evasión de estudiantes de grado de las IES privadas de Belém-PA.

Por lo tanto, a través de las pruebas realizadas con las técnicas de minería de datos, es posible concluir que el uso de diversas métricas y diversos algoritmos para analizar el conjunto de datos es viable y eficiente. Siendo posible identificar a través de experimentos el mejor algoritmo para cada proceso de minería, dependiendo del conjunto de datos obtenidos en la fase de preprocesamiento. Entre los experimentos realizados en este trabajo, el mejor resultado fue obtenido por el Árbol de Decisión (CART). A través de los experimentos el modelo generó una exactitud del 82,65% y

posibilitó identificar los factores de peso en la evasión de estudiantes de las IES de Belém do Pará. Entre los factores generados fue separado los diez más importantes (Fig. 5), posibilitando el análisis de cada uno individualmente por un especialista del área, con su grado de importancia entre 0-100.

Como trabajos futuros, se anhela:

- Consultar a un especialista para realizar una validación del conocimiento descubierto;
- Ampliar el conjunto de datos utilizados, considerando otras IES públicas y / o privadas de otras regiones;
- Crear un sistema web, con el modelo implementado para simulación de escenarios escogidos por el usuario.

Referencias

1. Camilo, C.O., Silva, J.C.: Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas. http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos /RT-INF_001-09.pdf, last accessed 2017/08/15 (2009)
2. Couto, D., Santana, A.: Mineração de Dados Educacionais Aplicada à Identificação de Variáveis Associadas à Evasão e Retenção. In: Araújo A.; Rebouças A.; Souza F.; Aguiar Y. II Congresso sobre Tecnologia na Educação, 1877, pp. 333–344 (2017)
3. Dekker, G., Pechenizkiy, M., Vleeshouwers, J.: Predicting Students Drop Out: A case Study. In: Barnes, T.; Desmarais, M.; Romero, C.; Ventura, S. 2nd International Conference On Educational Data Mining (EDM), 2, pp. 41–50 (2009)
4. Towards Data Science: Cross-Validation in Machine Learning. https://medium. com/towards-data-science/cross-validation-in-machine-learning-72924a69872f (2017)
5. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Cap. 2, pp. 47–103 (2006)
6. Martins, C.: Evasão dos alunos nos cursos de graduação em uma instituição superior.116f. Dissertação (Mestrado Profissional em Administração), Fundação Cultural Dr. Pedro Leopoldo, Minas Gerais (2007)
7. Ministério da Educação: http://portal.mec.gov.br/component/tags/tag/32123? limitstart=0 (2012)
8. Ministério da Educação: http://portal.mec.gov.br/index.php?option=com_docman &view=download&alias=71221-notas-sobre-censo-educacao-superior-2016-pdf& category_slug=agosto-2017-pdf&Itemid=30192 (2016)
9. Oliveira D., Kaestner, C.: Classificação Automática das Reclamações de Clientes de uma Empresa de Telecomunicações. In: Computer on the Beach, pp. 230–238 (2017)
10. Olson, D., Delen, D.: Advanced Data Mining Techniques. Springer (2008)
11. Santos, C.: Avaliação do uso de classificadores para verificação de atendimento a critérios de seleção em programas sociais, 87f. Dissertação (Programa de Pós Graduação em Modelagem computacional) (2017)
12. Santos, R.: Conceitos de Mineração de Dados na Web. http://www.lac. http:.br/~rafael.santos/Docs/WebMedia/2009/webmedia2009.pdf (2009)
13. Weiss, G. M.: Data Mining in Telecommunications. In: Data Mining and Knowledge Discovery Handbook, pp. 1189–1201 (2005)
14. Yukselturk, E., Ozekes, S., Turel Y.: Predicting Dropout Student: an Application of Data Mining Methods in an Online Education Program. European Journal of open, Distance and e-Learning, 17, pp. 119–133 (2014)

ISSN 1870-4069

Aprendizaje profundo de representaciones robustas para clasificación multi-instancia y multi-etiqueta de imágenes

Javier Roberto Veloz Centeno[1], Alfonso Rojas-Domínguez[3], Ivvan Valdez[2], Manuel Ornelas[1], Héctor Puga[1], Martín Carpio[1]

[1] Instituto Tecnológico de León, México

[2] Universidad de Guanajuato, México

[3] CONACYT Research Fellow, México

{veloz_c_22, jmcarpio61}@hotmail.com, alfonso.rojas@gmail.com, si.valdez@ugto.mx, manuel.ornelas@itleon.edu.mx, pugahector@yahoo.com

Resumen. Abordamos el 'Reto de Clasificación de Restaurantes de Yelp', que consiste en predecir los atributos que poseen restaurantes a partir de sus conjuntos de imágenes, etiquetados por la comunidad de Yelp para 9 posibles atributos. Este problema *multi-instancia* y *multi-etiqueta* nos permite explorar una variedad de ideas en el campo de aprendizaje de representaciones. Abordamos el aspecto multi-instancia del problema mediante la agregación de características de alto nivel del conjunto de imágenes de cada restaurante, creando un vector de características prototipo por establecimiento. Las características se extraen mediante Redes Convolucionales Profundas. Posteriormente, utilizamos los vectores de características para entrenar un sistema de clasificadores binarios, uno por atributo. Para mejorar el desempeño de nuestro modelo, inducimos una representación robusta mediante el cálculo de los vectores de características prototipo utilizando redes pre-entrenadas con 3 bases de datos distintas: ImageNet, Food-101 y MIT Places 2. Finalmente, dado que no toda representación individual es igualmente útil para realizar la predicción de un atributo, añadimos un clasificador final que aprende los pesos de predicción para cada representación de un atributo. Nuestra propuesta es un sistema automatizado de principio a fin, que logra un desempeño F1 en el conjunto de evaluación de 0.8177, haciendo a nuestro modelo competitivo con el mejor 10% de los competidores. Recomendaciones y directrices para trabajo futuro también se discuten.

Palabras clave: Redes neuronales convolucionales, aprendizaje de representaciones, clasificación multi-etiqueta, aprendizaje multi-instancia.

Robust Deep Representation Learning for Multi-Instance and Multi-Label Image Classification

Abstract. In this work we address the Yelp Restaurant Photo Classification Challenge which consists in predicting restaurant attributes given its corresponding, variable-size set of images; the restaurant images were provided by Yelp and the labels were annotated by the Yelp Community for the 9 different attributes. The multi-instance and multi-label nature of the problem permits to explore a variety of ideas in the field of representation learning. First, we tackle the multi-instance aspect of the problem by means of aggregating pre-trained CNN feature extractors of a restaurant image-set to create a restaurant prototype feature vector. We then use the aggregated restaurant features to train a system of binary classifiers, one for each attribute. In order to improve our model performance, we induce a robust representation by means of calculating the restaurant prototype features through the use of complementary VGG-16 feature extractors pre-trained on 3 different datasets, namely: Imagenet, Food-101, and MIT Places 2. Due to the fact that not every representation has equal importance for predicting a particular attribute, we add a final classifier which learns a prediction weight for each representation of a given attribute. Our proposal is an end-to-end system, that achieves a test-dataset performance F1-score of 0.8177, which makes our model competitive within the top 10% entries for the challenge. Finally, some recommendations for improvement and future work are discussed.

Keywords: Convolutional neural networks, multi-instance learning, multi-label classification, representation learning.

1 Introducción

En años recientes, apoyándose en un incremento de las capacidades computacionales tales como GPUs, las tareas de visión artificial han sido dominadas por modelos de aprendizaje profundo (DL, por sus siglas en inglés) [1] y en particular por redes neuronales convolucionales (CNNs, por sus siglas en inglés). Los retos de reconocimiento visual a gran escala como el ILSVRC [2] han estimulado la competencia para proponer modelos convolucionales como la red VGG [3], la GoogleNet [4] y las Redes Residuales [5]. Éstos están constituidos por dos secciones: una sección convolucional, para la extracción de características en niveles jerárquicos de complejidad, y la sección de clasificación, donde se aprende la interrelación entre características.

Una ventaja de DL es que los modelos se entrenan automáticamente de principio a fin, evitando el uso de extractores de atributos diseñados por humanos. Esto es posible porque los extractores de atributos de bajo nivel son usados en capas superiores para construir atributos de alto nivel que son específicos para la base de datos en cuestión. Adicionalmente, los pesos aprendidos por estos modelos pueden ser usados en otros

problemas como detección de objetos y segmentación [6] u otros problemas de clasificación. Una técnica que permite el uso de pesos previamente entrenados sobre una base de datos distinta a la tarea en cuestión, es *transfer-learning* mejorada con *fine-tuning* [7]. Aún existen preguntas por contestar sobre el aprendizaje de representaciones, como qué tan buena es la representación optimizada para un problema, dadas distintas bases de datos de entrenamiento.

La mayor parte de los modelos CNN a gran escala están enfocados a la tarea de clasificación de imágenes. El reto de clasificación de restaurantes de Yelp (RCRY)[1], definido sobre una base de datos de restaurantes etiquetados, es un problema de clasificación de imágenes *multi-instancia* (el número de imágenes por restaurante no es fijo) y *multi-etiqueta* (existen 9 posibles atributos por seleccionar). Esto nos permite explorar ideas en aprendizaje de representaciones; en particular, estudiamos cómo inducir características robustas para los restaurantes a través de CNNs pre-entrenadas sobre distintos conjuntos de entrenamiento.

Siguiendo el trabajo en [8], abordamos dicha pregunta por medio de la agregación de características de alto-nivel obtenidas de una red VGG para generar un vector de características por restaurante. Esto se realiza con los extractores de características pre-entrenados en 3 bases de datos distintas. Una vez que los vectores de características por restaurante son calculados, se aplica el procedimiento de fine-tuning en cada uno de los módulos de clasificación de las redes VGG para predecir, por medio de un sistema de clasificadores binarios, si un atributo aplica o no para un restaurante.

Para mejorar el desempeño del sistema, aplicamos aprendizaje por enjambre usando una estructura con 4 pliegues, resultando en un total de 36 módulos de clasificación entrenados. Nuestra metodología posee la ventaja de que maximiza el aprendizaje que se transfiere utilizando 3 bases de datos complementarias, induciendo una representación robusta por restaurante; otra ventaja es la incorporación de un clasificador final que integra las capacidades predictivas de cada representación por etiqueta.

El resto del artículo está organizado de la siguiente manera: La sección 2 presenta antecedentes teóricos. La sección 3 proporciona una descripción de nuestra propuesta. El diseño de experimentos se describe en la sección 4. Nuestros resultados en el PCRY se reportan en la Sección 5. Conclusiones y direcciones para trabajo futuro se presentan en la Sección 6.

2 Antecedentes

En los problemas *multi-instancia* (MIL, por sus siglas en inglés), un número arbitrario de instancias está asociado con una etiqueta de clase. Por lo tanto, el etiquetado de los datos de entrenamiento se vuelve más sencillo (ya que se realiza en conjunto, en vez de manera individual) con la desventaja de que se produce una base de datos débilmente supervisada [9]. En el PCRY cada restaurante está representado por un conjunto de imágenes que comparten la(s) etiqueta(s) de atributos de dicho

[1] https://www.kaggle.com/c/yelp-restaurant-photo-classification

Fig. 1. Ejemplos de imágenes de entrenamiento para 3 restaurantes aleatorios (en filas).

establecimiento; algunos ejemplos de los conjuntos de entrenamiento se muestran en la Fig. 1. Es necesaria una función que relacione las etiquetas por instancia con las etiquetas por conjunto. Bajo la suposición estándar multi-instancia (SMIA, por sus siglas en inglés) un conjunto pertenece a la clase 'positiva' si al menos una de sus instancias es 'positiva' [10]. Siguiendo la SMIA, un problema MIL puede ser visto como un problema de clasificación de una sola instancia, al asignar las etiquetas del conjunto a las instancias asociadas a éste. Sin embargo, ésta no es la única suposición posible para los problemas MIL [11] y para nuestro problema no resulta ventajosa. Una alternativa fue propuesta en [6], donde se obtiene la representación de atributos de un conjunto agregando las características extraídas de las instancias asociadas a éste, y dicha representación agregada se convierte en un vector prototipo para el conjunto. La ventaja es que la red puede aprender cuáles de estas características son útiles para discriminar entre clases. En este trabajo seguimos este último esquema.

Además de ser un problema multi-instancia, el PCRY es también un problema multi-etiqueta. Un enfoque para abordar los problemas multi-etiqueta es entrenar tantos clasificadores como la cardinalidad del conjunto potencia de las etiquetas [12]. Así, un sistema podría aprender los detalles de cada posible configuración. Sin embargo, este enfoque podría conllevar a una severa falta de representantes de cada configuración, debido a su naturaleza exponencial: para nuestro problema particular implicaría entrenar $2^9 = 512$ clasificadores utilizando solamente 2000 vectores de atributos donde no todas las configuraciones están presentes y por lo tanto no pueden ser aprendidas. Un enfoque más simple es convertir el problema multi-etiqueta en un problema de una sola etiqueta [13]. Esto puede lograrse por medio de un sistema de n clasificadores binarios 'aplica' / 'no aplica', uno por cada etiqueta disponible. Este último enfoque es el que seguimos en este trabajo.

La transferencia de extractores de atributos de alto nivel, aprendidos sobre bases de datos a gran escala como ImageNet, pueden ser útiles cuando se aborda un problema

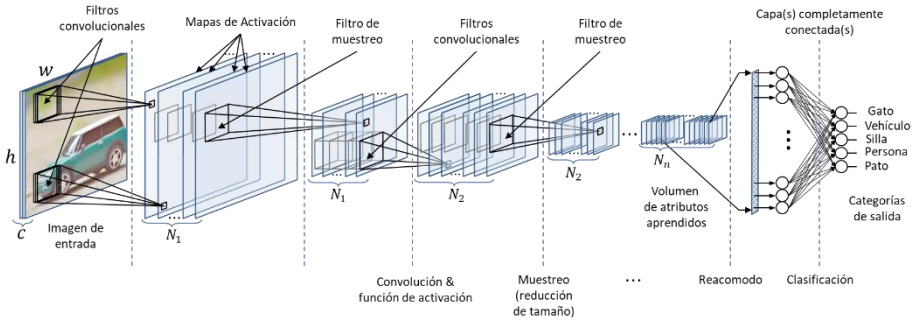


Fig. 2. Estructura de una Red Neuronal Convolucional para clasificación de imágenes.

similar de reconocimiento visual que presenta una cantidad limitada de datos de entrenamiento [6]. Uno de los primeros y más exhaustivos trabajos sobre la transferencia de parámetros aprendidos de una CNN es [14], donde los autores encontraron que la degradación del desempeño entre la tarea original y la nueva tarea esta dictada por la disimilitud entre ellas. Adicionalmente, llegaron a la conclusión de que los extractores de bajo nivel (primeras capas convolucionales) son genéricos en las tareas de visión artificial, mientras que los extractores de alto nivel (últimas capas convolucionales y capas completamente conectadas) son específicos a la base de datos de entrenamiento. En este trabajo, exploramos el uso de extractores de atributos previamente aprendidos, entrenados en 3 bases de datos distintas que consideramos son similares y complementarias a la base de datos del PCRY.

3 Propuesta

En esta sección se describe nuestra propuesta paso por paso; ésta trabaja bajo la suposición de que, usando diferentes bases de datos de entrenamiento, algunas de las características de alto nivel extraídas podrían ser más adecuadas para clasificar algún atributo particular de un restaurante. Por ejemplo, la etiqueta *outdoor seating* podría ser predicha con mayor precisión utilizando atributos aprendidos de una base de datos para clasificación de escenas; lo mismo se puede decir de las etiquetas *good for lunch* y *good for dinner* con respecto a una base de datos de alimentos. Nuestra metodología consiste en los siguientes pasos:

1. Aprovechamos las capacidades de representación de las arquitecturas CNN previamente entrenadas, al utilizar la técnica de transfer-learning sobre sus pesos. Para esta tarea decidimos utilizar la red VGG debido a que es la arquitectura CNN estándar con una precisión competitiva en las tareas de reconocimiento visual a gran escala (2^{do} lugar en la competencia ILSVRC 2014 [3]). La estructura de una CNN y una vista general de sus elementos se ilustran en la Figura 2. La red VGG-16 contiene 3 capas completamente conectadas (FC, por sus siglas en inglés), que comúnmente son referidas como: FC6, FC7 y capa de predicciones.

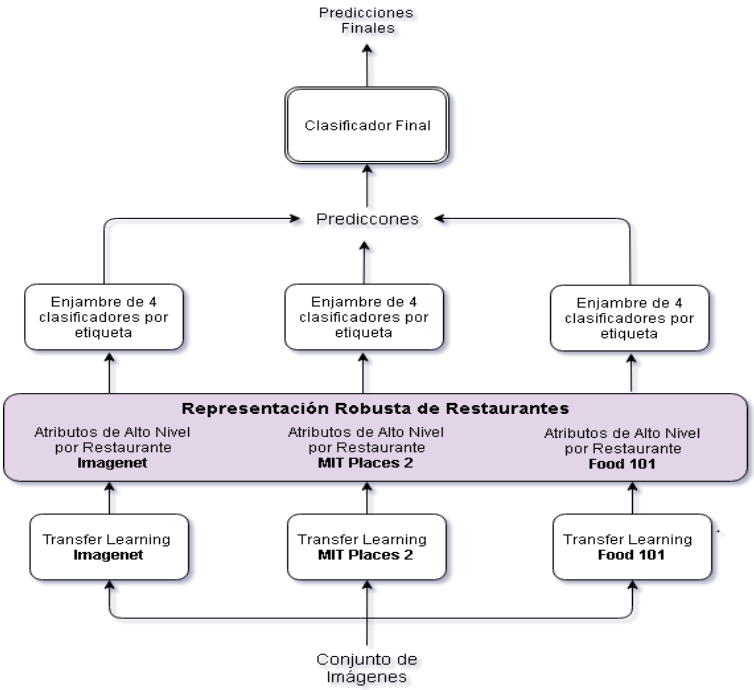


Fig. 3. Modelo propuesto para la clasificación de restaurantes.

2. Una vez cargados los pesos pre-entrenados, extraemos los atributos de alto nivel para cada imagen en la capa FC6. Escogimos la capa FC6 (atributos más específicos) en vez de la última capa convolucional (atributos más genéricos) como una forma de regularización, considerando que una gran proporción de los parámetros de la red VGG-16 conectan la última capa convolucional con la capa FC6 (100 millones de parámetros). Se asume que el aprendizaje de tantos parámetros utilizando una base de datos levemente supervisada conllevaría a un sobre-ajuste.

3. La representación extraída para la j-ésima instancia (imagen) del i-ésimo conjunto (restaurante) se denota como h_{ij}. De acuerdo a esta notación, la representación agregada de un conjunto está dada por: $\hat{h}_i = f(h_{i1}, h_{i2}, \dots, h_{in})$. La función f codifica el mapeo de los atributos a nivel instancia a los atributos a nivel conjunto. En este trabajo la función promedio $\hat{h}_i = avg_j(h_{ij})$ es utilizada para obtener el vector de atributos prototipo para el i-ésimo restaurante. Esta elección sigue la suposición de que activaciones similares se encontrarán presentes en los restaurantes que compartan una etiqueta de clase.

4. Una vez que se tienen los vectores de atributos por restaurante, procedemos a aplicar fine-tuning sobre la capa FC7 y la capa de predicción. Los pesos en la capa FC7 se inicializan con los valores óptimos aprendidos para la tarea original; la capa de

```
# Training of Representation-Dependent Binary Classifiers
for dataset ∈ {ImageNet, Places, Food-101} do:
    DBFC ← aggregated bag features, per restaurant
    shuffle the businesses index BI
    Instances n ← number of restaurants / number of folds
    for fold ∈ {0, 1, 2, 3} do:
        folds[fold] = BI[n×fold to n×(fold + 1)]
    end for
    for attribute j ∈ {0, 1, …, 8} do:
        for validation fold k ∈ {0, 1, 2, 3} do:
            for fold ∈ {0, 1, 2, 3} do:
                if fold is not k do:
                    append folds[fold] to train_idx
                end if
            val_idx ← folds[k]
            end for
            Train classifier_{j,k} on DBFC[train_idx]      // DBFC: Dataset Business FC
            ValProbs_{j,k} ← predict probabilities using val_idx
            ValClasses_{j,k} ← predict classes using val_idx
        end for
        datasetValProbs_j ← concatenate the ValProb_{j,k} on k axis
        datasetValClasses_j ← concatenate the ValClases_{j,k} on k axis
    end for
    datasetValProbs ← concatenate datasetValProbs_j on j axis
    datasetValClasses ← concatenate datasetValClasses_j on j axis
end for
Calculate validation accuracy with datasetValClasses
valProbs ← concatenate datasetValProbs on dataset
```

Fig. 4. Pseudocódigo del clasificador binario para cada representación

```
# Final classifier
valTargets ← get the restaurant target attributes
for validation fold k ∈ {0, 1, 2, 3} do:
    for fold ∈ {0, 1, 2, 3} do:
        if fold is not k do:
            append folds[fold] to train_idx
        end if
    end for
    val_idx ← folds[k]
    Train FinalClassifier_k on ValProbs[train_idx]
    Evaluate FinalClassifier_k on ValProbs[val_idx]
end for
Get the validation accuracy for the final classifier
```

Fig. 5. Pseudocódigo del clasificador final

predicción se modifica para contener una sola unidad sigmoide y así producir un clasificador binario por atributo.

5. Finalmente, ya que no toda representación es igualmente útil en la predicción de una etiqueta, implementamos un clasificador final con 9 neuronas sigmoides de salida y 27 unidades de entrada (9 por cada representación). De esta manera, añadimos el

módulo final a un sistema automático de principio a fin que evita el uso de heurísticas humanas para seleccionar los pesos apropiados de cada representación para una etiqueta dada.

Para el ajuste de hiper-parámetros se utilizó validación cruzada de 4 pliegues, que adicionalmente nos permite utilizar técnicas de aprendizaje por enjambre al entrenar 4 clasificadores por etiqueta. Las salidas de estos clasificadores se promedian para producir un vector de 9 probabilidades, una para cada uno de los 9 atributos.

Con el propósito de obtener una representación robusta de los atributos de un restaurante, los pesos aprendidos de diferentes tareas se utilizan para inicializar la arquitectura VGG-16. Las tareas elegidas son: ImageNet, con 1000 clases y más de un millón de imágenes de entrenamiento [2], MIT Places-2 con 365 clases de escenas naturales y humanas [15], y Food-101 con 101 clases de alimentos [16]. Por lo tanto, los pasos 1-4 deben repetirse por cada representación utilizada. Una representación gráfica de nuestro modelo se presenta en la Figura 3.

4 Diseño de experimentos

Los pesos entrenados de las bases de datos ImageNet y MIT Places 2 se descargaron de Caffe Model Zoo[2]. Para Food-101 no existen esos pesos, así que aplicamos *fine-tuning* sobre los pesos de ImageNet usando Food-101 hasta que la pérdida de validación convergiera.

Los folders de validación se usaron para el ajuste de híper-parámetros durante el entrenamiento de los clasificadores binarios. Se seleccionó una tasa de aprendizaje (lr) con una alta reducción de la perdida de validación durante las épocas iniciales que disminuye considerablemente conforme progresa el entrenamiento. Dado el poder de representación de la red y el potencial de sobre-ajuste, regularizamos los clasificadores con un $dropout\_rate = 0.5$ para los atributos de entrada y las activaciones FC7. Los híper-parámetros usados son: $lr = 10^{-6}$, $épocas = 100$. Estos parámetros se mantuvieron fijos para todos los clasificadores entrenados. El pseudocódigo para el entrenamiento de los clasificadores binarios se muestra en la Fig. 4 y el del clasificador final se muestra en la Figura 5.

El proceso de ajuste sobre los pesos en las capas FC es el siguiente: entrenar los pesos en la capa de predicción mientras se mantienen los pesos de FC7 fijos. Una vez que la pérdida de validación se estabiliza, los pesos de FC7 se descongelan.

El objetivo es obtener una mejor representación mediante los atributos de entrada FC6 en la capa FC7. Elegimos Adam como optimizador, pues se compara favorablemente a otros optimizadores [17]. La métrica oficial para el reto es la puntuación F1, definida para los C posibles atributos de los restaurantes basándose en las métricas de $Precisión$ y $Recall$ que se calculan como se muestra a continuación:

$$Precisión = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}, \quad (1)$$

[2] https://github.com/BVLC/caffe/wiki/Model-Zoo

$$Recall = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k}, \tag{2}$$

donde TP: Verdaderos Positivos, FP: Falsos Positivos y FN: Falsos Negativos. La $F1\ Score$ es la media harmónica entre $Precisión$ y $Recall$:

$$F1\ Score = \frac{2 \times Precisión \times Recall}{Precisión + Recall}. \tag{3}$$

5 Resultados

Primero presentamos los resultados de precisión en el conjunto de validación obtenidos para cada representación de manera individual: ImageNet, MIT Places 2, Food-101 y para el clasificador final, posteriormente presentamos los resultados sobre el conjunto de evaluación, obtenidos de la página web de la competencia.

Las Figuras 6-8 muestran la precisión de validación, promediada para los 4 pliegues, durante la fase de entrenamiento; la línea solida indica pesos congelados en la capa FC7, la línea punteada indica que todos los parámetros de la red son libres de actualizarse. Además, nótese que la caja de etiquetas esta ordenada de manera descendente por precisión de validación: de arriba-abajo, izquierda-derecha.

Una de las suposiciones al usar características de entrada de más bajo nivel FC6 (en vez de FC7) es que podrían conllevar a una mejor representación en la capa oculta FC7 mediante el uso de la técnica fine-tuning; sin embargo, observamos que el uso de las características extraídas en la capa FC7 (representados por la línea solida) no afecta la precisión de validación. Conjeturamos que esto es porque el conjunto de entrenamiento no es suficientemente grande para entrenar tantos parámetros y el optimizador no induce una mejor representación.

La precisión de validación por representación y para el clasificador final se muestra en la Tabla 1. Analizando la tendencia general de las representaciones individuales observamos que la precisión de validación para Food-101 es más alta en casi todos los atributos, un fenómeno similar había sido observado en el trabajo de [14].

Debido a que aplicamos fine-tuning sobre los pesos de ImageNet para la tarea de reconocimiento de alimentos, los extractores de atributos se volvieron más robustos, incorporando conocimiento de las 1000 clases originales de ImageNet y la tarea objetivo que contenía 101 variedades de alimentos.

Otra tendencia que persiste es el menor desempeño en los clasificadores entrenados con la representación de MIT Places 2. La hipótesis es que esto es debido a la discrepancia entre las tareas: en el PCRY los conjuntos de imágenes consisten principalmente en imágenes de alimentos y raramente en escenas naturales o humanas. A pesar de que asumimos que la etiqueta '*outdoor seating'* podría hacer uso de atributos encontrados en MIT Places 2, notamos que consistentemente esta etiqueta es la más difícil de predecir debido a que contiene poca, e inclusive ambigua, información visual sobre si la etiqueta aplica o no. La predicción del atributo '*has table service*' es considerablemente mejor en todas las representaciones, en particular con las representaciones de ImageNet/Food-101. Esto tiene sentido por dos razones: en primer

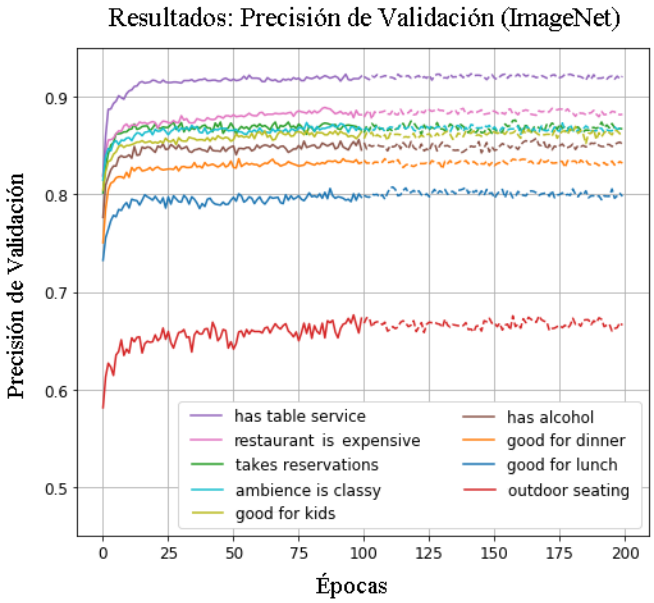


Fig. 6. Resultados (Validación) Precisión para la Representación ImageNet.

Tabla 1. Precisión de Validación para las Representaciones Individuales; el mejor desempeño se muestra en negritas.

	Representación			
Etiqueta	**ImageNet**	**MIT Places**	**Food 101**	**Clasificador Final**
Good for lunch	80.2%	79.8%	81%	**81.6%**
Good for dinner	82.6%	83.6%	**84.2%**	83.7%
Takes reservations	87.1%	86.8%	87.3%	**88%**
Outdoor seating	65.6%	65.7%	68.1%	**69.4%**
Restaurant is expensive	87.5%	87.8%	**89.2%**	89%
Has alcohol	84.6%	84.8%	85.4%	**86.4%**
Has table service	91.9%	89.3%	92%	**92.4%**
Ambience is classy	**86.6%**	85.4%	85.9%	86.5%
Good for kids	85.9%	85.8%	86.7%	**87.2%**
Desempeño	83.6%	83.2%	84.4%	**84.9%**

lugar existe una correlación consistente entre la etiqueta y la presencia de menús lo que la convierte en una clase fácil de predecir, además, la clase menú forma parte de las 1000 clases originales de ImageNet. Pertinente a esta misma idea es que aun con el fine-tuning sobre la representación original de ImageNet, sus capacidades de representación en Food-101 no se perdieron, lo que nos lleva a concluir que la capacidad de aprendizaje de VGG-16 excede a la base de datos Food-101 y por lo tanto la mayoría de los extractores de atributos mantuvieron sus pesos originales.

En los resultados para el clasificador final observamos que el desempeño por atributo, en casi todos los casos, es tan bueno como el mejor desempeño de las

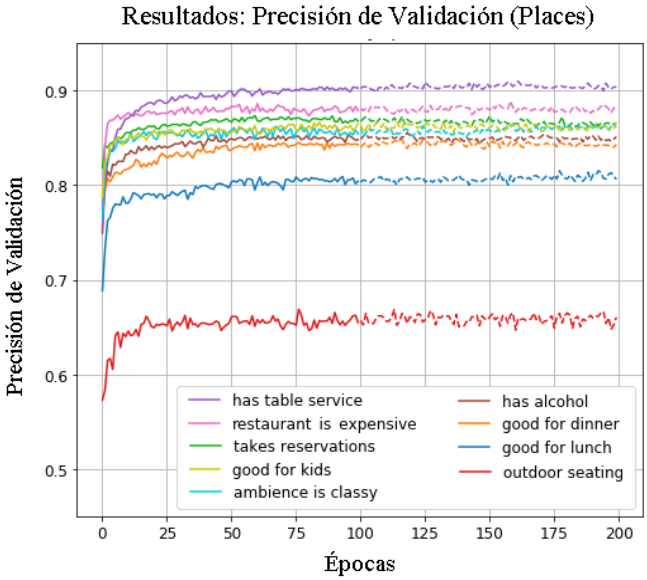


Fig. 7. Resultados (Validación) Precisión para la Representación MIT Places 2.

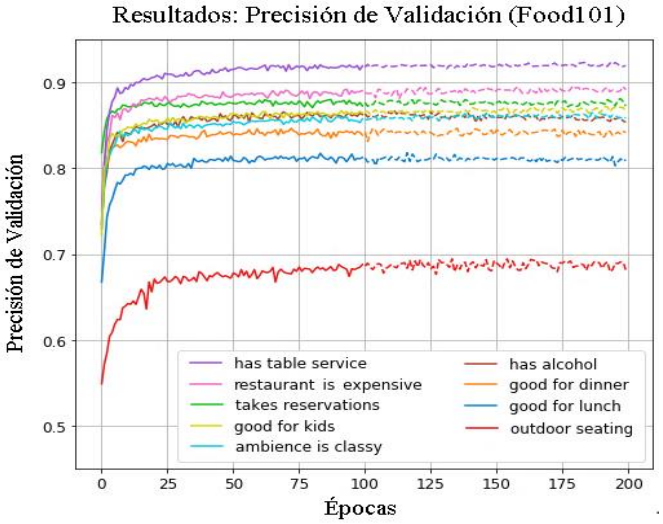


Fig. 8. Resultados (Validación) Precisión para la Representación Food-101.

representaciones individuales. De un análisis detallado de los pesos, observamos que el clasificador final esta asignando más importancia a aquella representación que presenta un mejor desempeño en la precisión de validación. La única etiqueta en la que el desempeño del clasificador final es significativamente peor es 'good for dinner'.

Pensamos que este es el caso por la fuerte dependencia del atributo respecto de las imágenes de comida, y de manera específica, de imágenes de platillos caros, como platos de porcelana (una clase que no está representada en ImageNet/MIT Places 2 pero que aparece en Food-101 como un atributo útil).

Una vez que el enjambre de clasificadores por atributo y el clasificador final fueron entrenados, aplicamos el modelo completo en los restaurantes de prueba, y enviamos las predicciones a la página de la competencia.

Dos resultados fueron obtenidos: una puntuación F1 pública de 0.8091 (en 30% de los datos de prueba), y una puntuación F1 privada de 0.8177 (en 70% de los datos de prueba). Este resultado coloca a nuestro modelo en el 10% superior de los competidores, donde la mejor puntuación fue de 0.83177. La principal ventaja de nuestro sistema es su simplicidad, apoyada por un uso uniforme de redes neuronales que son sistemas automatizados de principio a fin. El resultado final es un sistema modular y de propósito general para abordar tareas de clasificación multi-instancia y multi-objetivo.

6 Conclusiones

Observamos que la representación de características por restaurantes, obtenida mediante la agregación de características de alto nivel de su conjunto de imágenes, es útil cuando se aborda un problema multi-instancia debido a la conformidad con la suposición de que ciertas características con magnitud similar aparecen para restaurantes con una misma etiqueta de clase. La representación de alto nivel influencia de manera directa el desempeño de cada clasificador de atributos, esto se ejemplifica claramente con la etiqueta '*has table service'*, donde los pesos aprendidos mediante la base de datos original extraen atributos de objetos que están fuertemente correlacionados (a saber, los menús) con la etiqueta. La etapa de fine-tuning adicional con la base de datos Food-101, utilizando pesos pre-entrenados con ImageNet, resultó útil para mejorar el desempeño de la representación, debido a su habilidad para la discriminación de las clases en ambas bases de datos.

Mientras más pesos de redes CNN pre-entrenadas se hagan disponibles esperamos lograr un mejor desempeño utilizando modelos de principio a fin que extraen el desempeño óptimo por representación.

Aunque se necesita más evidencia y argumentación, consideramos que la presente metodología se podría extender y aplicarse en otras tareas de clasificación. Como trabajo futuro contemplamos la inclusión de un esquema de aumentación de datos con la finalidad de eludir uno de los factores principales que limitaron el desempeño de nuestro modelo.

Agracecimientos. Este trabajo se llevó a cabo gracias al auspicio del Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, a través de los apoyos 604421 (J. Veloz) y CÁTEDRAS-2598 (A. Rojas). Los autores agradecen a Yelp por hacer la base de datos PCRY disponible públicamente y a Kaggle por hospedar la competencia correspondiente.

Referencias

1. LeCun, Y., Bengio, Y.; Hinton, G.: Deep learning. Nature Research, 521(7553), pp. 436–444 (2015)
2. Russakovsky, O., Deng, J., Su, H., Krause, J., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), pp. 211–252 (2015)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. Szegedy, C., Liu, W., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv: 1512.03385 (2015)
6. Wu, J., Yu, Y., Huang, C., Yu, K.: Deep multiple instance learning for image classification and auto-annotation. In: Proceedings of the IEE Conference on Computer Vision and Pattern Recognition, pp. 3460–3469 (2015)
7. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1717–1724 (2014)
8. Baan, J: A Deep Learning Ensemble approach to the Yelp Restaurant Classification Challenge. (2016)
9. Dietterich, T. G., Lathrop, R. H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1), pp. 31–71 (1997)
10. Foulds, J., Frank, E.: A review of multi-instance learning assumptions. The Knowledge Engineering Review, 25(1), pp. 1–25 (2010)
11. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML`07), pp. 406–417 (2007)
12. Tsoumakas, G., Katakis, I., Vlahavas, I.: Data mining and knowledge discovery handbook. (2009)
13. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks?. In: Ghahramani, Z., Welling, M., et al. (eds), Advances in neural information processing systems, pp. 3320–3328 (2014)
14. Zhou, B., Lapedriza, A., Khosla, A, Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
15. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101: Mining discriminative components with random forests. In: European Conference on Computer Vision (2014)
16. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

ISSN 1870-4069

Análisis del comportamiento de diferentes algoritmos de aprendizaje automático para catalogar delitos en la zona metropolitana

Belém Priego Sánchez, Stephany Anaya García, José A. Reyes-Ortiz

Universidad Autónoma Metropolitana unidad Azcapotzalco,
Departamento de Sistemas,
México

{abps,jaro}@azc.uam.mx, stephany.anaya@hotmail.com

Resumen. Actualmente nos encontramos en la era de la información en donde se produce, se difunde y se emplea gran cantidad de ésta diariamente, la cual puede ser estudiada y analizada con el fin de obtener nueva información. Aunado a este hecho se encuentran las redes sociales en las cuales se puede expresar una idea, concepto, opinión o evento. El análisis de las ideas, comentarios de Twitter, por ejemplo, escritas por la comunidad a despertado el interés de muchos investigadores. En este artículo, se está interesado en catalogar información, de delitos ocurridos en la zona metropolitana, reportada en Twitter a partir de tres algoritmos de clasificación. Teniendo como objetivo presentar los resultados obtenidos tras su ejecución; estos resultados evidencian que si se tienen los parámetros correctos se puede clasificar y obtener nueva información a partir de una ya existente.

Palabras clave: Aprendizaje automático, algoritmos de clasificación, textos cortos, delitos.

Analysis about the Behavior of a Different Machine Learning Algorithms in order to Catalog Crimes in the Metropolitan Area

Abstract. Nowadays we live in the information era where a massive information is produced, disseminated and used daily, which can be studied and analyzed in order to obtain new information. In addition, we have the social networks, in which you can express an idea, concept, opinion or event. Twitter opinions analysis - twitter comments, for example- written by the community has awake the interest of many researchers. Therefore, this paper is focused on classify the information about crimes that occurred in the metropolitan area of Mexico City, reported on Twitter from three types of classification algorithms. Having in mind to present the results obtained after their execution, we observed that when the correct parameters are set, then it is possible to classify and to obtain new information from a preexistent one.

Keywords: Machine learning, classification algorithms, short texts, crimes.

1. Introducción

Indudablemente, nos encontramos en la era de la información, en donde se produce, difunde y emplea información en abundantes cantidades diariamente. El estímulo que originó esta era se dio con el auge de las tecnologías de la información y la comunicación. Un campo encargado del estudio de dicha información es el Procesamiento del Lenguaje Natural, denotado de aquí en adelante por PLN, gracias a que proporciona un análisis de los textos convenientes para detectar, recuperar y computar información subjetiva de forma metódica. Dentro del PLN se encuentra el área de la Minería de Textos la cual, actualmente, ha despuntado debido a la creciente demanda en el uso de las redes sociales. Oportunamente, las empresas y otras instituciones se han beneficiado de ello, estudiando y examinando las colosales cantidades de datos que se producen y difunden por los usuarios a través de Internet, esto diferenciando la composición del sentimiento en un texto.

La detección de las opiniones expresadas por los usuarios de las redes sociales, brinda la facultad de responder interrogantes en un gran número de dominios de aplicación. Así mismo, dentro de las redes sociales emergen los denominados *influencer*, los cuales son individuos con características y aptitudes sociales favorables, que contribuyen en la toma de decisiones de un gran número de personas, guiados por las opiniones de estos mismos. Cuando se origina y difunde una opinión pública, en la que la cual puede estar en actuación diferentes circunstancias, desde el comportamiento en la bolsa de valores de una empresa o bien la toma de decisiones en la política de un país, se analiza el comportamiento de estas opiniones dentro de las redes sociales que brinda la capacidad de informar acerca de sucesos o acontecimientos venideros. Sin embargo, el ordenamiento y estructuración de los textos capturados, resultaría irrealizable para el ser humano, de tal suerte que se cuenta con algoritmos, el propósito de disponer de estos algoritmos es conjeturar la condición objetivo, mediante el análisis del conjunto de datos capturados. Es decir, etiquetar y organizar con una valoración la información difundida en el texto, el cuál es la labor que se llevará a cabo en este trabajo de investigación.

Los algoritmos con los cuales se disponen, principalmente, son los de clasificación y agrupamiento (*agrupamiento*). Los algoritmos de clasificación persiguen predecir la clase objetivo por medio del análisis del conjunto de datos de entrenamiento. Por otro lado, los algoritmos de agrupamiento aspiran a reunir los elementos haciendo uso de diferentes medidas de afinidad o similitud. Es por ello que el aprendizaje automático se apoya de estos algoritmos, que se destinan para evolucionar los métodos de procesamiento y posibilitan a las computadoras a aprender a predecir acontecimientos en los cuales se involucran distintos métodos matemáticos. Al situar estos métodos a la práctica se instruyen de los modelos de entrenamiento para realizar su predicción.

En esta investigación se realiza una comparativa de tres algoritmos de aprendizaje automático, nos apoyamos en los algoritmos de clasificación y agrupamiento. Se selecciona, analiza y compara determinados modelos algorítmicos, con la finalidad de identificar la precisión y exhaustividad de estos algoritmos dentro de un mismo caso de estudio. El caso de estudio en el que se implementarán los algoritmos, será enfocado en clasificar seis tipos de delitos en el área metropolitana de México que de acuerdo con el Instituto Nacional de Estadística y Geografía [1] son los incidentes que más ocurren en México. Estos son reportados por diversas fuentes de información oficiales vía Twitter. Los *tweets* que contienen la información de los delitos fueron etiquetados manualmente, previamente, de acuerdo a la categoría a la cual pertenecen. Las categorías de etiquetado son: Homicidio, Suicidio, Asalto, Violación, Explotación y Secuestro. Con la elaboración de esta investigación se señala y demuestra el algoritmo con mayor precisión y exhaustividad dentro del análisis de textos orientado a los tweets que contengan esta información.

El resto del artículo se presenta como sigue: en la Sección 2 se da una breve motivación del porque realizar este trabajo, en la Sección 3 se describen algunos de los trabajos reportadosen el estado del arte, en la Sección 4 se presenta la metodología propuesta que da solución a la temática del artículo y en la Sección 5 se presentan los resultados obtenidos tras ejecutar dicha metodología. Fianlmene, se presentan las conclusiones obtenidas.

2. Motivación

En este trabajo de investigación se está interesado, principalmente, en analizar el comportamiento de los algoritmos de clasificación y agrupamiento, orientados a la clasificación automática de los seis delitos (homicidio, suicidio, asalto, violación, explotación y secuestro) en el área metropolitana de México. Este análisis nos permitirá determinar el mejor algoritmo en cuanto a precisión y exhaustividad.

El objetivo de realizar un estudio de diferentes algoritmos de clasificación es debido a que en la actualidad el análisis de los textos difundidos a través de las redes sociales, contribuye en la predicción de sucesos o bien permite modelar situaciones. Giovanni Cherubini, Jens Jelitto y Vinodh Venkatesan su artículo prospectivo *Cognitive Storage for Big Data* [2] proponen la noción de un sistema de "almacenamiento de datos cognitivos", en el cual se habitúa a criterios como la estimación y obsolescencia de los datos. Con la meta de desarrollar un algoritmo de aprendizaje que contribuya a la clasificación de los datos, en varias clases de relevancia y trabajar en conjunto con una arquitectura de almacenamiento multinivel para personalizar la ubicación de los datos.

El almacenamiento de los datos de forma cognitiva considera y reúne los datos, los cuales contribuyen al procedimiento de análisis de estos mismos. De esta manera, esta investigación contribuirá a sealar el algoritmo optimo a efectuar en la situación donde se cuente con datos afines a los del caso de estudio manifestado. Así mismo, se está interesado en contribuir en el análisis predictivo

de los seis principales delitos del área metropolitana de México. Lo cual, además, permitirá modelar la situación delictiva en esta región del país.

3. Estado del arte

En esta sección, se describen los trabajos reportados en la literatura relacionados con diferentes algoritmos de aprendizaje automático para el análisis de opiniones. Si bien es cierto que existen bastantes investigaciones que se dedican a la comparativa de diferentes algoritmos de clasificación, en este apartado hemos seleccionado algunas de las muchas investigaciones. Esto debido a que el objetivo es mostrar que esta temática es un punto de referencia clave a la hora de utilizar diferentes algoritmos y que es una investigación atrayente a los investigadores dedicados al análisis de opiniones.

En [8] se proporciona una pesquisa sobre los desafíos y la visión general de algunos algoritmos de clasificación y agrupación utilizados para el análisis sentimental y de la minería de opiniones. La similitud, entre la propuesta y dicho artículo, radica en la intensión de los algoritmos, sin embargo, difiere en la implementación y las herramientas de software a utilizar. En [7] se aborda el tema de minería de opiniones y análisis de sentimientos, la cual es una tarea de procesamiento de lenguaje natural e información que identifica las opiniones de los usuarios explicadas en forma de comentarios positivos, negativos o neutrales y citas subyacentes al texto. La similitud radica al utilizar distintos algoritmos supervisados o basados en datos. No obstante, en este caso se realiza para el análisis de sentimientos y también considera la precisión de la clasificación del sentimiento. En [6] se describe un trabajo de estudio sobre la eficiencia del lenguaje R en la minería de opiniones, la similitud de este artículo con la propuesta presentada radica en el uso del lenguaje R así, mismo, como el uso de un corpus extraído de Twitter. No obstante, difiere en la aplicación de algoritmos de aprendizaje automático.

En el trabajo [3] se aplicaron técnicas de PLN e implementaron cuatro diferentes procesos de Minería de Textos. La principal afinidad que se tiene con esta investigación y con el que se propone en esta propuesta, es el uso de diferentes algoritmos como máquinas de soporte vectorial (SVM, por sus siglas en inglés), Naïve Bayes, J48 y K-vecinos más cercanos para realizar la minería de texto y se compararán los resultados generados. Sin embargo, la diferencia radica en el uso del software Weka para implementar dichos algoritmos, mientras que en el caso de esta propuesta se implementaron en el software R. En la propuesta [4] se aplicaron tres algoritmos agrupamiento a los genes activos e inactivos, asociados con tumores pediátricos de 60 pacientes para poder determinar con ello que tipo de tumores se presentaba en cada paciente. La similitud con el presente trabajo se tiene en la aplicación de algoritmos de agrupamiento y el uso del software R, este trabajo es muy similar al que presentamos pero los objetivos planteados, resultados y los datos empleados son algunas de las características diferentes que se tienen. Además del análisis y resultados completamente distintos. En [5] se diseñó un sistema para la clasificación de artículos científicos en idioma

inglés, utilizando el formato PDF y se presentaron los resultados por medio de la tecnología Web Java Servlets. La similitud con este trabajp es la clasificación de un conjunto de datos, utilizando el algoritmo "K-Means", mientras que la diferencia radica en los datos a emplear. De igual forma, existe una diferencia con las herramientas a utilizar.

En la siguiente sección, se describe la metodología propuesta que dará solución al análisis de los diferentes algoritmos que se utilizarán para la tarea de clasificación automática.

4. Metodología propuesta

El objetivo de este trabajo, de investigación, es el de clasificar seis tipos de delitos. Para lograr dicha meta, se propone una metodología que está conformada de cuatro etapas principales.

1. *Agrupamiento de los tweets.* El objetivo, de esta etapa, es conformar la colección de textos, éstos formarán el corpus. Partiremos de un aproximado de 2,500 tweets que han sido, previamente, etiquetados manualmente de acuerdo a la etiqueta tipo de delito. Ésta corresponde al delito que el tweet tiene.
2. *Preprocesamiento de los tweets.* Se cubrirá, principalmente, el análisis gramatical de la oración para leer el texto, es decir, se analizará el cómo están formados los tweets. De la misma manera, se analizan dichos textos por estructuras.
3. *Minería de textos.* Se extrae información utilizando diferentes herramientas, este proceso encuentra las similitudes entre los datos que tiene el mismo significado y así poder obtener información sobre ellos. Dentro de esta etapa, se tienen dos subetapas. Con respecto a la primera etapa, "Aplicación del algoritmo", se tienen diferentes procesos. En la Figura 1, se muestran de forma gráfica estos procesos, los cuales están definidos de forma lineal.
   Las subetapas que se llevan a cabo son:
   a) Aplicación del algoritmo. Los algoritmos que serán implementados son: *K-Means, Naïve Bayes* y *k Nearest Neighbor*. Los procesos, o pasos, por los cuales se deberá someter a cada algoritmo son:
      1) Selección del algoritmo.
      2) Construcción del modelo.
      3) Implementación del modelo.
      4) Generación del modelo y coordinación.
      5) Análisis.
   b) Modelado de la información. En este proceso, se muestra de forma gráfica la información que los algoritmos suministren, de tal forma que el análisis de esta información sea más comprensible para la comparación de los algoritmos.

En la siguiente sección, se presentan los resultados obtenidos tras llevar a cabo la metodología propuesta. Los experimentos se ejecutaron mediante el entorno de R[9].

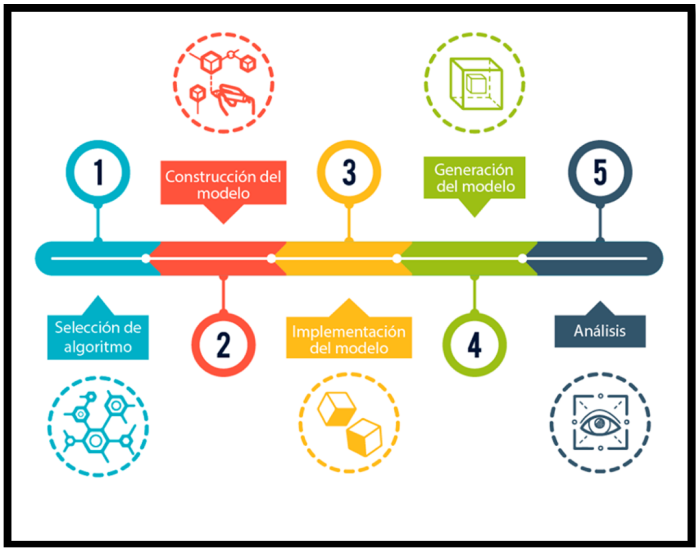


Fig. 1. Diagrama del proceso de análisis por cada algoritmo.

5. Resultados experimentales

Con el fin de mostrar los resultados obtenidos a lo largo del desarrollo de esta investigación, se ha divido esta sección en tres subsecciones principales: a)Conjunto de datos, b)Algoritmos empleados y c)Resultados obtenidos. A continuación, se describe cada una de éstas.

5.1. Conjunto de datos

En esta subsección se describe el conjunto de datos, utilizado y construido, denominado "Eventos Crimen México". Este conjunto de datos corresponde a encabezados (*headlines*) de noticias y tweets en Español sobre seguridad. Tanto los encabezados como los tweets provienen de periódicos electrónicos mexicanos como: "El universal, Excelsior, La Jornada, Noticias MVS, Reforma" y sus respectivas cuentas en Twitter.

En este corpus de noticias se tienen 1,500 encabezados de periódicos mexicanos y 1,500 tweets, los cuales fueron recolectados entre el 17 de noviembre del 2017 y el 18 de enero de 2018. Cada cabecera de noticia está compuesta de 35 palabras en promedio y cada mensaje de Twitter de máximo 140 caracteres que es lo que permite la red social.

El conjunto de datos está dividido en seis categorías: asalto, trata de personas, homicidio, suicidio, secuestro y violación. Para obtener las categorías se llevo a cabo un proceso de anotación, el cual consistió en asignar una etiqueta al encabezado o tweet de acuerdo a la categoría del evento. Este etiquetado fue realizado por humanos, quienes leen el encabezado de la noticia o tweet y

determinan la categoría. Una misma cabecera de una noticia o un tweet es asignado(a) a dos etiquetadores; si estos anotadores no se ponen de acuerdo entonces se realiza un proceso de decisión que consiste en que dicha cabecera se le asigna a un nuevo etiquetador y éste es quien decide la categoría final.Una vez finalizado este proceso, los 1,500 encabezados y 1,500 tweets han quedado distribuidos en seis categorías, mencionadas anteriormente, de acuerdo a la Tabla 1.

Tabla 1. Distribución de encabezados de noticias y tweets del conjunto de datos utilizado.

Categoría	Número de encabezados	Número de tweets	Total de textos
Asalto	346	386	732
Secuestro	210	218	428
Violación	185	164	349
Trata de personas	195	180	375
Homicidio	365	366	731
Suicidio	199	186	385

A manera de ejemplo, en la Figura 2 se muestra la estructura de un tweet para la categoría *homicidio*. En el caso de las demás categorías, se sigue el mismo diseño y patrón de formato xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<tweets>
<tweet id="935272038603870213"> La fiscal Yendi Torres Castellanos <evento tipo="Homicidio">fue asesinada</evento>
                                en <espacio>Pánuco #Veracruz.</espacio> https://t.co/0si97IsW5b </tweet>
</tweets>
```

Fig. 2. Estructura de un encabezado y de un tweet.

5.2. Algoritmos empleados

Las técnicas de aprendizaje automático supervisado son capaces de aprender el proceso humano para clasificar delitos con base en las características alimentadas, al clasificador, y los parámetros que se les pueden asignar. Con el fin de tener una perspectiva del tipo de clasificador que puede tratar mejor el problema de clasificación de delitos, se han seleccionado los siguientes tres algoritmos de aprendizaje:

1. *K-Means*: es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.
2. *Naïve Bayes*: es un clasificador probabilístico basado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales.

3. *K-Nearest Neighbor*: el método de los k vecinos más cercanos es un método de clasificación supervisada que sirve para estimar la función de densidad $F(x/C_j)$ de las predictoras x por cada clase C_j.

Los resultados que se obtuvieron al tratar de clasificar delitos en la Zona Metropolitana son presentados en la siguiente subsección.

5.3. Resultados obtenidos

Para llevar a cabo la ejecución de los algoritmos seleccionados, K-Means, Naïve Bayes y K-Nearest Neighbor, se han empleado diferentes parámetros. En esta sección se presentan los resultados óptimos, es decir, los mejores resultados que se obtuvieron al variar los parámetros.

En el caso de la clasificación con el algortimo de K-Means, en la Figura 3 se presenta un enfoque con un vali de $k = 3$ que corresponde al número de agrupamientos en los cuales se van a agrupar los tweets; éstos son el resultado de la agrupación de acuerdo a la parametrización de los tweets. Se presenta esta figura, debido a que con tres agrupamiento se observó que al parametrizar dichos tweets, los elementos a clasificar (tweets) son mejor agrupados a diferencia de la utilización de más o menos agrupamientos.

Además, en la Figura 4 se presenta de forma gráfica el dendograma que corresponde al agrupamiento de la calsificación de los tweets. Como puede observarse en la misma figura, Figura 4, las palabras más frecuentes que se encuentran en el corpus de tweets han sido agrupadas de acuerdo a la frecuencia de dichas palabras. Teniendo como resultado que las palabras *asesinato* y *robo* son de las palabaras con más repeticiones en el corpus y se realiza el agrupamiento de forma individual. Realizando una comparativa con el conjunto de palabras restantes, *sexual, año, hombre, cdmx, policía, dos, mujer* y *detienen*, se agrupan de manera que conforman un sólo agrupamiento.

Con respecto, al algoritmo de clasificación Naïve Bayes el conjunto de datos utilizado correspondió al 70 % para el entrenamiento y 30 % para pruebas, tomando una frecuencia de términos igual a treinta, lo cual significa que el algoritmo toma los términos con treinta repeticiones. En la Tabla 2 se muestra la matriz de confusión, tras ejecutar dicho algoritmo, la cual representa la manera en la que fueron agrupados los tweets y la predicción que tuvo el algortimo. La exactitud, accuracy, del algoritmo correspondió a un 87.35 %, el cual corresponde a un resultado de agrupamiento aceptable. Estos resultados se muestran de manera gráfica en la Figura 5, dicha figura categoriza los tweets de acuerdo a los delitos que se tienen.

Los resultados obtenidos con el algoritmo, de clasificación sueprvisada, K-Nearest Neighbor se han obtenido mediante los vecinos considerados es igual a tres. Debido a que es el modelo que más eficiencia tiene al clasificar los comentarios, de Twitter, dentro de la categoría a la cual pertenecen. En la Tabla 3, muestra la matriz de confusión, se permite visualizar el grado de efectividad de dicho algoritmo. Además, es importante mencionar que la exactitud fue del 84.36 %. Si realizamos una comparación con el mejor resultado obtenido, 87.35 %

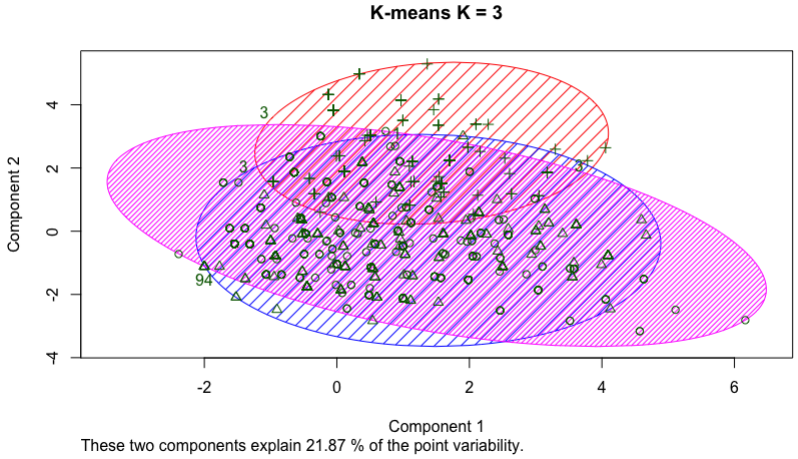


Fig. 3. Clasificación mediante el algortimo K-Means en tres agrupamientos de los delitos.

Tabla 2. Matriz de consfusión al ejecutar el algoritmo Naïve Bayes.

Predicción	Asalto	Explotación	Homicidio	Secuestro	Suicidio	Violación
Asalto	344	0	19	1	4	1
Explotación	1	0	0	0	0	0
Homicidio	23	5	388	12	14	25
Secuestro	0	0	0	7	0	0
Suicidio	1	0	7	1	12	3
Violación	1	1	0	0	0	71

resultado con Naïve Bayes, se puede observar que no existe una alta dispersión entre el resultado obtenido con K-Nearest Neighbor, sin embargo, existe.

5.4. Discusión

Cada uno de los algoritmos analizados en este artículo contiene factores a favor y factores en contra que influyen en el momento de emitir un dictamen positivo o negativo acerca de su exactitud, rapidez y facilidad de ejecución, dado que estos factores son los que concierne debatir.

El algoritmo K-means resulto ser un algoritmo, sencillo al implementar, con requerimientos mínimos de hardware, originando que el algoritmo fuese de los más veloces en tiempo de ejecución. No obstante, su desventaja radica en la exactitud que proporciona al clasificar los elementos del conjunto de datos; ya que el constante ajuste que se realiza a los centroides proporciona márgenes de error considerables. En contra parte, el algoritmo K-nearest Neighbors resultó ser el algoritmo con un tiempo de ejecución muy amplio, alrededor de quince minutos, por cada ejecucin del modelo propuesto. De igual manera, se empleó

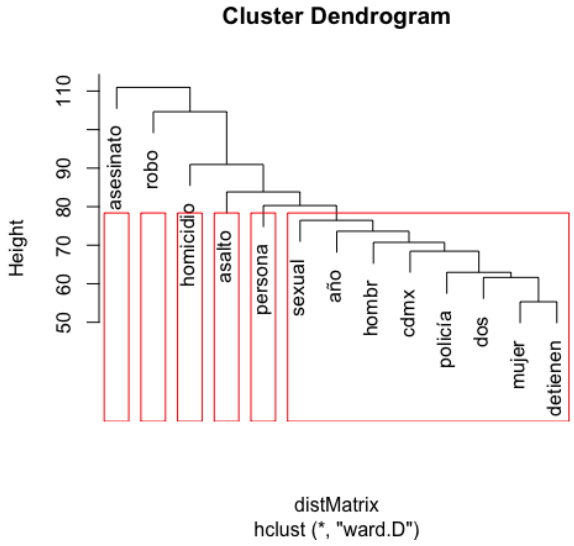


Fig. 4. Dendograma de agrupamiento mediante el algortimo K-Means en una clasificación de seis elementos.

Tabla 3. Matriz de consfusión al ejecutar el algoritmo K-Nearest Neighbor.

Predicción	Asalto	Explotación	Homicidio	Secuestro	Suicidio	Violación
Asalto	259	0	7	0	2	2
Explotación	0	3	0	0	0	0
Homicidio	77	3	451	11	23	17
Secuestro	0	0	1	13	0	0
Suicidio	0	0	2	0	10	0
Violación	1	0	0	0	1	57

la denominada distancia euclidiana, para calcular la distancia de vecindad entre un elemento y el conjunto de elementos ya agrupados o bien conjunto de datos de entrenamiento. En el lado intermedio, de ello, se encuentra el algoritmo Naïve Bayes, el cual no resulta ser el más rápido en tiempos de ejecución, sin embargo, tampoco resulta ser el algoritmo que más demora para obtener los resultados de la clasificación.

En términos de exactitud, el algoritmo que presenta mayor exactitud (*accuraccy*) es el algoritmo de Naïve Bayes, con una exactitud de clasificación de los Tweets muy aproximada a los valores reales expresados. En comparación con los dos algoritmos presentados, K-nearest Neighbors y K-means. En el algoritmo Naïve Bayes se consideran espacios probabilísticos ligados a dos eventos, en los cuales se calcula la probabilidad condicional de que ocurra el evento A dado el evento B.

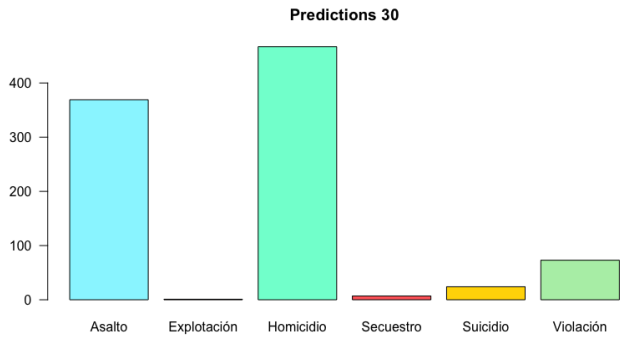


Fig. 5. Estructura de un encabezado y de un tweet.

A base de los elementos y hechos mencionados se dictamina que el algoritmo que presenta mejores resultados es el algoritmo de Naïve Bayes. Esto porque se demostró que presenta una adaptabilidad al modelo de datos presentados; resultando que el aprendizaje del algoritmo fuese rápido y certero al clasificar los Tweets. Los dos algoritmos restantes se adaptaron de forma correcta al modelo de datos, sin embargo, presentaron deficiencias en funciones esenciales como el tiempo de ejecución y de la clasificación correcta de los elementos.

6. Conclusiones y perspectivas

En este artículo se presentaron los resultados obtenidos tras ejecutar tres algoritmos de clasificación, K-Means, Naïve Bayes, K-Nearest Neighbor, para el agrupamiento de delitos sucedidos en la Zona Metropolitana y reportados en Twitter. Los resultados evidencian que cuando se configuran, o encuentran, los parámetros adecuadamente de los algoritmos empleados, éstos nos permiten obtener metainformación relevante de un conjunto de datos. Para nuestro caso, el conjunto de datos utilizado son los comentarios reportados en Twitter de delitos y de ellos se obtuvo, a manera de ejemplo, la semejanza existente entre las palabras utilizadas en uno u otro delito. Debido a la disperción existente entre las palabras de los tweets se observó más sencillez de implementación en los algoritmos de K-Means y Naïve Bayes con respecto a la K-Nearest Neighbor, esto debido al funcionamiento diferente de los algoritmos. El mejor resultado obtenido fue de 87.35 % para el caso del algoritmo,de clasificación, Naïve Bayes; de lo cual podemos concluir que dicho algoritmo es el que mejor responde para la tarea que tenemos como objetivo en este artículo, la clasificación de delitos.

Es importante destacar la efectividad de cada algoritmo, al realizar el aprendizaje automático, debido a que cada uno de ellos cuenta con parámetros diferentes para tomar en cuenta al momento de realizar la clasificación, estos parámetros

contribuyen a obtener diversos resultados. El parámetro que establece el algoritmo Naïve Bayes para calcular la probabilidad de un evento, corresponde a la proporción de ocurrencia del evento fraccionado por el número total de casos admisibles. De forma distinta el algoritmo k-nearest neighbors reúne los casos disponibles con la finalidad de clasificar los casos nuevos identificando afinidades entre los casos disponibles y los nuevos casos, proyectando puntos de los datos no clasificados en los conjuntos definidos. Por otro lado, K-means agrupa los puntos de datos en clases o clústers homogéneos. No obstante poniendo en comparación los algoritmos Naïve Bayes y K-Nearest Neighbor se puede diferenciar y destacar la efectividad en torno a la clasificación de los Tweets del algoritmo Naïve Bayes frente a K-Nearest Neighbor. Debido a que ambos algoritmos se basan en el aprendizaje supervisado. Por otro lado el algoritmo k-Means tiene un buen comportamiento frente a los dos algoritmos mencionados, sin embargo, el algoritmo que tiene una mejor presencia y exactitud al clasificar los textos es Naïve Bayes.

Como persectivas, se tiene que es posible probar otros algoritmos de clasificación y con un conjunto de datos diferente pero que corresponda al mismo dominio, los delitos. Ademaás, hoy en día nos encontramos en una turbulencia financiera, por ejemplo, esto debido a las situaciones políticas y económicas mundiales. Por lo cual se considera de suma importancia la capacidad de poder predecir acontecimientos financieros en torno a los mercados y bolsa de valores, los algoritmos de aprendizaje automático son una herramienta sustancial que nos permitirán poder realizar una predicción aceptable en este rubro y poder con ello contribuir de manera activa en el avance financiero del país. Entonces, los algoritmos empleados en este artículo podrían ser explotados en este campo dado que hemos observado que su método de predicción es fiable y aceptable.

Referencias

1. CED: Clasificación Estadística de Delitos. www3.inegi.org.mx, 2017. [Online]. Disponible: http://www3.inegi.org.mx/sistemas/clasificaciones/delitos.aspx. [Accedido: 01- Nov- 2017].
2. Cherubini, G., Jelitto, J., Venkatesan, V.: Cognitive Storage for Big Data, 49(4), pp. 40–51, (2016)
3. Paniagua, J.: Algoritmos de aprendizaje automático para el análisis de opiniones a partir de textos en español. Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco (2017)
4. Zinzun, J.: Comparativa de la clasificación de tumores obtenida por medio de los algoritmos k-Means, PAM, AGNES. Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco (2016)
5. López, J.: Sistema para la clasificación de artículos científicos mediante el algoritmo KMeans utilizando características semánticas. Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco (2015)
6. Khanna, P.: Sentiment Analysis: An Approach to Opinion Mining from Twitter Data Using R. International Journal of Advanced Research in Computer Science, 8(8), pp. 1–5 (2017)

7. Khairnar, J., Kinikar, M.: Machine Learning Algorithms for Opinion Mining and Sentiment Classification. (IJSRP), 3(6) pp. 1–6 (2013)
8. Sneka, G.: Algorithms for Opinion Mining and Sentiment Analysis: An Overview. International Journal of Advanced Research in Computer Science and Software Engineering, 6(2), pp.1–5 (2016)
9. R: Past and Future History: Disponible: https://cran.rproject.org/doc/html/interface98-paper/paper.html (2017)

ISSN 1870-4069

Predicción de la generación de residuos sólidos urbanos en la Ciudad de México

Ester Calderón-Casanova[1], Mariana López-Ortíz[1], Patricia Galán[1], Esaú Villatoro-Tello[1,2], Raúl R. García-Aguilar[1,2], Brenda García-Parra[1,2]

[1] Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, Maestría en Diseño, Información y Comunicación, México

[2] Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa, División de Ciencias de la Comunicación y Diseño, México

{estercalderon7,maris1589,patriciag.lara}@gmail.com
{evillatoro,rgarcia,bgarcia}@correo.cua.uam.mx

Resumen. La gestión de los residuos sólidos urbanos (RSU) en la Ciudad de México tiene fundamento en normas federales y locales, donde se señala a las delegaciones como las responsables del manejo integral de los RSU. La política del gobierno de la CDMX va encaminada hacia la prevención y minimización de los RSU a través del diseño de planes y programas para el manejo de los mismos. En este sentido, resulta necesario contar con instrumentos de apoyo que permitan conocer y entender el fenómeno, así como sus características, para el diseño de soluciones. El objetivo central de este trabajo fue desarrollar un modelo de predicción de RSU a través de técnicas de Inteligencia Artificial. Para la realización de los experimentos se buscó, identificó y extrajo información de fuentes institucionales. Los resultados obtenidos muestran que los modelos de regresión generados predicen de manera aceptable los niveles de RSU de la CDMX.

Palabras clave: Residuos sólidos urbanos, gobierno electrónico, inteligencia artificial, aprendizaje automático, modelos de regresión.

Towards the Prediction of Urban Solid Waste Generation in Mexico City

Abstract. Mexico City's (CDMX) urban solid waste (USW) management is driven through local and federal laws, which make countyes responsible for their own USW management. CDMX's public policies aim to prevent and reduce the USW generation by means of designing prevention plans and sustainable programs. In this context, local government decision makers can take advantage of support tools to facilitate the implementation of strategic approaches on this regard. In this paper, our

main goal was to develop a predictive USW model using Artificial Intelligence techniques. We collected a large data set regarding the USW management plans from the CDMX. For our experiments we worked with two well-know regression techniques. Performed experiments indicated that USW prediction is possible using a small amount of features.

Keywords: Urban solid waste, e-governance, artificial intelligence, machine learning, regression models.

1 Introducción

Los residuos sólidos urbanos (RSU) son todos aquellos que se generan de actividades humanas, específicamente domésticas y comerciales en comunidades de todos los tamaños, desde aquellas con características urbanas hasta las rurales. El incremento constante en la generación de los RSU está asociado estrechamente al aumento de la población y representa un problema a nivel mundial debido a las afectaciones que ocasiona en la salud pública, la contaminación al medio ambiente en términos de tierra, aire, agua y la explotación desmesurada de recursos naturales [15].

México no se excluye de tal situación, de acuerdo con el Instituto Nacional de Geografía Estadística e Informática al 2016 se generan casi 104 mil toneladas diarias de RSU. Cabe señalar que el incremento en la población, y actividad económica son los principales factores asociados a la generación de RSU. En ese sentido la Ciudad de México (CDMX), conformada por 16 delegaciones, concentra el 30 % de la población urbana nacional, es considerada como la tercera aglomeración urbana más habitada del mundo además que representa el principal centro político, económico, científico y cultural del país; esto combinado con la gran cantidad de actividades, sitios de interés, situación laboral, etc., otorgan a la ciudad un ritmo de vida acelerado que se ve con una repercusión directa en la generación de RSU [3], representando el 13.44 % de la generación nacional [8].

Ante tales circunstancias la CDMX enfrenta un reto importante en cuanto a la Gestión Integral de Residuos Sólidos (GIRS), y en consecuencia el desarrollo de instrumentos de planeación que permitan a las delegaciones tomar mejores decisiones respecto a los requerimientos y necesidades de la población y su medio ambiente. Al respecto, organizaciones internacionales recomiendan el uso y aprovechamiento de datos en el tema para la toma de decisiones. Entre ellas, las Naciones Unidas[3] (ONU) a través del Programa 21, que estipula los datos como un requisito esencial para poder seguir de cerca los cambios en cantidad y tipo de residuos y sus consecuencias para la salud y el medio ambiente. Del mismo modo, la Organización Económica para la Cooperación y el Desarrollo[4] (OECD) recolecta, analiza y explora datos relativos a diversas cuestiones medioambientales, incluyendo los residuos sólidos, con el fin de predecir tendencias a futuro

[3] www.un.org/spanish/esa/sustdev/agenda21/agenda21spchapter21.htm
[4] https://www.oecd.org/about/

y proporcionar a los gobernantes entendimiento sobre las problemáticas y sus cambios en busca de soluciones.

A pesar de los esfuerzos que se realizan en distintos países, y las observaciones y recomendaciones hechas por organizaciones de carácter mundial, la CDMX aún carece de herramientas que permitan el aprovechamiento y explotación de la información disponible para gestionar los RSU. Así entonces, en este trabajo se propone y evalúa un modelo de Inteligencia Artificial (IA) que realiza la predicción de la generación de RSU dentro de la CDMX. Nuestro modelo se entrenó a partir de información que fue recolectada de fuentes institucionales, específicamente la SEDEMA[5]. Los experimentos realizados muestran que es posible entrenar modelos automáticos para la predicción de RSU a partir de un conjunto acotado de variables (atributos), mismas que son ya actualmente recolectadas y documentadas por las autoridades de la CDMX. Además de esto, el análisis realizado en este proceso de minería de datos mostró que la generación de RSU depende de aspectos que van mas allá del aumento en la población, indicando una fuerte correlación con variables que no eran evidentes hasta el momento.

El resto del artículo se organiza de la siguiente manera. La sección 2 describe el trabajo relacionado, en la sección 3 se describe el proceso de recolección de datos con los cuales se entrenó y evaluó nuestro modelo de predicción. La sección 4 describe la metodología empleada para la realización de los experimentos, mientras que la sección 5 describe los experimentos y resultados obtenidos. Finalmente, la sección 6 plantea las conclusiones obtenidas y las líneas de trabajo futuro.

2 Trabajo relacionado

Actualmente, existe un creciente interés tanto en países de primer mundo como en países en vías de desarrollo por aprovechar las técnicas de inteligencia artificial en predicción de la generación de RSU [7,5,6,12,4,1,9,11,13,2]. A continuación se da una breve descripción de los trabajos más recientes y que se asemejan en gran medida a la propuesta aquí planteada. Es importante mencionar que debido a que no existe una base de datos estándar con la cual sistemas de predicción de RSU puedan ser evaluados, resulta difícil tener una comparación directa entre los distintos esfuerzos realizados hasta el momento.

En [6] los autores desarrollan un modelo de optimización de regresión basado en maquinas de vectores de soporte para la planificación de la gestión de residuos sólidos municipales (RSM) en los distritos urbanos de Beijing, China. El modelo desarrollado no solo puede predecir la cantidad futura de generación de residuos de la ciudad, sino que también refleja características dinámicas, interactivas e inciertas del sistema de gestión de RSU. Los autores evaluaron cuatro funciones de kernel; lineal, polinomial, base radial y un perceptron de capa múltiple. En sus experimentos, el kernel polinomial les permite obtener los mejores resultados.

[5] `http://www.sedema.cdmx.gob.mx/`

Por otro lado, en [4] los autores presentan un análisis sobre las variables que inciden en la estimación de residuos municipales recolectados en Venezuela, para lo cual emplearon técnicas de análisis mutivariado, por ejemplo, correlación de Pearson. Para su análisis, emplearon información de 175 municipios, identificando que atributos como son el tamaño de la población urbana, número y tipo de vivienda son variables altamente correlacionadas con la generación de RSU. De forma similar, en [9] los autores evaluaron la relevancia de variables como son: número de residentes, edad de la población, esperanza de vida urbana, en un modelo de predicción de residuos sólidos municipales en Iasi Rumania. Al contrario de otros trabajos, en [9] evalúan la predicción de RSU a un nivel más fino, es decir, buscan generar modelos de predicción por tipo de RSU, por ejemplo, papel, plástico, metal, vidrio, biodegradables y otros residuos.

En [1] los autores evaluaron cuatro modelos de inteligencia artificial como máquinas de vectores soporte (SVM), sistema adaptativo de inferencia neuro-difusa (ANFIS), redes neuronales (ANN), y vecinos más cercanos (kNN). El objetivo fue determinar el mejor modelo para predecir mensualmente la generación de residuos en el Ayuntamiento de Logan en Queensland Australia. De sus experimentos, los autores reportan que el modelo denominado ANFIS fue el que mejor comportamiento obtuvo alcanzando un $R^2 = 0.98$.

Mas recientemente, en [11] los autores emplean información de los censos municipales de Ontario, Canada. Las variables que los autores consideran son de información demográfica y socio-económica que describe a los municipios considerados (220 en total). Para la evaluación de sus modelos emplearon técnicas basadas en árboles de decisión y redes neuronales (ANN), llegando a la conclusión de que las ANN son mejores predictores alcanzando un $R^2 = 0.86$. Por otro lado, en [13] lo autores evalúan la importancia de variables como son el tamaño de la población, la edad, los ingresos y el número de turistas en un modelo de predicción de RSU. Los resultados que alcanzan reportan valores de $R^2 = 0.96$ empleando técnicas de redes neuronales.

Finalmente, en un estudio más ambicioso reportado en [2], los autores evaluaron el comportamiento de un modelo general de regresión basado en redes neuronales para la predicción de RSU en 44 países. Contrario a los trabajos descritos anteriormente, este artículo tuvo como principal objetivo determinar el impacto de las crisis económicas en la generación de RSU. Para sus experimentos emplearon tanto indicadores socio-económicos, datos demográficos, así como indicadores de planes de sustentabilidad.

Como es posible observar, muchos trabajos reportan resultados alentadores en sus modelos de predicción implementados. Sin embargo, nótese que no hay una evidencia clara sobre qué método de regresión puede ser el mejor, pues los resultados de la predicción dependen en gran medida de los datos disponibles. Varios de estos trabajos previos coinciden en la dificultad que representa tener datos confiables para la realización de estos ejercicios, por lo mismo no hay una evidencia clara de qué factores (variables o indicadores) son los más relevantes para la construcción de los modelos de predicción. Inspirados por estos esfuerzos, nuestro trabajo propone y evalúa un modelo de inteligencia artificial que realiza

la predicción de la generación de RSU dentro de la CDMX, el cual, contrario a los trabajos previos, emplea información sobre gestión de los RSU que es recolectada por las distintas delegaciones de la ciudad, con el objetivo de determinar la pertinencia de estos indicadores en el proceso de predicción. Agregado a lo anterior, es conveniente mencionar que hasta donde sabemos, este tipo de trabajos no se ha realizado previamente en la CDMX, por lo cual, este artículo representa un trabajo innovador en muchos sentidos para el contexto nacional.

3 Datos

Uno de los esfuerzos más relevantes para la realización de este trabajo fue la búsqueda y recolección de datos relativos a RSU en México, específicamente en la CDMX. Los datos, como insumo y núcleo del proyecto, resultan muy valiosos debido a la escasez de éstos, así como su función para representar y visibilizar el problema de la gestión de los RSU. Varios trabajos [4,2] reportan la misma situación, la falta de datos en el tema, sin embargo esto no ha sido obstáculo para la formulación de propuestas de solución utilizando enfoques de inteligencia artificial. Motivados por estos esfuerzos previos, este artículo tiene como finalidad determinar hasta que punto es posible entrenar un modelo de predicción de RSU en el contexto nacional, específicamente para la CDMX.

El primer paso para la obtención de datos fue la ubicación de las fuentes institucionales que los generan y publican. Se identificaron diversas organizaciones a nivel nacional como INEGI, SEGOB y CONAPO[6] las cuales cuentan con información relacionada, por ejemplo, generación de RSU, población e ingreso per capita a nivel municipal y delegacional. Sin embargo, para la realización de nuestro estudio se utilizaron los datos proporcionados por la Secretaría del Medio Ambiente (SEDEMA) de la CDMX.

Dichos datos, proporcionado por este organismo, contiene características que la hacen muy valiosa y útil para nuestros propósitos, por ejemplo su periodicidad, diversidad y cantidad de información relativa a RSU. Es importante mencionar que la SEDEMA inició desde el 2006 el monitoreo de información relacionada a la gestión de los RSU en las 16 delegaciones de la CDMX. Con esta información se genera una publicación anual, denominada Inventarios de Residuos Sólidos de la Ciudad de México. Para el trabajo presentado en este artículo se utilizó la información extraída de estos inventarios.

En la tabla 1[7] se muestra el inventario de variables monitoreadas por la SEDEMA para cada una de las 16 delegaciones de la CDMX, así como los años para los cuales existe información de dichas variables ('-' no existe información, 'x' si existe información). En total son 39 variables que refieren a distintos aspectos de la gestión de los RSU como son: equipo, infraestructura, personal, planes de manejo y generación de RSU, puntos de recolección, etc.

[6] `https://www.gob.mx/conapo`

[7] Es importante mencionar que existen muchas más variables, sin embargo para los experimentos reportados en este trabajo sólo se emplearon las variables que tienen información desglosada a nivel delegacional.

A pesar de la validez de esta información, observe que no todas las variables tienen información para los años considerados, i.e., 2006-2016. La razón a la que atribuimos esta ausencia de información es debido a que las distintas gestiones de la CDMX han decidido prestar atención a aspectos diferentes, por lo cual algunas variables desaparecen mientras que otras van surgiendo. El resultado de esto es la carencia de información en gran parte de las variables entre los años 2006 y 2010 principalmente, sin embargo note que la variable de interés, i.e., la que interesa modelar ("Generación de residuos sólidos (ton/día)") está presente para todos los años considerados.

En su totalidad, la base de datos que se logró compilar contiene 176 instancias (16 delegaciones × 11 años) representadas a través de 39 atributos (variables), es decir un total de 6864 registros.

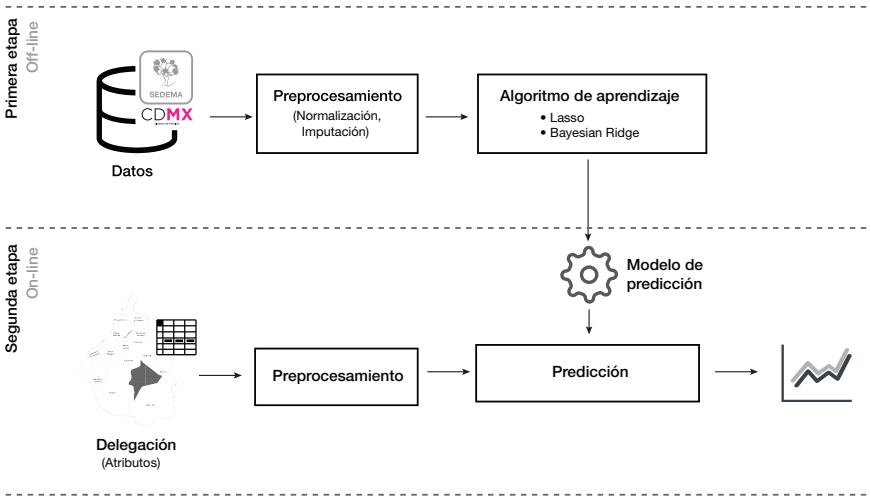


Fig. 1. Metodología propuesta para la predicción de la generación de RSU.

4 Metodología propuesta

La figura 1 muestra de manera esquemática la arquitectura del método propuesto para hacer la predicción del valor de generación de RSU. Contrario a un problema de clasificación, nuestro objetivo implica determinar un valor continuo, el valor de "Generación de residuos sólidos (ton/día)"; por lo tanto abordamos el reto como un problema de regresión. A continuación describimos en forma breve cada uno de los módulos de nuestra metodología empleada.

Tabla 1. Inventario y clasificación por año de variables identificadas. *PES (Puntos específicos de recolección separada). *PMGRS (Planes de manejo y generación de residuos sólidos).

Variable	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Rutas con recolección separada	x	x	x	x	x	x	x	x	-	-	-
Rutas totales de recolección	x	x	x	x	x	x	x	x	x	x	x
Colonias con separación	x	x	x	x	x	x	x	x	-	-	-
Colonias totales por delegación	x	x	x	x	x	x	x	-	x	x	x
Planes de manejo	x	x	x	-	x	x	x	x	x	x	x
Generación de residuos sólidos (ton/día)	x	x	x	x	x	x	x	x	x	x	x
Eficiencia de la separacion de r. organicos	-	-	-	-	-	-	x	x	-	-	-
Eficiencia de la recoleccion de r. organicos	-	-	-	-	-	-	-	-	x	x	x
Total de vehículos recolectores	-	x	x	x	x	x	x	x	x	x	x
Tipo de vehículo: Carga trasera	-	-	-	x	-	-	x	x	x	x	x
Tipo de vehículo: Rectangular	-	-	-	-	-	-	x	x	x	x	x
Tipo de vehículo: tubular	-	-	-	-	-	-	x	x	x	x	x
Tipo de vehículo: Volteo	-	-	-	-	-	-	x	x	x	x	x
Tipo de vehículo: Frontal	-	-	-	-	-	-	-	x	x	x	x
Tipo de vehículo: Doble compartimento	x	x	x	x	x	x	x	x	x	x	x
Otros vehículos	-	-	-	x	-	x	x	x	x	x	x
Barrido manual número de barredores	-	-	-	-	-	-	-	x	x	x	x
Barrido manual número de carritos	-	-	-	-	-	-	-	x	x	x	x
Barrido manual número de rutas o tramos	-	-	-	-	-	-	-	x	x	x	x
Número de barredoras mecánicas	-	-	-	-	-	-	-	x	x	x	x
Chóferes	-	-	-	-	-	-	-	x	x	x	x
Voluntarios	-	-	-	-	-	-	-	x	x	x	x
PES en unidades habitacionales	-	-	-	-	-	-	x	x	x	x	x
PES en escuelas	-	-	-	-	-	-	x	x	x	x	x
PES en mercados	-	-	-	-	-	-	x	x	x	x	x
PES centro comercial	-	-	-	-	-	-	x	x	x	x	x
PES en terminales	-	-	-	-	-	-	x	x	x	x	x
PES en parques y plazas	-	-	-	-	-	-	x	x	x	x	x
PES de servicios e industria	-	-	-	-	-	-	x	x	x	x	x
PES Otro	-	-	-	-	-	-	x	x	x	x	x
PMGRS por categoría A	-	-	-	-	-	x	x	x	x	x	x
PMGRS por categoría B	-	-	-	-	-	x	x	x	x	x	x
PMGRS por categoría C	-	-	-	-	-	x	x	x	x	x	x
PMGRS por categoría D	-	-	-	-	-	x	x	x	x	x	x
PMGRS por categoría E	-	-	-	-	-	x	x	x	x	x	x
PMGRS por sector comercio	-	-	-	-	-	-	x	x	x	x	x
PMGRS por sector industria	-	-	-	-	-	-	x	x	x	x	x
PMGRS por sector servicios	-	-	-	-	-	-	x	x	x	x	x

4.1 Preprocesamiento

Como se describió en la sección 3, la base de datos empleada está conformada con información proveniente de la SEDEMA. La tabla 1 muestra la existen-

cia/ausencia (x/-) de los 39 indicadores (atributos) que han sido monitoreados en el periodo de 2006-2016. Nótese un problema importante de datos faltantes, específicamente de los 6864 registros, 3323 son vacíos, es decir, el 48 %.

Tener una cantidad importante de datos faltantes representa un reto para todo algoritmo de aprendizaje automático. Por tal razón, como parte del pre-procesamiento de los datos se aplicaron técnicas de imputación. El proceso de imputación refiere a remplazar datos faltantes por un valor substituto. Para los experimentos reportados en este trabajo se utilizó como técnica de imputación de cada atributo k el valor de la media ($\overline{x}_k$).

Como operación adicional del pre-procesamiento de los datos se aplicó un proceso de normalizacion. Específicamente se aplicó la norma l^2 o norma euclidiana, la cual se calcula como se observa en la ecuación (1):

$$|\boldsymbol{X}| = \sqrt{\sum_{k=1}^{n} |x_k|^2}, \tag{1}$$

donde x_k representa al atributo k y n es el número total de atributos. Así entonces, al final el valor de cada atributo k se re-calcula como $x_k = \frac{x_k - \overline{x}_k}{|\boldsymbol{X}|}$.

4.2 Métodos de regresión

Los métodos de regresión tienen como finalidad principal construir un modelo que sea capaz de predecir el valor cuantitativo de una variable a partir de un conjunto de datos históricos. Al final, el modelo será capaz de hacer una predicción sobre un nuevo dato del cual se desconoce su respuesta. En otras palabras, el modelo predictivo puede ser descrito como el problema de aproximar una función f con variables de entrada X a su correspondiente valor continuo de salida y. A esto se le conoce como el problema de encontrar la función de aproximación $f(X) = y$.

A la variable y se le conoce como la variable (cuantitativa) de respuesta, mientras que X es el conjunto de atributos o variables predictores. Así entonces, lo que se busca es encontrar los coeficientes β que satisfacen a: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$. Para los experimentos realizados en este trabajo empleamos como métodos de regresión el algoritmo de Lasso [14] y Bayesian-Ridge [10].

Lasso. Es un tipo de regresión lineal que utiliza técnicas de "encogimiento" (shrinkage), el cual tiene como objetivo obtener el conjunto de predictores que minimizan el error de predicción de la variable de respuesta (ecuación (2)). El encogimiento significa que los valores de los predictores son llevados a un punto central, por ejemplo la media o incluso cero:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{2}$$

Contrario a un método de mínimos cuadrados ordinarios (OLS), Lasso impone restricciones en el encogimiento, específicamente penalizando por medio

de considerar el valor absoluto de la magnitud de los coeficientes, en otras palabras por el valor de la norma l^1 (segundo término de la ecuación (2).

Bayesian Ridge. El método de Ridge es muy similar al método de regresión de Lasso, la diferencia radica principalmente en la penalización por medio de considerar el cuadrado de la magnitud de los coeficientes (vea ecuación (3)), en otras palabras, la norma l^2.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j^2|. \tag{3}$$

El aspecto probabilístico (bayesian) de este método es debido a que los valores de los coeficientes son estimados a priori. Para nuestros experimentos, los valores a priori son estimados bajo una distribución Gaussiana. Este tipo de técnica es recomendada cuando existen una insuficiencia de datos.

En general, ambos métodos de regresión se recomiendan cuando los datos muestran altos niveles de correlación, el cual es el nuestro caso. Para los experimentos realizados se utilizó la implementación de Lasso y Bayesian Ridge disponibles en scikitlearn[8].

4.3 Evaluación

Para la evaluación del desempeño del modelo de predicción se utilizaron métricas estándar, como son el error medio absoluto (MAE), el coeficiente de determinación (R^2), y la varianza (EV), las cuales se calculan como se muestra en las ecuaciones (4), (5) y (6) respectivamente:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}, \tag{5}$$

$$\text{EV} = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}, \tag{6}$$

donde n es el número de instancias, $\hat{y}$ el el valor predicho por el modelo, y es el valor real, $\overline{y}$ es el valor medio de y, y Var es la varianza estadística σ^2. Tanto EV como R^2 son métricas que, entre más cercano a 1.0, significa que mejor es el desempeño del modelo de predicción. Por otro lado, MAE es una medida que crece conforme más errores comete el método de predicción, por lo cual, un valor cercano a 0 es preferible.

Finalmente, es conveniente mencionar que para la realización de los experimentos se utilizó como técnica de validación una estrategia de validación cruzada de 10 pliegues.

[8] `http://scikit-learn.org/`

Tabla 2. Resultados obtenidos de los experimentos realizados.

Experimento	Algoritmo	#Inst	#Atts	Prep-Op.	MAE	R^2	V
EXP-1	Lasso	176	39	l^2	281.26	0.40	0.40
	B-Ridge			l^2	267.62	0.45	0.45
	Lasso			l^2, $\overline{x}_k$	**258.53**	**0.52**	**0.52**
	B-Ridge			l^2, $\overline{x}_k$	266.97	0.52	0.52
EXP-2	Lasso	64	39	l^2	**152.45**	**0.78**	**0.78**
	B-Ridge			l^2	181.50	0.68	0.68
	Lasso			l^2, $\overline{x}_k$	181.50	0.68	0.68
	B-Ridge			l^2, $\overline{x}_k$	204.37	0.64	0.64
EXP-3	Lasso	64	35	l^2	**159.89**	**0.76**	**0.76**
	B-Ridge			l^2	208.65	0.65	0.65
	Lasso			l^2, $\overline{x}_k$	187.37	0.64	0.64
	B-Ridge			l^2, $\overline{x}_k$	220.44	0.61	0.61
EXP-4	Lasso	64	10	l^2	**160.71**	**0.79**	**0.79**
	B-Ridge			l^2	166.67	0.77	0.77

5 Experimentos y resultados

En esta sección se describen los experimentos realizados así como los resultados obtenidos. Cuatro grandes experimentos fueron realizados:

- **EXP-1.** El objetivo de este experimento fue comprobar el impacto de entrenar un modelo de predicción empleando toda la base de datos disponible.
- **EXP-2.** El objetivo de este experimento fue evaluar el impacto de eliminar de la base de datos los años que presentaban el porcentaje más alto de datos faltantes, i.e., 2006 a 2012.
- **EXP-3.** En este experimento se evaluó el impacto de eliminar aquellas variables ausentes del conjunto de datos resultante del EXP-2; en total 3 variables son descartadas (Total de rutas con recolección separada, total de colonias con recolección separada y eficiencia de la separación de los residuos orgánicos por delegación).
- **EXP-4.** Para este experimento se midió la correlación de Pearson entre las 35 variable restantes contra la variable de interés (i.e., Generación de residuos sólidos (ton/día)). Para la construcción de la representación conservamos sólo aquellas variables con correlación mayor a 0.5,

Los resultados de los experimentos se muestran en la tabla 2. La segunda columna especifica el algoritmo de regresión empleado, la columna **#Inst** indica el número de instancias presentes en el experimento, **#Atts** refiere el número de atributos con que cada instancia es representada, **Pre-Op** indica las operaciones aplicadas a los datos, como son normalización (l^2) e imputación ($\overline{x}_k$).

En primer lugar, observe que tener un porcentaje tan alto de datos vacíos (EXP-1) impacta de forma negativa en el desempeño del método de regresión. Nótese que cuando no se realiza imputación, el desempeño obtenido es de un $R^2 = 0.45$ con el método de B-Ridge. Por otro lado, cuando se hacen operaciones

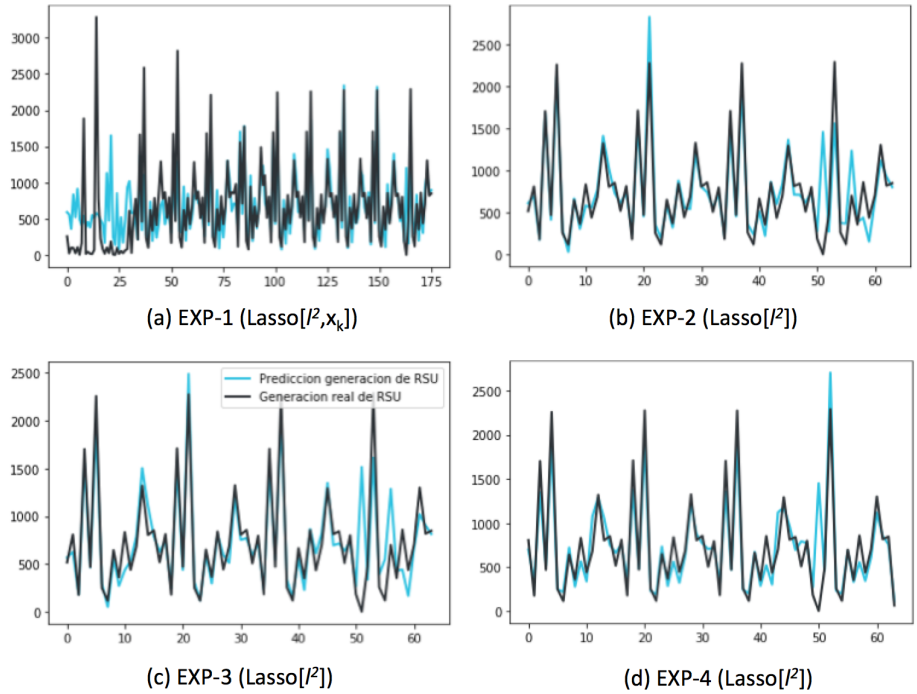


Fig. 2. Gráficas de comparación entre las mejores configuraciones de cada experimento realizado. El eje de las $y's$ representa el valor de RSU, mientras que el eje de las $x's$ representa las instancias, las cuales están ordenadas cronológicamente.

de imputación el coeficiente de determinación sube a $R^2 = 0.52$ con el método Lasso. En segundo lugar, los resultados de los experimentos EXP-2 y EXP-3 muestran que haber hecho la eliminación de las instancias de los años 2006 a 2012 permite al modelo de regresión obtener un mejor desempeño ($R^2 = 0.78$). La eliminación de estas instancias se justifica debido a que la gran mayoría de ellas tenían ausencia de datos para gran cantidad de variables. Finalmente, los resultados obtenidos del EXP-4 indican que hacer una selección de atributos permite mejorar el valor del coeficiente de determinación ($R^2 = 0.79$). Este resultado indica, hasta cierto punto, que es suficiente emplear solo diez atributos para lograr tener un modelo de predicción de RSU con un desempeño aceptable.

En la figura 2 se muestra gráficamente el desempeño de cada uno de los experimentos realizados en su mejor configuración. La línea más obscura (negra) denota el valor real de RSU mientras que la línea más tenue (azul) es el valor que se predice. Observe para el EXP-1 (a), las primeras instancias, las cuales corresponderían a los años de 2006 a 2009 aproximadamente, son las que peor comportamiento tienen. Por otro lado, el experimento que mejor desempeño tuvo es el (d) EXP-4, donde se puede observar que el comportamiento del modelo de predicción es mejor que (b) y (c), sobre todo hacia las últimas instancias.

6 Conclusiones

En este artículo se describe un método para la predicción de residuos sólidos urbanos dentro de la CDMX. El método propuesto busca impactar de manera positiva en un problema de relevancia mundial, i.e., la generación de RSU. Hasta donde se sabe, nuestro trabajo es pionero en buscar aplicar técnicas de IA para atender este problema dentro del contexto nacional y en particular en una de las urbes más pobladas del mundo, la Ciudad de México.

Los experimentos realizados muestran que es posible hacer una predicción adecuada de los valores de RSU, alcanzando desempeños de $R^2 = 0.79$. Como parte de los experimentos, se identificó un conjunto de 10 variables con las cuales se puede hacer esta predicción. Contrario al trabajo reportado en la literatura, nuestro método no emplea información socio-económica ni demográfica para hacer la representación de los municipios/delegaciones, con lo cual se muestra que hay otros factores involucrados en la generación de RSU. El modelo desarrollado puede servir para unidades territoriales similares a las delegaciones de la Ciudad de México, por ejemplo como son todos los municipios del país, siempre y cuando cuenten con la información mínima de las 10 variables/atributos requeridas para la predicción.

Es conveniente reflexionar sobre la importancia que tiene la disponibilidad de información. Es necesario resaltar la obligación de los gobiernos actuales para invertir en iniciativas que les permita monitorear de forma periódica aspectos relevantes a la gestión de RSU. Contar con mecanismos constantes permitirá generar información que eventualmente podría favorecer a los métodos de predicción. En este mismo sentido, el análisis de los datos obtenidos a través de la SEDEMA permitió identificar los cambios de intereses entre distintos periodos de gestión de la CDMX. Esto se vio reflejado debido a la aparición/desaparición de distintas variables a lo largo de los años. Esto evidencia, hasta cierto punto, una falta de planificación a largo plazo.

Como trabajo futuro haremos más experimentos incorporando información del tipo socio-demográfico, económico y/o de educación. En el trabajo previo reportado hasta el momento, se ha mostrado que este tipo de variables impacta positivamente a los modelos de regresión. Además de esto, es de nuestro particular interés integrar el modelo desarrollado en un sistema de información para la gestión de los RSU en la CDMX, el cual servirá como herramienta de apoyo en el análisis del estado actual de las delegaciones, se visualizará información sobre infraestructura, servicios y la generación de los RSU en la CDMX[9].

Agradecimientos. El trabajo de las primeras tres autoras fue parcialmente financiado por el CONACyT a través de las becas de maestría 616127, 616489, 775648 respectivamente. Agradecemos además el apoyo otorgado por la Coordinación de la Maestría en Diseño, Información y Comunicación (MADIC) de la UAM Cuajimalpa, así como el apoyo de la División de Ciencias de la Comunicación y Diseño de la UAM Cuajimalpa.

[9] El sistema de información estará disponible en: `www.rsucdmx.com`

Referencias

1. Abbasi, M., El Hanandeh, A.: Forecasting municipal solid waste generation using artificial intelligence modelling approaches. Waste Management 56, 13–22 (2016)
2. Adamović, V.M., Antanasijević, D.Z., Ristić, M.Đ., Perić-Grujić, A.A., Pocajt, V.V.: Prediction of municipal solid waste generation using artificial neural network approach enhanced by structural break analysis. Environmental Science and Pollution Research 24(1), 299–311 (2017)
3. del Medio Ambiente de la CDMX-SEDEMA, S.: Programa de gestión integral de los residuos sólidos 2016-2020. Tech. rep., SEDEMA (2016)
4. Chassaigne, G., Pinto, G.: Determinación de variables que inciden en la estimación de residuos y desechos sólidos municipales recolectados en venezuela. Interciencia 39(12) (2014)
5. Chung, S.S.: Projecting municipal solid waste: The case of hong kong sar. Resources, Conservation and Recycling 54(11), 759–768 (2010)
6. Dai, C., Li, Y., Huang, G.: A two-stage support-vector-regression optimization model for municipal solid waste management–a case study of beijing, china. Journal of environmental management 92(12), 3023–3037 (2011)
7. Dyson, B., Chang, N.B.: Forecasting municipal solid waste generation in a fast-growing urban region with system dynamics modeling. Waste management 25(7), 669–679 (2005)
8. Geografía (INEGI), I.N.d.E.y.: Censo nacional de gobiernos municipales y delegacionales 2017. CNGDM, `http://www.beta.inegi.org.mx/proyectos/censosgobierno/municipal/cngmd/2017/`
9. Ghinea, C., Drăgoi, E.N., Comăniţă, E.D., Gavrilescu, M., Câmpean, T., Curteanu, S., Gavrilescu, M.: Forecasting municipal solid waste generation using prognostic tools and regression analysis. Journal of environmental management 182, 80–93 (2016)
10. Haitovsky, Y., Wax, Y.: Generalized ridge regression, least squares with stochastic prior information, and bayesian estimators. Appl. Math. Comput. 7(2), 125–154 (Sep 1980), `http://dx.doi.org/10.1016/0096-3003(80)90002-8`
11. Kannangara, M., Dua, R., Ahmadi, L., Bensebaa, F.: Modeling and prediction of regional municipal solid waste generation and diversion in canada using machine learning approaches. Waste Management (2017)
12. Srivastava, A.K., Nema, A.K.: Fuzzy parametric programming model for multi-objective integrated solid waste management under uncertainty. Expert Systems with Applications 39(5), 4657–4678 (2012)
13. Sun, N., Chungpaibulpatana, S.: Development of an appropriate model for forecasting municipal solid waste generation in bangkok. Energy Procedia 138, 907–912 (2017)
14. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996)
15. Un-Habitat: Solid waste management in the world's cities: water and sanitation in the world's cities 2010. UN-HABITAT (2010)

ISSN 1870-4069

Enfoque para la clasificación de vegetación polinífera usando imágenes multiespectrales y redes neuronales

Juan Jose Negron-Granados, Ricardo Legarda-Sáenz, Víctor Uc-Cetina

Universidad Autónoma de Yucatán, Facultad de Matemáticas, México

a08001373@alumnos.uady.mx, {rlegarda, uccetina}@correo.uady.mx

Resumen. La apicultura es una actividad pecuaria muy importante para México. Yucatán es uno de las entidades federativas que más aporta en producción y exportación de miel. Sin embargo, la apicultura se ve amenazada por una deficiente organización de los apicultores, mala ubicación de los apiarios y reducción de la vegetación, lo que afecta a la flora polinífera, principal fuente de alimento de las diferentes especies de abejas. Este trabajo hace la propuesta de utilizar información espectral para identificar las diferentes plantas productoras de polen utilizando visión remota, procesamiento de imágenes y redes neuronales artificiales. En este trabajo se sientan las bases para realizar un software que contribuya a los apicultores en la cuantificación de la flora polinífera que existan alrededor de sus apiarios y les sirva como herramienta para prevenirse ante hambrunas, optimización en la ubicación estratégica de colonias de abeja y mejorar una de las actividades pecuarias más importantes de México.

Palabras clave: Índices de vegetación, redes neuronales, imágenes multiespectrales.

Approach for the Classification of Polinífera Vegetation Using Multispectral Images and Neural Networks

Abstract. The apiculture is a very important livestock activity for Mexico. Yucatan is one of the states that contributes most in the production and export of honey. However, the apiculture is threatened by a poor organization of beekeepers, poor location of the apiaries and reduction of vegetation, which affects the polliniferae flora, main source of food for different species of bees. This work makes the proposal to use spectral information to identify the different pollen producing plants using remote viewing, image processing and artificial neural networks. In this work, the foundations are laid for making software that contributes to beekeepers in the quantification of the polliniferae flora that exist around their apiaries and serves them as a tool to prevent themselves from famines,

optimization in the strategic location of bee colonies and improve one of the most important livestock activities in Mexico.

Keywords: Vegetation indices, neural networks, multispectral images

1. Introducción

La península de Yucatán es, por tradición, una importante región productora de miel a nivel mundial, ya que 95 % de su producción se destina al mercado internacional [12]. La apicultura ha sufrido un grave deterioro en la región. Ha disminuido la posibilidad de recursos néctar-polinívoros y, por tanto, la alimentación de las abejas, con la consiguiente baja en la producción. Por lo que existe la necesidad de mejorar los sistemas de comercialización y de diversificación de la actividad y actualizar las técnicas de producción y administración del proceso productivo por parte de los apicultores para obtener la calidad de la miel requerida por el mercado [12].

Una manera de mejorar la producción y la calidad de la miel es caracterizando la vegetación que se encuentra alrededor de una colonia de abejas. La dieta de las abejas es muy específica (Tabla 1), por lo que se espera que 5 km a la redonda de un apiario se encuentre una abundante vegetación adecuada para evitar los problemas de producción de miel.

Tabla 1. Especies de plantas más importantes para la producción de miel en Yucatán.

Nombre común	**Nombre científico**
Ts'its'ilche'	Gymnopodium floribundum
Taj	Viguiera dentata
Ja'abin	Piscidia piscipula
Box katsim	Acacia gaumeri
Sac katsim	Mimosa bahamensis
Tsalam	Lysiloma latisiliquum
X-tabentún	Turbina corymbosa
Kitimché	Caesalpinia gaumeri
Enredaderas	Varias especies de la familia Convolvulaceae

La clasificación de plantas en salidas de campo es un proceso complicado y nada eficiente. Es común que para la evaluación y clasificación de la vegetación suelen usarse imágenes satelitales por su alto contenido de información. Sin embargo, la distancia de los sensores evita tener imágenes de buena resolución, pues en una imagen multiespectral satelital un pixel puede representar un área de 30 m^2. Obtener imágenes aéreas con un dron mejora resolución espacial. La cámara Sequoia Parrot permite obtener imágenes con una con una relación de 3,7 y 18,6 centímetros por pixel para vuelos de 30 y 150 metros de altura respectivamente.

Con la información de los índices de vegetación se pueden calcular los Valores medios, desvíos estándares, máximos y mínimos valores entre índices espectrales que se utilizan como el vector de características de la red neuronal.

En este artículo se ha presentado en la sección de introducción la descripción del problema que se tiene con la flora polinífera, y la importancia y los beneficios que conlleva la clasificación de dicha flora. A continuación, en la sección del estado del arte, se describen los trabajos relacionados que se utilizó para explorar las diferentes técnicas empleadas para resolver problemas similares y justificar la metodología empleada. Posteriormente, se describe la metodología propuesta y las herramientas a utilizar para cumplir nuestro objetivo de clasificar diferentes plantas apícolas. Luego se presentan los resultados preliminares que dan soporte de factibilidad a la metodología propuesta.

2. Trabajos en el área de clasificación de imágenes espectrales

En la literatura consultada podemos encontrar trabajos similares, en los cuales se utilizan algoritmos de aprendizaje automático para el reconocimiento de superficies usando imágenes satelitales. A continuación se mencionan los trabajos enfocados a resolver problemas de clasificación de coberturas usando métodos de aprendizaje profundo (Deep Learning); estos trabajos se caracterizan por hacer uso de imágenes hiperespectrales. Las imágenes hiperespectrales están compuestas de imágenes contiguas en todo el espectro electromagnético, pudiendo llegar a ser un conjunto de más de 100 imágenes.

Los trabajos más recientes acerca de la clasificación general de la cobertura terrestre utilizan técnicas de aprendizaje profundo y se encuentran en las investigaciones de Chen y Lin, del Instituto de Tecnología Harbin, China. En el año 2013 Lin et al. hacen la propuesta de utilizar autoencoders para la clasificación de imágenes hiperespectrales [14]. En el 2014, Chen et al. reportaron sus resultados de aplicar los autoencoders [5] para el reconocimiento de diferentes regiones terrestres, logrando identificar con mucho éxito: suelo desnudo, prados, ladrillos, sombras, grava, agua, diferentes tipos de pantano, maleza y pinos. Un año después, Chen et al. realizaron el mismo experimento, cambiando el método autoencoder por una red de creencia profunda (DBN por sus siglas en inglés, Deep Belief Network) [6]. Los resultados fueron bastante buenos, la red pudo identificar las mismas regiones que detectó la técnica de autoencoder, sin embargo la arquitectura del autoencoder supera ligeramente a DBN. En promedio el autoencoder tuvo una efectividad del 96 % y DBN del 93 %.

También es posible encontrar trabajos que empleen técnicas de Redes Neuronales Convolucionales (CNN, por sus siglas en inglés), por ser métodos efectivos para el procesamiento de imágenes y la extracción de características. En las pruebas de Makantasis [17] y Maggiori [16] obtienen resultados de efectividad del 99 %.

Los trabajos anteriores describen la metodología para clasificar las coberturas utilizando métodos de aprendizaje profundo utilizando imágenes hiperespectra-

les, las cuales resultaron ser altamente eficientes. Sin embargo, en la literatura hay metodologías que utilizan imágenes multiespectrales, este tipo de imágenes consta de un conjunto entre 4 y 20 imágenes espectrales, dependiendo del satélite o sensor que se utilice. Para el procesamiento de imágenes multiespectrales en vegetación se suelen usar 4 bandas del espectro de la luz: verde (530-570 nm), rojo (640-680 nm), borde rojo (730-740 nm) e infrarrojo cercano (770-810 nm). A continuación se mencionan algunos de los trabajos citados que utilizan las bandas espectrales mencionadas para la identificación de las diferentes coberturas; además cabe mencionar que se utiliza una Red Neuronal Artificial como método de clasificación y sensores satelitales para la obtención de la información espectral.

El uso de redes neuronales artificiales en la clasificación de datos de sensores remotos ha crecido en los últimos años [15]. En la literatura citada se han usado para identificar diferentes coberturas terrestres, como agua, asfalto, suelo desnudo, vegetación [3, 26].

Entre las aplicaciones más importantes se encuentra el análisis multitemporal [11] que consiste en la clasificación de las coberturas pero analizando las variaciones espectrales a través del tiempo para determinar los cambios en las características biofísicas de las regiones, determinar el avance de la restauración de zonas ecológicas [23] o el monitoreo de parques nacionales [25]. Del mismo modo, la técnica de combinar imágenes multiespectrales con redes neuronales ha llevado al análisis de cultivos, por ejemplo el de caña de azúcar [24] o el análisis de las condiciones del suelo, como el nivel de salinidad en la tierra [19].

Debido a que varias coberturas terrestres pueden presentar respuestas espectrales similares, se han realizado variantes que complementan la información de las imágenes con información cartográfica [18] como modelos digitales de elevación (MDE), mapas temáticos de la región o datos de precipitación atmosférica [1], mejorando la efectividad aproximadamente en un 10 %.

En los trabajos presentados anteriormente se describen las ventajas de utilizar las imágenes multiespectrales y las redes neuronales para la clasificación de diferentes coberturas del suelo, sin embargo, una de las limitaciones de utilizar imágenes satelitales es el costo de la adquisición y procesamiento de las imágenes, pues una imagen puede tener una dimensión mayor a 27000 pixeles^2 con una resolución de 11 bits (QuickBird) por cada una de las bandas que opera el sensor en el satélite. Otra limitante que es común observar es la resolución del píxel, en el satélite QuickBird se tiene una resolución de $61cm$ de resolución, esto quiere decir que un pixel representa un área de $0{,}36m^2$; esta resolución puede ser útil cuando se trata de clasificar superficies extensas, sin embargo para detectar plantas específicas requiere una resolución más precisa.

Adicionalmente, en los trabajos presentados se identifica el proceso general para la clasificación de coberturas usando imágenes espectrales: adquisición de las imágenes, un preprocesamiento en la imagen, como el cálculo de los índices de vegetación o las correcciones y calibración de las imágenes satelitales, finalmente se hace el uso de técnicas de algoritmos de aprendizaje máquina para la clasificación.

3. Materiales y métodos

A diferencia de la literatura citada, este trabajo propone utilizar utilizando un dron y la cámara multiespectral Sequoia Parrot para muestrear las áreas de interés para la recolección de las muestras de la flora apícola. Esto debido a que mejora la resolución espacial de las imágenes y se evita influencias de gases en la atmósferas y nubes. Posteriormente se propone calcular índices de vegetación de las imágenes muestreadas; estos índices son el índice de vegetación de diferencia normalizada, el índice de vegetación verde normalizada y el índice de clorofila. A partir de los índices de vegetación calculados se construirá el vector de características a partir de datos estadísticos: promedio, moda, mínimo y máximo de cada uno de los índices. En total, un vector de 12 características para cada una de las imágenes. Luego, se generará una base de datos etiquetada con los vectores de características calculados. A continuación se diseña, programa y entrena una red neuronal completamentamente conectada para la clasificación de la vegetación utilizando la base de datos etiquetada. Una vez entrenada la red se probará el clasificador y finalmente, se analizarán los resultados. En las siguientes subsecciones se describen los materiales necesarios para llevar a cabo la metodología propuesta.

3.1. Determinación de los datos espectrales

Las imágenes a utilizar corresponden al sensor multiespectral Sequoia Parrot (Figura 1). Este sensor está compuesto por dos módulos: el cuerpo (*Body*) y el sensor de luz (*Sunshine sensor*). El cuerpo está integrado por cuatro cámaras espectrales de 1.2 Mpx, una cámara RGB de 16 Mpx, WiFi, IMU y magnetómetro. El sensor de luz cuenta con 4 sensores espectrales con los mismos filtros que el cuerpo, GPS, IMU y magnetómetro. La información de las diferentes frecuencias de la luz que se pueden obtener son las bandas: verde (530-570 nm), rojo (640-680 nm), borde rojo (730-740 nm) e infrarrojo cercano (770-810 nm); además de tomar fotos a color (Imágenes RGB). Las imágenes de las 4 bandas espectrales tienen formato RAW de 10 bits en un archivo TIFF. La imagen RGB se guarda en formato JPG. En total son 5 imágenes por cada foto.

Con la información de las imágenes multiespectrales es posible calcular índices espectrales que constituyen una herramienta útil para evaluar áreas extensas [2]. Recientes avances en el aprendizaje profundo han hecho que hoy en día sea posible utilizar estos índices para resolver problemas relacionados con la identificación de elementos en una imagen, con métodos coherentes, precisos y fiables [9].

3.2. Análisis de datos

Existen combinaciones de las imágenes espectrales obtenidas para obtener información relevante de la vegetación, a esto se le conoce como índices de vegetación. Estos índices permiten estimar variables biofísicas de la vegetación a partir de la luz que las plantas emiten o reflejan, minimizando la influencia de

Fig. 1. Cuerpo del sensor (Izquierda). Sensor de luz (Derecha).

perturbaciones como las debidas al suelo y a las condiciones atmosféricas [10,21].

La selección de bandas multiespectrales depende del índice que se desee calcular [10]; sin embargo, es frecuente el uso de las bandas espectrales del rojo (del campo visible) y el infrarrojo cercano, puesto que las plantas reflejan fuertemente la banda de luz infrarroja cercana debido a una capa gruesa en la superficie inferior de la hoja (Figura 2). A continuación se describen los índices de vegetación a utilizar para la extracción de características.

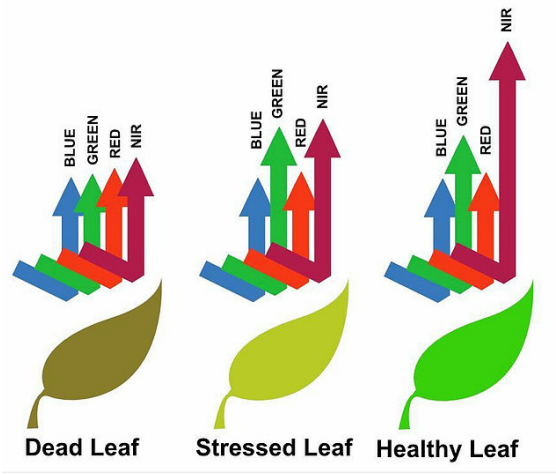


Fig. 2. El pigmento de las hojas de las plantas, la clorofila, absorbe con fuerza la luz visible para su uso en la fotosíntesis. La estructura celular de las hojas, por otro lado, refleja intensamente luz infrarroja cercana [20].

Índice de Vegetación Normalizado Rouse propuso el Índice de Vegetación de Diferencia Normalizada (Normalized Difference Vegetation Index, NDVI) en

1974 y hasta la fecha es uno de los más usado [22]. Un aspecto interesante de este índice frente al cociente simple es que toma valores entre -1 y 1, lo cual facilita notablemente su interpretación [13]. El NDVI es la diferencia normalizada de las bandas infrarroja (NIR) y roja (R) (Ecuacion 1):

$$NDVI = \frac{NIR–R}{NIR + R}. \tag{1}$$

Este índice es uno de los más usados, pues está fuertemente relacionado con la salud de las plantas. Una planta sana tiene las paredes de las células en las hojas llenas de agua; por lo que reflejan fuertemente la luz infrarroja. Por otro lado, una planta que sufre de estrés por plagas o sequías, reflejan débilmente la banda de infrarrojo, pues sus hojas no cuentan con suficiente agua.

Índice de Vegetación de Diferencia Normalizada Verde El Índice de Vegetación de Diferencia Normalizada Verde (Green Normalized Vegetation Index, GNVI) es un índice de la actividad fotosintética, y es uno de los índices de vegetación más comúnmente utilizados para determinar la absorción de agua y nitrógeno en el follaje del cultivo [7]. Es similar al NDVI, sin embargo se sustituye la banda roja del espectro visible por la banda verde del espectro visible (V) (Ecuación 2). Siendo uno de los índices de vegetación más comúnmente utilizados para determinar la absorción de agua y nitrógeno en el follaje del cultivo [4]:

$$GNVI = \frac{NIR - V}{NIR + V}. \tag{2}$$

Índice de Vegetación de Clorofila El índice de vegetación de clorofila (Chlorophyll vegetation index, CVI) es un estimador de clorofila [27]. Se calcula usando las bandas del infrarrojo cercano, roja y verde (Ecuación 3):

$$CVI = NIR \cdot \frac{R}{G^2}. \tag{3}$$

3.3. Red neuronal artificial

Las redes neuronales artificiales (RN) son algoritmos matemáticos basados en el funcionamiento del cerebro. Es un método de aprendizaje es supervisado, es decir que tiene una fase de entrenamiento. Existe un gran número de arquitecturas de RN; la propuesta para este trabajo es una perceptrón multicapa con dos capas ocultas completamente conectadas (Figura 3), debido a que la más empleada en percepción remota [8] por ser más eficiente que otros métodos para la clasificación de coberturas terrestres [15].

Para el entrenamiento de la red se propone extraer información estadística de los índices de vegetación para generar las entradas a la red neuronal, a estas entradas se le denomina como vector de características. Los parámetros estadísticos propuestos son los valores de media, moda, máximo y mínimo de cada índice utilizado. En total el vector de características esta compuesto de 12 parámetros: media, moda, máximo y mínimo de NDVI, GNDVI y CIR.

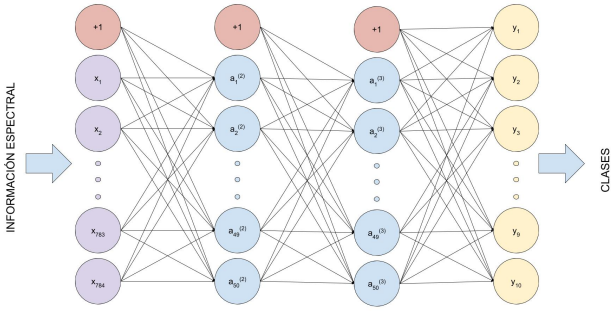


Fig. 3. Estructura de la red neuronal.

4. Análisis espectral

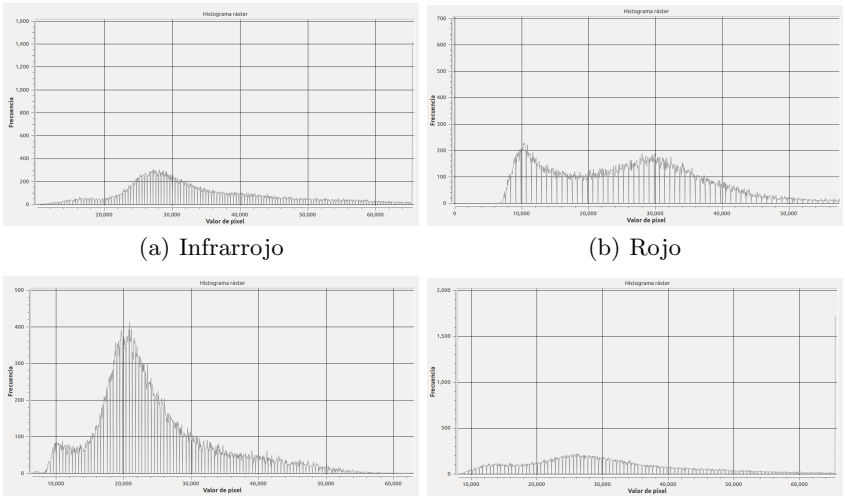
Como se ha mencionado en la sección 2, la literatura citada utiliza imágenes satelitales para la clasificación de diferentes texturas, sin embargo, dentro la textura vegetación no se ha reportado la clasificación de plantas específicas, posiblemente debido a la resolución especial de los sensores en los satélites. En la sección 3, se explican las razones de utilizar imágenes multiespectrales aéreas en los drones para obtener una mejor resolución espacial, las diferencias espectrales de cada planta nos sirve como un indicador único de información para cada tipo de planta, a esto se le conoce como firma espectral. En este apartado se hace una comparación de diferentes datos espectrales entre 2 plantas cítricas, limón y naranja (Figura 4), para marcar la diferencia espectral que existe en la diversidad de la flora, lo que evitará datos sesgados en el entrenamiento de la red neuronal y por lo tanto, mejor eficiencia en el clasificador.

Se calcularon los histogramas de las diferentes bandas espectrales (Figura 5), en donde se pudo observar que existen dos montículos en la frecuencia de algunos histogramas, estando la más notoria en la banda roja. Sin embargo, también es posible observar en la banda del borde rojo pero con mayor diferencia en número de la frecuencia. En las bandas infrarrojo y verde, solo es notoria las frecuencias que están relacionadas con el árbol de limón. Luego se calcularon los índices de vegetación: NDVI, GNVI y CVI para analizar los parámetros biofísicos de los árboles (Figura 6). De las imágenes de los índices calculados podemos decir que el árbol de limón tiene mayor densidad de biomasa y mayor concentración de nitrógeno, agua y clorofila en sus hojas que el árbol de naranja. Por lo tanto, ambos tienen información única que permitirá construir un clasificador eficiente.

5. Conclusiones

En este trabajo se propone utilizar una cámara multiespectral colocada en un dron para la captura de imágenes. Al usar imágenes dentro de la atmósfera terrestre eliminamos la interferencia que pueden tener los gases, la atmósfera o

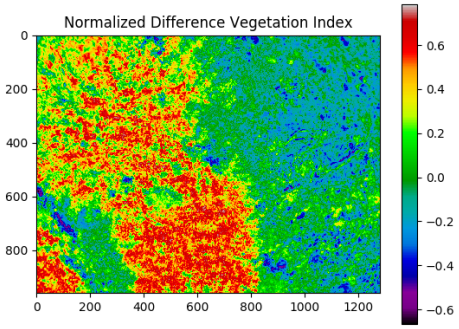
Fig. 4. En la parte superior izquierda de la imagen se encuentra un árbol de naranja, el árbol inferior es de limón.



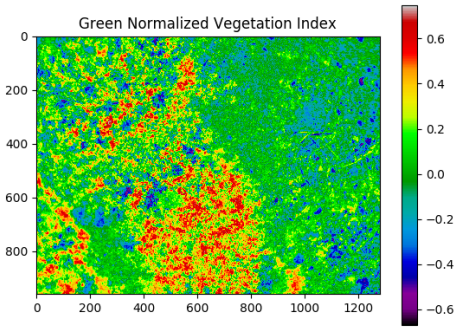
(a) Infrarrojo (b) Rojo

(c) Borde rojo (d) Verde

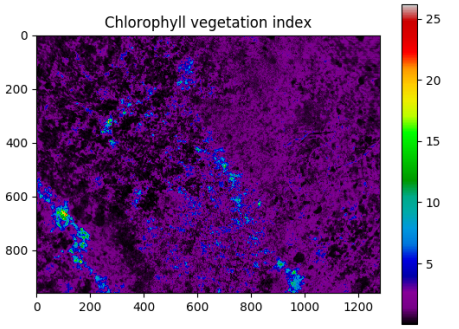
Fig. 5. Histogramas para las diferentes bandas espectrales.



(a) NDVI



(b) GNVI



(c) CVI

Fig. 6. Imágenes de los diferentes índices de vegetación.

las nubes en las imágenes espectrales. La corta distancia permite tener mejor resolución de las imágenes y por lo tanto obtener información más detallada de la cobertura terrestre. Utilizar un vector de descriptores minimiza el tiempo del entrenamiento. El uso de técnicas de aprendizaje máquina, en este caso redes neuronales, y procesamiento de imágenes mejora el análisis de la información espectral. Este trabajo contribuye en la innovación de los procesos en las actividades propias de la apicultura. Mejoraría la organización de los productores. Lograr una distribución óptima de las colonias.

6. Trabajos futuros

En el presente trabajo se propone la metodología de implementar una red neural para la clasificación de plantas apícolas específicas; se ha demostrado la factibilidad de usar redes neuronales e imágenes multiespectrales a baja altura para la clasificación; así como también se fundamenta el uso de los índices de vegetación como principales descriptores de la vegetación. Esta metodología podría ser capaz de aportar datos relevantes para impulsar la apicultura endémica de la península de Yucatán, al poder cuantificar de una manera precisa la densidad de los recursos de la flora alrededor de una colmena, incluso ser capaz de identificar el estado de salud de plantas; lo que permitiría optimizar las épocas de cosechas y crianza de la crianza (la crianza de abejas es una actividad en la que el apicultor proporciona alimento a las colmenas debido a la escasez de recursos) de abejas. Sin embargo basado en otros trabajos, sería interesante observar si la factibilidad puede mejorar utilizando arquitecturas de redes artificiales más recientes, como las redes convolucionales profundas o técnicas de deep learning.

En trabajos futuros se puede complementar la información del sensor espectral utilizado información espectral de las cámaras en los satélites; y mejorar o actualizar la información de la floración de vegetación polinífera. De igual manera se podría hacer un análisis de observar como los plaguicidas afectan el comportamiento de las plantas, realizar un estudio de la disminución de las areas acuicolas en busca de mejorar la distribución de los apiarios.

Referencias

1. Bunrit, S., Chanklan, R., Boonamnuay, S., Kerdprasop, N., Kerdprasop, K.: Neural network-based analysis of precipitation and remotely sensed data. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. vol. 1 (2016)
2. Buzzi, M.A., Rueter, B.L., Ghermandi, L.: Múltiples índices espectrales para predecir la variabilidad de atributos estructurales y funcionales en zonas áridas. Ecología austral 27(1), 55–62 (2017)
3. Carvajal Ramírez, F., Aguilar Torres, M.Á., Agüera Vega, F., Aguilar Torres, F.J.: Clasificación de una imagen multiespectral de satélite de alta resolución espacial mediante redes neuronales artificiales

4. Chang, J., Clay, D.E., Dalsted, K., Clay, S., O'Neill, M.: Corn (l.) yield prediction using multispectral and multidate reflectance. Agronomy journal 95(6), 1447–1453 (2003)
5. Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y.: Deep learning-based classification of hyperspectral data. IEEE Journal of Selected topics in applied earth observations and remote sensing 7(6), 2094–2107 (2014)
6. Chen, Y., Zhao, X., Jia, X.: Spectral–spatial classification of hyperspectral data based on deep belief network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8(6), 2381–2392 (2015)
7. Evora Jiménez, E.: Sistema de procesamiento de imágenes NIR e IR aéreas para agricultura de precisión. Ph.D. thesis, Universidad Central"Marta Abreu"de Las Villas. Facultad de Ingeniería Eléctrica. Departamento de Automática y Sistemas Computacionales (2016)
8. Foody, G.M.: Thematic mapping from remotely sensed data with neural networks: Mlp, rbf and pnn based approaches. Journal of Geographical Systems 3(3), 217–232 (2001)
9. Galárraga Cañizares, J.L.: Clasificador de hojas mediante Deep Learning. Ph.D. thesis, ETSI_Informatica (2017)
10. Gilabert, M.A., González-Piqueras, J., García-Haro, J.: Acerca de los índices de vegetación. Revista de teledetección 8(10) (1997)
11. Gómez-Casero, M., López-Granados, F., Peña-Barragán, J., Jurado-Expósito, M., García-Torres, L.: Caracterización espectral multitemporal de cultivos de regadío aplicando análisis discriminante y redes neuronales
12. Güemes Ricalde, F.J., Echazarreta González, C., Villanueva, R., Pat Fernández, J.M., Gómez Alvarez, R.: La apicultura en la península de yucatán. actividad de subsistencia en un entorno globalizado. Revista Mexicana del Caribe 8(16) (2003)
13. Li, F., Gnyp, M.L., Jia, L., Miao, Y., Yu, Z., Koppe, W., Bareth, G., Chen, X., Zhang, F.: Estimating n status of winter wheat using a handheld spectrometer in the north china plain. Field Crops Research 106(1), 77–85 (2008)
14. Lin, Z., Chen, Y., Zhao, X., Wang, G.: Spectral-spatial classification of hyperspectral image using autoencoders. In: Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on. pp. 1–5. IEEE (2013)
15. Lizarazo, I.: Clasificación de la cobertura y del uso del suelo urbano usando imágenes de satélite y algoritmos supervisados de inteligencia artificial. UD y la geomática (2), 4–18 (2008)
16. Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P.: Convolutional neural networks for large-scale remote-sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 55(2), 645–657 (2017)
17. Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N.: Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. pp. 4959–4962. IEEE (2015)
18. Mas, J.F.: Un método para combinar datos espectrales e información auxiliar en una red artificial neuronal. Anais XII Simpósio Brasileiro de Sensoriamento Remoto pp. 3543–3549 (2005)
19. Meza, R.R.B., Acuña, J.R.: Clasificación de la salinidad del suelo mediante imágenes de satélite y las redes neuronales artificiales. ECIPeru: Revista del Encuentro Científico Internacional 10(1), 4–8 (2013)
20. Morris, E.P., Gómez Enri, J., et al.: Teledetecion de Habitats Bentonicos en la Bahia de Cadiz, Espana. Ph.D. thesis, Universidad de Cádiz (2011)

21. Paruelo, J.M.: La caracterización funcional de ecosistemas mediante sensores remotos. Revista Ecosistemas 17(3), 4–22 (2008)
22. Paruelo, J.M.: La caracterización funcional de ecosistemas mediante sensores remotos. Revista Ecosistemas 17(3) (2008)
23. Ponce Suarez, C.Y.: Análisis multitemporal de la cobertura vegetal del valle interandino del chota e identificación de zonas de restauración ecológica. B.S. thesis (2017)
24. Schneider, G., Hadad, A.J., Kemerer, A.: Implementación de un software para el análisis de imágenes aéreas multiespectrales de caña de azúcar [implementation of software for the analysis of multispectral aerial images sugarcane]. Ventana Informática (28) (2013)
25. Suarez, A., Jiménez, A., Franco, M.C., Cruz-Roa, A.: Clasificación automática de coberturas del suelo en imágenes satelitales utilizando redes neuronales convolucionales: Un caso aplicado en parques nacionales naturales de colombia
26. Tovar, S.A.O., Valenzuela, D.M.Q., Ortega, Y.D.M., Pastrana, D.A.M., Medina, W.A.O., Portilla, D.E.P.: Zonificación de unidades productivas con el uso de herramientas satelitales y actividades de formación interdisciplinar. Revista del Sistema de Ciencia Tecnología e Innovación (SENNOVA) 2(1), 98–121 (2017)
27. Vincini, M., Frazzi, E., D'Alessio, P.: A broad-band leaf chlorophyll vegetation index at the canopy scale. Precision Agriculture 9(5), 303–319 (2008)

ISSN 1870-4069

Redes neuronales aplicadas al control de riego usando instrumentación y análisis de imágenes para un micro-invernadero aplicado al cultivo de Albahaca

Martín Gerardo Vázquez Rueda, Marlen Ibarra Reyes,
Francisco Gerardo Flores García, Héctor Aurelio Moreno Casillas

Instituto Tecnológico de la Laguna,
Tecnológico Nacional de México Torreón Coahuila,
México

mart2vazquez@yahoo.com, marlen_mir@hotmail.com,
{francisco.floresgarcia, honerom}@gmail.com

Resumen. El presente artículo muestra el desarrollo de un control con Inteligencia Artificial para un micro-invernadero utilizando redes neuronales artificiales multicapa de retro-propagacion. En el invernadero se analizan las plantas y se obtienen datos por medio de sensores, que permiten controlar y automatizar el riego. Mediante la interfaz se pueden observar los datos que monitorea el Sistema: la temperatura ambiente y la humedad relativa en el invernadero; la temperatura y humedad de cada una de las plantas, el tamaño de la planta, asi como el porcentaje sano y dañado de cada una de las plantas. El experimento fue realizado durante 44 días, probando las redes neuronales en un rango de humedad de 20 a 21% y una temperatura estable promedio de 30°C. Con estas condiciones se encontró que las plantas de albahaca (Ocimum basilicum), se desarrollan favorablemente. El error para el control de Riego de la Red Neuronal Artificial (RNA) fue de solo 1.67%.

Palabras clave: Control inteligente, micro invernadero, red neuronal artificial, temperatura, humedad, riego.

Neural Networks Applied to Irrigation Control Using Instrumentation and Image Analysis for a Micro-Greenhouse Applied to the Cultivation of Basil

Abstract. This article shows the development of a control using Artificial Intelligence for a micro-greenhouse using multilayer Backpropagation Artificial Neural Networks. In the micro-greenhouse the plants are analyzed and data obtained by sensors allowing to automate and control irrigation. Through the interface, the user can monitor the data of the system, the application shows: ambient temperature and relative moisture inside the greenhouse, temperature

and moisture for each plant, the size and percentage of healthy and damaged state of each plant. The experiment was performed for 44 days, testing the Artificial Neural Networks (ANN) within a moisture range of 20% to 21%, and an average temperature of 30°C. With these conditions it was found that basil plants (*Ocimum basilicum*) developed favorably. Error for the Irrigation Control of the ANN was only of 1.67%.

Keywords: Intelligent control, micro greenhouse, artificial neural network, temperature, moisture, irrigation.

1. Introducción

1.1. Agricultura de precisión

El desarrollo de la agricultura ha derivado el uso de herramientas, actividades, materiales y estructuras que se emplean para la protección de los cultivos, conocido como la agricultura protegida. La agricultura protegida busca principalmente evitar las restricciones que el medio ambiente impone al desarrollo de las plantas cultivadas y se puede definir como "toda estructura cerrada, cubierta por materiales transparentes o semitransparentes, que permite obtener condiciones artificiales de microclima para el cultivo de plantas y/o flores, mediante la cual se tiene el propósito de alcanzar un adecuado crecimiento vegetal, aumentar rendimientos, mejorar la calidad y obtener excelentes cosechas" [1].

La agricultura protegida en México está en amplio crecimiento y desarrollo. De acuerdo con la SAGARPA, en el año 2000 solo existían 790 hectáreas de agricultura protegida en el país, con un crecimiento anual de 1,500 hectáreas para el 2015 el mismo organismo público un total de 23, 521 hectáreas lo que representa un crecimiento de 22, 461 hectáreas en solo 15 años. Según datos de la Asociación Mexicana de Horticultura Protegida (AMHPAC), hay un total de 25,814 instalaciones activas de agricultura protegida [2].

La comarca lagunera cuenta con una extensión territorial de 500,000 hectáreas y está situada en la parte suroeste del estado de Coahuila y noroeste del estado de Durango, la cual se caracteriza por pocas precipitaciones de lluvia, limitados recursos hídricos y su clima seco, caluroso en verano y frío en invierno, en donde las temperaturas siempre cambiantes y extremas llegan a alcanzar los 0°C a 45°C, dependiendo de la estación del año, para combatir estos retos ambientales algunos productores han optado por la alternativa del uso de la agricultura protegida. La región lagunera del estado de Coahuila es conocida por actividades específicas de agricultura, cuenta con la mayor superficie de riego en el estado, sobresalen los cultivos de forrajes, melón, nuez, algodón y maíz: así como la producción de hortalizas, como tomate y chile en invernaderos y malla sombra [3].

Coahuila representa el 1.6% de hectáreas en agricultura protegida a nivel nacional, con el 0.8% de instalaciones activas. Siendo la comarca lagunera de Coahuila la región del estado con la mayor superficie que hace uso de la agricultura protegida, representando 81% de hectáreas cubiertas y el 37.14% de instalaciones en el estado, de

las instalaciones que se encuentran en la comarca lagunera el invernadero es el tipo de agricultura protegida más utilizada con un 66.12% [1].

Con un crecimiento promedio anual del 12% la agricultura protegida mexicana cuenta con una infraestructura instalada cuyo valor es mayor a los 3,500 millones de dólares [2]. De las estructuras empleadas para proteger cultivos, los invernaderos permiten modificar y controlar de forma más eficiente los principales factores ambientales que intervienen en el crecimiento y desarrollo de las especies vegetales [4]. Los invernaderos y casa sombra se convierten en una de las economías más pujantes del sector primario. Siendo el invernadero el elemento cualitativamente más importante del sistema de producción en agricultura protegida, debido a que de él depende en gran medida la capacidad productiva [5].

Los sistemas que tienen alta Tecnología: en este nivel se incluyen instalaciones que cuentan con control climático automatizado (mayor independencia del clima externo), riegos, automatizados y de precisión, inyecciones de CO2, para ello cuentan con sensores y dispositivos que operan los sistemas de riego y ventilación, pantallas térmicas para el control de la iluminación y cultivo en sustratos [6].

El uso de invernaderos de alta tecnología se justifica mediante la corriente mundial de calidad en la que se vive. Los mercados son cada vez más exigentes en calidad, inocuidad, presentación y certificación del contenido [7].

Los invernaderos modernos son acondicionados con mecanismos y equipos necesarios para controlar temperatura, luminosidad, humedad ambiental y del sustrato, ventilación, aireación, aporte de CO2, riegos y fertilización.

El desarrollo de tecnologías aplicadas a invernaderos fomenta el aumento de calidad y conocimiento respecto a los cultivos. Si el monitoreo y análisis se realizan de forma automática crea una herramienta importante para incluir a las características de un invernadero de alta tecnología, que permitirá no solo cultivar para producción, sino que también se obtenga la capacidad de realizar análisis de cultivos, mejorando la toma de decisiones y generando una comprensión mayor del desarrollo de las plantas y cultivos.

1.2. Inteligencia artificial

La inteligencia artificial nace en una reunión celebrada en el verano de 1956 en Dartmouth (Estados Unidos) en la que participaron los que más tarde han sido los investigadores principales del área. Existen muchas definiciones de lo que es la inteligencia artificial. Sin embargo, todas ellas coinciden en la necesidad de validar el trabajo mediante programas [8].

Las redes neuronales son programas de la inteligencia artificial capaces de simular algunas de las funciones de aprendizaje del ser humano. Una red neuronal obtiene experiencia analizando automática y sistemáticamente los datos para determinar reglas de comportamiento; con base en ellas, puede realizar predicciones sobre nuevos casos. Estas técnicas se aplican a problemas de clasificación y series de tiempo e identifican conexiones con cosas que otras técnicas no pueden, porque utilizan relaciones lineales y no lineales [9].

Estas redes surgieron a partir de las ideas en la publicación de McCulloch y Pitts, donde se postula que las neuronas funcionan como dispositivos booleanos. Este

postulado fue criticado como teoría biológica, pero permitió generar una neurona como un modelo lineal seguido de una función activación booleana.

Aquí, la función lineal representa la sinapsis (unión entre neuronas) y la agregación de la información, mientras que la función no lineal representa el procesamiento que hace la neurona. La función lineal se ve en la ecuación (1) [10]:

$$r=f(\sum_{i=1}^{n} x_i w_i + b), \quad (1)$$

donde x_i son los datos de entrada, w_i son los pesos sinápticos y b es un factor de polarización, el resultado de r es procesado de tal manera que a la salida de un valor de uno o cero, utilizando f como función de transferencia *harlim*.

El proceso de aprendizaje consiste en hallar los pesos que codifican los conocimientos. Una regla de aprendizaje hace variar el valor de los pesos de una red hasta que estos adoptan un valor constante, cuando esto ocurre se dice que la red ya "ha aprendido" [11].

El perceptrón fue el primer modelo de red neuronal, el cual utiliza la neurona artificial de la Figura 1, además de que tenía su arquitectura en tres capas y un algoritmo de aprendizaje. Fué probado como un detector de caracteres ópticos, por lo que el campo de redes neuronales se inició como una forma de procesar imágenes [10].

El algoritmo backpropagation es el método de aprendizaje más frecuentemente utilizado en el entrenamiento de las redes. Este algoritmo de aprendizaje es una generalización de la regla de corrección de error de Widrow-Hoff. El error es la diferencia entre la salida que proporciona la RNA y la salida que se pretende obtener. Los diferentes pesos de las conexiones de las neuronas son corregidos mediante iteraciones que pretenden minimizar el error [12].

El presente trabajo se centra en el desarrollo de tecnologías aplicadas a invernaderos para el análisis de cultivos que permitan la obtención de conocimiento. Desarrollando una estación de trabajo automática que permitan el control de un microinvernadero de pruebas mediante una programación de inteligencia artificial que por medio de la obtención de variables realicen una toma de decisiones eficaces para optimizar el uso de agua para producir un cultivo de mejor calidad.

2. Materiales y métodos

2.1 Micro-invernadero

La estación de trabajo consta de una estructura con paredes de nylon transparente, contando con un sistema de circulación de aire. Emulando un invernadero en la cual se monitorea y controla el desarrollo de las plantas. Para esto se cuenta con diversos sistemas electrónicos, informáticos y estructurales, los cuales permiten procesamiento de imágenes, lecturas de sensores, controles para temperatura y riego. La estructura del invernadero está diseñada para contener 6 plantas. El sistema es de diseño propio.

Se cuenta con cinco sistemas diferentes: Sistema base, sistema de sensores, sistema riego, sistemas de riego, sistema de iluminación. Los sistemas están programados en el programa Labview para su interfaz y parte del control y cuentan con programas de Matlab y microcontroladores para los elementos matemáticos y electrónicos.

El sistema base, es el que se encarga de la obtención de imágenes de las plantas mediante una cámara Microsoft LifeCam HD-6000, y el sistema ventilación del invernadero.

El sistema sensores engloba un sensor infrarrojo MXL90614 para medir la temperatura del ambiente dentro del invernadero y la temperatura objeto, es decir de cada una de las plantas, así como los seis sensores de humedad YL-69, cada uno correspondiente a una planta.

El sistema de riego es el encargado de suministrar determinadas cantidades de agua a las plantas. El módulo cuanta con tres válvulas solenoides 2/2, tubería de agua, una bomba de agua, y un contenedor de agua, que proporciona el gasto adecuado acorde al tipo de planta [13].

2.2 Metodología de experimento

Para realizar estas pruebas se dividieron las plantas en 3 grupos quedando distribuidos como se muestra en la Tabla 1. La división de estos grupos se realizó debido a que se necesitaban obtener diferentes estados en las plantas y suministrando las diferentes cantidades de agua a cada sección, para lograr el objetivo de tener diferentes casos para el entrenamiento de la red neuronal artificial.

Tabla 1. Distribución de grupos para plantas de Albahaca.

Grupo	Plantas
1	Planta 1 y 6
2	Planta 2 y 5
3	Planta 3 y 4

Cada planta cuenta con un sensor de humedad en suelo YL-69, los cuales se insertaron de forma transversal en la maceta aproximadamente a la mitad de lo alto de la maceta. Una vez realizado el análisis, la lectura del sensor de humedad de agua en suelo arroja valores que van de 0 para tierra seca, es decir sin ninguna cantidad de agua, y hasta 25 que es la saturación del sensor y representa el 25% de humedad. Siendo el 100% de humedad el líquido directamente.

El sensor de temperatura infrarrojo MXL90614 es calibrado de fábrica para entregar temperaturas en rangos de -40 a 125 °C para temperaturas ambiente y -70 a 380 °C para temperaturas objeto. También cuenta con el sensor de humedad relativa HIH-4030 entrega valores de 0.8V a 3.9V escalados en porcentajes (0% a 100%) [14].

A partir del procesamiento de imagen se obtiene un porcentaje de follaje en buen estado y mal estado de la vista superior de la planta, además se proporciona la cantidad de área analizada de la planta solamente, discriminando las partes que no corresponden a la planta. La captura de la imagen es realizada por la cámara Microsoft LifeCam HD-6000.

Para el procesamiento de imagen se tuvieron en cuenta las siguientes consideraciones [15]: El color de fondo alrededor de la planta a analizar deber ser blanco, esto para facilitar la detección del área que abarca la planta y discriminar el resto. La iluminación del invernadero debe estar encendida al momento de realizar el análisis de las plantas.

Tabla 2. Horario de toma de muestras durante el análisis.

M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8
00:30	03:30	06:30	09:30	12:30	15:30	18:30	21:30

Tabla 3. Promedio de datos recopilados por día de la planta 1.

Día	Temperatura Ambiente	Humedad Relativa	Temperatura Objeto	Humedad en Suelo	% Bueno	% Malo	Área Total
1	28.43	17.12	29	21.00	98.75	1.25	159872.00
2	29.35	17.83	28.01	21.00	98.69	1.31	159213.00
3	32.56	18.53	30.85	20.22	97.27	2.73	156719.00

Tabla 4. Promedio de datos recopilados por día de la planta 6.

Día	Temperatura Ambiente	Humedad Relativa	Temperatura Objeto	Humedad en Suelo	% Bueno	% Malo	Área Total
1	28.43	17.12	29.07	20.86	97.09	2.91	159878.00
2	29.35	17.83	28.41	20.34	96.98	3.02	159226.00
3	32.56	18.53	30.47	20.00	93.74	6.26	157718.00

El algoritmo de procesamiento de imagen se centra en la detección de zonas amarillas en las hojas. El color amarillo en las hojas de una planta significa que hay un déficit de clorofila (clorosis) en la misma, se puede deber a varios motivos, los más comunes son: plagas que consuman la clorofila de la planta, falta de nutrientes, insuficiente exposición a la luz solar, falta de agua suficiente. Se debe tener cuidado que el campo de visión de la cámara y la captura realizada por esta contenga solamente a la planta a analizar.

Grupo de Estudio 1, Perteneciente a la sección 1 de riego. Sujeto de Estudio: Planta 1 y Planta 6. Las pruebas se realizaron del día 1 de Mayo del 2017 al día 14 de Junio del 2017, total 44 días.

Se realizaron 8 muestreos al día los cuales se seccionaron durante las 24 horas obteniendo así un muestro cada 3 horas. En la Tabla 2, se indica la hora a que se realizó el muestreo.

Total de muestras: Al finalizar se obtuvieron un total de 750 lecturas**,** siendo 350 correspondientes a la Planta 1 y 350 correspondientes a la Planta 6. Para este análisis se estableció un riego diario a las 14:30 horas y para este grupo de estudio se estableció 0 ml de agua.

Los datos recopilados por ambas plantas son demasiados para ser presentados de forma completa, en la Tabla 3 y 4, se muestran algunos de los datos recopilados por día y para las seis regiones bajo prueba.

La figura 1, muestra las imágenes obtenidas para algunos casos en distintas plantas en diferentes etapas del proceso y bajo diferentes condiciones de humedad, las imagines son preprocesadas con algoritmos de tratamiento de imágenes, de las cuales se extraen el área total de la planta y el índice de PB porcentaje en buen estado.

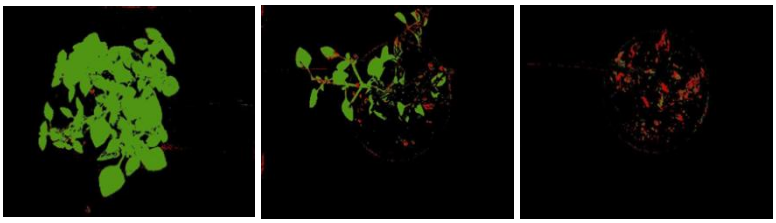


Fig. 1. Muestra de las imágenes originales en diferentes etapas del proceso.

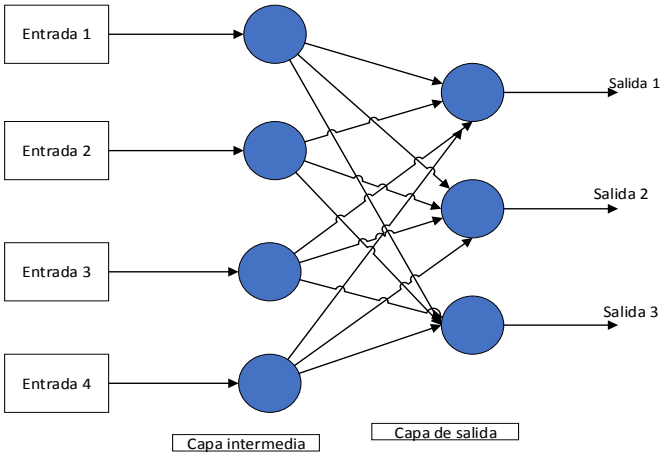


Fig. 2. Red Neuronal Artificial (RNA) para control de riego

Una red neuronal de retropropagación emplea el algoritmo del gradiente descendente, tal como la regla de aprendizaje Widrow-Hoff, en la cual, los pesos de la red se mueven a través de la parte negativa del gradiente de la función de desempeño. El término, retropropagación, se refiere a la forma en que el gradiente es calculado para redes multicapa no lineales, donde el error se calcula para ajustar los valores de los pesos propagándose hacia atrás [16]. Hay un gran número de variaciones, en el algoritmo básico, que están basadas en las técnicas de optimización estándar, tal como gradiente conjugado o el método de Newton [12].

Se empleó el ambiente de programación Matlab, para la generación, prueba y validación de las redes. Usando el algoritmo de Levenberg-Marquardt para su entrenamiento; función de aprendizaje por gradiente descendente con momento; como función de desempeño el cuadrado medio del error; una función de transferencia Hardlim para las capas ocultas y función de transferencia Hardlim para la capa de salida.

La red se diseñó con tres salidas, que pertenecen a tres niveles de riego, Bajo, Normal o Estable y Máximo o Excesivo. Con los datos obtenidos del experimento, donde las plantas tienen un desarrollo favorable en condiciones estables, se utilizaron para simular las redes neuronales del control de riego. La figura 2 muestra la arquitectura de la red.

3. Resultados y discusión

Resultados de RNA de control de riego en sección 1. Tomando los datos de la humedad en suelo (HS) y el porcentaje de estado bueno (PB) de las plantas 1 y 6 como entradas para la red neuronal de la sección 1 del riego se procesaron los 20 casos y se obtuvo la siguiente respuesta de la red, representada en la Tabla 5.

Tabla 5. Respuesta de la RNA para riego de la sección 1 del micro-invernadero.

Caso	Entrada 1 HS1	Entrada 2 HS6	Entrada 3 PB1	Entrada 4 PB6	Salida Deseada	Salida de la Red
1	21	21.5	98.75	98.1	0 1 0 = Humedad Estable	0 1 0 = Humedad Estable
2	20	21.5	98.69	98.27	0 1 0 = Humedad Estable	0 1 0 = Humedad Estable
3	21	21	98.27	97.91	0 1 0 = Humedad Estable	0 1 0 = Humedad Estable
4	22	21.6	98.59	98.4	0 0 1 = Humedad Excesiva	0 1 0 = Humedad Estable

Tabla 6. Respuesta de la RNA para riego de la sección 2 del micro-invernadero**.**

Caso	Entrada 1 HS2	Entrada 2 HS5	Entrada 3 PB2	Entrada 4 PB5	Salida Deseada	Salida de la Red
1	21	20.7	98	97.87	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**
2	21	20.7	98.51	97.77	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**
3	21	20.74	97.99	97.77	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**

De la tabla 5 con todos los datos extraídos en la sección de riego 1 del micro-invernadero la red da un 95% de aciertos en su salida, 5% de error.

Resultados de RNA de control de Riego en sección 2. Tomando los datos de la humedad en suelo y el porcentaje de estado bueno de las plantas 2 y 5 como entradas para la red neuronal de la sección 2 del Riego se procesaron los 20 casos y se obtuvo la siguiente respuesta de la red, representada en la Tabla 6.

De la tabla 6 se muestra que en la sección de riego 2 del micro-invernadero la red fue 100% asertiva en su salida.

Resultados de RNA de control de riego en sección 3. Tomando los datos de la humedad en suelo y el porcentaje de estado bueno de las plantas 3 y 4 como entradas para la red neuronal de la sección 3 del Riego se procesaron los 20 casos y se obtuvo la siguiente respuesta de la red, representada en la Tabla 7.

Tabla 7. Respuesta de la RNA para riego de la sección 3 del micro-invernadero.

Caso	Entrada 1 HS3	Entrada 2 HS4	Entrada 3 PB3	Entrada 4 PB4	Salida Deseada	Salida de la Red
1	21.3	21.2	98.28	97.6	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**
2	21	21	98.1	97.78	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**
3	21	21.2	98.4	97.87	0 1 0 = Humedad Estable	**0 1 0 = Humedad Estable**

De la tabla 7 se muestra que en la sección de riego 3 del invernadero la red fue 100% asertiva en su salida.

Tomando en consideración las tres secciones, se puede concluir que la RNA del control de Riego tiene un margen de error del 1.67%. La cantidad de agua requerida por las plantas de Albaca, así como las condiciones necesarias para su crecimiento [13].

4. Conclusiones y trabajo futuro

Se cumplio con el objetivo principal de diseñar e implementar un control inteligente a la estación de trabajo del micro-invernadero, el cual permite mantener un riego eficiente para mejorar el desarrollo de los cultivos, sin tener que estar al pendiente las 24 hrs del invernadero para realizar los cambios manualmente. La implementación del control inteligente permite: Un control del sistema de riego en el cual según el estado de humedad de la planta es la cantidad de agua que suministra, si la planta se encuentra normal sigue con la misma cantidad (60ml) si la planta esta deshidratada aumenta la cantidad de agua hasta que la planta se regula a su estado estable y si la planta se encuentra sobre hidratada disminuye la cantidad de agua suministrada hasta que la planta vuelva a estar en condiciones estables [13].

Visualizar las últimas imágenes de las seis plantas capturadas por la cámara y las seis imágenes después del procesamiento de imagen, permitiendo conocer el estado actual de las plantas de forma visual.

Permite tener un historial de análisis de cada planta a través de reportes los cuales son guardados en tablas de Excel. Con los reportes se puede observar las lecturas de los sensores como lo son la temperatura ambiente, temperatura directa, humedad relativa y humedad de agua en suelo, además de los resultados arrojados por el procesamiento de imagen, las fechas y horas de análisis.

Como trabajos a Futuro se tiene en cuenta lo siguiente: Implementar un sistema de calefacción. Añadir al sistema de sensado de macronutrientes esenciales. Que integrándolo en un sistema con inteligencia artificial pueden mejorar el proceso de producción de plantas, generando diferentes estrategias de control con inteligencia artificial incluyendo otras variables y procesos.

Agradecimientos. Se agradece al soporte brindado por el CONACYT, al Tecnológico Nacional de México, al Instituto Tecnológico de la Laguna, por el apoyo brindado para el desarrollo de este proyecto.

Referencias

1. Santos-Bielinski, M., Obregón-Henner, A., Salamé-Teresa P.: Producción de Hortizales en Ambientes Protegidos: Estructuras para la Agricultura Protegida, Publicación del Departamento de Horticultural Sciences, UF/IFAS Estension, University of Florida, 2(5), pag. 99–110 (2016)
2. AMHPAC: Asociación Mexicana de Horticultura Protegida Agricultura Protegida en México. http://www.amhpac.org/es/index.php/homepage/agricultura-protegida-en-mexico (2015)
3. Secretaria de Desarrollo Rural del Gobierno de Estado de Coahuila de Zaragoza.: Programa Estatal de Desarrollo Rural 2011-2017. http://coahuila.gob.mx/archivos/pdf/Publicaciones /DESARROLLO%20RURAL.pdf (2017)
4. Juárez-López, P., Bugarín-Montoya, R., Castro-Brindis, R., Sánchez-Monteón, A.L., Cruz-Crespo, E., Juárez-Rosete, C.R., Alejo-Santiago, G., Balois-Morales, R.: Estructuras utilizadas en la agricultura protegida. Revista Fuente, Año 3, 8, pp. 21–27 (2011)
5. Ortega-Martínez, L.D., Ocampo-Mendoza, J., Sandoval-Castro, E., Martínez-Valenzuela, C., Huerta-De La Peña, A., Jaramillo-Villanueva, J.L.: Caracterización y funcionalidad de invernaderos en Chignahuapan, Puebla. Colegio de Postgraduados Campus Puebla, Revista Bio Ciencias, pp. 261–270 (2014)
6. SAGARPA: Servicio de Información Agroalimentaria y Pesquera. Agricultura Protegida. http://www.gob.mx/siap/documentos/agricultura-protegida (2017)
7. Pacheco, A.J: Fundamentos técnicos para el diseño y construcción de invernaderos. (AGRO), 51, pp. 12–19 (2008)
8. Torra, V.: La Inteligencia Artificial. Lychnos 7, pp. 14–19 (2011)
9. García-Fernández, L.A.: Usos y aplicaciones de la inteligencia artificial. La ciencia y el hombre, 17(3), pp. 29–32 (2004)
10. Ramírez, Q., Juan, A., Chacón, M., Mario, I.: Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. (RIEE&C), Revista de Ingeniería Eléctrica, Electrónica y Computación, 9(1), pp. 7–16 (2011)
11. Sotolongo, G., Guzmán, M.V.: Aplicaciones de las redes neuronales. El caso de la bibliometría, Ciencias de la Información, 32(1), pp. 27–34 (2001)
12. Martín del Brío, B., Sanz-Molina, A.: Redes Neuronales y Sistemas Difusos. Editorial Rama (2006)
13. Martinez-Beranal, L.F.: Determinación del requerimiento hídrico de la albahaca (Ocimun basilicum) y comparación de dos sistemas de microirrigación (microaspersión y goteo) en condiciones de clima frio, bajo invernadero https://www.researchgate.net/ publication/306091339_Determinacion_del_requerimiento_hidrico_de_la_albahaca_Ocimun_basilicum_y_comparacion_de_dos_sistemas_de_microirrigacion_microaspersion_y_goteo_en_condiciones_de_clima_frio_bajo_invernadero (2018)
14. Galicia-Reyes, J.I.: Aplicación para control vía remota, integración de sensores y sistema de riego a invernadero de pruebas automático para el monitoreo de plantas. Tesis de maestría, Instituto Tecnológico de la Laguna (2015)
15. Avilés de León, Y.: Estación de Trabajo Automática para el Monitoreo de Plantas, Tesis de maestría, Instituto Tecnológico de la laguna (2014)

16. Valencia Reyes Marco Antonio, Yáñez Márquez Cornelio, Sánchez Fernández Luis Pastor.: Algoritmo Backpropagation para Redes Neuronales: conceptos y aplicaciones, IPN CIC 125(1), 1–14, 2006.

ISSN 1870-4069

Sesgo cognitivo y redes neuronales artificiales aplicados en una BCI para clasificación de señales neuronales biológicas a palabras dicotómicas "SI-NO" obtenidas mediante un EEG: Speechless Talk

Bladimir Serna[1], Rosario Baltazar[1], Martha Rocha[1],
Delia Irazú Hernández Farías[2], Miguel Ángel Casillas-Araiza[1], Victor Zamudio[1]

[1] Tecnológico Nacional de México,
Instituto Tecnológico de León,
México

[2] Universidad Politecnica de Valencia, Valencia,
España

{bladimir.serna, marthaalicia.rocha}@itleon.edu.mx, charobalmx@yahoo.com.mx,
dhernandez1@dsic.upv.es, miguel.casillas@gmail.com, vic.zamudio@ieee.org

Resumen. La tecnología, al igual que las ciencias exactas y la computación juegan un papel muy importante en la vida diaria, así mismo, son de gran ayuda para la resolución de problemas de diferente índole, tales como problemas médicos, tecnológicos, la industria armamentista, escolares, entre otros. En este ocasión, se muestra un enfoque orientado a mejorar la comunicación no verbal de las personas y, a su vez, brindar una alternativa de mejora en la calidad de vida de pacientes con problemas de comunicación verbal. La unión de éstas desencadena en aplicaciones de gran utilidad y brinda alternativas eficaces de comunicación. Por tal motivo, en este trabajo se presenta el desarrollo del proyecto denominado *Speechless Talk* cuyo principal objetivo consiste en interpretar de manera dicotómica palabras tales como "SI" o "NO" a través de señales neuronales biológicas obtenidas con el uso de EEG (Electroencefalogramas), que posteriormente son clasificadas utilizando Redes Neuronales Artificiales.

Palabras clave: Redes neuronales artificiales, EEG (electroencefalograma), MindFlex, dicotomía, interfaces cerebro máquina.

Cognitive Bias and Artificial Neural Networks Applied in a BCI for Classification of Biological Neural Signals to Dichotomous Words "YES-NO" Obtained Through an EEG: Speechless Talk

Abstract. Technology, like the exact sciences and computers play a very important role in daily life, likewise, they are of great help for the

resolution of problems of different nature, such as medical, technological problems, the arms industry, school , among others. On this occasion, an approach is shown aimed at improving the non-verbal communication of people and, at the same time, providing an alternative to improve the quality of life of patients with verbal communication problems. The union of these triggers in applications of great utility and offers effective alternatives of communication. For this reason, this paper presents the development of the project called *Speechless Talk* whose main objective is to interpret in a dichotomous way words such as "YES" or "NO" through biological neural signals obtained with the use of EEG (Electroencephalograms), which are subsequently classified using Artificial Neural Networks.

Keywords: ANN (articial neural network), EEG (electroencephalogram), MindFlex, dichotomous, BCI (brain computer interfaces).

1. Introducción

Según el INEGI (Instituto Nacional de Estadística y Geografía), para el 2010 existían un total de 4,527,784 casos documentados con personas que presentan algún tipo de discapacidad en México, de los cuales el 8.93 % (es decir, aproximadamente 401,538 personas [12]), presentan algún tipo de problema para comunicarse de manera oral.

Lo cual trae consigo una gran problemática en varios sectores, partiendo desde el paciente que se encuentra postrado en cama y no puede expresar de manera adecuada sus necesidades o síntomas (dificultando con esto la comunicación paciente-médico), así como la frustración que puede llegar a sentir dicho paciente al no poder expresar de manera adecuada lo que requiere, hasta la persona que necesita realizar sus actividades cotidianas lo mas normal posible y no tiene forma alguna de comunicarse.

Por otra parte, personas que se encuentran en estado vegetativo, mientras conserven la conciencia, podrían tener una alternativa de comunicación con sus familiares, médicos y personal que los atiende.

La tecnología juega un papel muy importante en nuestras actividades cotidianas y cada día es más frecuente acceder a esta, por tal motivo, se considera de suma importancia la implementación de soluciones tecnológicas a problemas comunes y apoyar a mejorar la calidad de vida de las personas.

Un buen ejemplo, es el caso del científico Steven Hawking quien padecía ELA (Esclerosis Lateral Amiotrófica)[3]. Para mejorar la forma de comunicación de Hawking, se creó una versión mejorada de la tecnología *Swiftkey*, que consiste en la interpretación de frases a través de palabras detectadas por una serie

[3] ELA es una enfermedad degenerativa de tipo neuro-muscular,que se caracteriza por destruir células y neuronas motoras, tanto en extremidades como en la musculatura bulbar [1], de manera paulatina

de sensores colocados tanto en la mejilla derecha así como en la garganta del usuario, para que cuando este emule hablar, los sensores interpreten las palabras que intentó pronunciar.

Dichas palabras se someten al software *Swiftkey* el cual intenta colocarlas de la mejor manera posible. Para realizar dicha predicción, el software tiene precargadas todas y cada una de las publicaciones realizadas por Hawking hasta el día de su muerte (14-03-2018), para así encontrar coincidencias en lo que comúnmente es llamado "forma de escribir" y determinar la frase exacta de lo que se emula pronunciar.

Una vez que se cuenta con la interpretación, una computadora colocada en la silla de ruedas del científico se encargaba de leer a través de los altavoces, lo que Hawking ha querido pronunciar.

El nombre de dicho sistema es *ACAT (Assistive Context-Aware Toolkit)*, desarrollado por la empresa INTEL [13]; una de las principales desventajas de este completo sistema es su elevado precio de producción, lo cual lo hace inaccesible para el público en general.

Otro gran avance con el que contamos hoy en día es el proyecto denominado *Brain-To-Text* [2] desarrollado por la Universidad de Bremen en Alemania, el cual consiste en la traducción de ondas cerebrales a texto, mediante un algoritmo diseñado por ellos mismos.

Para poder realizar dicha prueba, Herff y su equipo de trabajo tuvieron que movilizarse a un hospital en EEUU especializado en pacientes con epilepsia. Los experimentos se consideran un método totalmente invasivo, ya que los electrodos son colocados directamente en el encéfalo para obtener los datos; como resultado se obtuvieron un total de 7 palabras bien clasificadas de diez almacenadas en una base de datos, pero cuando se incrementan las palabras a 100, únicamente 43 de ellas fueron correctamente clasificadas [2] . Al ser un método totalmente invasivo, se considera poco funcional y de difícil acceso al público en general.

El proyecto denominado "habla imaginada" desarrollado por la red ICA (Inteligencia Computacional Aplicada) en el INAOE (Instituto Nacional de Astrofísica, Óptica y Electrónica), consiste en la clasificación de cinco palabras (Izquierda, Derecha, Arriba, Abajo, Seleccionar) [3] obtenidas de la actividad cerebral apoyados en potenciales visuales evocados; dicho proyecto cuenta con un 70 % de clasificación correcta lo que lo hace un aporte bastante confiable, para este caso se utiliza una diadema denominada *EMOTIV Epoc* que cuenta con 14 canales para la obtención de señal y tiene un costo alto, adicionalmente, se requiere de un entrenamiento especial para poder implementar dicho sistema.

Con el objetivo solucionar los problemas de comunicación oral, se propone la implementación de un sistema de comunicación no verbal que permita a pacientes con comunicación oral limitada o nula, expresar de manera sencilla sus necesidades básicas, así como expresar incomodidades o responder cosas tan básicas como si tiene hambre, le duele algo o tiene alguna incomodidad en el momento, a través de un simple "SI" o "NO".

Por tal motivo, se crea **Speechless Talk**, que es una aplicación de escritorio que permite a personas con comunicación limitada a nula, expresar palabras

básicas, tales como: "SI" o "NO", solamente con pensarlo; Apoyados en tecnicas computacionales inteligentes.

2. Marco referencial

Para brindar una alternativa de comunicación no verbal y ofrecer una mejora a las técnicas computaciones existentes hasta hoy en día, se presenta el proceso de comunicación a nivel neuronal, que toma en cuenta que necesitamos realizar un prototipo económico y que no requiera de estímulos externos diferentes al de pensar. Dado que en psicología existe un término conocido como "sesgo cognitivo", el cual indica que existe cierta información básica almacenada en algunas neuronas y dicha información está presente casi por inercia (tales como nuestro nombre, nuestra edad, nombre de nuestros padres, lugar donde vivimos, entre otros) también indica que existe información que requiere mayor esfuerzo para acceder a ella o para procesarla por ejemplo la raíz cuadrada de 1326, recordar que comimos ayer por la mañana, o pensar de que color estaba vestido el profesor de matemáticas, etc. [5] Esta información a su vez puede ser causante de algunos errores en la toma de decisiones de la vida diaria.

Bajo este principio y las teorías que indican que a nivel neurológico la comunicación entre neuronas se realiza a través de pulsaciones eléctricas enviadas de neurona a neurona [6, 7, 10, 11], se infiere que cuando nos hacen preguntas básicas tales como: ¿Te llamas Carlos?, ¿Ya comiste?, ¿El cielo es azul?, etc. nuestra respuesta por inercia es un "SI" o "NO". Dicha respuesta no requiere de una gran cantidad de actividad cerebral, por lo cual con la tecnología de electroencefalografía con la que se cuenta hoy en día, es sumamente difícil de localizar la parte exacta donde se procesó, en un universo de mas de 100 000 000 000 de neuronas según Belmonte [11]. Por otro lado, si la pregunta fuera: ¿Cuál es la raíz cuadrada de 1536?, la respuesta requeriría de un mayor nivel de concentración y mayor activación cerebral [5], lo cual facilita localizar y/o identificar un patrón que indique a nivel cerebral qué se está pensando en algo, aunque se desconozca en que se piensa, la actividad cerebral es más fácil de localizar.

Por tal motivo y con base en este conocimiento, se infiere que si dedicamos especial atención a pensar en repetidas ocasiones ciertas respuestas, la actividad cerebral generada se captaría con mayor facilidad. Considerando lo anterior, se propone crear un sistema denominado Speechless Talk que consiste en la creación de una aplicación que permita a personas con problemas de comunicación expresar palabras dicotómicas tales como "SI" y "NO", utilizando un EEG desarrollado por la empresa NeuroSky y generando mayor actividad cerebral al pensar en repetidas ocasiones en la misma respuesta.

3. Desarrollo

Para la creación del sistema *Speechless Talk* se requirieron conocimientos básicos en electrónica, programación, redes neuronales artificiales y biológicas,

neurociencias y psicología así como de la diadema de EEG Mindwave, diseñada por NeuroSky, arduino y demás herramientas de trabajo, puesto que es requerido crear una interfaz completa, útil, económica y poco invasiva.

Para lograrlo se utilizó "Mindwave", un EEG [8] que consta de una diadema de 1 Electrodo tipo seco [16]), este se coloca en la posición Fp1 según el sistema internacional 10-20 EEG para la colocación de electrodos o lóbulo frontal; de forma más precisa, en la parte superior del ojo izquierdo, esto debido a que aquí se encuentra la corteza frontomedial (CFM) que es la que se encarga de los procesos de inhibición en la detección y solución de conflictos [14], así como de la regulación y el esfuerzo empleado en la concentración [14]; esto para recibir pulsaciones eléctricas provenientes de la actividad cerebral generada al concentrarnos y meditar; adicionalmente, se cuenta con otros dos electrodos que se colocan en los lóbulos de la oreja, con la finalidad de obtener GND y asegurar la correcta postura de la diadema [8].

Dicha diadema incluye la tecnología ThinkGear [8], dentro de la cual se procesa la señal obtenida por los sensores, además se ejecuta el algoritmo eSense (Algoritmo diseñado por NeuroSky para obtener en una escala del 1 al 100 [9], los niveles de concentración y meditación, en tiempo real así como brindar las señales de las diferentes frecuencias) de la diadema podemos extraer 11 datos, que son: Nivel de Señal, Niveles de Atención, Niveles de Meditación, Delta (1-3Hz), Theta (4-7Hz), Low Alpha (8-9Hz), High Alpha (10-12Hz), Low Beta (13-17Hz), High Beta (18-30Hz), Low Gamma (31-40Hz), High Gamma [4]; cuyo comportamiento similar al mostrado en la Fig. 1, las cuales son preprocesados en su paso por el algoritmo eSense para discriminar la mayor cantidad posible de ruidos [4].

La diadema recibe a través de los sensores conectados, los valores de las diferentes frecuencias presentadas en ese momento y posteriormente son procesadas con el algoritmo eSense incluido en el chip de la diadema, con tecnología Thinkgear [9].

Prototipo Hardware. Para cumplir el objetivo de clasificar correctamente las palabras "SI" y "NO" a través de la actividad cerebral, es necesario acceder a los datos que se procesan en el chip de la diadema, de tal manera que para desarrollar un prototipo se requiere contar con todos los materiales tales como Diadema EEG NeuroSky (podría ser MindFlex o MindWave), Arduino uno, Matlab, Software Arduino, Arduino Brain Library [15]. En la Figura 2 se muestran los componentes de la diadema, y en la Figura 3 se muestra el circuito de conexión de la diadema con el arduino, así como los puertos del chip ThinkGear para realizar la conexión entre la diadema y el módulo de arduino.

Para establecer la comunicación entre el EEG y la computadora, es requerido conectar por medio de USB el modulo arduino, seguido de una librería desarrollada por Erik Mika [4] en el software de arduino para la recepción de los datos del EEG [15]; para esto, se debe asegurar que la comunicación entre ambos dispositivos se realice con éxito.

Para identificar las señales del EEG y para clasificarlas se emplea una red neuronal. Se utiliza la función **patternnet** de Matlab; dicha función crea una

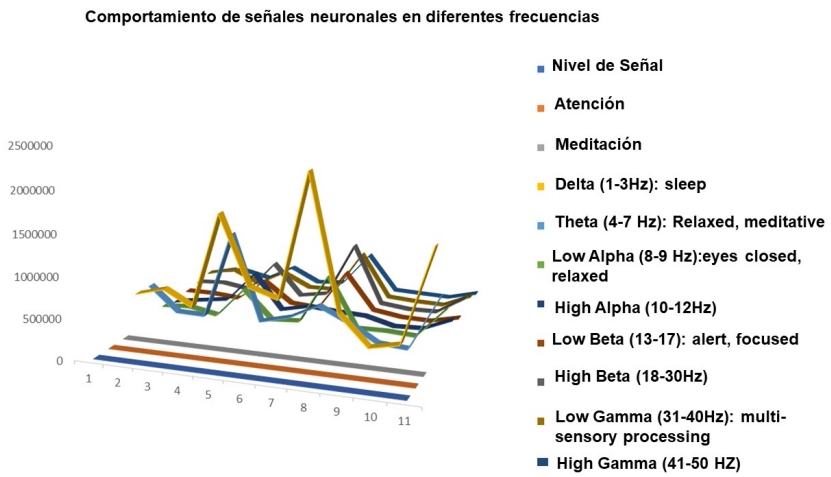


Fig. 1. Diagrama de comportamiento de la señal en las diferentes frecuencias, en 10 segundos.

red neuronal artificial de tipo feedforward cuya función objetivo consiste en diferenciar un "SI" de un "NO"; la estructura de la red la podemos observar en la Fig. 4.

La metodología para diferenciar entre "SI" y "NO", se define de la siguiente manera:
Etapa de entrenamiento:

- Seleccionar un espacio adecuado para pruebas, se recomienda que sea un lugar cerrado o con pocos distractores.
- Explicar al usuario las instrucciones y pedirle que las cumpla al pie de la letra.
- Iniciar la aplicación Speechless Talk en una computadora (desarrollada en Matlab).
- Conexión de diadema Mindwave modificada en el usuario.
- Realizar la conexión del modulo arduino de la diadema y la computadora.
- La aplicación debe comenzar a mostrar una cantidad de 60 preguntas al usuario, las cuales podrán ser escuchadas o leídas por el usuario (según su elección).
  Las preguntas deben ser divididas en bloques de 30, cuyas respuestas sean obvias para que la respuesta forzosamente sea un "SI" y otras 30 preguntas cuyas respuestas sean forzosamente un "NO".
- Una vez que el usuario lee y/o escucha la pregunta, debe comenzar a pensar en la respuesta en repetidas ocasiones, hasta que aparezca la siguiente pregunta (aproximadamente 10 segundos). Esto con la finalidad de generar a través de la actividad cerebral una respuesta "SI" o bien una respuesta "NO" y diferenciar ambas con mayor facilidad.

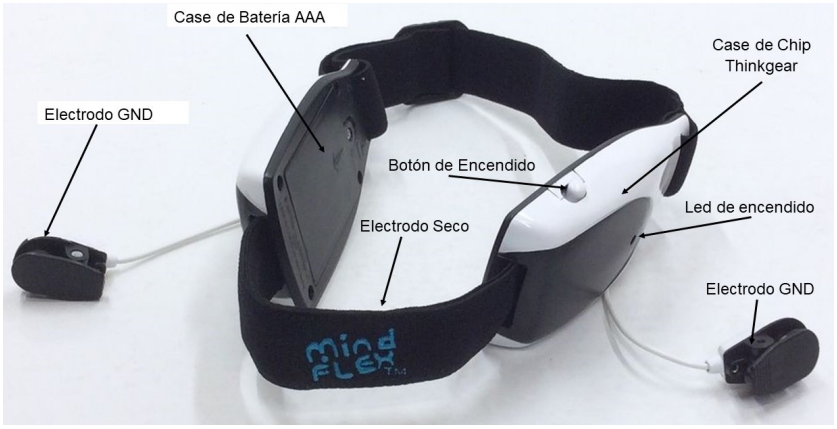


Fig. 2. Diagrama de los componentes de la Diadema Mindflex.

- A través de la diadema EEG, se obtiene la información de la actividad cerebral generada en los 10 segundos que tiene el usuario para responder cada pregunta; esta se obtiene en forma de una señal eléctrica que se introduce al arduino y posteriormente a la computadora, esto se hace continuamente durante las 60 preguntas (aproximadamente 15 minutos, por usuario). La nueva matriz de datos obtenida al finalizar las 60 preguntas será de dimensiones 600*11, haciendo referencia a las 600 lecturas obtenidas por el EEG y las 11 características que proporciona el EEG.
- Con la nueva matriz obtenida, entrenar nuestra red neuronal.

Para la etapa de clasificación:

- Realizar una pregunta adicional al usuario.
- Clasificar dentro de la red el resultado a la pregunta realizada.
- Imprimir en pantalla el resultado obtenido.
- Repetir los últimos 3 pasos, 9 ocasiones mas, haciendo un total de 10 preguntas realizadas al usuario; por prueba.
- Comparar la respuesta real esperada con la respuesta de clasificación de la red.

4. Pruebas y resultados

Las pruebas se realizan en el Laboratorio de Ambientes Inteligentes del Instituto Tecnológico de León, esto debido a que presenta el ambiente requerido, puesto que no existe demasiado ruido ni mucho trafico peatonal. No fue necesario realizar modificación alguna al laboratorio, tampoco se impidió el acceso a personas mientras se realizaban las pruebas, únicamente se les solicito hablar en voz baja para evitar distraer al usuario; el motivo por el que se realiza una

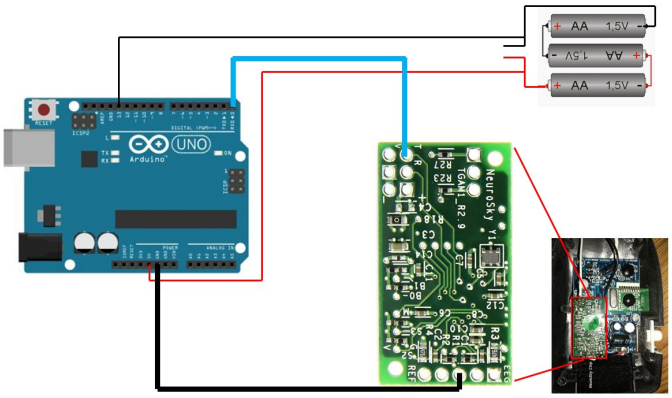


Fig. 3. Circuito Eléctrico de Conexión diadema-Arduino-Alimentación de energía.

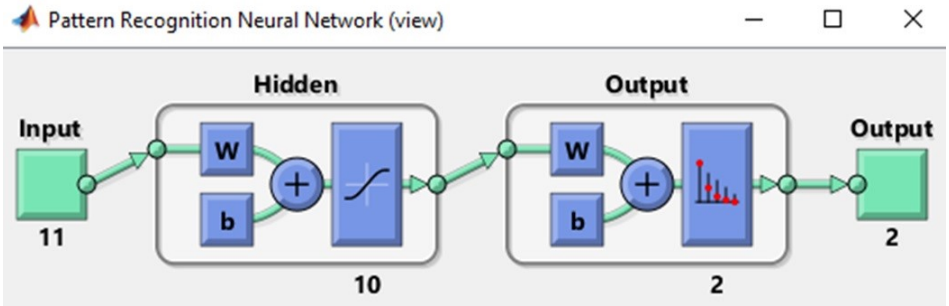


Fig. 4. Estructura de la Red Neuronal.

gran cantidad de preguntas para entrenamiento (60) es precisamente para que sí en algún momento existen factores externos que distraigan al usuario, esto no afecte en su resultado final. Se realizaron pruebas con sujetos de diferente nivel educativo, oficio, género y edad, esto para cubrir distintas características en la población, así como en diferentes horarios y bajo diferentes situaciones, tales como: después de comer, por la mañana, al anochecer, antes de comer, en situaciones de estrés.

Para realizar las pruebas, se indicaron las instrucciones a los usuarios: desde concentrarse en las respuestas, pensar su respuesta en repetidas ocasiones hasta que se le dicte la siguiente pregunta, no poner atención a lo que ocurra en su entorno, solamente pensar y no hablar las respuestas, entre otras, posteriormente, se ejecutó la aplicación y el usuario debe concentrarse solamente en responder con el pensamiento.

Los conjuntos de datos obtenidos por cada sujeto prueba, dentro de la etapa de entrenamiento se utilizaron para entregar y calibrar la red neuronal, la cual

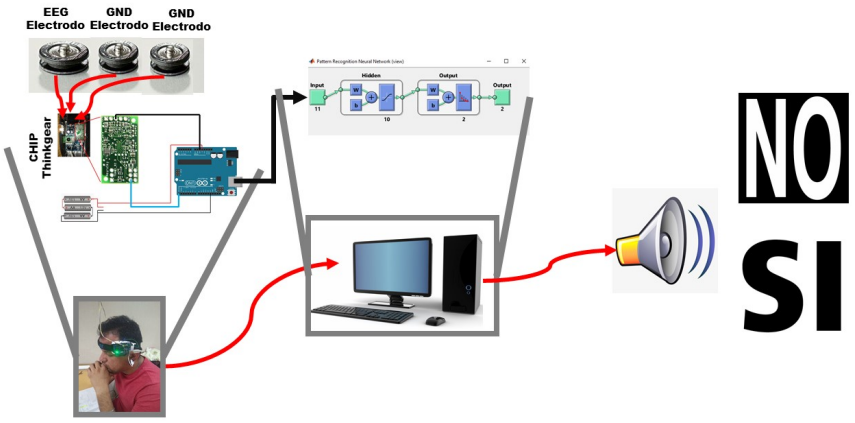


Fig. 5. Diagrama de Speechless Talk.

demostró comportarse como se presenta en la Fig. 6.

De dicha base de datos de señales se hizo una partición de 70 % para entrenamiento de la red, 15 % para validación y 15 % restante se utiliza para pruebas, obteniendo un porcentaje de clasificación del 94.3 % lo cual puede ser corroborado en la Fig. 6. A continuación, en la Fig. 7, vemos el comportamiento de las curvas ROC (receiver operating characteristic curve) utilizadas para la representación de falsos positivos frente a la razón de verdaderos positivos.

Como resultado de la experimentación, se obtiene una aplicación de escritorio que se conecta por un puerto serial a un EEG colocado en el cráneo del usuario, dicha aplicación incluye una red neuronal de tipo feedforward que se encarga de clasificar los pensamientos del usuario y obtener una respuesta a las preguntas en cuestión; la primer versión solamente clasifica las palabras “SI” y “NO”; dicha red, así como toda la programación de la aplicación, se encuentra desarrollada en Matlab.

El sistema se encarga de leer y mostrar a los usuarios una serie de preguntas a las cuales se debe responder solamente pensando, después de una serie de 60 preguntas, viene una serie adicional de preguntas con base a lo que se desea saber, cuyas únicas respuestas deben ser “SI” o “NO” (en esta parte es donde se predice o se realiza la comunicación en tiempo real con el usuario). Para la parte de prueba, se sometieron al experimento 15 personas y se observó el siguiente comportamiento en los resultados de clasificación entre las respuestas esperadas y las respuestas que clasificó la el sistema Speechless Talk.

Cabe mencionar que se tienen 106 preguntas correctamente clasificadas de un total 150, lo que hace un 71.33 % de clasificación correcta.

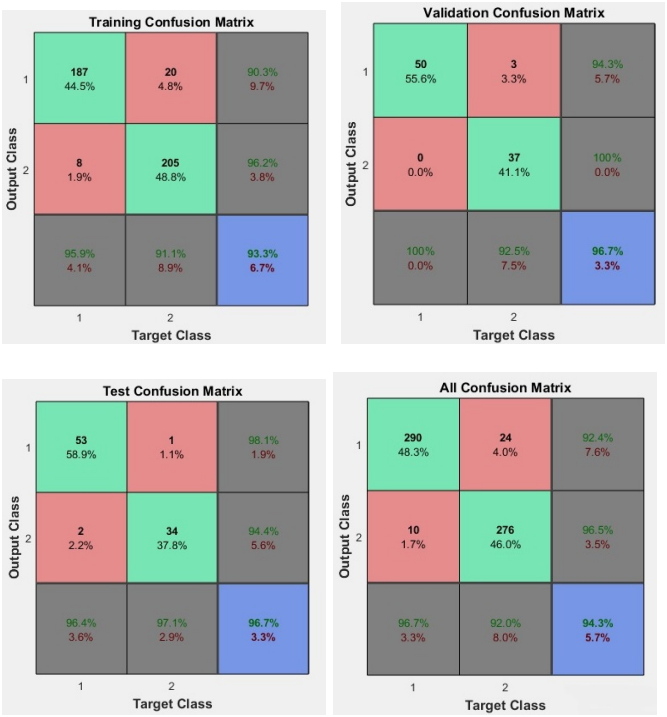


Fig. 6. Matriz de confusión.

5. Conclusiones

El trabajar con señales que provienen de EEG, es un tema complejo, además, existe una serie de factores que afectan los resultados esperados y se deben de tomar en cuenta, como la alimentación, el estado anímico, la hora del día, estado de salud, así como el entorno que rodea a los usuarios, ya que si bien, por la mañana, probablemente el usuario se encuentre en un estado más activo, presente diferencias a la hora de concentrarse, así mismo, dependiendo lo que el usuario haya comido puede llegar a acelerar la actividad neuronal, lo cual generaría interferencia para pruebas futuras, en un conjunto de datos ya entrenado.

Al realizar las 60 preguntas de entrenamiento al usuario e inmediatamente presentar las 10 preguntas de prueba, los resultados presentaban cierto sesgo hacia las últimas preguntas realizadas, esto debido a que los niveles de somnolencia o cansancio juegan un papel muy importante al momento de clasificar y es considerado como factor decisivo en la prueba en tiempo real, pues el paciente se encuentra en el mismo estado que cuando presentó las ultimas 30 preguntas; por consiguiente las clasificación de la red neuronal artificial se inclina por la respuesta pensaba en las ultimas 30 preguntas de entrenamiento.

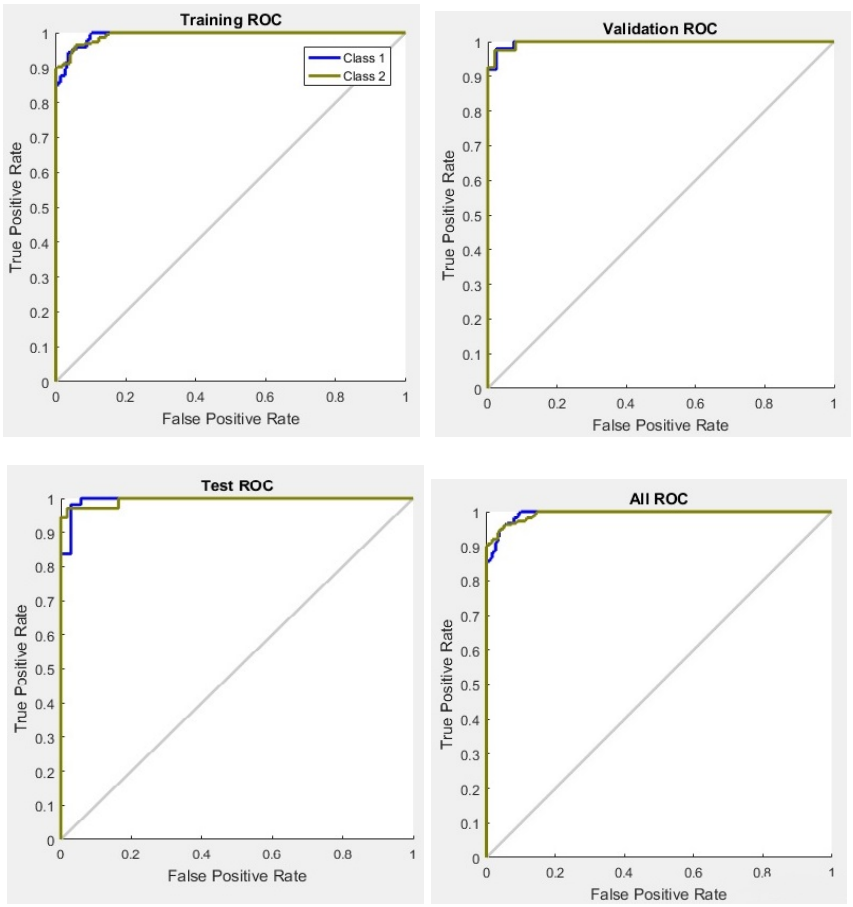


Fig. 7. Análisis de curvas ROC (receiver operating characteristic curve).

Otro factor a considerar para la mejora de resultados es la disponibilidad del usuario, debido a que después de algunos minutos los usuarios para prueba llegan a sentir aburrimiento o cansancio o no prestan la atención necesaria a la prueba, generando con esto, resultados erróneos. Para el correcto funcionamiento y mejorar la cantidad de respuestas correctas, se solicita a los usuarios pongan todo el empeño necesario en concentrase.

Se analizaron diferentes técnicas de inteligencia computacional para la correcta clasificación de señales neuronales biológicas, así como métodos de tratamiento de señal, optando por emplear una red neuronal artificial de tipo feedforward; se diseñó una BCI amigable con el usuario para obtener las pulsaciones eléctricas de los usuarios y enviarlas a una computadora para ser analizadas y clasificadas en la red neuronal programada en Matlab, dando como resultado un 71.33 % palabras dicotómicas “SI” y “NO” obtenidas al pensar, las cuáles son respuestas

Tabla 1: Comparativa de respuesta paciente vs respuesta Speechless Talk. PD: 0 indica una respuesta NO y 1 Indica una respuesta SI.

	Speechless Talk		
	Respuestas Paciente	Respuesta Speechless Talk	Porcentaje de clasificación
Sujeto 1	1 0 1 1 0 0 1 0 1 1	1 0 1 1 0 1 1 0 1 1	90 %
Sujeto 2	1 0 0 1 1 0 1 0 1 1	1 1 0 1 0 0 1 0 1 1	80 %
Sujeto 3	0 1 0 1 0 1 0 0 0 0	0 1 0 0 0 1 1 0 0 0	80 %
Sujeto 4	0 0 0 1 0 1 0 0 0 0	0 0 0 1 0 0 0 1 0 0	80 %
Sujeto 5	0 0 0 0 1 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0	80 %
Sujeto 6	00 0 0 0 1 0 0 1 0 0	0 0 0 0 0 0 0 1 1 0	80 %
Sujeto 7	0 0 0 1 0 0 0 0 0 1	1 0 1 1 0 1 0 0 0 1	70 %
Sujeto 8	1 1 0 1 1 1 0 0 1 1	0 0 1 1 1 1 0 0 1 1	70 %
Sujeto 9	0 1 0 0 1 0 0 0 0 1	0 1 0 0 1 1 0 0 1 0	70 %
Sujeto 10	1 1 1 1 0 0 1 0 0 0	1 1 0 1 0 0 1 0 1 1	70 %
Sujeto 11	0 1 0 1 0 1 1 0 0 0	0 0 1 1 1 1 1 0 0 0	70 %
Sujeto 12	0 0 0 0 1 1 1 1 1 1	0 0 1 0 1 1 0 1 0 1	70 %
Sujeto 13	1 0 1 1 1 0 1 1 0 1	0 0 0 1 1 0 1 1 0 0	70 %
Sujeto 14	1 0 0 1 0 1 1 1 0 0	1 1 1 1 0 0 1 0 1 1	60 %
Sujeto 15	0 1 1 1 1 1 1 0 0 1	0 0 0 0 0 0 0 0 0 0	30 %

a una serie de preguntas realizadas a 15 usuarios diferentes, en tiempo real.

Speechless Talk es una aplicación pensada en apoyar a todas aquellas personas que cuentan con algún problema de comunicación verbal, la cual a través de una interfaz BCI obtiene señales obtenidas al pensar y estimular los niveles de concentración para posteriormente a través de técnicas computacionales inteligentes procesar las señales obtenidas y clasificar lo mejor posible las señales que conforman el pensamiento dicotómico para las palabras “SI” y “NO”.

6. Trabajo futuro

Después del trabajo realizado, se sugieren las siguiente mejoras para las posteriores versiones:

- Implementar filtros de tratamiento de señal.
- Experimentar con diferentes técnicas de clasificación para mejorar resultados.
- Incrementar la cantidad de palabras a clasificar.
- Realizar un prototipo hardware portátil para brindarle mayor movilidad e independencia al paciente.
- Implementar Speechless Talk en aplicaciones médicas.

Referencias

1. Zapata-Zapata, C.H., Franco-Dager, E., Solano-Atehortua, J.M., Ahunca-Velasquez, L.F.: Amyotrophic lateral sclerosis: update/Esclerosis lateral amio-

trofica: actualizacion/La esclerose lateral amiotrofica, Universidad de Antioquia, Facultad de Medicina (2016)
2. Herff, C., Heger, D., Pesters, A., Telaar, D.: Brain-to-text : decoding spoken phrases from phone representations in the brain. Frontier in Neuroscience, 2(5), pp. 1–11 (2015)
3. Torres-García, A.A., Reyes-García, C.A., Villaseñor-Pineda, L., García-Aguilar, G.: Implementing a fuzzy inference system in a multi-objective EEG channel selection model for imagined speech classification. El Sevier, pp. 1–12 (2016)
4. Brain-Hack, M.E.: `http://www.frontiernerds.com/brain-hack` (2018)
5. Kohan, N.C.: `http://www.redalyc.org/articulo.oa?id=299023503010` (2011)
6. Gupta, K.C.: Neural Network Structures. Neural Networks for RF and Microwave Design. pp. 61–103 (2000)
7. Bear M., Connors, B., Paradiso, M.: Neurociencia La Exploración del Cerebro. The Plant Journal, 41 (2005)
8. NeuroSky: `http://developer.neurosky.com/docs/doku.php?id=what_is_thinkgear` (2018)
9. NeuroSky: `http://developer.neurosky.com/docs/doku.php?id=esenses_tm&s[]=esense` (2018)
10. Purves, D., Augustine, G., Fitzpatrick, D., Hall, W., Lamatia, A., Mcnamara, J.W.M.: Neurociencia. 918 (2008)
11. Belmonte, C.: La exploración del cerebro : Una aventura de futuro (2012)
12. INEGI: `http://www.beta.inegi.org.mx/temas/discapacidad/` (2017)
13. INTEL: `https://01.org/acat/documentation-list` (2017)
14. Flores-Lázaro, J. C.: Neuropsicología de Lóbulos Frontales, Funciones Ejecutivas y Conducta Humana. Revista Neuropsicología, Neuropsiquiatría Y Neurociencias, 8(1), pp. 47–58 (2008)
15. NeuroSky: `http://developer.neurosky.com/docs/doku.php?id=projects` (2018)
16. Vea-Murgía, V.P.: Investigación del Funcionamiento de Electrodos Secos y Gorro de Diseño Propio contra Gorro Comercial con Electrodos Húmedos Aplicando Filtros CSP a Tareas de Movimiento. 34 (2011)

ISSN 1870-4069

Algoritmos de aprendizaje supervisado para la clasificación de géneros musicales caracterizados mediante modelos estadísticos

Arturo Tepepa Cantero, Héctor Manuel Pérez Meana, Mariko Nakano Miyatake

Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán,
Sección de Estudios de Posgrado e Investigación,
México

sas_19_93@hotmail.com, {mnakano, hmperezm}@ipn.mx

Resumen. En nuestros días es común tener música almacenada en formato digital. Sin embargo debido a la gran cantidad de información que se tiene, es imposible realizar una clasificación adecuada de toda la música existente sin algún tipo software. Tomando en cuenta esta problemática, en el presente trabajo se desarrolló un algoritmo para realizar la clasificación automática de géneros musicales usando un proceso de segmentación empleando características espectrales tales como *centroide* (SC), *flatness* (SF) y *spread* (SS) así como una temporal, tal como la tasa de cruces por cero (ZCR); obteniendo vectores característicos de las pistas de audio. En la etapa de clasificación se utilizaron 4 clasificadores KNN, SVM, LDA y árboles de decisión, observando que la mejor clasificación para nuestros vectores se obtuvo usando la máquina de soporte vectorial (SVM). Finalmente se utilizó el proceso de *voting* para mejorar la clasificación obtenida usando segmentos individuales; dando como resultado un grado de clasificación mayor al 90%. Finalmente se clasificaron canciones en las cuales se utilizaba un solo instrumento musical por lo cual se obtuvieron mejores resultados muy próximos al 100% de clasificación.

Palabras clave: Clasificación géneros musicales, segmentación, clasificador, voting.

Supervised Learning Algorithms for the Classification of Musical Genres Characterized by Statistical Models

Abstract. Nowadays is common to have the music stored using some digital format. However, because the large amount of data, it is impossible to make an accurate classification of all existing music without some kind of software. Attending to this requirement, in this work we developed a musical genres using a segmentation process together with some spectral characteristics such as centroid (SC), flatness (SF) and spread (SS) as well as temporary characteristic

such as zero crossing rate (ZCR) getting the characteristic vectors of the music. In the step of classification 4 classifiers were used KNN, SVM, LDA and decision trees. From these results we noticed that the best classification using the estimated characteristics vectors was using support vector machine (SVM). To further improve the classification obtained using each individual segment, we use a voting method which provides a classification performance higher than 90%. Finally we classify several songs played with only one kind of musical instrument, obtaining classification results closely to 100%.

Keywords: Music genre classification, segmentation, classifier, voting.

1. Introducción

Durante los últimos años se ha tendido a almacenar las pistas de audio para su posterior uso ya sea en discos CD-DVD, Discos Duros HDD-SSD así como en internet, lo cual implica un desafío poder clasificar la información ya sea en línea o fuera de línea. Para realizar lo anterior se debe hacer un etiquetado de las pistas de audio se dice que las etiquetas son textos basados en la información semántica del sonido [1]. Así el análisis de la música se puede hacer de varias formas [2] donde se identifica la música por su género, artista, instrumentos y estructura, mediante el etiquetado, el cual puede ser manual o automático. El etiquetado manual permite una visualización del comportamiento de una pista de audio ya sea en dominio del tiempo o en dominio de la frecuencia tal como en el espectrograma, haciendo posible clasificar las canciones sin necesidad de escucharlas. Sin embargo la realización de este proceso conlleva mucho tiempo y esfuerzo, incluso problemas en la salud [3] en donde se muestra que "el volumen, la sensibilidad acústica, el tiempo y costo requeridos para un proceso manual de etiquetado es en general prohibitivo. Por su parte, para la realización de un etiquetado automático se necesitan 3 pasos fundamentales: el pre-procesamiento, la extracción de características y la clasificación [4]. En el presente trabajo se desarrolló un esquema de clasificación en el cual, inicialmente se procesa la señal de entrada para reducir el ruido, seguidamente se segmenta la señal, cuyos segmentos se procesan mediante dos esquemas de caracterización, una en el dominio de frecuencia y otra en el dominio del tiempo [5].

Se han hecho trabajos en la segmentación de audio [6] utilizando características básicas como la tasa de cruces por cero por siglas en inglés "ZCR" además del cálculo de energía en un periodo de tiempo muy corto "centroide", utilizando ventanas de 2.4s, donde se reportó una precisión de 98% en la clasificación. Existen además desarrollos en el procesamiento digital de imágenes [7] enfocándose en el espectrograma cuyo objetivo es la clasificación multiclase, donde el clasificador empleado fue la máquina de soporte vectorial (SVM) obteniendo resultados de 85% de clasificación multi-clase, en donde el sistema determina a cual clase pertenece la señal de audio bajo análisis. Aquí el clasificador SVM a pesar de ser un clasificador creado para clasificación binaria obtiene muy buenos resultados debido a la previa extracción de características en la pista de audio.

En este trabajo se utilizaron algunas características propuestas por Tzanetakis y Cook [8] cómo lo es el centroide espectral (punto donde el espectro se encuentra en equilibrio) y *ZCR* (valor promedio de las veces que la señal cruza cero en el eje x estando dominio temporal). Además se usaron otras características como *Spectral Flatness* (Valor de la cantidad de cambios en la frecuencia por trama) y *Spectral Spread* (Potencia alrededor de cada *Spectral Centroid* y la relación de un centroide con los demás).

2. Método propuesto

El sistema propuesto se muestra en la Fig. 1, el cual clasifica un conjunto de pistas de audio divididas entre 5 géneros. En particular durante el entrenamiento se emplearon 10 pistas de audio de cada género musical, las cuales fueron Cumbia, Pop, Rap, Rock y Salsa con frecuencia de muestro fs = 44100Hz, 16 bits de profundidad, con una duración de 5 – 10 minutos y formato wav. Se extrajeron 3 características espectrales *Spectral Centroid, Spectral Spread y Spectral Flatness*, así como una característica temporal *Zero Crossing Rate*. Finalmente se utilizaron 4 clasificadores: Decision Trees, Discriminant Analysis, Support Vector Machine (Gaussian), Nearest Neighbor Classifier.

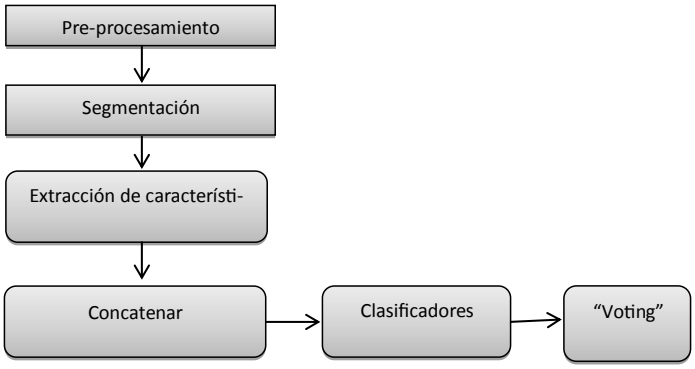


Fig. 1. Método de clasificación señales de audio.

2.1. Pre-procesamiento

En la etapa del análisis de datos, se observa que una parte importante de la música conserva información relevante a partir de una tercera parte de la canción hasta un poco más de las dos terceras partes, debido al silencio que usualmente se tiene en las pistas de audio al inicio y al final así que se toma como un factor importante la cantidad de muestras para enfocarse en esa región y observando que las canciones de mayor duración tienen el doble de duración entonces el número de muestras mínimas es la mitad de la duración máxima de las canciones que se tienen en la base de datos. Así la forma el número de muestras se calculad como se muestra en la ecuación (1):

$$\text{Numero de Muestras} = fs * \text{Tiempo}. \quad (1)$$

Las muestras para la canción más corta fueron alrededor de $13x10^6$, por lo que el máximo de muestras para la canción de mayor duración serán $26x10^6$. Estos valores son muy importantes para la segmentación que se hará posteriormente debido a que se tratará de eliminar partes de las canciones que no aportan información importante para su fututa clasificación.

2.2. Segmentación

En este proceso se toma la parte deseable de la canción "D" y se descartan las partes

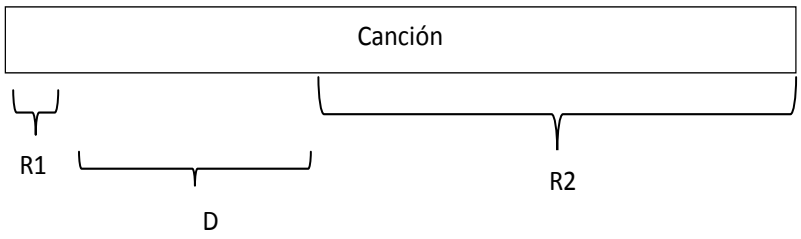


Fig. 2. Obtención de sección de análisis.

"R1y R2" correspondiente a posibles silencios como se muestra en la Fig. 2.

El valor R1 contiene de $.5x10^6$ muestras, para calcular el valor en segundo se usa la ecuación (2):

$$\text{Calculo de muestras en tiempo} = \text{número de muestras} / fs. \quad (2)$$

Son 11.2 segundos descartados (R1). La parte deseada "D" consistirá de 15 "*segmentos"* que sumados no deberán superar $13x10^6$ muestras, valor de muestras máximo de la canción con menos duración, así que se escoge el valor de $.332800x10^6$ el cual simboliza el número de muestras para cada "*segmento*". El tamaño de "D" esta expresado por la ecuación (3). Es importante señalar que los 15 "*segmentos"* deben ser etiquetas con el mismo género de la canción:

$$D = .3328x10^6 * 15. \quad (3)$$

D resulta de un tamaño de $4.992x10^6$ muestras, aplicando la ecuación (2) nos da un valor de 113 segundos de música para analizar sin el problema de silencio, aumentando la base de datos 15 veces y obteniendo segmentos del mismo tamaño sin importar la duración de la canción.

Finalmente cada uno de los 15 segmentos de "D" se divide en 650 "*sub-segmentos*" obteniendo 512 muestras por "*sub-segmento*", a las cuales se les aplicará la extracción de características. El número de vectores que serán utilizados para el entrenamiento está dado por la ecuación (4):

#_Vectores = #_Géneros * #_pistas_género*#_segmentos (4)
#_ Vectores = 5 * 10* 15 = 750.

2.3. Extracción de características

Como se mencionó en el inicio de este capítulo se utilizaran 4 modelos matemáticos que obtienen diversas características de la pista de audio, 1 de ellos se encuentra en un dominio temporal y 3 son espectrales así que se tendrá que calcular la FFT de las ultimas 3.

Zero Crossing Rate Este caracterizador es del tipo temporal, a cada instante de tiempo se le asigna un valor obtenido por un micrófono llamándose "muestra" que tiene valores positivos y negativos, que serán utilizados para calcular el número de cruces por cero con la ecuación (5):

$$\text{ZCR} = \frac{1}{2(N-1)} \sum_{m=1}^{N-1} |sgn[x(m+1)] - sgn[x(m)]|, \quad (5)$$

donde x es el conjunto de muestras, m es la posición de la muestra y N el total de muestras. El objetivo del algoritmo es sumar las veces que cambia de signo una muestra con respecto a la anterior, significando que la señal de audio atravesó de valores positivos a negativos o viceversa en el eje x, se suman los valores obtenidos y se normaliza dividendo entre 2(N-1).

Como se observó en la sección de segmentación se obtuvieron 650 sub-segmentos con una longitud de 512 muestras a las cuales aplicando el algoritmo ZCR se obtienen 650 componentes normalizadas con 2 veces el total de muestras es decir 1024 formando un vector característico con su respectiva etiqueta.

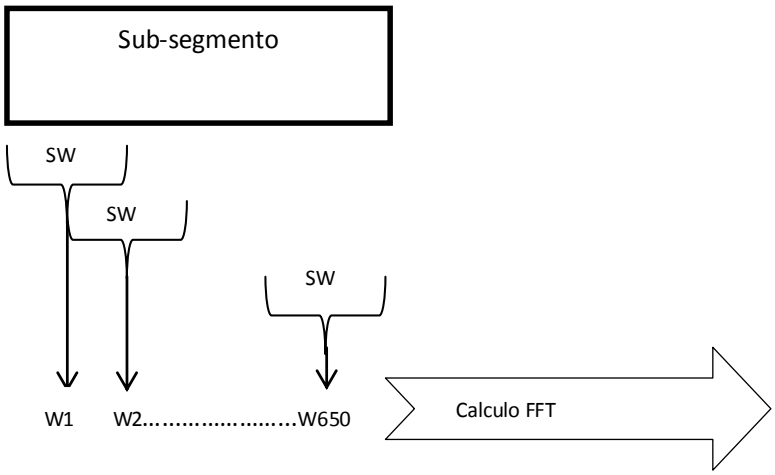


Fig. 3. Conversión dominio temporal a frecuencia.

Spectral Centroid Antes de calcular la Transformada Rápida de Fourier (FFT) y transformar los valores del dominio temporal al espectral es necesario un "ventaneo" de la

señal con un traslape, esto para disminuir el denominado efecto de "Gibbs" producido al cortar de forma abrupta la señal. Con esta finalidad se ocupó una venta Hamming de tamaño de 1024 (SW) que es el doble de tamaño de la cantidad de muestras para así traslapar al 50%, el esquema se muestra en la Fig. 3. Así después de "ventanear" la señal y obtener valores con un tamaño de 1024 se aplica FFT para posteriormente calcular *Spectral centroid* con la ecuación (6):

$$\text{SC} = \frac{\sum_{m=1}^{N-1} X(m) f(m)}{\sum_{m=1}^{N-1} X(m)}, \tag{6}$$

donde X representa los valores obtenidos de la FFT y f resulta de crear un vector de 1-1024 valores y dividir cada valor entre 1024, se crea una nueva escala que representa f.

Calculando los centroides en el "sub-segmento" se obtiene otro vector característico con un tamaño de 650 valores que se asemeja con el vector obtenido de *ZCR*, sin embargo esta característica representa la energía que se tiene en cada una de las ventanas.

Spectral Spread *Spectral Spread* representa la concentración de energía alrededor de cada *Spectral centroid*, Una característica importante es que entre mayor sea SS, significa que habrá una gran cambio en las frecuencias, se calcula enseguida de tener SC sobre la ventana de 1024 usando la ecuación (7):

$$\text{SS} = \sqrt{\frac{\sum_{m=1}^{N-1} \left(f(m) - SC(m)\right) * |X(m)|}{\sum_{m=1}^{N-1} |X(m)|}}. \tag{7}$$

La única diferencia en el cálculo de SS es que se realiza una sustracción a f con el SC que se obtuvo, además de calcular su raíz cuadrada en esa ventana.

Spectral Flatness Este rasgo pertenece al conjunto de características básicas [4] la cual indica que tan "plano" es el espectro con una serie de valores que expresan la energía del espectro dentro de una banda de frecuencia pre-definida, se ocupa la ecuación (8):

$$\text{SF} = \frac{\sqrt{\prod_{m=1}^{N-1} |X(m)|}}{\frac{1}{N} \sum_{m=1}^{N-1} |X(m)|}. \tag{8}$$

2.4. Concatenar vectores

Se tienen 4 vectores característicos cuyo tamaño es de 650, estos se concatenan y se obtiene 1 vector de tamaño de 2600, el orden será ZCR-SC-SS-SF-Etiqueta, este proceso se hará con los 750 vectores, obteniendo el descriptor final que se clasificará por 4 métodos.

2.5. Clasificadores

La clasificación se hace en cada sub-segmento cuya duración será de 11.2 ms, es decir habrá 750 clasificaciones y como se ha mencionado en el artículo se utilizaron 4 clasificadores; obteniéndose los valores de clasificación mostrados en la Figura 1. De estos resultados se observa que, como podría esperarse, empleando un solo sub-

Clasificadores	
Tipo	Precisión
Decision Tree	33.9 %
Discriminant Analysis	49.7 %
Support Vector Machine	58 %
K-Nearest Neighbor	49.3%

Fig. 1. Porcentaje de clasificación géneros musicales.

Gaussian SVM					
	Cumbia	Pop	Rap	Rock	Salsa
Cumbia	71	43	5	8	23
Pop	15	80	24	3	28
Rap	12	18	102	3	15
Rock	12	5	8	92	33
Salsa	20	8	17	15	90

Fig. 2. Matriz de confusión de géneros musicales.

segmento de 11.2 ms el porcentaje de acierto no es satisfactorio. Por otro lado se observa que el valor máximo de clasificación es obtenido usando SVM en una clasificación multiclase, donde el clasificador es requerido a determinar a cuál de las diferentes clases pertenece la señal de entrada. Así observando la matriz de confusión mostrada en la Figura 2. se tendrá una tendencia que aprovecharemos para utilizar el método denominado "Voting".

En la mayoría de los casos el valor obtenido en la diagonal principal de la matriz de confusión supera por más del doble y en el mejor de los casos (rap) la tendencia es 6 veces mayor que la mayor de las otras 4 opciones. Además observando los valores de la curva ROC del género cumbia (Representación gráfica de la comparación entre sensibilidad eje y contra la especificidad eje x y donde el valor máximo es 1) se aprecia una excelente clasificación Fig. 3. El valor de la curva ROC para todos los géneros se muestra en la Figura 4.

2.6. Voting

Una vez obtenida la clasificación de cada sub-segmento es decir 15 vectores clasificados de la misma canción se procede a escoger el valor que más se repite dentro de los 15 sub-segmentos como se observa en la figura 5.

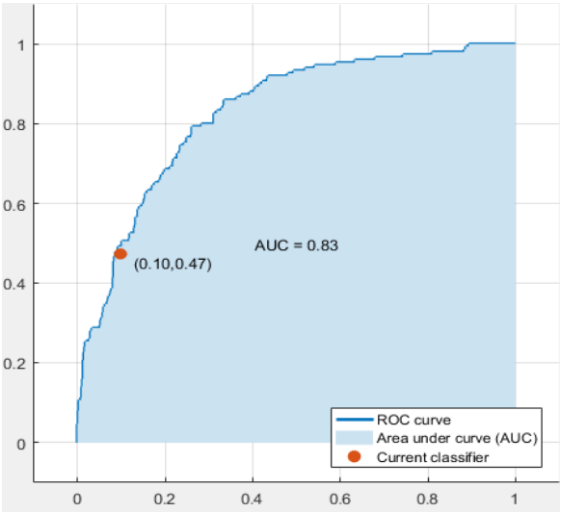


Fig. 3. Curva ROC clasificación géneros musicales.

Género	Valor curva ROC
Violín	0.99
Piano	0.96
Guitarra	0.97
Flauta T.	0.99

Fig. 4. Curva ROC clasificación géneros musicales.

C1	C1	C2	C1	C3	C1	C1	C1	C4	C5	C2	C1	C1	C1	C1

Fig. 5. Sub-segmentos clasificados.

En el caso de este ejemplo se tiene que hubo 10 valores para C1, 2 para C2, 1 para C3, 1 para C4 y 1 para C5, se escoge C1 debido a que es el que más se repite. Se aplicó esta clasificación para las 50 canciones de nuestra base de datos para obtener de resultado un 96% de clasificación, equivocándose en 2 canciones de cumbia, confundidas con pop.

3. Pruebas y resultados

Durante el proyecto se obtuvo una clasificación multiclase de 96% utilizando el Gaussian SVM y el método de voting con señales de audio que tienen componentes espectrales muy parecidas, se hizo el mismo procedimiento para clasificación de 10 canciones tocadas con 4 instrumentos musicales diferentes obteniendo excelentes

Clasificadores	
Tipo	Precisión
Decision Tree	71.2 %
Discriminant Analysis	80.8 %
Support Vector Machine	93 %
K-Nearest Neighbor	76%

Fig. 5. Porcentaje de clasificación instrumentos musicales.

Gaussian SVM				
	Violín	Piano	Guitarra	Flauta T.
Violín	146	1	2	1
Piano	0	137	10	3
Guitarra	12	0	135	3
Flauta T.	6	2	3	139

Fig. 6. Matriz de confusión instrumentos musicales.

Género	**Valor curva ROC**
Violín	0.99
Piano	0.96
Guitarra	0.97
Flauta T.	0.99

Fig. 7. Curva ROC clasificación instrumentos musicales.

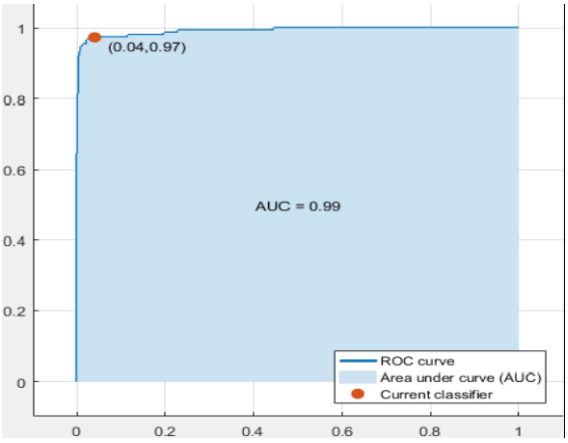


Fig. 8. Curva ROC clasificación instrumentos musicales.

resultados con los 5 clasificadores donde el SVM fue también el clasificador que proporción los mejores resultados, como se muestra en la Figura 5.

La matriz de confusión obtenida es bastante buena debido a la diagonal principal que contiene un porcentaje muy alto, como se muestra en la Tabla 5. Esto se observa también en la Tabla 6 y en la curva ROC Figura 6.

Aplicando voting la clasificación fue de 99% para instrumentos musicales obteniendo resultados satisfactorios y observando que dependiendo la cantidad de frecuencias involucradas (instrumentos y voz) será menor el índice de clasificación, aun así con el método de voting mejora notablemente.

4. Conclusiones

Para realizar una correcta clasificación es necesario conocer los vectores de entrada y discriminar defectos que pueden tener, además se observa que la clasificación depende en gran medida de que tan parecidos sean los vectores (géneros o instrumentos musicales), en estos casos las dos pruebas comprobaron esta idea aunque fuera el mismo tamaño, segmentación y extracción de características, los valores cambiaron ampliamente. El proceso de voting es muy eficiente si se tiene una clasificación de regular a buena en varios segmentos a pesar del nivel bajo obtenido por SVM.

Finalmente se puede asegurar que la utilización de los modelos estadísticos como lo es el SC, SF, SC y ZCR a pesar de tener un coste computacional bajo sirvieron bastante bien, esto nos da la pauta para aplicar posteriormente sistemas más robustos que obtengan características más eficientes que provoquen una mayor clasificación, inclusive no tan general como géneros sino como la pista en cuestión.

Referencias

1. Panagakis, Y., Kotropoulos, C.: Automatic music tagging via PARAFAC2. (ICASSP), IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings (2011)
2. Mitrovic, D., Zeppelzauer, M., Eidenberger, H.: Analysis of the Data Quality of Audio Features of Environmental Sounds. Knowledge Creation Diffusion Utilization, pp. 4–17 (2006)
3. Lau, A., Mason, R., Pham, B., Richards, M., Roe, P., Zhang, J.: Monitoring the environment through acoustics using smartphone-based sensors and 3G networking. IEEE international conference on distributed computing in sensor systems (2008)
4. Greece-Duan, S., Zhang, J., Roe, P.: A survey of tagging techniques for music, speech and environmental sound, pp. 637–661 (2014)
5. Stowell, D., Plumbley, M.: A survey of UK birdsong and machine recognition for music researchers. Tech. Rep., pp. 09–12 (2011)
6. Lu, L., Zhang, H.J., Li, S.: Digital Object Identifier Multimedia Systems Content-based audio classification and segmentation by using support vector machines. Multimedia Systems, pp. 482–492 (2003)
7. Faisal-Ahmed, P.P., Paul, M.G.: Music Genre Classification Using a Gradiente-Based Local Texture descriptor. Springer International Publishing Switzerland, pp. 99–110 (2016)
8. Tzanetakis, G.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, pp. 293–302 (2002)

ISSN 1870-4069

Comparación de dos métodos para reconocimiento de dígitos manuscritos fuera de línea

María Cristina Guevara Neri[1], Osslan Osiris Vergara Villegas[1], Vianey Guadalupe Cruz Sánchez[1], Juan Humberto Sossa Azuela[2, 3]

[1] Universidad Autónoma de Ciudad Juárez, Chihuahua, México

[2] Instituto Politécnico Nacional (CIC), México

[3] Instituto Tecnológico de Estudios Superiores Monterrey (Campus Guadalajara), México

mc_guevara_neri@hotmail.com, {overgara, vianey.cruz}@uacj.mx, hsossa@cic.ipn.mx

Resumen. En el presente artículo, se muestra el resultado de la comparación del desempeño de dos métodos para el reconocimiento de dígitos manuscritos fuera de línea. El primer método, es una red de perceptrones con la cual se clasificaron las imágenes tras realizar una comparación por pares de clases; el segundo, es un método novedoso que realiza una comparación pixel por pixel entre la imagen por clasificar, y las imágenes de referencia. Para las pruebas, se utilizó un subconjunto de 450 imágenes de la base de datos MNIST. Cada método fue evaluado en dos partes: primero, con un conjunto de 100 imágenes de entrenamiento, y segundo, con un conjunto de 350 imágenes de prueba. Con el primer clasificador se obtuvo una exactitud del 93.86%, y con el segundo se consiguió una del 95.14%. Después del análisis de los resultados obtenidos se demuestra que el segundo método se desempeñó mejor que el primero. La fortaleza del método novedoso radica principalmente en su robustez y tiempo de ejecución.

Palabras clave: Reconocimiento de dígitos manuscritos, red de perceptrones, comparación vector con vector, MNIST.

Comparison of Two Off-Line Handwritten Digits Recognition Methods

Abstract. In this paper, the results of the comparison between two off-line handwritten digits recognition methods are presented. The first method is a network of perceptrons with which the images were classified after making a comparison by pairs of classes; the second, is a new method that performs a pixel by pixel comparison between the image to be classified, and the reference images. For the tests, a subset of 450 images from the MNIST database was used.

Each method was evaluated in two parts: first, with a set of 100 training images, and second, with a set of 350 test images. With the first classifier, an accuracy of 93.86% was obtained, and with the second, an accuracy of 95.14%. After the analysis of the results, it is shown that the second method outperformed the first. The strength of the new method lies mainly in its robustness and execution time.

Keywords: Handwritten digits recognition, perceptron network, comparison between vectors, MNIST.

1. Introducción

En la última década (2008-2018), se ha observado un incremento en la digitalización de documentos, por lo que, el reconocimiento de dígitos manuscritos, y caracteres en general, es un área de interés para científicos, académicos e industriales [1]. El reconocimiento de caracteres manuscritos ha ganado popularidad en el área de investigación, y ha generado el reto entre los investigadores de lograr emular el sistema de procesamiento visual humano, mediante el desarrollo de algoritmos computacionales.

El reconocimiento de dígitos manuscritos es una rama correspondiente a la tecnología de reconocimiento óptico de caracteres, en la que el reto consiste en cómo utilizar un dispositivo procesador para reconocer automáticamente los dígitos [2]. La lectura de caracteres manuscritos representa una tarea complicada para una máquina, ya que no cuenta, como lo hace un ser humano, con la posibilidad de tomar decisiones fuera de un esquema establecido.

Las técnicas de reconocimiento de dígitos manuscritos se clasifican en dos grandes áreas que son en línea y fuera de línea. En la primera técnica, el flujo de dígitos es reconocido en el momento en el que es escrito, mientras que, en la segunda, que suele ser más compleja, los dígitos son reconocidos mediante la captura o escaneo de imágenes. Usualmente, el reconocimiento de dígitos manuscritos consiste en las siguientes etapas: 1) adquisición, 2) pre-procesamiento, 3) segmentación, 4) extracción de características, y 5) clasificación [3].

La etapa de adquisición o digitalización consiste en adquirir la imagen de los dígitos manuscritos a través de una fotografía o escaneo. En el pre-procesamiento se aplican diversos algoritmos de mejoramiento con la finalidad de mejorar la calidad de la información. Por ejemplo, en las imágenes de dígitos manuscritos se puede presentar ruido debido al color del papel, la textura, imágenes de fondo, emborronamiento, distorsiones por la perspectiva de la imagen, variaciones en la iluminación, el cual debe ser eliminado en la medida de lo posible [4].

La segmentación de los objetos de la imagen es la fase donde se tiene como objetivo separar el área correspondiente al dígito, y el área del fondo [5]. En la extracción de características se transforma la información de entrada en descriptores esenciales para distinguir un dígito de otro [6]. Finalmente, la clasificación representa la predicción de la clase (o etiqueta) para un objeto, basado en la similitud del dígito contra los dígitos de referencia [7]. Uno de los algoritmos más simples de clasificación es el vecino más

cercano (k-NN, del inglés k-Nearest Neighbor), el cual se basa en el uso de distancias (Euclidiana, Manhattan, Hamming, etc.) [8]. Por otro lado, las Redes Neuronales Artificiales (RNA) tienen como objetivo procesar información de la misma manera que el cerebro humano lo hace, por lo que cuentan con una fase de entrenamiento y aprendizaje y han sido muy utilizadas en la etapa de clasificación de dígitos manuscritos [1, 2, 7]. La técnica de Máquina con Vectores Soporte (SVM, del inglés Support Vector Machine), se basa en el descubrimiento de un hiperplano que se usa para separar los datos de las clases existentes [9]. En [10] se puede consultar un estudio de las diferentes técnicas de clasificación de dígitos manuscritos.

Aun cuando en la literatura se han presentado diversos trabajos sobre el reconocimiento de dígitos manuscritos, la tarea sigue representando un reto en el área de visión por computadora, debido a la diversidad de estilos de escritura asociados a diferentes personas, las herramientas utilizadas para escribir (pluma, lápiz), y la falta de cuidado al realizar los trazos. Por lo tanto, en el presente trabajo se realiza una comparación entre dos métodos de clasificación para el reconocimiento de dígitos manuscritos fuera de línea: un método fundamentado en el uso de una RNA de perceptrones, y un nuevo método basado en comparaciones similar al k-NN.

2. Materiales y métodos

La metodología para la clasificación de dígitos manuscritos utilizada en el presente artículo es similar a la explicada en la sección 1 y se puede observar en la Figura 1.

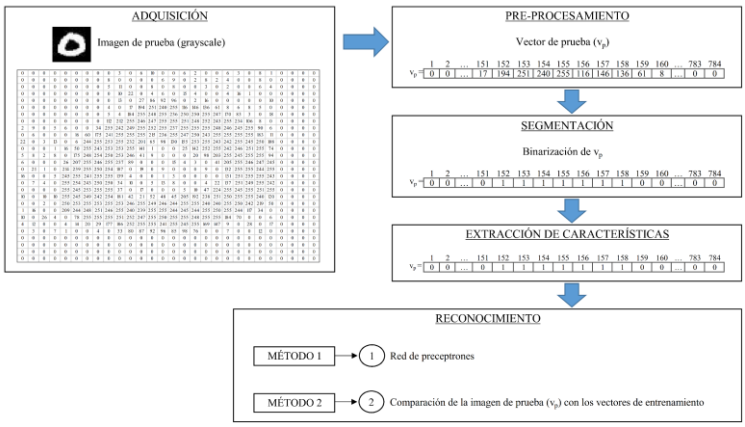


Fig. 1. Etapas propuestas para el reconocimiento de dígitos manuscritos

En la literatura sobre reconocimiento de dígitos manuscritos, existen diversas bases de datos tales como: The Street View House Numbers (SVHN) [11], The National Institute of Standards and Technology (NIST) Special Database 19 [12], y The Modified National Institute of Standards and Technology (MNIST) database [13, 14], que se han utilizado como estándar para probar la robustez de los métodos propuestos. Sin embargo, MNIST creada por Lecun [14], es la que más se utiliza en la literatura,

Fig. 2. Ejemplo de algunos dígitos del subconjunto seleccionado de MNIST.

por lo que fue seleccionada en el presente artículo. La base de datos MNIST se forma por un conjunto de 60000 imágenes de entrenamiento y 10000 de prueba de dígitos manuscritos del 0 al 9, las cuales están normalizadas, es decir, cada dígito tiene un tamaño de 20 x 20 pixeles, y se encuentra sobre un fondo de 28 x 28 pixeles (centrado mediante el centro de gravedad). Para probar los dos métodos de clasificación del presente artículo, se seleccionó al azar un subconjunto de 450 imágenes, 100 para la fase de entrenamiento, 10 por cada dígito desde el 0 al 9, y 350 para la fase de pruebas, 35 por cada dígito desde el 0 al 9. En la Figura 2, se muestra un ejemplo de algunos de los dígitos seleccionados.

Cada una de las 450 imágenes fue preprocesada por medio de una transformación (reacomodo) que genera una representación vectorial de 1 x 784 píxeles. En seguida, cada vector (que se encuentra en escala de grises, con valores entre 0 y 255) es segmentado por medio de un proceso de binarización. Para binarizar las imágenes se aplicó la regla siguiente: para cada pixel de la imagen de entrada (Im), si el valor del pixel es menor que 100, que es el valor del umbral[1], se le asigna un valor de 0, y si el valor del pixel es mayor o igual que 100, entonces se le asigna un valor de 1, como se puede observar en las Ecuaciones 1 y 2.

$$Si\ Im(i,j) < 100, \quad Im(i,j) = 0\ con\ i,j\ de\ 1\ a\ 28, \tag{1}$$

$$Si\ Im(i,j) \geq 100, \quad Im(i,j) = 1\ con\ i,j\ de\ 1\ a\ 28. \tag{2}$$

Los 784 valores obtenidos del proceso de segmentación son utilizados como el vector de características para describir cada uno de los dígitos, cabe mencionar que en el presente trabajo no se realizó una etapa de selección de características. Finalmente, se realiza el reconocimiento de los dígitos por medio de dos métodos diferentes.

Antes de describir los métodos utilizados para clasificación es importante mencionar que: a) El primer método, utiliza un concepto conocido: el perceptrón, b) El segundo método, es un método nuevo propuesto en el presente trabajo, el cual consiste en la aplicación de una regla de comparación entre imágenes, c) Ambos métodos se aplicaron a los mismos conjuntos de imágenes, d) Los experimentos para ambos métodos se realizaron bajo las mismas condiciones.

[1] Después de un análisis exhaustivo donde se analizó el valor de cada pixel de la imagen, se obtuvo que el umbral 100 permite que todos los dígitos queden segmentados de manera adecuada.

2.1. Método 1: red de perceptrones

El primer método utilizó una red de perceptrones para la clasificación de las imágenes. Un perceptrón es un algoritmo de aprendizaje binario que permite determinar si el dígito a clasificar pertenece o no a determinada clase [15].

Para crear la red, se utilizó la función *perceptron* incluida en la librería estándar de MATLAB, la cual emplea como base para la clasificación la función de transferencia *hardlimit*. La función *perceptron* requiere una fase de entrenamiento (*train*), y, utiliza el parámetro *epoch*, que es la aplicación de la regla de aprendizaje de la red a cada ejemplo en el conjunto de datos (1 *epoch* = 1 aplicación de la regla). El método utilizó, en promedio, alrededor de 6 *epochs* por perceptrón.

Como el perceptrón realiza la clasificación entre dos tipos de clases, se construyó una red de 90 perceptrones para poder realizar todas las comparaciones entre la imagen de prueba y las 10 clases posibles como se muestra en la Figura 3.

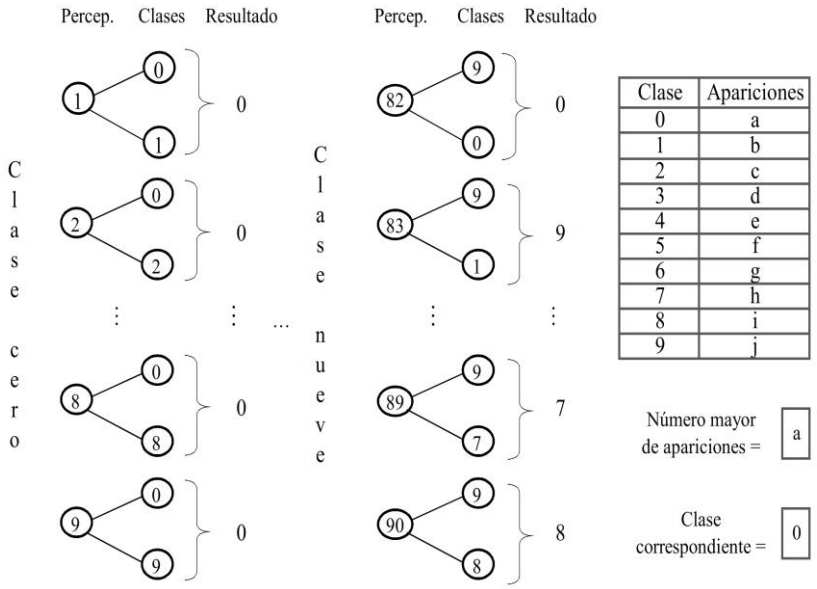


Fig. 3. Red de perceptrones utilizada en el método 1.

Las comparaciones para decidir cuál era la clase a la que más se parecía la imagen de prueba se realizaron por pares, y finalmente, se realizó la clasificación con la clase más repetida en todas las comparaciones hechas. La Tabla 1 muestra un ejemplo con una imagen de prueba correspondiente a la clase cero.

Cada renglón de la Tabla 1 corresponde a dos perceptrones, donde las columnas referentes a las clases indican cuáles fueron las clases analizadas (por pares), y la columna de clasificación muestra cuál fue el resultado obtenido. En la Tabla 2, se muestra el número de apariciones de cada una de las clases predichas por el clasificador para la imagen de la Tabla 1.

Tabla 1. Clasificación de una imagen de prueba correspondiente a un cero.

Clase 1	Clase 2	Clasificación	Clase 1	Clase 2	Clasificación
cero	uno	cero	cinco	cero	cero
cero	dos	cero	cinco	uno	cinco
cero	tres	cero	cinco	dos	cinco
cero	cuatro	cero	cinco	tres	tres
cero	cinco	cero	cinco	cuatro	cinco
cero	seis	cero	cinco	seis	cinco
cero	siete	cero	cinco	siete	cinco
cero	ocho	cero	cinco	ocho	cinco
cero	nueve	cero	cinco	nueve	cinco
uno	cero	cero	seis	cero	cero
uno	dos	dos	seis	uno	seis
uno	tres	tres	seis	dos	seis
uno	cuatro	cuatro	seis	tres	tres
uno	cinco	cinco	seis	cuatro	cuatro
uno	seis	seis	seis	cinco	cinco
uno	siete	siete	seis	siete	siete
uno	ocho	ocho	seis	ocho	ocho
uno	nueve	nueve	seis	nueve	seis
dos	cero	cero	siete	cero	cero
dos	uno	dos	siete	uno	siete
dos	tres	tres	siete	dos	dos
dos	cuatro	dos	siete	tres	tres
dos	cinco	cinco	siete	cuatro	cuatro
dos	seis	seis	siete	cinco	siete
dos	siete	dos	siete	seis	siete
dos	ocho	dos	siete	ocho	ocho
dos	nueve	dos	siete	nueve	siete
tres	cero	cero	ocho	cero	cero
tres	uno	tres	ocho	uno	ocho
tres	dos	tres	ocho	dos	dos
tres	cuatro	tres	ocho	tres	tres
tres	cinco	cinco	ocho	cuatro	ocho
tres	seis	tres	ocho	cinco	cinco
tres	siete	tres	ocho	seis	ocho
tres	ocho	tres	ocho	siete	ocho
tres	nueve	tres	ocho	nueve	ocho
cuatro	cero	cero	nueve	cero	cero
cuatro	uno	cuatro	nueve	uno	nueve
cuatro	dos	dos	nueve	dos	dos
cuatro	tres	tres	nueve	tres	tres
cuatro	cinco	cinco	nueve	cuatro	cuatro
cuatro	seis	cuatro	nueve	cinco	cinco
cuatro	siete	cuatro	nueve	seis	nueve
cuatro	ocho	cuatro	nueve	siete	siete
cuatro	nueve	cuatro	nueve	ocho	ocho

Tabla 2. Resumen del número total de aparición de clases para la imagen de la Tabla 1.

Clase	Apariciones	Clase	Apariciones
cero	18	cinco	14
uno	0	seis	5
dos	10	siete	7
tres	15	ocho	9
cuatro	9	nueve	3

Como se puede observar en la Tabla 2, la clase que más se predijo fue la clase cero, con 18 apariciones (respecto al total de los 90 resultados), por lo cual la imagen fue clasificada como un cero. Cabe mencionar que en ninguna de las pruebas se presentó algún caso de empate entre el número mayor de apariciones de las clases, por lo que no se generó un criterio de desempate.

2.2. Método 2: comparación vector contra vector

El segundo método consiste en la aplicación de un nuevo algoritmo de comparación creado en MATLAB. El método se basa, como su nombre lo indica, en una comparación directa entre el vector de prueba (v_p) y los vectores de entrenamiento (v_e). Los vectores de entrenamiento se componen por 10 vectores renglón por dígito, cada uno de ellos correspondiente a una imagen, conformando así 100 vectores de entrenamiento en total. El vector de prueba corresponde al vector renglón que representa la imagen que se quiere clasificar. Es importante aclarar que los vectores de entrenamiento hacen referencia a los vectores que se están tomando como punto de referencia para la comparación que realiza el método, por lo tanto, el método propuesto al igual que un k-NN **no tiene** una fase de aprendizaje, además el método propuesto no necesita una medida de distancia.

El vector de prueba es comparado contra cada uno de los vectores de entrenamiento, es decir, cada una de sus componentes se contrasta contra cada una de las componentes de los 100 vectores, para calcular a cuál se parece más. Por lo que, el número de comparaciones hechas por cada imagen de prueba que se compara, contra cada clase, es de 78400 (784 componentes del vector de prueba por 100 vectores de entrenamiento). En cada comparación, se asigna un valor específico que se obtiene de la siguiente manera: por cada componente correspondiente de los vectores que sea exactamente igual, a la comparación se le fija un valor 0, y por cada que sea distinta se le fija un valor 1, como se muestra en las Ecuaciones 3 y 4.

$$Si\ v_p(1,i) = v_e(1,i), \qquad a = 0\ con\ i\ de\ 1\ a\ 784, \tag{3}$$

$$Si\ v_p(1,i) \neq v_e(1,i), \qquad a = 1\ con\ i\ de\ 1\ a\ 784. \tag{4}$$

Finalmente, todos los valores de cada comparación entre componentes se suman y generan como resultado un valor de comparación final, el cual se almacena en un vector (z), como se muestra en la Ecuación 5, de manera que cuando se realizan las 100 comparaciones, se selecciona aquella con el valor más pequeño (ver Ecuación 6), y se le determina una clase del cero al nueve, de acuerdo con la posición del valor mínimo.

Cabe mencionar que en las pruebas realizadas no se presentó algún caso de empate entre los valores más pequeños, por lo que no se generó un criterio de desempate.

$$z(j,1) = \sum_{i=1}^{784} a_i \ , con\ j\ de\ 1\ a\ 100, \tag{5}$$

$$Clase \leftarrow \min(z). \tag{6}$$

En la Figura 4 se muestra de manera visual un ejemplo del funcionamiento del nuevo método propuesto.

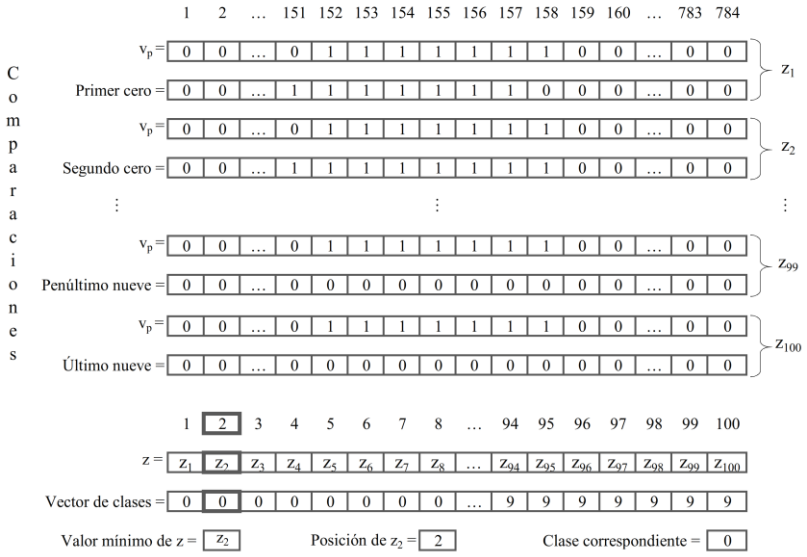


Fig. 4. Funcionamiento del método de comparación de vector contra vector.

3 Experimentación y resultados

Para cada uno de los métodos se realizaron experimentos con los mismos dos conjuntos de imágenes: 1) entrenamiento (100), y 2) prueba (350). En ambos casos se presentan los resultados del entrenamiento y de la prueba. Todas las pruebas se realizaron en una laptop HP con procesador Intel Core i5 con memoria RAM de 8Gb y las implementaciones de los dos métodos se realizaron con MATLAB.

La Tabla 3 muestra la notación, fórmula y descripción de las variables medidas. Es importante aclarar que el comportamiento de ambos métodos a lo largo de los experimentos siempre fue el mismo, por lo que los datos presentados en las siguientes subsecciones corresponden a una única corrida por imagen.

3.1. Resultados obtenidos con el método 1

Tabla 3. Variables de medición obtenidas de [16].

Variable (notación)	Fórmula	Descripción
Verdaderos Positivos (VP)	-	Casos que fueron predichos SÍ, y realmente fueron SÍ
Verdaderos Negativos (VN)	-	Casos que fueron predichos NO, y realmente fueron NO
Falsos Negativos (FN)	-	Casos que fueron predichos NO, y realmente fueron SÍ
Falsos Positivos (FP)	-	Casos que fueron predichos SÍ, y realmente fueron NO
Exactitud	(VP+VN) / total	En general, ¿qué tan frecuente el clasificador acierta?
Tasa de error	(FP+FN) / total	En general, ¿qué tan frecuente el clasificador NO acierta?
Sensibilidad	VP / SÍ real	Cuando es un SÍ real, ¿qué tan frecuente el clasificador predice un SÍ?
Especificidad	VN / NO real	Cuando es un NO real, ¿qué tan frecuente el clasificador predice un NO?
Precisión	VP / SÍ predicho	Cuando el clasificador predice SÍ, ¿qué tan frecuente está en lo correcto?

Tabla 4. Método 1: resultados obtenidos con 100 imágenes de entrenamiento.

	CLASE										
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Total**
VP	9	10	10	10	10	10	10	10	10	10	99
VN	90	90	90	90	90	89	90	90	90	90	-
FP	0	0	0	0	0	1	0	0	0	0	-
FN	1	0	0	0	0	0	0	0	0	0	1
SÍ reales	10	10	10	10	10	10	10	10	10	10	100
NO reales	90	90	90	90	90	90	90	90	90	90	-
SÍ predichos	9	10	10	10	10	11	10	10	10	10	-
NO predichos	91	90	90	90	90	89	90	90	90	90	-
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Media**
Exactitud	0.9900	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000	1.0000	1.0000	1.0000	0.9980
Tasa de error	0.0100	0.0000	0.0000	0.0000	0.0000	0.0100	0.0000	0.0000	0.0000	0.0000	0.0020
Sensibilidad	0.9000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9900
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	0.9889	1.0000	1.0000	1.0000	1.0000	0.9989
Precisión	1.0000	1.0000	1.0000	1.0000	1.0000	0.9091	1.0000	1.0000	1.0000	1.0000	0.9909

Primero, se midió la capacidad para aprender del método 1. Los resultados obtenidos con las 100 imágenes de entrenamiento se muestran en la Tabla 4.

Los resultados obtenidos al aplicar el método 1 a las 350 imágenes de prueba se muestran en la Tabla 5.

Tabla 5. Método 1: resultados obtenidos con 350 imágenes de prueba.

	CLASE										
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Total**
VP	31	31	18	27	27	21	23	23	22	19	242
VN	308	314	303	308	305	289	302	307	298	309	-
FP	7	1	12	7	10	26	13	8	17	6	-
FN	4	4	17	8	8	14	12	12	13	16	108
SÍ reales	35	35	35	35	35	35	35	35	35	35	350
NO reales	315	315	315	315	315	315	315	315	315	315	-
SÍ predichos	38	32	30	34	37	47	36	31	39	25	-
NO predichos	312	318	320	316	313	303	314	319	311	325	-
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Media**
Exactitud	0.9686	0.9857	0.9171	0.9571	0.9486	0.8857	0.9286	0.9429	0.9143	0.9371	0.9386
Tasa de error	0.0314	0.0143	0.0829	0.0429	0.0514	0.1143	0.0714	0.0571	0.0857	0.0629	0.0614
Sensibilidad	0.8857	0.8857	0.5143	0.7714	0.7714	0.6000	0.6571	0.6571	0.6286	0.5429	0.6914
Especificidad	0.9778	0.9968	0.9619	0.9778	0.9683	0.9175	0.9587	0.9746	0.9460	0.9810	0.9660
Precisión	0.8158	0.9688	0.6000	0.7941	0.7297	0.4468	0.6389	0.7419	0.5641	0.7600	0.7060

Tabla 6. Método 2: resultados obtenidos con 100 imágenes de entrenamiento.

	CLASE										
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Total**
VP	10	10	10	10	10	10	10	10	10	10	100
VN	90	90	90	90	90	90	90	90	90	90	-
FP	0	0	0	0	0	0	0	0	0	0	-
FN	0	0	0	0	0	0	0	0	0	0	0
SÍ reales	10	10	10	10	10	10	10	10	10	10	100
NO reales	90	90	90	90	90	90	90	90	90	90	-
SÍ predichos	10	10	10	10	10	10	10	10	10	10	-
NO predichos	90	90	90	90	90	90	90	90	90	90	-
Variables	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Media**
Exactitud	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Tasa de error	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sensibilidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Especificidad	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Precisión	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

3.2. Resultados obtenidos con el método 2

Los resultados obtenidos al aplicar el método 2 a las 100 imágenes de entrenamiento se muestran en la Tabla 6.

Los resultados obtenidos al aplicar el método 2 a las 350 imágenes de prueba se muestran en la Tabla 7.

Tabla 7. Método 2: resultados obtenidos con 350 imágenes de prueba.

	CLASE										
VARIABLES	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**TOTAL**
VP	33	34	22	23	28	23	28	27	24	23	265
VN	305	301	312	310	308	304	310	303	312	300	-
FP	10	14	3	5	7	11	5	12	3	15	-
FN	2	1	13	12	7	12	7	8	11	12	85
SÍ reales	35	35	35	35	35	35	35	35	35	35	350
NO reales	315	315	315	315	315	315	315	315	315	315	-
SÍ predichos	43	48	25	28	35	34	33	39	27	38	-
NO predichos	307	302	325	322	315	316	317	311	323	312	-
VARIABLES	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**MEDIA**
Exactitud	0.9657	0.9571	0.9543	0.9514	0.9600	0.9343	0.9657	0.9429	0.9600	0.9229	0.9514
Tasa de error	0.0343	0.0429	0.0457	0.0486	0.0400	0.0657	0.0343	0.0571	0.0400	0.0771	0.0486
Sensibilidad	0.9429	0.9714	0.6286	0.6571	0.8000	0.6571	0.8000	0.7714	0.6857	0.6571	0.7571
Especificidad	0.9683	0.9556	0.9905	0.9841	0.9778	0.9651	0.9841	0.9619	0.9905	0.9524	0.9730
Precisión	0.7674	0.7083	0.8800	0.8214	0.8000	0.6765	0.8485	0.6923	0.8889	0.6053	0.7689

3.3. Discusión

De acuerdo con los resultados obtenidos con el primer conjunto de imágenes (de entrenamiento) mostrados en las Tablas 4 y 6, se observa que el método 2 presentó un mejor desempeño, ya que clasificó correctamente todos los casos, mientras que el método 1 clasificó correctamente 99 de las 100 imágenes. Aunque los resultados arrojados por el método 1 son (relativamente) buenos, no son los esperados, ya que se esperaba que pudieran ser reconocidas todas las imágenes del conjunto, tal como lo hizo el método 2.

Con los resultados obtenidos en los experimentos hechos sobre el conjunto de imágenes de prueba mostrados en las Tablas 5 y 7, se observa que el método 2 es mejor que el método 1. La comparación de las variables obtenidas mediante la aplicación de ambos métodos se muestra en la Tabla 8.

Tabla 8. Comparación de los métodos.

Variables	Método 1	Método 2
Exactitud	0.9386	0.9514
Tasa de error	0.0614	0.0486
Sensibilidad	0.6914	0.7571
Especificidad	0.9660	0.9730
Precisión	0.7060	0.7689

Como se puede observar en la Tabla 8, con el método 1 se obtuvo una sensibilidad de 69%, es decir, clasificó de manera correcta 242 de las 350 imágenes, mientras que con el método 2 se obtuvo una sensibilidad de 75%, es decir, de las 350 imágenes, clasificó correctamente 265. De igual forma, el método 2 presentó mejores resultados en exactitud, pues el clasificador fue mejor al momento de clasificar de manera general

Fig. 5. Ejemplos de resultados para las imágenes de prueba.

las imágenes, es decir, acertó de manera más frecuente sobre cuándo una imagen pertenece a una clase, y cuándo no pertenece a una clase.

La Figura 5 ilustra algunos ejemplos tomados del conjunto de imágenes de prueba, con su respectiva clasificación (se debe observar que existen dígitos que a simple vista son difíciles de reconocer). El número a la izquierda (en negritas) representa la clase correcta a la que pertenece la imagen, los siguientes dos números refieren a la clase dada por ambos clasificadores, (método 1/método 2).

Aunque los resultados entre los dos métodos se encuentran relativamente cerca, la principal diferencia radica en el tiempo de ejecución. El método 2 fue significativamente más rápido para la prueba y entrenamiento, en comparación con el método 1. Para crear una idea sobre qué tan significativa fue la diferencia respecto a los tiempos de ejecución, considere lo siguiente: el método 2 tardó en clasificar las 450 imágenes en un tiempo menor a 10 s, mientras que el método 1 tardó en clasificar las mismas imágenes, y bajo las mismas condiciones, alrededor de 90 min. Por lo que se si se utilizaran, por ejemplo, 42000 imágenes de MNIST, el método 2 terminaría de clasificarlas todas, en el mismo tiempo que el método 1 solamente clasificaría 70.

Con respecto a la bondad del método presentado contra los trabajos de la literatura se observa que es difícil realizar una comparación real. Aun cuando existen trabajos que utilizan MNIST, normalmente, no se utiliza el mismo subconjunto de imágenes y se tendrían que programar los otros métodos. Sin embargo, para el caso de la clasificación nuestro trabajo obtuvo una exactitud de 0.9514 con el nuevo método propuesto, mientras que en el trabajo de [2], se utilizó una RNA de pico de descenso aproximado normalizado y se obtuvo una exactitud de 0.9817; en el trabajo de [5], se utilizó una SVM y se obtuvo una exactitud de 0.9691, en la investigación de [7], se utilizó una red de creencias profundas con aprendizaje Q y se obtuvo una exactitud de 0.9918, además se reporta un tiempo de 21.46 s para la clasificación de 100 imágenes de MNIST. Finalmente, en el trabajo presentado en [9], se utilizó un híbrido de una SVM y una red neuronal convolucional en la que se reporta una exactitud de 0.9981. Como se puede observar, el método 2 presentado en este artículo obtuvo resultados de exactitud competitivos contra los presentados en el estado del arte y además buen tiempo de ejecución, con la característica adicional que nuestro método es computacionalmente menos complejo.

4. Conclusiones

Se presentó una comparación entre dos métodos para el reconocimiento de dígitos manuscritos fuera de línea. Para poder realizar una comparación justa para ambos métodos se utilizó un subconjunto de 450 imágenes de MNIST. En el método 1 se implementó una red de perceptrones, mientras que en el método 2 se presentó un nuevo

algoritmo que consiste en la comparación entre las características de imágenes de prueba y las características de imágenes de referencia similar al k-NN con la diferencia de que no se calculan distancias.

De acuerdo con los resultados obtenidos se concluye que el nuevo método propuesto (2) tiene un mejor desempeño comparado con el método 1. La conclusión se obtiene a partir de la eficiencia del clasificador al realizar la clasificación con los diversos conjuntos de imágenes, así como el tiempo de ejecución que utiliza cada método para realizar la tarea asignada. Además de la eficiencia en la clasificación es importante destacar que, el método propuesto en el presente trabajo está basado en el uso de operaciones de comparación, en donde se conoce qué está siendo comparado, y cómo se realiza la comparación, mientras que en el método 1 no ocurre lo mismo, puesto que al utilizar el perceptrón se conoce qué se está clasificando, pero no cómo está siendo clasificado.

Como trabajo futuro se considera: aumentar el tamaño del conjunto de imágenes de entrenamiento y de prueba, y modificar el método 1 para reducir el tiempo de ejecución, a través de la reducción del número de perceptrones utilizados, mediante el descarte temprano de algunas clases. Además, será importante realizar pruebas con otros tipos de clasificadores.

Agradecimientos. H. Sossa agradece al Instituto Politécnico Nacional y al CONACYT, a través de los apoyos económicos en el marco de fondos SIP 20180730 y 65 (Fronteras de la Ciencia). María Cristina Guevara Neri agradece al CONACYT por la beca otorgada para la realización de sus estudios de doctorado.

Referencias

1. Kulkarni, S., Rajendran, B.: Spiking neural networks for handwritten digit recognition-supervised learning and network optimization. Neural Networks, 103, pp.118–127 (2018)
2. Sueiras, J., Ruiz, V., Sanchez, A., Velez, J.: Offline continuous handwriting recognition using sequence to sequence neural networks. Neurocomputing, 289, pp. 119–128 (2018)
3. Kumar, M., Jindal, M., Sharma, R., Rani, S.: Character and numeral recognition for non-Indic and Indic scripts: A survey. Artificial Intelligence Review, pp. 1–27 (2018)
4. Mandal, S., Mahadeva, S., Sundaram, S.: GMM posterior features for improving online handwriting recognition. Expert Systems with Applications, 97, pp. 421–433 (2018)
5. Gattal, A., Chibani, Y., Hadjadji, B.: Segmentation and recognition system for unknown-length handwritten digit strings. Pattern Analysis and Applications. 20(2), pp. 307–323 (2017)
6. Surinta, O., Karaaba, M., Schomaker, L., Wiering, M.: Recognition of handwritten characters using local gradient feature descriptors. Engineering Applications of Artificial Intelligence, 45, pp. 405–414 (2015)
7. Qiao, J., Wang, G., Li, W., Chen, M.: An adaptive deep Q-learning strategy for handwritten digit recognition. Neural Networks. Article in Press, pp. 1–18 (2018)
8. Bhattacharya, G., Ghosh, K., Chowdhury, A.: An affinity-based new local distance function and similarity measure for kNN algorithm. Pattern Recognition Letters. 33(3), pp. 356–363 (2012)
9. Niu, X., Suen, C.: A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognition, 45(4), pp. 1318–1325 (2012)

10. Al-Helali, B., Mahmoud, S.: Arabic online handwriting recognition (AOHR): A survey. ACM Computing Surveys, 50(3), pp. 1–35 (2017)
11. The Street View House Numbers (SVHN) Dataset: http://ufldl.stanford.edu/housenumbers/, last accessed (2018)
12. NIST: National Institute of Standards and Technology. NIST Special Database 19 (2018)
13. The MNIST database: http://yann.lecun.com/exdb/mnist/ (2018)
14. LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, A., Sackinger, E., Simard, P., Vapnik, V.: Learning algorithms for classification: A comparison on handwritten digit recognition. Neural Networks: The Statistical Mechanics Perspective, pp. 261–276 (1995)
15. Kurkova, V., Sanguineti, M.: Probabilistic lower bounds for approximation by shallow perceptron networks. Neural Networks, 91, pp. 31–41 (2017)
16. Markham, K.: Simple guide to confusion matrix terminology, http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (2018)

ISSN 1870-4069

Algoritmo de aprendizaje eficiente para tratar el problema del desbalance de múltiples clases

J. Monroy-de-Jesús, A. Guadalupe-Ramírez, J.C. Ambriz-Polo, E. López-González

Tecnológico de Estudios Superiores de Jocotitlán, Estado de México, México

gr.lizeth.a@gmail.com, juan_120990@hotmail.com, {juan.monroy, erika.lopez}@tesjo.edu.mx

Resumen. En este trabajo se presenta un enfoque de muestreo dinámico (I-SDSA) para tratar el problema de desequilibrio de múltiples clases. ISDSA es una modificación del algoritmo *Backpropagation*, que se enfoca en hacer un mejor uso de las muestras de entrenamiento para mejorar el rendimiento de clasificación del perceptrón multicapa (MLP). I-SDSA usa el error cuadrático medio y una función gaussiana para identificar las mejores muestras para entrenar la red neuronal. Los resultados que se muestran en este artículo destacan que I-SDSA aprovecha mejor el conjunto de datos de entrenamiento y mejora el rendimiento de clasificación de MLP. En otras palabras, I-SDSA es una técnica exitosa para lidiar con el problema del desequilibrio de múltiples clases. Además, los resultados presentados en este trabajo indican que el método propuesto es muy competitivo en términos de rendimiento de clasificación con respecto a los métodos clásicos de muestreo y otros enfoques dinámicos de muestreo, incluso en el tiempo de entrenamiento y el tamaño de la base de datos es mejor que en los métodos de muestreo.

Palabras clave: Backpropagation, MLP, múltiples clases, error cuadrático medio.

Efficient Learning Algorithm to Face the Multi-Class Imbalance Problem

Abstract. In this paper, a dynamic sampling approach (I-SDSA) is presented to deal with the problem of imbalance of multiple classes. I-SDSA is a modification of the Backpropagation algorithm, which focuses on making better use of training samples to improve multilayer perceptron (MLP) classification performance. I-SDSA uses the mean square error and a Gaussian function to identify the best samples to train the neural network. The results shown in this article highlight that I-SDSA takes better advantage of the training data set and improves the MLP classification performance. In other words, I-SDSA is a successful technique for dealing with the problem of imbalance of multiple classes. In addition, the results presented in this paper indicate that the proposed method is very competitive in terms of classification performance with respect to classical sampling methods

and other dynamic sampling approaches, even in the training time and the size of the base of data is better than in sampling methods.

Keywords: Backpropagation, MLP, multi-class, mean square error.

1. Introducción

El desbalance entre clases resulta ser un problema fundamental en las redes neuronales artificiales o ANN (Artificial Neural Network por sus siglas en inglés) ya que afecta el rendimiento del clasificador, no solo a su capacidad de generalización sino también en el costo computacional asociado a la fase de entrenamiento cuando éste es entrenado con métodos iterativos. El problema del desbalance de clases se presenta con mucha frecuencia en el mundo real, por ejemplo, en la detección de llamadas fraudulentas, imágenes de percepción remota, medicina, entre otras [1].

Se han propuesto diversos métodos para resolver el desbalance de clases, por ejemplo, el uso de técnicas de muestreo, los cuales duplican o eliminan patrones o muestras de entrenamiento hasta alcanzar un relativo equilibrio entre el número de muestras de las distintas clases (Over-sampling o Under-sampling) [2].

Uno de los métodos más comunes de sobre-muestreo (over-sampling), es la técnica de SMOTE (técnica de muestreo sintético) propuesta por Chawla et al. [3]. El funcionamiento de esta técnica consiste principalmente en generar nuevas muestras sintéticas interpoladas entre patrones de la clase minoritaria. Asimismo, el SMOTE ha servido de base para el desarrollo de otros métodos de sobre muestreo. Por ejemplo, *Bordeline-SMOTE, Adaptive Synthetic Sampling (ADASYN), SMOTE Editing Nearest Neighbor, Safe-LevelSMOTE, DBSMOTE, SMOTE + Tomeks Links*, entre otros.

Por otro lado, en el dominio de las técnicas de sub-muestreo (under-sampling), el método de sub-muestreo aleatorio ha sido reportado como uno de los más efectivos [4]. No obstante, se han generado numerosas aproximaciones a los métodos de sub-muestreo, las cuales se caracterizan por incluir un mecanismo heurístico en su funcionamiento. Básicamente este componente heurístico tiene como objetivo eliminar o cambiar las etiquetas de patrones ya sean ruido, atípicos o redundantes [5]. Por ejemplo, los métodos *Neighborhood Cleaning Rule, Onesided selection, Tomek links* y *Condensed Nearest Neighbor Rule*.

Actualmente, se está incrementado el interés por el estudio y desarrollo de métodos de muestreo dinámico, en los cuales la proporción y selección de muestras a duplicar o eliminar se realiza durante el entrenamiento del clasificador. Por ejemplo, en [6] se propone un nuevo algoritmo para determinar el nivel de equilibrio de clases, y además incluyen un mecanismo de selección de patrones de entrenamiento difíciles de aprender, con el propósito de mejorar la capacidad de generalización del MLP entrenado con el algoritmo Backpropagation.

Chawla et al. [7] propone un paradigma del tipo WRAPPER para determinar el nivel de sobre y sub-muestreo a aplicar en cada base de datos desbalanceada, que va ser usada para entrenar el clasificador. Wang and Jean [8] proponen el método SNOWBALL para entrenar redes del tipo MLP con datos desbalanceados, básicamente este método repite el entrenamiento de las muestras de las clases minoritarias hasta que el clasificador las identifica adecuadamente

En este sentido mientras que el problema de desbalance entre dos clases ha sido ampliamente estudiado, en dominios de múltiples clases, este inconveniente ha sido muy poco tratado.

2. Trabajos relacionados

En los últimos años el problema del desbalance de clases se ha abordado de muchas maneras y enfoques diferentes, sin embargo, los métodos más estudiados han sido los métodos de muestreo, por ejemplo, véanse las referencias [6, 9]. Estos métodos suelen ser eficaces y son independientes del clasificador.

Los métodos de muestreo pueden ser tan simples y claros como ROS o RUS [10], pero mientras que el primero replica muestras existentes en la clase minoritaria es más probable que ocurra sobre ajuste [11], y el segundo puede quitar tantas muestras permita la proporción de desbalance de clases, que, en algunos escenarios, sería inapropiado debido a la enorme pérdida de información en la base de datos.

Por lo tanto, se han desarrollado otros métodos de muestreo inteligentes que incluyen un mecanismo heurístico [12], como SMOTE, que crea muestras artificiales de la clase minoritaria mediante la interpolación de muestras existentes cerca de ellas [11] y de esta forma evitar la sobre especialización.

Otra técnica propuesta para superar las deficiencias de las técnicas de muestreo como ROS o SMOTE es Borderline-SMOTE [13], la cual selecciona muestras de la clase minoritaria que están en el límite, realizando sólo SMOTE en esas muestras. El muestreo sintético adaptativo (ADASYN) es una extensión de SMOTE, creando en la región límite más muestras entre las dos clases que en el interior de la clase minoritaria [14]. SMOTE Editing Nearest Neighbor (ENN) consiste en aplicar SMOTE y, a continuación, la regla ENN [15]. Safe-Level-SMOTE generan muestras de clase minoritaria sintéticas situadas más cerca del mayor nivel de seguridad, entonces todas las muestras sintéticas sólo se generan en regiones seguras [16].

SMOTE + Tomek Links (TL) [17] es la combinación de SMOTE y TL [15], Neighborhood Cleaning Rule usa la regla ENN, pero sólo elimina las muestras de la clase mayoritaria. Condensed Nearest Neighbor rule (CNN) [18] y One-sided selection eliminan las muestras redundantes, pero esta última usa TL. Show-Jane y Yue-Shi [19] presentan un nuevo método de sub-muestreo basado en métodos de agrupamiento para seleccionar los datos representativos como datos de entrenamiento para mejorar la precisión de clasificación para la clase minoritaria.

Otros enfoques de muestreo importantes se estudian en las referencias [9, 20, 21]. Se han propuesto métodos más sofisticados para tratar el problema del desbalance entre varias clases. Por ejemplo, el costo sensitivo (CS), que es uno de los temas más relevantes en la investigación en aprendizaje automático [22], es una buena solución para el problema de desbalance de clases [23].

El CS utiliza los costos asociados con la clasificación errónea de las muestras, emplea varias matrices de costos que definen los costos de clasificación errónea de cualquier muestra de datos [20]. Sin embargo, en estos métodos el costo de clasificación errónea debe ser conocido, pero en un problema de clasificación real, el costo de clasificación errónea es a menudo desconocido [24]. Zhi-Hua y Xu-Ying [22]

proporcionan un marco unificado para el uso de CS para abordar el desbalance de clases.

Los métodos ENSEMBLE es otro enfoque para tratar de resolver el problema de desbalance de clases. Esta técnica entrena múltiples componentes y luego combina sus predicciones [23, 25]. Sun et al. [24] emplean un conjunto de Máquinas de Vectores de Soporte, y el margen máximo se adopta para guiar el procedimiento de aprendizaje de conjuntos para la clasificación de imágenes de percepción remota. Galar et al. [26] presentan una revisión exhaustiva sobre los ensambles para el problema de desbalance de clases.

Recientemente, se han propuesto métodos de muestreo dinámico para resolver el problema de desbalance de múltiples clases. Estos establecen automáticamente la tasa de muestreo, por ejemplo, Fernández-Navarro et al. [27, 28] combinan métodos a nivel de datos con técnicas de entrenamiento dinámico. Utilizan algoritmos gen éticos para obtener la mejor relación de sobre-muestreo. Chawla et al. [7] proponen un paradigma Wrapper que descubre automáticamente la cantidad de sub-muestreo y tasa de sobre-muestreo para un conjunto de datos basado en optimización de las funciones de evaluación.

3. Análisis de muestras seguras, promedio y de frontera

En la literatura especializada sobre el problema de desbalance entre clases, se busca el interés de encontrar las mejores muestras para construir los clasificadores, eliminando aquellas muestras con alta probabilidad de ser ruido o muestras superpuestas [10, 29], es decir, aquellos cercanos a la decisión límite [13, 14] (este último ha sido menos explorado). Por lo tanto, en la literatura se pueden identificar básicamente tres categorías de muestras:

- Ruido y muestras raras o extrañas. Los primeros son casos con errores en sus etiquetas [30] o valores erróneos en sus rasgos que los describen, y los últimos son muestras minoritarias y raras situadas dentro de la clase mayoritaria [31].
- Las muestras fronterizas o superpuestas. Son aquellas localizadas donde se cruzan las regiones fronterizas de decisión [32].
- Las muestras seguras. Son aquellas con alta probabilidad de ser correctamente clasificadas y están rodeados de muestras de la misma clase [31].

Sin embargo, hay otras muestras que podrían ser de interés, las muestras situadas cerca de la decisión límite y lejos de las muestras seguras. Estas muestras se conocen como muestras" promedio".

En este trabajo en primera instancia se realizó la identificación de las muestras promedio, utilizando la salida de la red neuronal para analizar las muestras de entrenamiento, así como una función gaussiana γ, para identificar el tipo de muestra.

Esto se puede observar en las siguientes funciones:

$$\gamma(diff) = \exp(\frac{||diff - \mu||^2}{2\sigma^2}). \quad (1)$$

La variable $diff$ es la diferencia normalizada entre la salida real de la ANN para la muestra q:

$$diff = \frac{z_{min}^{q}}{\sqrt{(z_{min}^{q} - z_{maj}^{q})^2}} - \frac{z_{maj}^{q}}{\sqrt{(z_{min}^{q} - z_{maj}^{q})^2}}, \quad (2)$$

donde z_{min}^{q} y z_{maj}^{q} son las salidas reales de la ANN correspondientes a las clases minoritarias y mayoritarias (respectivamente) para una muestra q. La variable μ se calcula bajo la siguiente consideración: las salidas de la ANN se codifican usualmente en valores 0 y 1. Por ejemplo, para un problema de dos clases (clase A y clase B) las salidas ANN deseadas se codifican como (1; 0) y (0; 1)), respectivamente. Estos valores son las salidas objetivo de la ANN y los valores finales esperados son emitidos por la ANN después del entrenamiento. Por lo tanto, de acuerdo con este entendimiento, los valores esperados por μ son:

a) μ = 1.0 para muestras seguras, ya que se espera que la ANN se clasifique con alto nivel de precisión, las salidas de la ANN para todas las neuronas son valores cercanos a (0,1) o (1,0). Por lo tanto, si se aplica la Ecuación 2 el valor esperado (idealmente) es 1,0, por lo tanto, la función γ (Ecuación (1)) obtiene su valor máximo.
b) μ = 0.0 para muestras de frontera, ya que se espera que el clasificador no clasifique correctamente, es decir, las salidas esperadas para todas las neuronas son valores cercanos a (-0.5 o 0.5), por lo que en la ecuación (2) es aproximadamente 0.0, y la función γ (Ecuación 1) obtiene su valor máximo para estas muestras.
c) μ = 0.5 para muestras promedio, ya que se espera que la ANN se clasifique con menos exactitud, debido a que las muestras promedio se encuentran entre las muestras seguras (μ = 1.0) y frontera (μ = 0.0).

La función γ está propuesta para dar un cierto grado de prioridad a cada tipo de muestras. El objetivo es identificar cada tipo de muestra para ese valor μ. La ecuación. (2) da valores altos a las muestras cuando su *diff* (Ecuación (1)) es cercano a μ y valores bajos cuando el *diff* está lejos de μ.

Básicamente, el proceso para seleccionar las muestras es el siguiente: Antes de la formación de la ANN, el conjunto de datos de entrenamiento es equilibrado al 100% mediante una técnica eficaz de sobre-muestreo. Durante el entrenamiento, el método propuesto selecciona las muestras usando la ecuación (1) para actualizar los pesos de la red neuronal, elige desde el conjunto de datos de entrenamiento equilibrado sólo las mejores muestras para usar en el entrenamiento de la red neuronal.

3.1. Enfoque selectivo de muestreo dinámico (I-SDSA)

SDSA se basa en la idea de utilizar sólo las muestras más apropiadas durante la etapa de entrenamiento del MLP (muestras promedio), para mejorar el rendimiento del clasificador. I-SDSA funciona de la siguiente manera:

1. Antes del entrenamiento: Los datos de entrenamiento son balanceados al 100% mediante una técnica eficaz de sobre-muestreo.

2. Durante el entrenamiento: Del conjunto de datos de entrenamiento balanceados, I-SDSA elige las mejores muestras para ser usados en el entrenamiento del

MLP. Con el objetivo de identificar las mejores muestras, para esto utiliza la siguiente función:

$$\gamma(\Delta^q) = exp(-\frac{||\Delta^q - \mu||^2}{2\sigma^2}). \quad (3)$$

La variable Δq es la diferencia normalizada entre las salidas reales (Z), a y b de la red neuronal, para una muestra q:

$$\Delta^q = \frac{z_a^q}{\sqrt{(z_a^q - z_b^q)^2}} - \frac{z_b^q}{\sqrt{(z_a^q - z_b^q)^2}}, \quad (4)$$

$z\frac{a}{q}$ = max {zkq}; zbq = maxk=6 a {zkq} donde k = 1,2, 3..., K y K representan el número de clases en la base de datos. $z\frac{a}{q}$ y $z\,\frac{b}{q}$ son las dos salidas reales máximas de la red neuronal, correspondientes a una muestra q como se menciona en la ecuación (4).

Para seleccionar el valor de la variable μ, se aplica el siguiente proceso después de la iteración i: Obtener el nuevo MSE (MSE_i), si el $MSE_i < MSE_{(i-1)}$ se aplica la siguiente función, $\mu = \mu - \mu \cdot \epsilon$, donde $(0 < \epsilon < 1)$, y μ= 1 en la primera iteración (*i*=1); *i=1,2*, 3, I.

El enfoque selectivo de muestreo dinámico(I+SDSA) es detallado en el algoritmo 1.

Algoritmo 1 Enfoque selectivo de muestreo dinámico (I+SDSA) basado en el algoritmo estocástico Backpropagation.

Entrada: Datos de Entrenamiento **X**;

Salida: Pesos **W** y **U**;

INIT():

1: Leer archivo de configuración del MLP;
2: $i = 1, \mu = 1, \epsilon = 0.001$;
3: Generar pesos iniciales aleatorios entre −0.5 y 0.5;
LEARNING( ):
4: **while** (i < I) o (MSE_i > 0.0001) **do**
5: **for** q = 1 **to** Q **do**
6: $x^p \leftarrow$ elegir aleatoriamente una muestra **X;**
7: **FORWARD(**x^p**);**
8: $z_a^q = max_{(k=1,2,\ldots,K;)} \{z_k^q\}$;
9: $z_b^q = max_{(k=1,2,\ldots,K;\ k\neq a)} \{z_k^q\}$;
10: $\Delta^q = \frac{z_a^q}{\sqrt{(z_a^q - z_b^q)^2}} - \frac{z_b^q}{\sqrt{(z_a^q - z_b^q)^2}}$;
11: $\gamma(\Delta^p) = exp\ (-\ ||\Delta^q - \mu||^2/2\sigma^2)$;
12: **if** (**Random( )** <= $\gamma(\Delta^q)$) **then**
13: **UPDATE(**x^p**);**
14: **end if**

```
15:  end for
16: if (MSE_i < MSE_(i-1) and i > 1) then
17:     μ = μ − μ · ϵ;
18:  end if
19: i + +;
20: end while
```

Para la etapa experimental se usaron quince conjuntos de datos, que fueron obtenidas de cinco bases de datos de percepción remota procedentes del mundo real, (Cayo, Feltwell, Satimage, Segment y 92AV3C).

Los conjuntos de datos originales fueron alterados uniendo y/o reduciendo al azar el tamaño de algunas clases con el fin de obtener conjuntos de datos de múltiples clases desbalanceadas con varias clases de distribución.

El conjunto de datos 92AV3C utilizado en este trabajo es una versión reducida del conjunto de datos original con seis clases (2, 3, 4, 6, 7 y 8) y treinta y ocho atributos. La Tabla 1 muestra las principales características de este proceso.

Se aplicó el método de validación cruzada de diez veces a todos los conjuntos de datos empleados en el proceso experimental.

Para entrenar al MLP se utilizó el Backpropagation y cada proceso de entrenamiento se realizó diez veces, en otras palabras, los pesos fueron iniciados aleatoriamente diez veces.

Se eligió el algoritmo estocástico Backpropagation, ya que suele ser mucho más rápido y a menudo resulta en mejores soluciones que el Backpropagation por lotes, y se puede utilizar para el seguimiento de los cambios [33], que permite directamente aplicar el mecanismo de selección del método propuesto (ecuación (3)).

La razón de aprendizaje (η) se fijó en 0.1, y se estableció el criterio de detención a 500 iteraciones o si el valor MSE es inferior a 0.001. Se utilizó una sola capa oculta y para el número de neuronas en la capa oculta para cada conjunto de datos se estableció mediante prueba y error. El número de neuronas se fijaron en: 7 para MCAA, MCAB y MCAC; a 6 para MFEA, MFEB y MFEC; a 12 para MSAA, MSAB y MSAC; a 10 para MSEA, MSEB, MSEC, M92A, M92B y M92C. La variable *k* en SMOTE se estableció en cinco como en [11] ya que fue un trabajo que se tomó como referencia de comparación.

Para evaluar y comparar el rendimiento clasificador del método propuesto y los otros enfoques, se utilizó una versión para problemas de múltiples clases (ver ecuación (5)) del área bajo la curva, característica del receptor (MAUC) [34], es un método para la validación de clasificadores en escenario de desbalance de múltiples clases. El MAUC es define como:

$$MAUC = \frac{2}{\|J\|(\|J\|-1)} \sum_{j_i, j_k \epsilon J} AUC_R(j_i, j_k) \quad (5)$$

donde $AUC_R(j_i \; y \; j_k)$ es el AUC para cada par de clases $j_i \; y \; j_k$.

Tabla 1. Resumen de las principales características del conjunto de datos.

Base de Datos	#Ejemplos.	#Atributos.	# Ejemplos por clases						
			1	**2**	**3**	**4**	**5**	**6**	**7**
MCAA	6019	4	2941	293	2283	369	133	–	–
MCAB			3310	293	2283	133	–	–	–
MCAC			3074	293	2283	369	–	–	–
MFEA	8536	15	3531	2441	91	2295	178	–	–
MFEB			5972	178	91	2295	–	–	–
MFEC			5826	2441	91	178	–	–	–
MSAA	4697	36	1508	1533	104	1358	93	101	–
MSAB			3041	101	104	1358	93	–	–
MSAC			2866	1533	104	101	93	–	–
MSEA	1470	19	330	50	330	330	50	50	330
MSEB			660	50	330	330	50	50	–
MSEC			660	50	330	330	50	50	–
M92A	5063	38	190	117	1434	2468	747	106	–
M92B			190	117	1434	3215	106	–	–
M92C			190	117	3902	106	747	–	–

Por otro lado, con el fin de fortalecer los resultados del análisis, se aplicaron las pruebas estadísticas no paramétricas de Friedman e Iman - Davenport, para saber si existe diferencia estadística significativa en los resultados.

Finalmente, cuando existe alguna diferencia significativa entre los métodos individuales utilizados, se aplicó las pruebas post hoc de Holm [35] y Shaffer [36] con el fin de encontrar el par de métodos particulares que producen diferencias estadísticas significativas. En las referencias. [37,38] se presenta un estudio exhaustivo de estos métodos estadísticos no paramétricos. Se emplearon las pruebas de Friedman, Iman - Davenport, Holm y Shaffer con $\alpha = 0.05$ para el nivel de confianza, esto con ayuda del software KEEL [39].

4. Resultados

La experimentación se basó principalmente en la comparación con diferentes métodos. En primer lugar, I-SDSA se compara con métodos paralelos (SDSA y DyS). En segundo lugar, con tres enfoques de muestreo convencionales (SMOTE, ROS y RUS), que han demostrado su capacidad para tratar el problema de desbalance de clases

BD	I-SDSA-S	SDSA-R	SMOTE	SDSA-S	ROS	I-SDSA-R	DYS	DOS	RUS	STANDARD
MCAA	0.897	0.896	**0.900**	0.897	0.896	0.894	0.869	0.837	0.860	0.689
MCAB	0.923	0.924	0.926	**0.930**	0.925	0.923	0.908	0.910	0.891	0.687
MCAC	0.904	0.902	0.905	0.903	**0.907**	0.903	0.854	0.890	0.870	0.742
MFEA	**0.935**	0.927	0.929	0.927	0.928	0.930	0.927	0.917	0.902	0.780
MFEB	0.945	0.935	0.941	**0.949**	0.937	0.940	0.930	0.915	0.919	0.729
MFEC	**0.932**	0.928	0.929	**0.932**	0.912	0.913	0.922	0.928	0.910	0.750
MSAA	0.832	0.834	0.835	0.830	0.827	0.821	0.824	0.831	0.815	0.734
MSAB	0.840	0.824	0.841	**0.852**	0.813	0.816	0.823	0.829	0.826	0.686
MSAC	**0.864**	0.838	**0.864**	0.853	0.830	0.841	0.823	0.848	0.798	0.710
MSEA	0.945	**0.958**	0.943	0.946	0.951	0.945	0.949	0.945	0.923	0.922
MSEB	0.949	**0.951**	0.943	0.942	0.945	0.943	0.950	0.944	0.915	0.916
MSEC	0.941	0.943	0.935	0.930	0.938	0.941	0.937	0.935	0.912	0.910
M92A	0.842	0.866	0.837	0.814	0.859	0.857	0.861	0.796	0.773	0.734
M92B	0.849	0.860	0.846	0.845	0.868	0.863	0.857	0.789	0.803	0.715
M92C	0.889	0.911	0.885	0.898	0.906	0.906	0.924	0.887	0.887	0.819
MAUC	**0.899**	**0.900**	**0.897**	**0.897**	**0.896**	**0.896**	**0.891**	**0.880**	**0.867**	**0.768**
Ranking	**4.300**	**4.567**	**4.833**	**5.233**	**5.300**	**6.200**	**6.800**	**7.900**	**10.167**	**12.367**

Fig. 1. Rendimiento de la clasificación del Backpropagation usando MAUC. Los números en negrita representan los mejores valores.

[9, 11, 21], prácticamente son métodos que balancean las muestras antes de la clasificación.

La comparación se desarrolló a partir de tres enfoques: 1) rendimiento de clasificación, 2) muestras utilizadas en el entrenamiento, y 3) tiempo de entrenamiento. De la misma forma, se incluyeron pruebas estadísticas no paramétricas para informar cuando existe diferencia estadística significativa en los resultados.

La Figura 1 muestra los promedios MAUC y rangos Friedman, obtenidos en la etapa de clasificación de los métodos estudiados. Friedman clasifica el método establecido en el rango 1 al mejor algoritmo, 2 en el segundo mejor, 3 en el tercer mejor, sucesivamente para todos los casos; si existen semejanzas, se calcula el rango promedio [37, 38].

Por otra parte, en la Figura 1 se observa que todos los métodos estudiados mejoran el rendimiento de la clasificación (considerando como referencia los resultados estándar del Backpropagation (STANDARD) y la familia SDSA (ISDSA-R, I-SDSAS, SDSA-R, SDSA-S), los métodos de sobre-muestreo (ROS y SMOTE) producen prácticamente los mismos resultados. Por lo tanto, de acuerdo con los rangos de Friedman, I-SDSA-S presenta una tendencia a obtener mejores resultados que los otros métodos, pero SDSA-R en términos de promedio de MAUC es mejor que I-SDSA-S.

Sin embargo, SDSA-S y SDSA-R necesitan un modelo independiente de validación para funcionar correctamente, prueba diferentes valores de μ para obtener el mejor. Para la experimentación se utilizaron los siguientes valores de μ para SDSA-S y SDSA-R: 0.125, 0.25, 0.375, 0.5, 0.625, 0.75 y 0.875.

Para emplear una validación independiente, se involucraron dos problemas: 1) ¿Necesita costo computacional adicional, y 2) Cuantos valores μ diferentes necesitamos probar? En I-SDSA-S se obtiene automáticamente el valor apropiado de μ para cada conjunto de datos durante el entrenamiento de MLP. Para ello se utilizó el MSE del Backpropagation.

De la misma forma, podemos ver en la Figura 1 que I-SDSA presenta un comportamiento opuesto a SDSA, por ejemplo, I-SDSA-S es mejor que I-SDSA-R y SDSA-S es peor que SDSA-R. La explicación de esto es precisamente que SDSAR y

SDSA-S aplican para cada conjunto de datos el mismo valor en (μ = 0. 125), mientras que I-SDSA utiliza diferentes valores para cada conjunto de datos. En el mismo sentido en la Figura 1, se observa que, SMOTE es mejor que ROS. Por otro lado, SDSA es consistente con los resultados de la Referencia [40], es decir, para utilizar ROS como método de sobre-muestreo en SDSA es mejor que aplicar SMOTE.

DyS y DOS no mejoran los resultados de la familia SDSA, mucho menos ROS y SMOTE, sin embargo, sus resultados no son malos en sí mismos. ROS y SMOTE son efectivos para tratar el problema de desbalance de clases, sin embargo, SMOTE y ROS necesitan más muestras (Fig. 1) y tiempo (Fig. 2) en la etapa de entrenamiento que los otros métodos estudiados. La principal ventaja de estos métodos de sobre-muestreo es que son los más simples. La tendencia de RUS es obtener mejores resultados que el STANDARD, sin embargo, su desempeño es peor que los otros métodos. El bajo rendimiento de clasificación de RUS podría explicarse por el número de muestras borradas en el conjunto de datos de formación para este método, debido a la perdida de información pertinente (Fig.1).

Por otro lado, la Fig. 1 muestra que la familia DyS y SDSA hacen un mejor uso de las muestras de entrenamiento, es decir, no requieren todas las muestras de entrenamiento. La familia SDSA utiliza menos muestras que STANDARD, pero la familia SDSA gasto aproximadamente 50% más de tiempo en la etapa de entrenamiento que la STANDARD (ver Fig. 2). Esto se debe a que la familia SDSA no elimina ninguna muestra durante el entrenamiento. Por el contrario, DyS utiliza considerablemente menos muestras (Fig. 1) y paso mucho menos tiempo en la etapa de entrenamiento (al rededor del 50% de las muestras de entrenamiento y el tiempo con respecto al STANDARD, ver Figura. 2).

Sin embargo, de acuerdo con la Tabla 2, DyS no muestra una tendencia a superar el rendimiento de clasificación de los métodos de la familia ROS, SMOTE y SDSA. Por su parte, el DOS emplea menos muestras y gasto menos tiempo que ROS y SMOTE, pero necesita más muestras y tiempo que DyS y SDSA

(Fig. 1 y 2). Con respecto al rendimiento de clasificación, DOS es solo mejor que RUS y STANDARD. Su principal inconveniente es que utiliza el número de iteraciones como parámetro para actualizar la ratio de desbalance, por lo que el rendimiento del clasificador se vincula a este número. En [41] los autores utilizan 5000 ´épocas en el entrenamiento, en la experimentación del desarrollo del trabajo se aplican solo 500, por esta razón DOS usa más muestras en la etapa de entrenamiento que en la obra original.

Con el fin de reforzar el análisis de resultados se aplicó un análisis estadístico no paramétrico y considerando el rendimiento de reducción, se distribuye de acuerdo con la chi-cuadrado con 13 grados de libertad, la estadística de Friedman se establece en 100.638, y valor de *p* calculado por la prueba de Friedman es de 7.432E-11.

Considerando el rendimiento de reducción distribuido según la distribución *F* con 13 y 143 grados de libertad, la estadística de Iman y Davenport nos proporciona un valor de 19.996 y el valor *p* calculado por la prueba de Iman-Davenport es de -2.220E-16. Por lo tanto, la hipótesis nula es rechazada, es decir, las pruebas de Friedman e Iman-Davenport indican que existen diferencias significativas en los resultados.

Por otra parte, se utilizaron los procedimientos estadísticos de Holm y Shaffer para realizar el análisis estadístico no paramétrico post-hoc. Este análisis se realiza para averiguar qué algoritmos son diferentes entre todas las comparaciones $C \times C$ de los clasificadores *C*.

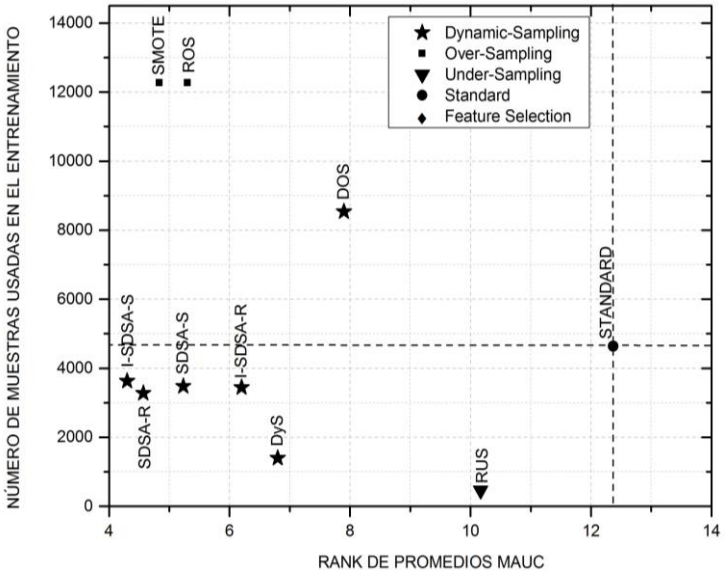


Fig. 1. Comparación de rangos medios MAUC versus muestras utilizadas en el entrenamiento. El tamaño se considera con referencias al tamaño de la base de datos original.

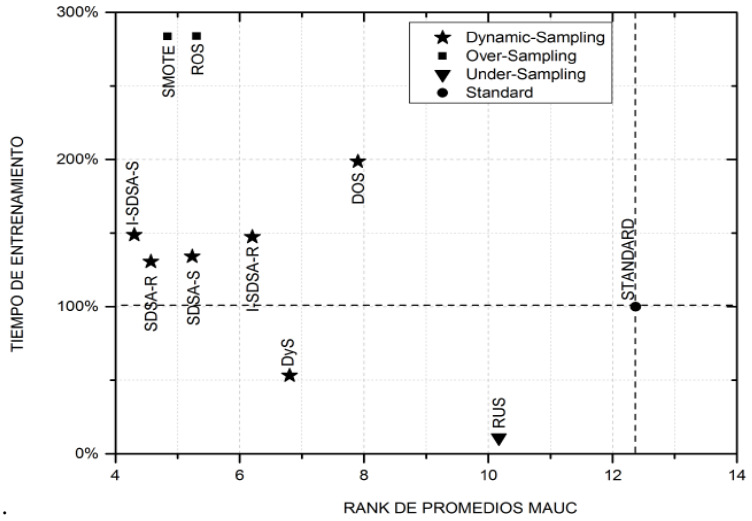


Fig. 2. Comparación de rangos medios de MAUC versus tiempo de entrenamiento. El tiempo (100%) corresponde al tiempo usado en la etapa de entrenamiento del Backpropagation estándar con el conjunto de datos original.

La Tabla 3. Presenta los valores de p no ajustados y los valores de p ajustados (α) para los procedimientos estadísticos de Holm (p-Holm) y Shaffer (p-Shaffer) considerando un α= 0.05.

Tabla 3. Valores de *P* ajustados y no ajustados para comparaciones de CXC, sobre 15 bases de datos, teniendo en cuenta un valor de efectividad de α= 0.05.

	STANDART	SMOTE	SDSA-S	SDSA-R	RUS	ROS	I-SDSA-S	I-SDSA-R	DyS	DOS
STANDARD		**0.000002**	**0.000009**	**0.000006**	0.11842	**0.000018**	**0.000001**	**0.000116**	0.00139	0.01373
		0.000602	**0.000685**	**0.000667**	0.00122	**0.000704**	**0.000581**	**0.000735**	0.00079	0.00091
		0.000641	**0.000746**	**0.000746**	0.00122	**0.000746**	**0.000641**	**0.000746**	0.00079	0.00091
SMOTE	0.73268		0.78841	0.00128	0.62559	0.94165	0.35387	0.11278	0.02046	
	0.00278		0.00333	0.00078	0.00208	0.01250	0.00152	0.00119	0.00094	
	0.00278		0.00333	0.00078	0.00208	0.01250	0.00152	0.00119	0.00094	
SDSA-S	0.94165	0.00399	0.88362	0.67832	0.55819	0.21340	0.04813			
	0.01667	0.00085	0.00625	0.00227	0.00192	0.00139	0.00104			
	0.16667	0.00085	0.00625	0.00227	0.00192	0.00139	0.00104			
SDSA-R	0.00316	0.82620	0.73268	0.51007	0.18768	0.04042				
	0.00082	0.00357	0.00263	0.00179	0.00132	0.00100				
	0.00082	0.00357	0.00263	0.00179	0.00132	0.00100				
RUS	0.00629	0.00099	0.02183	0.10213	0.36668					
	0.00088	0.00077	0.00096	0.00116	0.00156					
	0.00088	0.00077	0.00096	0.00116	0.00156					
ROS	0.57470	0.66055	0.27225	0.67278						
	0.00200	0.00217	0.00143	0.00109						
	0.00200	0.00217	0.00143	0.00109						
I-SDSA-S	0.31717	0.09711	0.01681							
	0.00147	0.00114	0.00093							
	0.00147	0.00114	0.00093							
I-SDSA-R	0.51007	0.16433								
	0.00172	0.00128								
	0.00172	0.00128								
DyS	0.46421									
	0.00167									
	0.00167									
DOS										

Las filas y columnas representan los métodos estudiados. Los valores *p* no ajustados y los métodos de *p*-Holm y *p*-Shaffer. Para cada método se muestran tres valores de *p*, el primero es el valor *p* no ajustado, el segundo es de *p*-Holm, y el ultimo es el valor de *p*-Shaffer.

El procedimiento de Holm rechaza aquellas hipótesis que tienen un valor *p* no ajustado ≤ *p*-Holm, y el procedimiento de Shaffer rechaza aquellas hipótesis que tienen un valor *p* no ajustado ≤ p-Shaffer. La hipótesis nula rechazada se escribe en negrita. En las Tablas 2 y 3 se observa que los métodos de ROS, SMOTE, DyS y SDSA clasifican mejor (con estadística significativa) que el ESTANDAR. ´ Con estadística significativa en sus resultados I-SDSA-S, SDSA-R y SMOTE, mejoran el rendimiento de clasificación que de RUS y ROS.

DOS y DyS presentan un rendimiento inferior al de ROS, SMOTE y SDSA, pero de acuerdo con la prueba post-hoc de Holm-Shaffer, sus resultados de clasificación no son estadísticamente diferentes.

ROS, SMOTE y SDSA son los mejores métodos estudiados en este trabajo, no obstante, sus resultados no muestran diferencia con significancia estadística entre ellos.

5. Conclusiones y trabajos futuros

Los resultados mostrados demuestran que I-SDSA es muy competitivo en el desempeño de la clasificación con respecto a los métodos de sobre-muestreo y sub-muestreo (ROS, SMOTE y RUS), y con enfoques similares como los métodos de muestreo dinámico (DyS) o SDSA.

I-SDSA es mejor en términos de muestras de entrenamiento, tiempo de entrenamiento y desempeño de clasificación.

DyS y RUS necesitan menos muestras, incluso menos tiempo que I-SDSA, pero la tendencia es que el desempeño de clasificación del método propuesto debe ser mejor. I-SDSA usa menos muestras que el Backpropagation estándar, pero requiere un 50% más de tiempo de entrenamiento, en este sentido I-SDSA es un enfoque exitoso para abordar el problema de desbalance de varias clases, porque hace un mejor uso de las muestras de entrenamiento que permite mejorar el desempeño de la clasificación. En conclusión, el algoritmo presentado en este trabajo (I-SDSA) es una buena estrategia para tratar el desbalance entre varias clases, ya que hace un mejor uso de las muestras de entrenamiento que permitiendo mejorar el desempeño no de la clasificación.

Como trabajos futuros se pretende usar aprendizaje profundo (Deep Lear Ning) para trabajar con bases de datos de gran tamaño (bit data) y mejorar los tiempos de entrenamiento.

Referencias

1. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. In: Emerging Artificial Intelligence Applications in Computer Engineering, pp. 3–24 (2007)
2. Batista, G., Prati, R., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. (SIGKDD) Explor. Newsl., 6, pp. 20–29 (2004)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W. P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res., 16, pp. 321–357 (2002)
4. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis, 6, pp. 429–449 (2002)
5. Batista, G., Silva, D., Prati, R: An experimental design to evaluate class imbalance treatment methods. In: International Conference on Machine Learning and Applications (ICMLA), 2, pp. 95–101 (2012)
6. Lin, M., Tang, K., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Netw. Learning Syst., 24(4), pp. 647–660 (2013)
7. Chawla, N., Cieslak, D., Hall, L., Ajay, J.: Automatically countering imbalance and its empirical relationship to cost. Data Min. Knowl. Discov., 17, pp. 225– 252 (2008)
8. Wang, J., Jean, J.S.N.: Resolving multifont character confusion with neural networks. Pattern Recognition, 26(1), pp. 175–187 (1993)
9. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling techniques. IEEE Transactions on Knowledge and Data Engineering, pp. 1041–4347 (2015)

10. Batista, G., Prati, R., Monard, M.: Balancing strategies and class overlapping. IDA, pp. 24–35 (2005)
11. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, vol. 16, pp. 321–357 (2002)
12. Prati, R.C., Batista, G., Monard, M.C.: Data mining with imbalanced class distributions: concepts and methods. In: Proceedings of the 4th Indian International Conference on Artificial Intelligence, (IICAI), pp. 359–376 (2009)
13. Han, H., Wang, W., Mao, B.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. (ICIC), pp. 878–887 (2005)
14. He, H., Bai, Y., Garcia, E., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. (IJCNN), pp. 1322–1328 (2008)
15. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. (SIGKDD), Explor. Newsl., 6, pp. 20–29 (2004)
16. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD09), 5476 of Lecture Notes on Computer Science, pp. 475–482, Springer-Verlag (2009)
17. Tomek, I.: Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, 7(2), pp. 679–772 (1976)
18. Hart, P.: The condensed nearest neighbour rule. IEEE Transactions on Information Theory, 14(5), pp. 515–516 (1968)
19. Show-Jane, L., Yue-Shi, Y.: Cluster-based under-sampling approaches for imbalanced data distributions. Expert Systems with Applications, 36, pp. 5718–5727 (2009)
20. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284 (2009)
21. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf. Sci., 250, pp. 113–141 (2013)
22. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. Computational Intelligence, 26(3), pp. 232–257 (2010)
23. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18, pp. 63–77 (2006)
24. Sun, T. Jiao, L., Feng, J., Liu, F., Zhang, X.: Imbalanced hyperspectral image classification based on maximum margin. IEEE Geosci. Remote Sensing Lett., 12(3), pp. 522–526 (2015)
25. Mirza, B., Lin, Z., Liu, N.: Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing, 149, pp. 316–329 (2015)
26. Galar, M., Fernández, A., Tartas, E.B., Sola, H.B., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybridbased approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 42(4), pp. 463–484 (2012)
27. Fernández-Navarro, F., Hervás-Martínez, C., Gutiérrez, P.A.: A dynamic oversampling procedure based on sensitivity for multi-class problems. Pattern Recogn., 44, pp. 1821–1833 (2011)
28. Fernández-Navarro, F., Hervás-Martínez, C., García-Alonso, C., Torres-Jiménez, M.: Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity. Expert Syst. Appl., 38(10), pp. 12483–12490 (2011)
29. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine, (AIME), pp. 63–66, Springer Verlag (2001)

30. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250, pp. 113–141 (2013)
31. Stefanowski, J.: Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In: Emerging Paradigms in Machine Learning, S. Ramanna, L. C. Jain, and R. J. Howlett (eds.), 13 of Smart Innovation, Systems and Technologies, pp. 277–306, Springer Berlin Heidelberg (2013)
32. Alejo, R., Valdovinos, R., García, V., Pacheco-Sanchez, J.H.: A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. Pattern Recognition Letters, 34(4), pp. 380–388 (2013)
33. Lecun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient BackProp. In: Neural Networks—Tricks of the Trade, G. Orr and K. Müller (eds.), 1524 of Lecture Notes in Computer Science, pp. 5–50, Springer Verlag (1998)
34. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett., 27, pp. 861–874 (2006)
35. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2), pp. 65–70 (1979)
36. Shaffer, J.: Modified sequentially rejective multiple test procedures. Journal of the American Statistical Association, 81(375), pp. 826–831 (1986)
37. García, S., Herrera, F.: An Extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all Pairwise Comparisons. Journal of Machine Learning Research, 9, pp. 2677–2694 (2008)
38. Luengo, J., García, S., Herrera, F.: A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests. Expert Systems with Applications, 36(4), pp. 7798–7808 (2009)
39. Alcalá-Fernández, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing, 17(2–3), pp. 255–287 (2011)
40. Alejo, R., Monroy-de Jesús, J., Pacheco-Sánchez, J., López-González, H., Antonio-Velázquez, J. A.: A selective dynamic sampling back-propagation approach for handling the two-class imbalance problem. Applied Sciences, 6(7), pp. 200 (2016)
41. Alejo, R., García, V., Pacheco-Sánchez, J. H.: An efficient over-sampling approach based on mean square error back-propagation for dealing with the multiclass imbalance problem. Neural Processing Letters, pp. 1–16 (2015)

ISSN 1870-4069

Estudio del impacto de un curso de nivelación en el desempeño de alumnos de ingeniería utilizando Minería de Datos Educacional

Beatriz A. González-Beltrán, Silvia B. González-Brambila, Lourdes Sánchez-Guerrero, Irma Ardón-Pulido, Josué Figueroa-González

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
{bgonzalez,sgb,lsg,ifap,jfgo}@azc.uam.mx

Resumen. En un entorno educativo se genera gran cantidad de información que analizada adecuadamente puede ser de utilidad en la toma de decisiones. La Minería de Datos Educacional consiste en utilizar técnicas de Minería de Datos para analizar información académica, de tal manera que se pueda obtener conocimiento de diversos aspectos educativos, siendo uno de los más estudiados, el desempeño escolar. En la Universidad Autónoma Metropolitana, se crearon cursos de nivelación para que los alumnos de recién ingreso a un programa de ingeniería, puedan tener un nivel adecuado de matemáticas básicas y de física. Estos cursos pueden acreditarse a través de un examen diagnóstico o cursándolos de manera presencial. Actualmente no se ha analizado si la existencia de estos cursos, ya sea que se cursen o acrediten a través de un examen, tiene un efecto positivo en el desempeño de los alumnos en cursos posteriores. Este trabajo presenta un análisis, utilizando Reglas de Asociación, del impacto que tiene cursar o aprobar mediante un examen el curso de nivelación de Matemáticas básicas en el desempeño de los alumnos en dos cursos posteriores; de tal manera que se pueda decidir si es adecuado que este curso pueda acreditarse mediante un examen, deba cursarse o incluso, pudiera desaparecer de los planes de estudio.

Palabras clave: Análisis de planes de estudio, cursos de nivelación, desempeño escolar, minería de datos educacional, reglas de asociación.

Study of the Impact of a Leveling Course on the Performance of Engineering Students Using Educational Data Mining

Abstract. In an educational environment, a large amount of information is generated, when properly analyzed, can be useful in decision-making. Educational Data Mining uses Data Mining techniques for analyzing academic information, in such a way that knowledge of different educational aspects can be obtained, being one of the most studied, the academic performance. In the Universidad Autónoma Metropolitana, leveling courses were created so that students who have recently entered

to an engineering program can obtain an adequate level of mathematics and physics. These courses can be approved through a diagnostic exam or attending the course. Actually, it has not been analyzed if the existence of these courses, whether they are taken or approved through an exam, has a positive effect on the student's performance in subsequent courses. This paper presents an analysis, using Rules of Association, of the impact of attending a Basic Mathematics leveling course or passing it through an exam, over the performance of students in two subsequent courses; in such a way that it could be possible to decide if it is appropriate that this course can be accredited through an exam, it must be taken or even, disappear from the curricula.

Keywords: Analysis of curricula, leveling courses, school performance, educational data mining, association rules.

1. Introducción

Con el rápido crecimiento de la tecnología, la cantidad de datos que se producen ha permitido obtener información valiosa a través de su análisis. El proceso de obtener conocimiento a través de los datos es el objetivo de la Minería de Datos, la cual se aplica a muchos ámbitos de la sociedad, economía, medicina, etc. Cuando se analizan datos en un entorno educativo con técnicas de Minería de Datos, se habla de una rama denominada Minería de Datos Educacional [7,8], la cual ha tenido un gran crecimiento en los últimos años y ha permitido analizar diversos factores relacionados con la educación [12], siendo uno de los más estudiados el desempeño escolar de los estudiantes. Desde 2008 en la Universidad Autónoma Metropolitana (UAM), Unidad Azcapotzalco, los alumnos de nuevo ingreso de los diez programas de ingeniería que se imparten, presentan un examen diagnóstico que mide conocimientos básicos de matemáticas y de física; los que no aprueban, deben tomar los cursos de "Taller de Matemáticas" e "Introducción a la Física" [4] pertenecientes al Tronco de Nivelación Académica. El objetivo de estos cursos es que el alumno adquiera los conocimientos y habilidades necesarias para futuros cursos más avanzados; sin embargo, no se analizado si estos cursos de nivelación están ayudando a los estudiantes en los siguientes cursos. Es importante resaltar que el hecho de cursar estas dos materias implica un trimestre más en el tiempo de finalización de los estudios de los alumnos.

En este trabajo se analiza si el curso "Taller de Matemáticas" realmente está ayudando a que los alumnos tengan un mejor desempeño en dos cursos posteriores de matemáticas, "Complementos de Matemáticas" e "Introducción al Cálculo" [5]. Con el fin de identificar y analizar la relación entre el desempeño en "Taller de Matemáticas" con el desempeño en otros cursos, se utiliza la técnica de Reglas de Asociación analizando factores como la calificación obtenida en "Taller de Matemáticas", la cantidad de veces que se cursó antes de aprobarlo y el tiempo transcurrido desde su aprobación hasta la inscripción de los siguientes cursos. El objetivo es determinar si la existencia de este curso y la forma en que

se puede aprobar está ayudando a los alumnos a estar mejor preparados para cursos más avanzados.

El contenido del documento es el siguiente: en la sección de Trabajos relacionados se presenta un análisis de los trabajos con un enfoque similar, en la sección de Metodología se presentan los pasos realizados para la obtención y análisis de las Reglas de Asociación y en la sección de Resultados y Análisis, se muestran las reglas obtenidas y su interpretación.

2. Trabajos relacionados

Existen diversos trabajos que a través de la Minería de Datos Educacional han analizado ciertas problemáticas en la educación superior. El trabajo de [3] propone descubrir Reglas de Asociación utilizando los algoritmos SLP-Growth (*Significant Least Pattern Growth* y la medida CRS (*Critical Relative Support*). En [1,2], a partir de los datos de los alumnos de computación para el ingreso 2008/2009 de la University Malaysia Terengganu, se analiza un conjunto de datos que contiene los registros de los programas preferidos que son seleccionados por los estudiantes aceptados en dicha universidad. Los resultados muestran que se puede extraer un menor número de reglas de asociación interesantes comparado con las medidas tradicionales (análisis de correlación). Además, sugieren que sus resultados sean analizados por las autoridades universitarias para ofrecer programas más apropiados a los estudiantes prospectos en lugar de que sea de manera aleatoria.

En [11] se propone el uso del valor añadido (o equivalente a la medida *lift*) y del coseno como medidas adecuadas para datos educacionales. Esta propuesta se aplica en un curso titulado "Conceptos Básicos y Formales de Ciencias de la Computación" en la plataforma Moodle de la University of Applied Sciences TFH-Berlin durante el semestre de invierno 2007-2008. El curso de Moodle ofrece recursos adicionales a los ofrecidos en la clase presencial y los profesores están interesados en saber si el uso de estos recursos tienen un impacto positivo en sus calificaciones. Este trabajo propone el uso del coseno en primer lugar, y utilizar *lift* si la regla de asociación es considerada como no interesante con la medida del coseno. Si los resultados son contradictorios, entonces los profesores deben decidir si se toma o no en cuenta la regla de asociación.

El uso de árboles de decisión como técnica de clasificación, se propone en [6], esto aplicado sobre los datos de los alumnos del Departamento de Aplicaciones Computacionales en los años 2007-2010 de la VBS Purvanchal University, Jaunpur (Uttar Pradesh). El objetivo de este trabajo es predecir el desempeño de los alumnos al final del semestre. Los datos utilizados fueron: asistencia, pruebas en los cursos, seminarios y calificaciones de tareas. Los resultados podrían utilizarse para identificar a los estudiantes que necesitan atención especial para reducir la tasa de fracaso y tomar las acciones apropiadas para las siguientes pruebas semestrales.

En [9] se aplican algoritmos de clasificación basados en Reglas de Asociación y Árboles de Decisión sobre los datos de los alumnos de nuevo ingreso del

Programa II de la Unidad Académica Preparatoria de la Universidad Autónoma de Zacatecas del año académico 2009-2010 para predecir el estado académico de los estudiantes al final del primer semestre. Los resultados podrían utilizarse en la detección de alumnos con riesgo, cuya medida pudiera ser la asignación de un profesor-tutor para evitar la deserción del alumno.

En [10], se utilizan Reglas de Asociación basadas en el método Apriori para descubrir las conexiones entre las actividades de los estudiantes y sus calificaciones finales. La propuesta fue aplicada sobre la bitácora de datos de los alumnos del curso de "Programación 2" extraídos del sistema Moodle de la Universidad de Rijeka. Los resultados muestran que existe una influencia entre las actividades y el éxito del curso y que la creación de video-clases está justificada. La Tabla 1 presenta una comparativa de los trabajos analizados.

Tabla 1. Tabla comparativa de los trabajos relacionados.

Trabajo	Técnica de clasificación	Aplicación
[1,2]	Reglas de asociación	Programas preferidos por los alumnos
[11]	Reglas de asociación	Relación del uso de los recursos en Moodle y la calificación obtenida en el curso
[6]	Árboles de decisión	Relación de notas obtenidas en los instrumentos de evaluación y nota final
[9]	Reglas de asociación y Árboles de decisión	Relación de datos de ingreso y nota al final del primer semestre
[10]	Reglas de asociación	Influencia entre las actividades y el éxito del curso

Como se presenta en los trabajos relacionados, el aplicar la Minería de Datos Educacional a información en un entorno educativo ha permitido encontrar aspectos que impactan en el desempeño escolar de los alumnos. La validez del curso de nivelación "Taller de Matemáticas" ha sido debatida en la UAM, por lo que es necesario realizar un estudio del impacto que pudiera tener. En este trabajo utilizamos la técnica de Reglas de Asociación para encontrar relaciones entre el desempeño en "Taller de Matemáticas" con dos cursos posteriores, estos primeros resultados serán validados en futuros trabajos con otro tipo de técnicas, por ejemplo, predictivas, las cuáles también han mostrado obtener resultados importantes en este tipo de análisis.

3. Metodología

Tomando como base la metodología CRISP-DM [13], se aplicaron los siguientes pasos para analizar la relación entre el curso de nivelación "Taller de Matemáticas" con los cursos "Complementos de Matemáticas" e "Introducción al Cálculo".

3.1. Planteamiento de los objetivos

En esta fase se definen los objetivos del estudio. El curso de nivelación "Taller de Matemáticas" tiene como objetivo académico que los alumnos que lo cursan adquieran un mejor nivel en conceptos de matemáticas para cursos futuros. Con esta idea, se puede suponer que:

- Un alumno que no cursa "Taller de Matemáticas" por haber aprobado el examen diagnóstico, tendrá un buen desempeño en los cursos de "Complementos de Matemáticas" e "Introducción al Cálculo".
- Un alumno que cursa y tiene un buen desempeño en "Taller de Matemáticas", también lo tendrá en los cursos de "Complementos de Matemáticas" e "Introducción al Cálculo".

Se busca obtener relaciones que permitan corroborar o refutar estas suposiciones.

3.2. Selección de los datos

Para este estudio se utilizó la siguiente información relacionada con los alumnos y su desempeño escolar:

- La modalidad en que se aprobó "Taller de Matemáticas", ya sea por examen diagnóstico o por inscripción.
- La calificación obtenida en "Taller de Matemáticas", ya sea que por inscripción o por examen diagnóstico. En caso de que lo haya inscrito, también se considera el número de oportunidades que utilizó el alumno para aprobar dicho curso.
- La calificación que obtuvo el alumno en su primera oportunidad al cursar "Complementos de Matemáticas".
- La calificación que obtuvo el alumno en su primera oportunidad al cursar "Introducción al Cálculo".

Los datos se obtuvieron de dos fuentes: el Archivo General de Alumnos (AGA) y el historial académico de los alumnos (llamado kardex en la UAM). Dado que los cursos "Taller de Matemáticas", "Complementos de Matemáticas" e "Introducción al Cálculo" deben aprobarlos todos los estudiantes de ingeniería, se analizó el historial de los alumnos de las diez ingenierías que se imparten en la UAM. Se obtuvieron datos a partir del año 2008, fecha en la que surgió el curso de nivelación "Taller de Matemáticas". De esta forma se analizaron 5,181 alumnos con las características mostradas en la Tabla 2.

Como primera aproximación del impacto de "Taller de Matemáticas" en "Complementos de Matemáticas" e "Introducción al Cálculo", se obtuvo una estadística que muestra la cantidad de alumnos que obtuvieron una de las cuatro posibles calificaciones que se manejan en la UAM: MB (Muy bien), B (Bien), S (Suficiente), NA (No Acreditada), según su desempeño en "Taller de Matemáticas". La Tabla 3 muestra los resultados para "Complementos de Matemáticas y la Tabla 4 para "Introducción al Cálculo".

Tabla 2. Desempeño de los alumnos en "Taller de Matemáticas".

Calificación en "Taller de Matemáticas"	No. alumnos que aprobaron en examen diagnóstico	No. alumnos que cursaron y aprobaron		
		en primera oportunidad	en segunda oportunidad	en tercera o mas oportunidades
Muy Bien	277	771	113	9
Bien	463	1384	210	30
Suficiente	0	1439	394	97

Tabla 3. Desempeño de los alumnos en "Complementos de Matemáticas" de acuerdo a la calificación en "Taller de Matemáticas".

Calificación en "Taller de Matemáticas"	Calificación en "Complementos de Matemáticas"			
	MB	B	S	N
MB en examen diagnóstico	108	73	35	51
B en examen diagnóstico	109	150	85	112
MB en primera oportunidad	197	211	157	194
B en primera oportunidad	180	321	306	514
S en primera oportunidad	133	259	308	739
MB en segunda oportunidad	15	32	28	36
B en segunda oportunidad	27	43	39	99
S en segunda oportunidad	27	72	90	205
MB en tercera oportunidad	0	0	4	5
B en tercera oportunidad	2	5	7	15
S en tercera oportunidad	6	25	13	53

Tabla 4. Desempeño de los alumnos en "Introducción al Cálculo" de acuerdo a la calificación en "Taller de Matemáticas".

Calificación en "Taller de Matemáticas"	Calificación en "Introducción al Cálculo"			
	MB	B	S	N
MB en examen diagnóstico	86	52	21	53
B en examen diagnóstico	69	103	78	142
MB en primera oportunidad	139	150	131	210
B en primera oportunidad	105	185	232	548
S en primera oportunidad	77	152	181	771
MB en segunda oportunidad	13	23	15	46
B en segunda oportunidad	12	34	24	106
S en segunda oportunidad	28	50	59	205
MB en tercera oportunidad	3	1	1	4
B en tercera oportunidado	0	2	4	20
S en tercera oportunidad	6	12	17	58

3.3. Preparación de los datos

Se generaron el conjunto de antecedentes y consecuentes necesarios para analizar la información de las Tablas 3 y 4 con la técnica de Reglas de Asociación. Tomando en cuenta el trimestre de ingreso del alumno, si éste, en ese mismo trimestre había cursado y aprobado "Taller de Matemáticas" y cursado, mas no necesariamente aprobado, las materias de "Complementos de Matemáticas" e "Introducción al Cálculo", significaba que había aprobado a través del examen diagnóstico la materia de "Taller de Matemáticas". Se generaron un conjunto de antecedentes y consecuentes para la relación entre "Taller de Matemáticas" y "Complementos de Matemáticas" y otro conjunto para "Taller de Matemáticas" e "Introducción al Cálculo".

Los antecedentes que se analizaron en ambos cursos fueron los siguientes:
Antecedente 1. Calificación del alumno en "Taller de Matemáticas" de acuerdo a la escala de calificaciones: MB, B y S.
Antecedente 2. La modalidad en que el alumno aprobó "Taller de Matemáticas":

- APROBO. Acreditó el curso a través del examen diagnóstico
- PRIMER. Aprobó el curso en su primera oportunidad
- SEGUNDO. Aprobó el curso en su segunda oportunidad
- MAS_DE_DOS. Necesitó de más de dos oportunidades para aprobar (el número máximo de oportunidades es de cinco)

Antecedente 3. El número de trimestres que el alumno dejó pasar antes de inscribir "Complementos de Matemáticas" o "Introducción al Cálculo":

- EXAMEN. Aprobó el examen diagnóstico y en el trimestre de ingreso cursó "Complementos de Matemáticas" e "Introducción al Cálculo"
- SIGUIENTE. Al siguiente trimestre que aprobó "Taller de Matemáticas", cursó "Complementos de Matemáticas" o "Introducción al Cálculo"
- UNO. Dejó pasar un trimestre antes de inscribir "Complementos de Matemáticas" o "Introducción al Cálculo"
- DOS. Dejó pasar dos trimestres antes de inscribir "Complementos de Matemáticas" o "Introducción al Cálculo"
- MAS_DE_DOS. Dejó pasar tres o más trimestres antes de inscribir "Complementos de Matemáticas" o "Introducción al Cálculo"

El consecuente estaba relacionado con la calificación obtenida en "Complementos de Matemáticas" o "Introducción al Cálculo" la primera vez que se cursaron, pudiendo tenerse cualquiera de las siguientes calificaciones: MB, B, S o NA.

Un ejemplo del formato de los datos procesados que representaría que: El alumno aprobó en examen diagnóstico "Taller de Matemáticas" con MB y obtuvo B en "Complementos de Matemáticas" es:

MB,APROBO,EXAMEN,B

De manera similar, se tiene: El alumno cursó "Taller de Matemáticas", lo aprobó al primer intento con B, dejó pasar un trimestre antes de inscribir "Introducción al Cálculo" y cuando la aprobó, obtuvo MB.

B,PRIMER,UNO,MB

En total se obtuvieron 5,095 conjuntos de antecedentes y consecuentes para la relación entre "Taller de Matemáticas" y "Complementos de Matemáticas" y 4231 para la relación "Taller de Matemáticas" e "Introducción al Cálculo". La diferencia se debe a que no todos los alumnos habían cursado alguna de las materias.

3.4. Modelado

Una vez obtenidos los conjuntos de antecedentes y consecuentes, se obtuvieron las reglas de asociación utilizando el algoritmo Apriori. Los resultados se presentan en la sección de Resultados y análisis.

3.5. Evaluación

Una vez obtenidas las reglas de asociación, se seleccionaron las más representativas. Existen varios criterios para determinar la validez e importancia de las reglas, siendo la más conocida el **soporte**, que representa el porcentaje de elementos X en un conjunto de transacciones D y se define en (1):

$$soporte(X) = \frac{|X|}{|D|}. \tag{1}$$

Otro criterio es la **confianza**, que dada una asociación X ⇒ Y, indica la cantidad de reglas que tienen a X como antecedente y a Y como consecuente. Este valor se define en (2):

$$confianza(X \Rightarrow Y) = \frac{soporte(X \cup Y)}{soporte(X)}. \tag{2}$$

El criterio más utilizado es ***lift***, que representa la frecuencia de un conjunto de elementos X en una determinada transacción que llevan a una consecuencia Y y está representado en (3):

$$lift(X \Rightarrow Y) = \frac{soporte(X \Rightarrow Y)}{soporte(X) * soporte(Y)}. \tag{3}$$

Cuando el valor de *lift* es igual a 1, significa que la relación entre antecedentes y consecuentes, puede deberse a factores aleatorios. Si el valor es menor a 1 significa que no hay tanta relación entre antecedentes y consecuentes y se dice que X y Y son sustitutos. Si el valor de *lift* es mayor a 1, se dice que hay una mayor relación entre X y Y y se denominan complementos.

Para la evaluación de las reglas, se consideraron aquellas reglas que tuvieran un valor de *lift* mayor o igual a 1.1 y un soporte mayor a 0.01. A partir de eso, se analizaron las reglas considerando un umbral de confianza de 0.3.

4. Resultados y análisis

Para facilitar el análisis, se asignaron los siguientes acrónimos a los antecedentes y consecuentes:

- CTA.Calificación en "Taller de Matemáticas"
- CUR_APR. Forma en que el alumno aprobó "Taller de Matemáticas" en examen
- TIEM. Tiempo antes de inscribir "Complementos de Matemáticas" o "Introducción al Cálculo"

Mientras que para los consecuentes:

- CCOM. Calificación en "Complementos de Matemáticas"
- CCAL.Calificación en "Introducción al Cálculo"

Tras el procesamiento, se obtuvieron un conjunto de 14 reglas para la relación entre "Taller de Matemáticas" y "Complementos de Matemáticas", se realizó un filtrado para eliminar la redundancia obteniéndose finalmente 6, las cuales se muestran en la Tabla 5.

Tabla 5. Reglas de asociación para la relación entre "Taller de Matemáticas" y "Complementos de Matemáticas".

No	Regla	Soporte	Confianza	Lift
1	{CTA=MB,CUR_APR=APROBO,TIEM=EXAMEN} ⇒ {CCOM=MB}	0.021	0.40	2.55
2	{CTA=B,CUR_APR=APROBO,TIEM=EXAMEN} ⇒ {CCOM=B}	0.029	0.32	1.40
3	{CTA=S,CUR_APR=PRIMER,TIEM=SIGUIENTE} ⇒ {CCOM=NA}	0.120	0.52	1.33
4	{CTA=S,CUR_APR=SEGUNDO,TIEM=SIGUIENTE} ⇒ {CCOM=NA}	0.021	0.51	1.29
5	{CTA=S,CUR_APR=SEGUNDO,TIEM=UNO} ⇒ {CCOM=NA}	0.011	0.47	1.20
6	{CTA=B,CUR_APR=SEGUNDO,TIEM=SIGUIENTE} ⇒ {CCOM=NA}	0.010	0.44	1.12

Para facilitar su interpretación, generamos una representación visual la cual se muestra en la Figura 1. Aquí, el tamaño del círuclo representa el valor del *lift*, a mayor *lift*, mayor el tamaño del círculo, mientras que el tono del círculo representa la confianza, entre más obscuro, mayor confianza.

De acuerdo a la primera y segunda regla de la Tabla 5 y a la Figura 1, se puede observar que el aprobar el examen diagnóstico con MB o B sin tener que cursar "Taller de Matemáticas", está relacionado con obtener MB o B en "Complementos de Matemáticas".

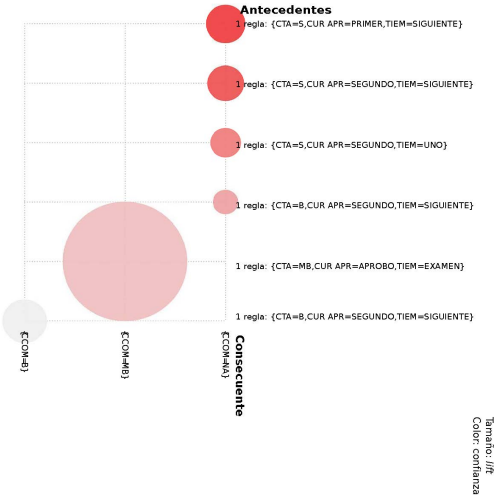


Fig. 1. Representación gráfica de las reglas entre "Taller de Matemáticas" y "Complementos de Matemáticas".

Estas reglas tienen un valor de *lift* mayor que el resto y un nivel de confianza para el primer caso de 40 % y de 32 % para el segundo caso.

Los casos con mayor ocurrencia fueron aquellos donde el consecuente tiene una calificación de NA en "Complementos de Matemáticas" y un antecedente de S en "Taller de Matemáticas", en la primera o segunda oportunidad. Esto se refleja en la tercera, cuarta y sexta regla, por lo que es posible determinar que el aprobar "Taller de Matemáticas" con S en una primera o segunda oportunidad u obtener B en la segunda oportunidad y tomar "Complementos de Matemáticas" al siguiente trimestre, tiene una mayor relación con una calificación de NA. Teniendo un nivel de confianza para el primer caso de 52 %, 51 % para el segundo y 44 % para el tercero.

De la quinta regla, se entiende que el aprobar "Taller de Matemáticas" con S en la segunda oportunidad y dejar pasar un trimestre para cursar "Complementos de Matemáticas" está relacionado con obtener una calificación de NA. Esto con un nivel de confianza de 47 %.

De manera similar, se obtuvieron 19 reglas para la relación entre "Taller de Matemáticas" e "Introducción al Cálculo", las cuales tras ser filtradas para eliminar las redundantes, dieron como resultado las 7 reglas mostradas en la Tabla 6.

La representación gráfica de estas relaciones se muestra en la Figura 2. La relación entre tamaño y color con confianza y *lift* es la misma que para la Figura 1.

Analizando las reglas de la Tabla 6 y de la Figura 2, observamos que las reglas en dónde la calificación en "Introducción al Cálculo" es MB tienen un *lift* mayor y son aquellas en las que se aprobó "Taller de Matemáticas" con MB en

Tabla 6. Reglas de asociación para la relación entre "Taller de Matemáticas" e "Introducción al Cálculo".

No	Regla	Soporte	Confianza	Lift
1	{CTA=MB,CUR_APR=APROBO,TIEM=EXAMEN} ⇒ {CCAL=MB}	0.020	0.40	3.17
2	{CTA=S,CUR_APR=PRIMER,TIEM=SIGUIENTE} ⇒ {CCAL=NA}	0.129	0.66	1.29
3	{CTA=S,CUR_APR=SEGUNDO,TIEM=SIGUIENTE} ⇒ {CCAL=NA}	0.025	0.65	1.27
4	{CTA=S,CUR_APR=PRIMER,TIEM=MAS_DE_DOS} ⇒ {CCAL=NA}	0.025	0.64	1.26
5	{CTA=S,CUR_APR=PRIMER,TIEM=DOS} ⇒ {CCAL=NA}	0.013	0.64	1.26
6	{CTA=B,CUR_APR=SEGUNDO,TIEM=SIGUIENTE} ⇒ {CCAL=NA}	0.011	0.60	1.19
7	{CTA=S,CUR_APR=PRIMER,TIEM=UNO} ⇒ {CCAL=NA}	0.014	0.58	1.15

el examen diagnóstico. Esto se ve reflejado en la primera regla, con un nivel de confianza del 40 %.

Por otra parte, obtener NA en "Introducción al Cálculo" está relacionado con cursar y aprobar "Taller de Matemáticas" con S o B. Esto se observa en las reglas dos y tres, en dónde aprobar con S al primer o segundo intento "Taller de Matemáticas" y cursar en el siguiente trimestre "Introducción al Cálculo" está relacionado con una calificación de NA. Estas reglas tienen un nivel de confianza del 66 % para el primer caso y del 65 % para el segundo.

En el caso de la cuarta y quinta regla, el cursar "Taller de Matemáticas" y obtener una calificación de S en la primera oportunidad y dejar pasar dos o más trimestres para cursar "Introducción al Cálculo" también está relacionado con una calificación de NA, esto con un nivel de confianza del 64 % para ambas reglas.

La sexta regla indica que cursar y aprobar "Taller de Matemáticas" con una calificación de B en la segunda oportunidad y cursar al siguiente trimestre "Introducción al Cálculo", está relacionada con una calificación de NA, esto con un nivel de confianza del 60 %.

Finalmente, la séptima regla señala que el cursar "Taller de Matemáticas", obtener una calificación de S al primer intento y dejar pasar un trimestre para cursar "Introducción al Cálculo", también se relaciona con una calificación de NA con un nivel de confianza del 58 %.

5. Conclusiones

En este trabajo se aplicó la Minería de Datos Educacional para determinar el impacto de un curso de nivelación en el desempeño escolar de los alumnos

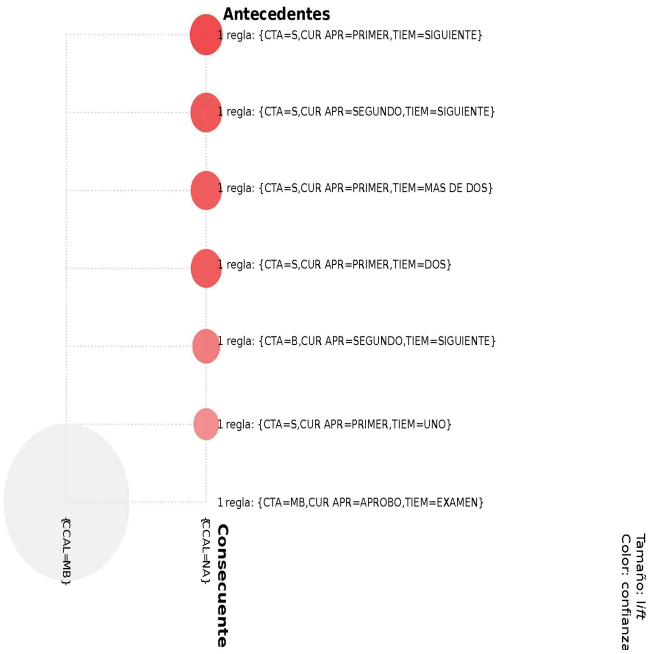


Fig. 2. Representación gráfica de las reglas entre "Taller de Matemáticas" e "Introducción al Cálculo".

de ingeniería de la Universidad Autónoma Metropolitana de la Unidad Azcapotzalco. Para esto, se utilizó la metodología CRISP-DM con el objetivo de analizar la relación entre el curso de nivelación "Taller de Matemáticas" con los cursos de "Complementos de Matemáticas" e "Introducción al Cálculo". Para el análisis de la información se utilizó la técnica de Reglas de Asociación empleando el algoritmo Apriori. Para esto, se establecieron varios supuestos a verificar o refutar con el análisis realizado.

El primer supuesto establecido fue que: «Un alumno que no cursa "Taller de Matemáticas" por haber aprobado el examen diagnóstico, tendrá un buen desempeño en los cursos de "Complementos de Matemáticas" e "Introducción al Cálculo"». Como resultado del análisis de este supuesto se cumple que los alumnos que obtuvieron una calificación de MB en el examen diagnóstico, obtuvieron un buen desempeño en los cursos posteriores anteriormente mencionados. Esto representa que aquellos alumnos que no cursan "Taller de Matemáticas" tienen un nivel aceptable de bases matemáticas, lo cual les permite tener un buen desempeño en sus siguientes cursos.

Sobre el segundo supuesto: «Un alumno que cursa y tiene buen desempeño en "Taller de Matemáticas", también lo tendrá en los cursos de "Complementos de Matemáticas" e "Introducción al Cálculo"», no se encontraron reglas lo suficientemente representativas que permitieran sustentar este supuesto, por lo que no se puede decir que esto esté pasando, lo que implicaría que "Taller de

Matemáticas" no cumple con el objetivo para el que fue creado.

Otro resultado obtenido del análisis mostró que los alumnos que aprobaron el examen diagnóstico con una calificación de B, posteriormente acreditaron la materia de "Complementos de Matemáticas" pero no la de "Introducción al Cálculo".

Los casos con mayor ocurrencia fueron aquellos alumnos que tuvieron que cursar "Taller de Matemáticas", obteniendo una calificación de S en su primera o segunda oportunidad y que no aprobaron "Complementos de Matemáticas" ni "Introducción al Cálculo". Otro factor presente en las reglas es que el tiempo que dejan pasar los alumnos después de aprobar la materia de "Taller de Matemáticas" e inscribir alguno de los otros cursos impacta en su desempeño de tal manera que el dejar pasar dos o más trimestres está relacionado fuertemente con no aprobar estos cursos.

Los resultados obtenidos sugieren que el contenido del curso "Taller de Matemáticas" debe de ser analizado para comprender porque no está ayudando en futuros cursos, en especial en "Introducción al Cálculo".

Como trabajo futuro, tenemos contemplado incluir más variables en el análisis, como la edad de ingreso del alumno, desempeño en el examen de admisión y escuela de procedencia. Asimismo, plantea realizar un análisis similar en otras seriaciones del plan de estudio de diversas licenciaturas que se imparten en la Universidad Autónoma Metropolitana Unidad Azcapotzalco.

Agradecimientos. Agradecimientos a Sistemas Escolares, a la Dirección de Ciencias Básicas e Ingeniería y a Secretaría Académica de la Universidad Autónoma Metropolitana Azcapotzalco por proporcionar la información necesaria para la realización del presente trabajo.

Referencias

1. Abdullah, Z., Herawan, T., Deris, M.M.: Mining significant least association rules using fast slp-growth algorithm. In: Kim, T.h., Adeli, H. (eds.) Advances in Computer Science and Information Technology. pp. 324–336. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
2. Abdullah, Z., Herawan, T., Deris, M.M.: Tracing significant association rules using critical least association rules model. International Journal of Innovative Computing and Applications 5(1), 3–17 (2013)
3. Abdullah, Z., Herawan, T., Deris, M.M.: Discovering interesting association rules from student admission dataset. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). pp. 135–142. Springer Singapore, Singapore (2014)
4. Azcapotzalco, U.A.M.: División de ciencias básicas e ingeniería. http://cbi.azc.uam.mx/es/CBI/Tronco_de_Nivelacion_Academica, Última consulta 20 Abr 2018
5. Azcapotzalco, U.A.M.: División de ciencias básicas e ingeniería. http://cbi.azc.uam.mx/es/CBI/Planes_Programa_Estudio_Com, Última consulta 20 Abr 2018

6. Baradwaj, B., Pal, S.: Mining educational data to analyze students' performance 2, 63–69 (10 2011)
7. Castro, F., Vellido, A., Nebot, À., Mugica, F.: Applying data mining techniques to e-learning problems. In: Evolution of teaching and learning paradigms in intelligent environment, pp. 183–221. Springer (2007)
8. Kumar, J.: A comprehensive study of educational data mining. International Journal of Electrical Electronics & Computer Science Engineering Special Issue-TeLMISR pp. 2348–2273 (2015)
9. Marquez-Vera, C., Romero-Morales, C., Ventura-Soto, S.: Mining educational data to analyze students' performance 8, 63–69 (2 2013)
10. Matetic, M., Bakaric, M.B., Sisovic, S.: Association rule mining and visualization of introductory programming course activities. In: Proceedings of the 16th International Conference on Computer Systems and Technologies. pp. 374–381. CompSysTech '15, ACM, New York, NY, USA (2015)
11. Merceron, A., Yacef, K.: Interestingness measures for association rules in educational data. In: Educational Data Mining 2008 - 1st International Conference on Educational Data Mining, Proceedings. pp. 57–66 (01 2008)
12. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40(6), 601–618 (2010)
13. Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. pp. 29–39. Citeseer (2000)

ISSN 1870-4069

Análisis metabólico predictivo en cultivos por lote en bioreactor utilizando redes neuronales artificiales

José Manuel Martínez Sánchez, Luis Bernardo Flores Cotera

CINVESTAV Zacatenco,
Departamento de Biotecnología y Bioingeniería,
México

{josem.martinez,lbcotera}@cinvestav.mx

Resumen. Un cultivo en lote de un microorganismo eucarionte operado en un bioreactor permite realizar mediciones en tiempo real de algunos parámetros biológicos y la colecta de muestras del cultivo para realizar mediciones a los párametros que no pueden ser medidos en tiempo real. En el presente artículo se muestran los resultados obtenidos de la aplicación de una red neuronal artificial para la predicción de condiciones limitantes en cultivos por lote en bioreactor de las tasas de incremeto de biomasa en la levadura *P. Rhodozyma*. La red neuronal atificial se construyó con una arquitectura multicapa, escrita en lenguaje `Python` donde se evaluaron datos obtenidos de pH, oxígeno disuelto y azúcares reductores de cultivos por lote arrojando una capacidad predictiva del 99 % para el patrón de crecimiento de biomasa, evidenciando así importante capacidad predictiva de estas herramientas para la biotecnología moderna.

Palabras clave: Bioingeniería, redes neuronales, bioprocesos, heurística, bioreactores por lote.

Predictive Metabolic Analysis in Batch-Culture in Bioreactor Using Artificial Neural Networks

Abstract. A batch culture of a eukaryotic microorganism operated in a bioreactor allows real-time measurements of some biological parameters and the collection samples to perform measurements to the parameters that can not be measured in real time. This article shows the results obtained from the application of an artificial neural red for the prediction of limitations in culture per batch in bioreactor for the biomass growth rate in the yeast *P. Rhodozyma*. The artificial neuronal network was built with a multi-layer architecture, written in Python language where the obtained data of pH, dissolved oxygen, biomass and crop reducing sugars per batch yielding a predictive capacity of 99 % for the biomass growth rate, evidencing as important predictive power of these tools for modern biotechnology.

Keywords: Bioengineering, neural networks, bioprocesses, heuristics, batch bioreactors.

1. Introducción

Un bioreactor es un instrumento diseñado para implementar y operar un ambiente biológicamente apropiado para el mantenimiento de lineas celulares especificas en condiciones controladas [7]. El bioreactor esta equipado con sensores y actuadores que permiten establecer y controlar con precisión parametros fisicoquímicos de interés en experimentos de optimización de medios, cultivos continuos, selección de cepas etc (Figura 1) [1]. Los cultivos por lotes en bioreactor son una forma de cultivo donde se utiliza un cantidad de biomasa conocida y una cantidad fija de nutrientes en un medio de cultivo con volumen definido, sin adición subsecuente de biomasa o nutrientes ni salida de productos [7,12].

Este tipo de cultivos son ampliamente utilizados en diversos laboratorios en experimentos de fermentación de diversas lineas celulares, incluyendo microorganismos y células animales y vegetales y se utilizan en aplicaciones variadas como la optimización de medios de cultivo, estudios de escalamiento de bioprocesos, selección de colonias, entre otros [4,12]. Los experimentos en bioreactor permiten registrar mucha información biológica importante mediante la captura de datos de los sensores asociados al sistema, tales como sensor de pH, sensor de oxígeno disuelto y sensor de nivel principalmente, además de permitir establecer configuraciones fijas o dinámicas de parámetros como la agitación, la temperatura y el flujo de aire mediante los actuadores disponibles [5,6].

La caracterización metabólica de un cultivo continuo incluye la captura de datos en linea de los sensores y actuadores disponibles registrando así los niveles de pH, oxígeno disuelto en tiempo real sin embargo, valores de parámetros como los gramos de biomasa producida, el consumo de azúcares y la producción de etanol y proteínas no pueden ser registrados en tiempo real e implican la aplicación de procedimientos *downstream* analizando las muestras recolectadas durante el experimento implicando también un costo asociado de reactivos y mano de obra [6]. El oxígeno disuelto presente en el bioreactor puede ser medido en tiempo real mediante un sensor polarográfico que registre cambios en la conductividad eléctrica de su membrana provocados por una reacción química acoplada a la cantidad de oxígeno presente en el medio (Figura 2). Mientras que las variaciones de pH pueden ser registradas en tiempo real con un sensor electroquímico. La medición de estos parámetros es trascendental para revelar información sobre el desarrollo del cultivo en el bioreactor y estan relacionados con una serie de variables biológicas (QO_2) y físicas (kLa) producto del crecimiento celular [3].

Un cultivo por lote en bioreactor puede presentar limitaciones provocadas de forma deliberada por el analista como parte del experimento [14] o pueden aparecer de manera eventual como resultado de ambientes no óptimos para las células en experimentos donde se desconocen los valores mas apropiados de cada parámetro y se realizan pruebas para encontrar las mejores condiciones de

cultivo [2,15]. En la actualidad, se han desarrollado diversos estudios de modelos de predicción y análisis que integran técnicas de inteligencia artificial, cuya estructura computacional flexible es capaz de identificar relaciones complejas no lineales entre los datos de entrada y salida [10,13]. Estudios recientes han desarrollado distintos modelos de redes neuronales artificiales para la predicción de cambios en variables biológicas muy diversas, como los estudios llevados a cabo por las corporaciones Verily Life Sciences y Google Inc. donde desarrollaron una red neuronal para aprendizaje profundo denominada *DeepVariant* que es capaz de identificar variaciones genéticas en bases de datos de secuenciación masiva de ADN revelando detalles hasta ahora desconocidos sobre el genoma [11], o los estudios realizados en el Centro de Investigación para las Ciencias de la Salud de Portugal, donde se utilizaron redes neuronales artificiales para optimizar la temperatura, flujo de metanol y concentración de dimetilsulfóxido en un cultivo en bioreactor para producir la enzima Catechol-O-Metiltransferasa. Sus resultados mostraron que la red neuronal artificial fue capaz de describir con gran precisión los efectos de las variables biológicas mencionadas en la producción de la enzima [12], ejemplificando así la reelevancia de estos modelos matemáticos en el desarrollo de muchas áreas de la biotecnología moderna.

P.Rhodozyma es un basidiomiceto facultativo con capacidades fermentativas de producción de etanol a partir de azúcar, además de representar una herramienta muy importante para la industria biotecnológica ya que es uno de los pocos microrganismos conocidos capaces de producir el oxo-carotenoide astaxantina, un pigmento con multiples aplicaciones y un elevado valor en el mercado [15]. Los cultivos por lote de este microorganismo pueden ser implementados en Bioreactor para estudios de diversa índole en procesos con duraciones típicas de mas de 80 horas, generando así grandes volumenes de información [15].

Con base en este panorama, el presente trabajo tiene como objetivo implementar una red neuronal artificial para realizar inferencias acerca de las tasas de crecimiento de biomasa en condiciones de cultivo predeterminadas de la levadura *P.Rhodozyma*, el modelo se construyo con 3 conjuntos de datos de entrada (pH,oxgeno disuelto y azucares reductores) y un conjunto de datos de salida (Biomasa) y se evaluó mediante parámetros estadísticos la respuesta de la red a la variación del número de capas (desde 1 hasta 5) y el número de neuronas por capa (desde 2 hasta 10); se muestran los resultados de la aplicación del modelo de la red neuronal seleccionada para la predicción de condiciones limitantes en cultivos por lote en bioreactor y cuyos resultados podrian utilizarse para la optimización de medios de cultivo y la selección de cepas.

2. Aspectos metodológicos

El proceso para la obtención de datos biológicos consistio en el desarollo de 10 cultivos por lote en bioreactor de la levadura *P.Rhodozyma* cada uno durante 84 horas con registro en tiempo real de las concentraciones de oxígeno disuelto y análisis posteriores de los niveles de biomasa y azúcares reductores.

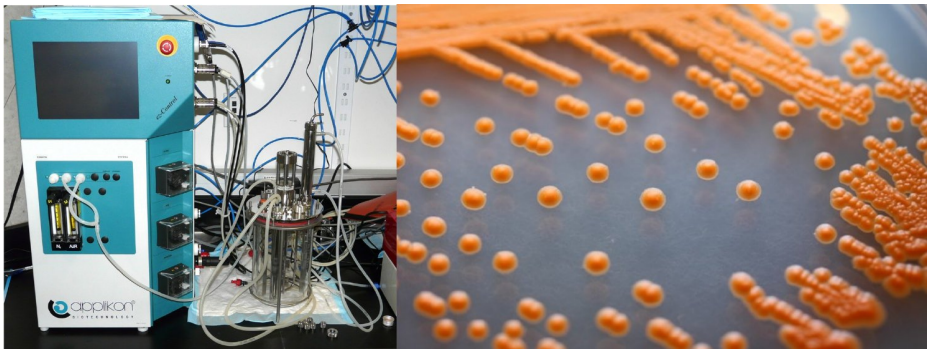


Fig. 1. Los cultivos por lote tienen muchas aplicaciones en biotecnología, a la iquierda, un bioreactor de 3L, a la derecha, colonias aisladas de *P.Rhodozyma*.

2.1. Datos de oxígeno disuelto y pH

La captura de las concentraciones de oxigeno disuelto en tiempo real se llevó a cabo mediante un sensor polarográfico de baja deriva AppliSens en intervalos de tiempo de 60 segundos durante todo el experimento y para cada cultivo, generando conjuntos de datos registrados y almacenados por una computadora acoplada al bioreactor, de igual forma, la captura de datos de pH se realizó con un sensor de pH esterilizable Applisens con registro de datos cada 60 segundos.

2.2. Datos de azúcares reductores

Los niveles de azúcares reductores se evaluaron mediante el método colorimétrico del ácido 3-5 Dinitrosalicílico[16] en muestras colectadas cada 6 horas durante todo el bioproceso.

2.3. Redes neuronales artificiales

Los modelos de redes neuronales artificiales (RNAs) son abstracciones matemáticas de la estructura del sistema nervioso, desarrollados para emular el funcionamiento de los sistemas neuronales biológicos. Estos modelos se basan en el óptimo tratamiento que realizan las millones de neuronas que conforman el cerebro a la información masiva, imprecisa y distorsionada proveniente del ambiente. El objetivo de emular estas estructuras neurobiológicas, es la construcción de sistemas de procesamiento paralelo de cálculo, memoria distribuida y adaptabilidad al entorno, mediante la interconexión de unidades de procesamiento virtuales.

Los modelos matemáticos obtenidos a partir de RNAs se caracterizan por su robustez y, capacidad de aprendizaje, generalización y tolerancia a fallas. Una red neuronal artificial (RNA) es fundamentalmente, un sistema reticular constituido por estructuras paralelas que contienen procesadores elementales,

htb]

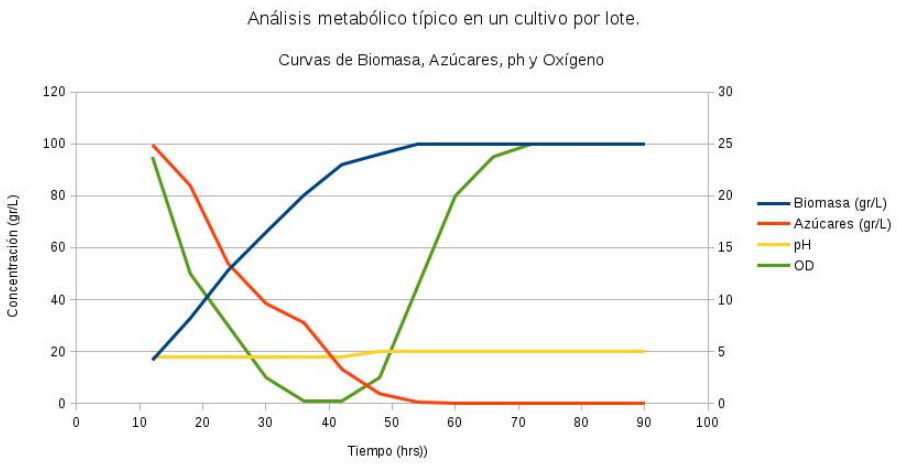


Fig. 2. Graficación de los valores de parámetros biológicos recolectados durante un cultivo por lote.

llamados neuronas, dispuestos en capas e interconectados a través de enlaces con diferentes pesos numéricos asociados a cada elemento. Los elementos fundamentales se toda red neuronal artificial son una capa de entrada, una capa de procesamiento y una capa de salida, donde se encuentran asociadas variables de entrada y de salida, pesos numéricos sinápticos, función de activación, función de propagación y función de salida (Figura 3). La configuración de estos elementos define la eficiencia del sistema[10].

Perceptrón multicapa. La red neuronal artificial utilizada para el desarrollo del programa se denomina red neuronal perceptrón multicapa (RNPM). Este modelo neuronal se caracteriza por ser unidireccional con conexiones *feedforward*, carente de interconexiones a otras neuronas de una misma capa o conexiones hacia neuronas de capas anteriores. En una RNPM el número de capas ocultas puede ser mayor o igual que uno, y difiere de la Red Neuronal Perceptrón (RNP) debido a la implementación de una función de activación no lineal. La arquitectura de la RPNM, dispone de 3 tipos de capas que son :

1. La capa de entrada: receptora de la información desde el exterior
2. Las capas ocultas: operadoras de las funciones de asociación entre la entrada y la salida.
3. La capa de salida: Devueleve las predicciones de la red neuronal (Ecuación 1).

La salida de una capa oculta del perceptrón multi capa, puede ser representada con la expresión:

$$f(x = GW^T x + b) f : R^D \rightarrow R^L, \tag{1}$$

donde D es el tamaño del vector de entrada x. L es el tamaño del vector de salida. G es la función de activación.

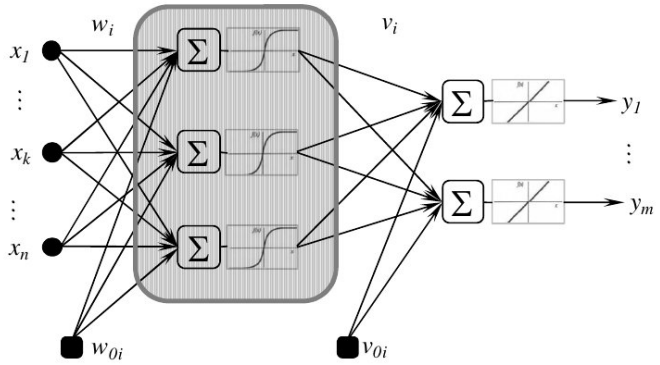


Fig. 3. Estructura general de una red neuronal perceptrón multicapa.

Algoritmo de implementación. El algoritmo para implementar la Red Neuronal Perceptrón Multicapa se implementó en el lenguaje `python 2.7.13` cuyas operaciones clave se describen a continuación.

```
Definir las clases para representar las neuronas de la red.

class LayerN():
    def __init__(self, numero de entradas, numero de neuronas):
        asigna los pesos sinapticos iniciales.

class ANN():
    def __init__(self, capa_1, capa_n-1, capa_n):
        Asigna las capas correspondientes al objeto.

def_funcionSigmoide(Neurona)
    return La salida de la neurona.

pesos_S = Inicializa los pesos para la matriz de la primer
                  sinapsis aleatoriamente.

Implementa cambios  en las matrices de salida de las capas.
```

```
def test_N(self, inputs):
Donde i = 0
        salida en capa i = producto punto
        (pesos sinapticos de  la capa 1,el array ingresado
 como parametro.)
        salida de la capa i + 1 = producto punto
        (salida de la capa i,los pesos sinapticos de la capa i + 1)
        Iterar sobre cada capa disponible.
        return salida en capa i, salida en capa n

Inicializamos parametros para construir la red.
x_0= Matriz de datos de entrada.
y_0= Vector de datos de salida.
ni= Numero de iteraciones.
nn = Numero de neuronas en la capa oculta.
Obtener conjunto de entrada.
Obtener conjunto de salida.

def entrenamiento (self, x_0, y_0, ni):
        for iteration in xrange(ni):
        salida en capa i= self.test_N(x_0)
        salida en capa i+1= self.test_N(x_0)
        Optimizacion para el error cuadratico medio mediante
        el algoritmo Levenberg  Marquardt y el criterio
   early stopping.
        Calcular error y deltas para cada capa.
        ajuste_capa_i = producto punto(x_0, delta capa 1)
        ajuste_capa_i+1 =  producto punto(x_0, delta capa 2)

        ajuste de pesos para capa i
        ajuste de pesos para capa i+1

Crear la capa i de la red con n neuronas y m entradas
capa_i = LayerN(n, m)
Repetir para cada capa de la red.

Crear la red con n neuronas.
ann = ANN(Numero de capas disponibles.)

Entrenar la red.
ann.entrenamiento(datos de entrada, datos de salida,
                  ciclos de entrenamiento)
```

2.4. Estructura y parámetros de la red

La estructura de la red se compone de 3 conjuntos de datos de entrada (pH,oxígeno disuelto y azúcares reductores) y un conjunto de datos de salida. Se utilizó el algoritmo Levenberg-Marquardt para optimizar el error cuadrático medio y mejorar el ajuste de reconocimiento de patrones debido a que para

este problema en particular, se considera que los parámetros para esta red no son numerosos, por lo tanto la elección de este algoritmo se justifica para ganar velocidad de procesamiento [17].

Se realizaron dos mil repeticiones de ejecución del programa hasta alcanzar la validación total de la información en conjunto con un gradiente con cinco revisiones de validación para evaluar la calidad del sistema. Se evaluó la respuesta de la red a la variación del el número de capas (desde 1 hasta 5) y el número de neuronas por capa (desde 2 hasta 10), generando así veinticinco versiones del modelo. El programa utiliza el 60 % de los datos de los periodos seleccionados para la fase de entrenamiento y los restantes para la validación y prueba del modelo (10 % para cada fase respectivamente). Los pesos iniciales fueron asignados aleatoriamente en el intervalo de (0.5,+0.5). En cada modelo se aplicó la función de entrenamiento y la posterior evaluación de las capacidades predictivas del sistema. Como entradas del modelo se utilizaron datos de pH,oxígeno disuelto y azúcares reductores y como salida del modelo se utilizaron los datos de biomasa.

2.5. Analisis estadístico y métricas de evaluación

Para evaluar la precisión de los modelos elegidos se implementó un programa en lenguaje R (Figura 4), manejado mediante un script en `python` como subproceso `subprocess.call()` que evalua los valores de los parámetros estadísticos reportados como métricas de evaluación en estudios de modelos de simulación, [8] los cuales son el coeficiente de correlación (CC), coeficiente de concordancia (CCC), raíz cuadrada del error cuadrático medio (RMSE), error porcentual absoluto de la media (MAPE), error medio cuadrático (MSE) y error medio absoluto (MAE) de los datos de salida de cada modelo disponible de la red neuronal, el programa almacena la información en disco para su análisis mediante objetos de la clase `shelve` y de acuerdo al siguiente esquema:

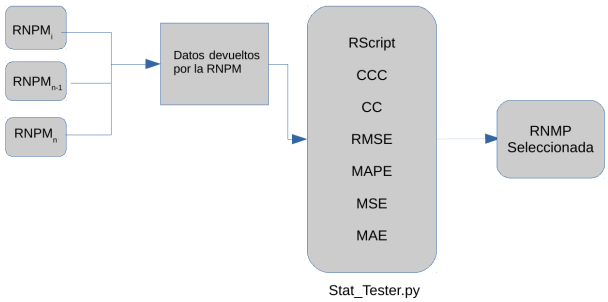


Fig. 4. Esquema de flujo para el análisis estadístico de resultados.

3. Resultados

Al analizar todos los criterios estadísticos antes mencionados entre todas las versiones del modelo, el modelo no. 25 ofreció mejores resultados a juzgar por el máximo coeficiente de correlación y por los valores de los otros criterios estadísticos antes mencionados, este modelo de la red se seleccionó debido a que presenta los mejores resultados favorables, con un máximo Coeficiente de Correlación (0.9958) revelando que dicho modelo presenta un 99 % de presición del pronóstico en la correlación de los datos de prueba contra los datos simulados por la red neuronal (Tabla 1).

Se observa así mismo el mayor coeficiente de concordancia (0.9958), menor error medio absoluto (0.0171) y menor error medio cuadrático (0.00060), analizando los valores obtenidos de error porcentual absoluto de la media se observa que el resultado de las predicciones del modelo 25 son aceptables al tener un valor menor al 5 %. En este modelo se implementaron 5 capas y 10 neuronas por cada capa y los resultados mostrados revelan la importancia de evaluar el número de capas y de neuronas para mejorar la predicción del sistema.

Tabla 1. Selección de los mejores modelos generados.

Modelo	CC	CCC	RMSE	MAPE	MSE	MAE
Mod1	0.9877	0.9877	0.0272	5.1920	0.00071	0.0199
Mod2	0.9751	0.9751	0.0270	5.1901	0.00079	0.0198
Mod3	0.9807	0.9807	0.0270	5.1881	0.00071	0.0198
Mod4	0.9817	0.9817	0.0268	5.0199	0.00074	0.0197
Mod5	0.9841	0.9841	0.0265	5.0190	0.00071	0.0197
Mod6	0.9839	0.9839	0.0265	5.0160	0.00073	0.0196
Mod7	0.9899	0.9899	0.0264	5.0030	0.00071	0.0196
Mod8	0.9905	0.9905	0.0265	5.0001	0.00072	0.0195
Mod9	0.9913	0.9913	0.0263	4.9980	0.00071	0.0193
Mod10	0.9916	0.9916	0.0260	4.9078	0.00078	0.0193
Mod11	0.9944	0.9944	0.0258	4.8950	0.00071	0.0189
Mod12	0.9945	0.9945	0.0254	4.8072	0.00071	0.0188
Mod13	0.9945	0.9945	0.0250	4.7033	0.00070	0.0182
Mod14	0.9946	0.9946	0.0243	4.5234	0.00070	0.0181
Mod15	0.9947	0.9947	0.0240	4.5099	0.00070	0.0181
Mod16	0.9949	0.9947	0.0238	4.4999	0.00070	0.0180
Mod17	0.9952	0.9952	0.0235	4.4556	0.00070	0.0180
Mod18	0.9955	0.9955	0.0226	4.4188	0.00069	0.0179
Mod19	0.9955	0.9955	0.0226	4.4102	0.00067	0.0178
Mod20	0.9956	0.9956	0.0225	4.4011	0.00068	0.0178
Mod21	0.9956	0.9956	0.0225	4.4001	0.00068	0.0176
Mod22	0.9957	0.9957	0.0225	4.3398	0.00068	0.0175
Mod23	0.9957	0.9957	0.0224	4.3399	0.00069	0.0174
Mod24	0.9957	0.9957	0.0224	4.3301	0.00069	0.0172
Mod25	0.9958	0.9958	0.0223	4.3121	0.00060	0.0171

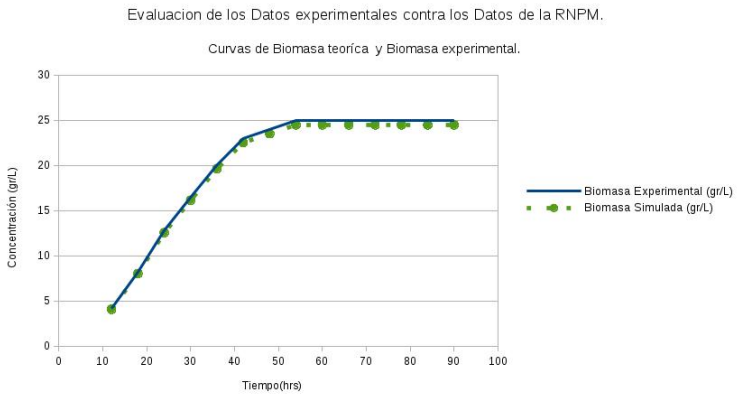


Fig. 5. Comparación entre los resultados de los niveles de biomasa experimentales contra los niveles de biomasa predichos por la red neuronal.

4. Conclusiones

La simulación de las tazas de aumento de biomasa en un bioreactor operado en cultivo por lote utilizando redes neuronales multicapa generó información estadística muy útil, presentada en la tabla 1 , para predecir el comportamiento del cultivo en función del tiempo bajo condiciones definidas, representando así una herramienta muy eficiente para predecir limitaciones en desarrollo del cultivo o cambios en los patrones de crecimiento en experimentos de selección de colonias, esta conclusión se encuentra soportada por el el analísis de los datos de salida de la red neuronal (Figura 5), cuya gráfica revela un patrón muy similar a la grafíca de los datos experimentales, donde una limitación en el crecimiento sería fácilmente identificable por pendientes negativas en la gráfica.

La implemetación de la red neuronal perceptrón multicapa en el lenguaje `Python 2.7.13` permitió obtener un modelo con coeficientes de correlación de pearson y concordancia de un 0.9958 reflejando una muy fuerte relación contre los datos experimentales y los datos simulados por la red neuronal y en conjunción con error porcentual absoluto de la media menor al 5 %, se obtuvo un modelo robusto con una elevada presición en la predicción. La aplicación de este tipo de modelos representa una importante alternativa para la generación de herramientas de detección de limitaciones metabólicas acoplado al sistema de aquisición de datos del bioreactor, que sea capaz de identificar cambios en patrones de los niveles de biomasa aportando información muy importante al experimentador para aplicar ajustes oportunos a las condiciones de cultivo o seleccionar cepas distintas de una misma linea celular optimizando condiciones de operación.

La propuesta elegida en el presente trabajo para inferir las tasas de variación de biomasa se encuentra justificada en el hecho de que una red neuronal permite modelar fenomenos de multiples variables en relaciones no lineales como es el

caso del problema abordado en el presente trabajo y los resultados obtenidos, de igual manera `Python` y R son lenguajes ampliamente usados para la ciencia de datos y visualización de información compleja y ofrecen flexibilidad y potencia en para el diseno de software científico. La capacidad de las redes neuronales multicapa para modelar fenónemos no lineales sería muy útil en investigaciones futuras basadas en el presente trabajo para desarrollar un modelo predictivo que involucre los niveles de expresión genética de los genes relacionados a los cambios metabólicos de interés con las condiciones de operación del cultivo, conectado así datos de expresión genética con variables fisicoquímicas del ambiente surgiendo la posibilidad de disponer de un modelo predictivo avanzado que permita optimizar mas finamente las condiciones de operación del cultivo para la producción metabolitos específicos en esta importante levadura de interés comerecial.

Agradecimientos. Los autores agradecen profundamente al Centro de Investigación y Estudios Avanzados del IPN por las facilidades prestadas en equipos e infraestructura para la realización del presente trabajo.

Referencias

1. Zhou, J.X., Huang, S.: Synthetic Biology- Tools and Applications. Chapter 5 - theoretical considerations for reprogramming multicellular systems, pp. 81–99, (2013)
2. Fritz, G., Buchler, N.E., Hwa, T., Gerland, U.: Designing sequential transcription logic: a simple genetic circuit for conditional memory. Systems and Synthetic Biology, 1, pp. 89–98 (2007)
3. Tu, B.P., Kudlicki, A., Rowicka, M., McKnight, S.L.: Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. Science, 310(5751), pp. 1152–1158 (2005)
4. Stano, P., De Souza, T.P., Kuruma, Y., Carrara, P., Luisi, P.L.: Synthetic Biology-Tools and Applications. Chapter 14 - semi-synthetic minimal cells: Biochemical, physical, and technological aspects, pp. 261–276 (2013)
5. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1996)
6. Abbas, A., Al-Bastaki, N.: Modeling of an RO water desalination unit using neural networks. Chem. Eng. J., 114, pp. 139–143 (2005)
7. inar, O., Hasar, H., Kinaci, C.: Modeling of submerged membrane bioreactor treating cheese whey wastewater by artificial neural networks. J. Biotechnol, 123, pp. 204–209 (2006)
8. Lin, L., Hedayat, A.S., Sinha, B., Yang, M.: Statistical methods in assessing agreement: Models, issues, and tools. Journal of the American Statistical Association, 97(457), pp. 257–270 (2002)
9. Da Silva, I.N., Flauzino, R.A.: An approach based on neural networks for estimation and generalization of crossflow filtration processes. Appl. Soft Comput., 8, pp. 590–598 (2008)
10. Harvey, R.L.: Neural networks principles. Prentice-Hall, Englewood Cliffs (1994)
11. Poplin, R.: Creating a universal SNP and small indel variant caller with deep neural networks. Cold Spring Harbor Laboratory bioRxiv (2018)

12. Pedro, A.Q., Martins, L.M., Dias, J.M., Bonifcio, M.J., Queiroz, J.A., Passarinha, L.A.: An artificial neural network for membrane-bound catechol-O-methyltransferase biosynthesis with *Pichia pastoris* methanol-induced cultures. 14(113) (2015)
13. Da Silva, I.N., Flauzino, R.A.: An approach based on neural networks for estimation and generalization of crossflow filtration processes. Appl. Soft Comput., 8, pp. 590–598 (2008)
14. Mickel, L., Jansen, A., et. al.: Saccharomyces cerevisiae strains for second-generation ethanol production: from academic exploration to industrial implementation. FEMS Yeast Research, 17 (2017)
15. Mata-Gómez, L.C., et. al.: Biotechnological production of carotenoids by yeasts: an overview. Microb Cell Fact, 13(12) (2014)
16. Miller, G.L., et. al.: Use of Dinitrosalicylic Acid Reagent for Determination of Reducing Sugar. Analytical Chemistry. 31, (1959)
17. Kermani, B.G., et. al.: Performance of the LevenbergMarquardt neural network training method in electronic nose applications. Sensors and Actuators B: Chemical, 110(1), pp. 13–22 (2005)

ISSN 1870-4069

Clasificación de clorosis en hojas de árboles de naranja mediante aprendizaje automático

Juan P-Salazar, Eddy Sánchez-DelaCruz, R.R. Biswal

Instituto Tecnológico Superior de Misantla, Posgrado en Sistemas Computacionales, Misantla, Veracruz,
México

jpsalazar1122@gmail.com, {esanchezd, rroshanb}@itsm.com.mx

Resumen. La fruta de naranja es considerada como uno de los principales productos agroalimentarios de mayor producción en México [13]. El volumen promedio producido es de 4.2 millones de toneladas anuales y el estado de Veracruz a nivel nacional es el primer productor aportando el 44.5% del volumen total [12]. La producción eficiente de este fruto se ve afectada por la diseminación de enfermedades en los campos de producción y es la principal causa de pérdidas económicas. Para solucionar esta problemática se han implementado técnicas computacionales con la capacidad de clasificar e identificar dichas enfermedades. Esta investigación propone evaluar un conjunto de clasificadores para su implementación en la detección de enfermedades usando características sintomáticas en hojas de árboles de naranja Valencia (*Citrus Sinensis)*. Las técnicas y métodos de procesamiento de imágenes para la creación de una base de datos y clasificación binaria aquí aplicados, nos llevaron a encontrar que tres clasificadores ensamblados (metaclasificadores) en combinación con el algoritmo de aprendizaje profundo perceptrón multicapa (Dl4jMlp: Deep Learning for java with Multilayer Perceptron) tienen la capacidad de lograr una clasificación del 100% en bases de datos de tamaño pequeño y mejoradas alcanzado por la aplicación de técnicas de procesamiento de imágenes y minería de datos.

Palabras clave: Técnicas computacionales, procesamiento de imágenes, algoritmos ensamblados, aprendizaje profundo, perceptrón multicapa, clasificación.

Classification of Chlorosis in Leaves of Orange Trees Using Machine Learning

Abstract. Oranges are considered to be one of the main agroalimentary products of major production in México [13]. The average volume produced is 4.2 million tons and Veracruz turns out to be the first contributing state with 44.5% of the total volume [12]. Efficient production of this fruit is affected by the spread of diseases in production fields resulting in major economic losses. To solve this problem, computer techniques have been implemented with the main objective to classify and identify these diseases. This research proposes to evaluate various

classifiers for implementation in disease detection using symptomatic features in leaves of Valencia orange trees (*Citrus Sinensis)*. The procedures and methods for the creation of a database and binary classification applied, led us to find out that three Metalearning Algorithms in combination with the Dl4jMlp (Deep Learning for java with Multilayer Perceptron) have the ability to accomplish a classification of 100% in small dimension and improved databases achieved by the application of image processing and data mining techniques.

Keywords: Computer techniques, image processing, metalearning algorithms, deep learning, multilayer perceptron, classification.

1. Introducción

La detección oportuna de enfermedades en cultivos es un factor determinante en una eficiente producción agrícola, por lo que es esencial la identificación de manera temprana. En la actualidad aunque el avance tecnológico aplicado al área agrícola crece de manera exponencial aún existe rezago en el estudio de muchas enfermedades. La implementación de herramientas computacionales para la clasificación se ha dirigido con gran intensidad a Huanglongbing (HLB), sin embargo, existen enfermedades a las cuales no se le ha dado importancia relevante como el caso de Clorosis Variegada de los Cítricos (CVC) y Virus Psorosis de los Cítricos (CPsV), estas enfermedades son consideradas destructivas y de peligro para la producción de cítricos y son comparadas con HLB por el impacto en pérdidas económicas [1]. Los síntomas visibles en las hojas de las plantas son los primeros que alertan sobre una posible plaga o alguna deficiencia nutrimental. El CVC al igual que el HLB, el CPsV y la deficiencia de Zinc se caracterizan por la decoloración de las hojas de cítricos [4], debido a esta similitud, es difícil confiar solo en los síntomas foliares para la identificación. Para solucionar esta problemática se han implementado técnicas como análisis y diferenciación fotoquímica entre HLB y deficiencia de Zinc [2], espectroscopia de infrarrojo medio para la detección de HLB [1], sensores de visión [10] y métodos de visión por computadora, procesamiento de imágenes y minería de datos avanzada empleando algoritmos para la clasificación de enfermedades de cítricos [11], técnicas aplicables a la clasificación de enfermedades mediante la presencia de características foliares como la clorosis.

Cualquier tipo de clorosis visible en las hojas de cítricos precisa de ser investigada. La pérdida de color verde es una indicación importante de problemas de crecimiento y pérdidas de productividad; los causantes pueden ser tanto bacterias, virus o deficiencias nutricionales. La CVC es causada por el agente *"Xylella fastidiosa"* [14]. La bacteria vive y se multiplica en la savia de las plantas de cítricos, bloqueando la absorción de agua. Los síntomas foliares del CVC son muy similares a la deficiencia de nutrientes y otras enfermedades como HLB y CPsV. Los primeros síntomas de la hoja se parecen a la deficiencia de Zinc con áreas cloróticas intervenales en la superficie superior, las hojas pueden ser más pequeñas de lo normal y los síntomas de la hoja como la decoloración de las áreas afectadas y las manchas obscuras o marrón con acumulación de gomosis en el envés se intensifican mucho más en hojas maduras, en la última etapa las manchas comienzan a extenderse hacia el borde y el tejido se seca [4,8]. Estas características sintomáticas de las hojas se pueden aprovechar para la clasificación de enfermedades empleando técnicas de procesamiento de imágenes, minería de datos y

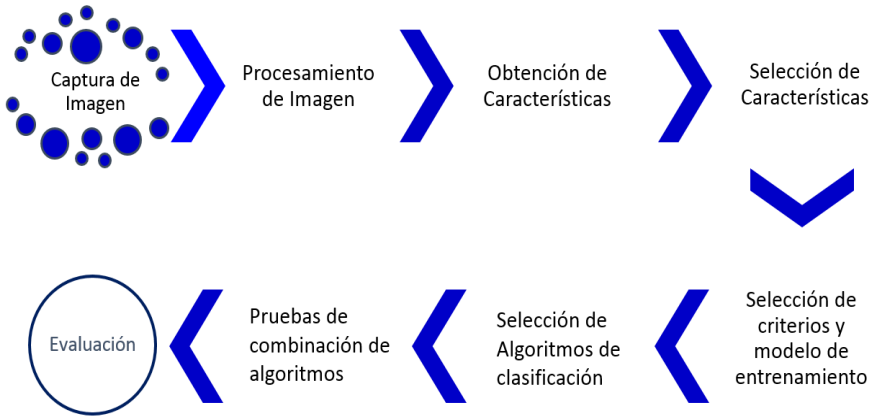


Fig. 1. Metodología propuesta.

aprendizaje profundo con la finalidad de obtener un clasificador eficientemente capaz de identificar las diferencias entre la presencia sintomática de clorosis y hojas visiblemente sanas.

1.1. Propuesta de solución

Esta investigación propone un método de procesamiento de imágenes [15] y minería de datos para la clasificación binaria de presencia sintomática de clorosis en hojas de naranja y hojas visiblemente sanas para la obtención de un clasificador eficiente. El diagrama siguiente proporciona una breve descripción de la metodología aplicada, la cual se describe a detalle en la sección cuatro (ver Figura 1).

2. Trabajos previos

Pourreza et al., en una de sus investigaciones desarrollaron un sensor de visión aprovechando la rotación del plano de polarización de la luz en algunas longitudes de onda para la identificación de síntomas de HLB; los resultados confirmaron que la iluminación polarizada de banda estrecha a 591 nm aumentó significativamente la precisión del diagnóstico en la clasificación y detección de HBL [10].

Camponez et al., aplicaron un método de diagnóstico para la detección de presencia de HLB y CVC en hojas de árboles de naranja dulce mediante espectroscopia infrarroja con reflectancia total atenuada de la transformada de Fourier y el clasificador inducido mediante regresión parcial de mínimos cuadrados. El modelo consideró cuatro clases de hojas, una con hojas saludables, con CVC sintomático, con HLB sintomático y con HBL asintomático en donde los resultados obtenidos arrojaron un porcentaje del 93.8% en la correcta identificación de los cuatro tipos de hojas estudiados [1].

Pydipati et al., propusieron un modelo empleando un método de co-ocurrencia (CCM) para determinar si el tono de color, la saturación y la intensidad (HSI) basados

en características de textura en conjunción con algoritmos de clasificación estadística podrían ser utilizados para identificar las hojas de cítricos enfermas y normales bajo condiciones de laboratorio. Se evaluaron muestras de hojas de cítricos normales y enfermas estas últimas con puntos grasos, melanosis y escamosis. El análisis discriminante de la muestra de hojas con características de textura CCM logró precisiones de clasificación de más del 95% para todas las clases cuando se utilizaron la tonalidad de color y la saturación de las características de textura [11].

Ceballos et al., desarrollaron y una técnica optimizada basada en el método GC-MS para el análisis de metabolitos no alcanzables en hojas de cítricos. Las muestras de HLB y deficiencia de zinc de hojas de árboles de naranja dulce fueron sometidas al proceso de micro extracción en fase sólida y tratamientos de derivatización anteriores al análisis GC-MS. El análisis de componentes principales alcanzó una correcta clasificación de todos los derivativos de extractos líquidos empleando biomarcadores para la rápida diferenciación de HLB y deficiencia de zinc [2].

Sankaran et al., muestran en su investigación un enfoque en la aplicación de espectroscopia en el infrarrojo medio para la detección de HLB. Se emplearon muestras de hojas saludables, con HLB y deficiencia de nutrientes en árboles mediante dos formas de procesamiento y analizándolas usando un espectrómetro portable y robusto. Se utilizaron datos de la absorbancia espectral en el rango de los 5.15–10.72nm y se procesaron mediante corrección de alineamiento básica, corrección de datos atípicos y remoción de datos de las longitudes de banda de la absorbancia del agua. El primero y el segundo proceso derivativo fueron calculados mediante el método de Savitzky–Golay y analizados mediante análisis de componentes principales y los records de este fueron clasificados usando el algoritmo de análisis discriminante cuadrático (QDA) y el algoritmo de vecinos más cercanos (KNN), determinando que el algoritmo KNN obtuvo por arriba de 95% en comparación con el algoritmo QDA, en este caso los datos analizados en bruto proporcionaron mejor precisión en la clasificación comparados con los datos derivativos del procesamiento uno y dos [12].

Mohanty et al., trabajaron en el desarrollo de un clasificador de imágenes de alta precisión con la finalidad de diagnosticar enfermedades de plantas empleando una base de datos del proyecto de recolección de hojas con presencia de enfermedades y saludables llamado PlantVillage del cual se emplearon 54,306 imágenes en la clasificación de 26 enfermedades distintas de 16 especies de cultivos mediante el enfoque de una red neuronal convolucional. El desempeño de los modelos de clasificación fue medido basado en su habilidad de predecir los pares correctos cultivo-enfermedad, dadas 38 posibles clases. El modelo con mejor desempeño alcanzo un puntaje medio F1 de 0.9934 es decir una precisión del 99.35% demostrando la factibilidad técnica de este enfoque [9].

Kadir et al., propusieron un método que captura información de color como un aspecto importante para la identificación de enfermedades en hojas. En esta investigación se incorporaron características de forma geométrica, color y textura de cada hoja; empleando una Red Neuronal Probabilística como clasificador. Lo resultados experimentales mostraron que el método de clasificación elegido obtuvo una precisión media del 93.75% cuando fue probado con la base de datos Flavia la cual contiene 32 tipos de hojas de plantas [7].

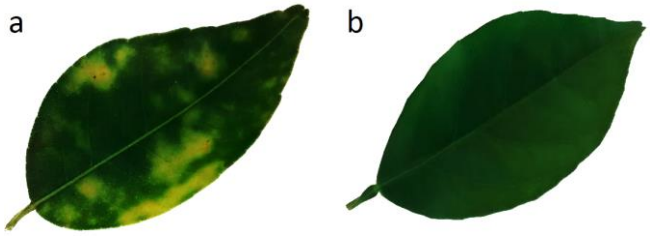


Fig. 2. Muestras de hojas (a) con presencia de clorosis y (b) saludable.

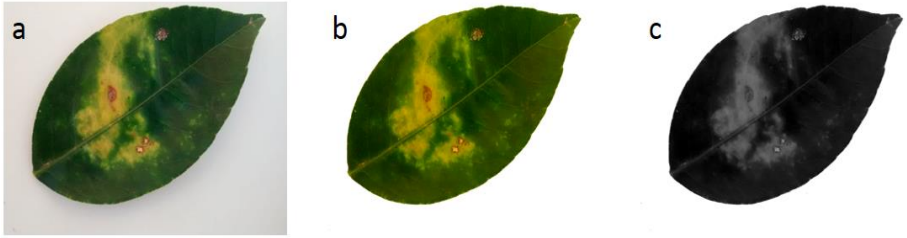


Fig. 3. Procesamiento de imágenes (a) original (b) segmentada y mejorada (c) escala de grises.

3. Materiales y método

3.1. Adquisición de la base de datos

Se recolectaron hojas enfermas con presencia de clorosis y hojas saludables de árboles de naranja dulce Valencia *(Citrus Sinensis)*, se tomaron fotografías individuales en un ambiente controlado para reducir el brillo y reflectancia en las imágenes, se empleó la cámara de un dispositivo móvil con calidad de 5 megapíxeles y una resolución de 2592x1456; las fotografías obtenidas se recopilaron en una base de datos, 60 imágenes de hojas saludables y 60 con clorosis sintomática (ver Figura 2.).

La obtención de características de las imágenes de hojas se realizó mediante algoritmos de reconocimiento de patrones aplicados a procesamiento digital de imágenes [15]. El primer paso fue la eliminación del brillo residual y una mejora de contraste mediante la ecualización del histograma, el siguiente fue realizar una segmentación para eliminar el fondo de la imagen empleando umbralización con el método de otsu, aprovechando la diferencia de los niveles de intensidad de las hojas y el fondo, esto nos dio como resultado la obtención de nuestra región de interés que es solo la hoja, posteriormente la imagen se convirtió a escala de niveles de gris para la obtención de los valores de cada pixel y características estadísticas estándar del histograma de frecuencias. Estos valores se guardaron en un archivo en formato CSV para la generación de nuestra base de datos (ver Figura 3.).

La base de datos obtenida contiene la información de las características más relevantes de las hojas, en primer término, las dos clases se distribuyen en forma de filas, 60 de ellas son nombradas saludable y 60 enferma, la primera para las hojas

Tabla 1. Resumen de las características de la base de datos obtenida.

Clase	Categoría	Características de Niveles de Gris	Número de Características
1	Saludable	Valores de Pixeles	256
1	Saludable	Características Estadísticas	6
2	Enferma	Valores de Pixeles	256
2	Enferma	Características Estadísticas	6
		Total de Características por Clase	262

Tabla 2. Conjuntos de entrenamiento y prueba.

Clase	Entrenamiento	Prueba	Total
Saludable	40	20	60
Enferma	40	20	60
Total	80	40	120

visiblemente saludables y la segunda para las hojas con presencia sintomática de clorosis respectivamente.

Cada columna de la base de datos contiene cada uno de los atributos de cada hoja iniciando con los valores de cada pixel del histograma de frecuencia de niveles de gris desde 0 hasta 255, es decir ocupa las primeras 256 columnas; las ultimas 6 columnas son ocupadas por las características estadísticas del histograma de frecuencia incluyendo la media, desviación estándar, curtosis, oblicuidad (índice de asimetría), media laplaciana y gradiente límite de la media [15]. Por último los nombres de las dos clases: *saludable* y *enferma*, se ubicaron en la última columna con el propósito evitar errores en su procesamiento. La tabla siguiente brinda una breve descripción de las características de la base de datos (ver Tabla 1).

3.2. Tratamiento de los datos

El procesamiento de la base de datos se realizó mediante una combinación de algoritmos ensamblados con el algoritmo de aprendizaje profundo perceptrón multicapa (Dl4jMlp). Para la realización de las pruebas se emplearon 3 criterios de evaluación el primero por validación cruzada, el segundo empleando una división de porcentaje 2/3 para entrenamiento y 1/3 para pruebas y por último por muestra reprentativa calculada de 92 instancias con un porcentaje de 23.33% del total de instancias. El procedimiento consistió en enfrentar todos los algoritmos metaclasificadores contra el algoritmo clasificador Dl4jMlp y evaluar el porcentaje de precisión que obtiene cada combinación, estos resultados se pueden apreciar en la sección 4. El criterio de división de porcentaje de 2/3 para entrenamiento y 1/3 para pruebas, con el que se obtuvo el valor óptimo, se muestra en la Tabla 2.

Los algoritmos utilizados para realizar la combinación con el algoritmo Dl4jMlp (Deep Learning for Java with Multilayer Perceptron) se categorizan dentro de los Algoritmos Metaclasificadores.

Algoritmo 1: Algoritmo MLP Simple

```
Initialize w and b to zero
repeat
    for l∈1..L do
        if yl(w·xl+b)≤β then
            for i∈1..N do
                if /vi·xl+di/≤l then
                    vi←vi+λylxl
                    di←di+λyl
                end if
            end for
            b←b+λyl
        end if
    end for
until termination criterion
```

Tabla 3. Resultados de la combinación de algoritmos metaclasificadores y Dl4jMlp.

Metaclasificadores	**Aprendizaje Profundo**	**Val. Cruzada 10iteraciones**	**2/3-1/3**	**Muestra Representativa**
----------	Dl4jMlp	97.5	97.5	92.2222
AttributeSelectedClassifier	Dl4jMlp	99.1667	100	98.8889
Bagging	Dl4jMlp	50	42.5	44.4444
ClassificationViaClustering	Dl4jMlp	62.5	67.5	
ClassificationViaRegression	Dl4jMlp	50	42.5	44.4444
CVParameterelection	Dl4jMlp	50	42.5	44.4444
FilteredClassifier	Dl4jMlp	99.1667	100	96.6667
LogitBoost	Dl4jMlp	50	57.5	55.5556
MultiClassClasifier	Dl4jMlp	50	42.5	44.4444
MultiScheme	Dl4jMlp	50	42.5	
MultiSearch	Dl4jMlp	50	42.5	44.4444
OrdinalClassClassifier	Dl4jMlp	97.5	97.5	92.2222
RandomCommite	Dl4jMlp	50	42.5	44.4444
RandomizableFilteredClassif	Dl4jMlp	54.1667	60	51.1111
RandomSubSpace	Dl4jMlp	96.6667	100	98.8889
Stacking	Dl4jMlp	50	57.5	44.4444
ThresholdSelector	Dl4jMlp	54	42.5	55.5556
Vote	Dl4jMlp	50		
WeightedInstancesHandlerW	Dl4jMlp	50	42.5	44.4444

La función de los Algoritmos Metaclasificadores consiste en mejorar el aprendizaje de los clasificadores convirtiéndolos en aprendices más poderosos. En los resultados,

los mejores clasificadores fueron tres, cada uno con un porcentaje perfecto de clasificación, a continuación se hace una descripción resumida del funcionamiento de estos algoritmos [6].

AttributeSelectedClassifier: Se encarga de seleccionar atributos reduciendo la dimensionalidad de los datos para eliminar datos redundantes y aplicar un clasificador al resultado del conjunto de datos obtenido; esto permite el uso de un método de selección de atributos y un algoritmo de aprendizaje específico como parte de un esquema de clasificación, asegurando que el conjunto de atributos elegidos es seleccionado solamente basado en los datos de entrenamiento y evaluado en los datos de prueba.

FilteredClassifier: Este algoritmo se encarga de aplicar un filtro al conjunto de datos antes de aplicar un algoritmo de aprendizaje, filtrando los datos de entrenamiento sin filtrar los de prueba además emplea un clasificador el cual usa los datos previamente filtrados con la finalidad de reducir el conjunto de datos a solamente los de mayor importancia. Este algoritmo se utiliza para la evaluación equitativa de discretización supervisada esto lo realiza solo en los datos de prueba usando intervalos de discretización computados para los datos de entrenamiento que en la práctica equivale al procesamiento de datos completamente frescos.

RandomSubSpace: Construye un ensamble de clasificadores, cada uno entrenado usando un subconjunto seleccionado aleatoriamente de atributos de entrada. Aparte del número de iteraciones y la inserción aleatoria usada, provee un parámetro de control del tamaño del subconjunto de atributos, reduciendo el tamaño de los conjuntos de datos para una eficiente clasificación.

El algoritmo *Dl4jMlp* se expresa como ecuación matemática de una forma razonablemente natural, este algoritmo tiene como base la función de regresión y clasificación mediante perceptrón multicapa, este algoritmo se explica de manera resumida a continuación [6].

Perceptrón Multicapa también llamado red neuronal de retroalimentación, es un modelo de aprendizaje profundo por excelencia, el objetivo de este algoritmo es aproximar una función f^*. Un ejemplo claro para un clasificador $y=f^*(x)$ mapea un valor de x a una categoría y. Una red de retroalimentación define un mapeo $y=f(x,\theta)$ y aprende los valores del parámetro θ el cual resulta en la mejor función de aproximación. En este algoritmo la información fluye a través de la función que ha sido evaluada desde x a través del cómputo intermediario para definir f hasta llegar a la salida y [5]. Esta clase de red neuronal se compone de una capa de entrada, capas ocultas intermedias y una capa de salida, en donde cada conexión tiene una ponderación (peso) y cada nodo realiza una suma ponderada de sus entradas, umbralizando el resultado; no contiene ningún ciclo y la salida de la red depende solamente de las instancias actuales de entrada; su ventaja es la capacidad de aprender a ignorar atributos irrelevantes y representa límites no lineares de decisión. El proceso de operación de este algoritmo se muestra a continuación [3].

4. Experimentos y resultados

El análisis realizado aplicando todos los Algoritmos Metaclasificadores en combinación con el algoritmo Dl4jMlp, arrojó los mejores resultados con los criterios

Tabla 4. Matriz de confusión para los tres pares de algoritmos.

a	b	Clasificadas como
23	0	a=Saludables
0	17	b=Enfermas

de división de 2/3 - 1/3 logrados por tres metaclasificadores, *AttributeSelectedClassifier*, *FilteredClassifier* y *RandomSubSpace*, todos con una precisión de clasificación del 100%. Las combinaciones completas y sus resultados se muestran en la Tabla 3.

Los experimentos se realizaron en un ordenador con: OS Windows 8, Intel(R) Core(TM) i3-2330M CPU 2.20 GHz, RAM 8.00 GB, HD 320 GB, sistema operativo de 64 bits y procesador x64. Además, para el procesamiento de imágenes y la ejecución de los algoritmos descritos en esta sección, utilizamos el lenguaje de programación MATLAB® y el software de minería de datos WEKA v.3.8 respectivamente.

5. Análisis y discusión

El porcentaje alcanzado de instancias clasificadas correctamente por los tres algoritmos metaclasificadores con Dl4jMlp se debe principalmente al tratamiento de datos realizado por cada clasificador, como se explica a continuación:

- El algoritmo *AttributeSelectedClassifier* realiza una reducción de la dimensionalidad de los datos de entrenamiento al seleccionar los atributos antes de pasarlos al clasificador.
- El algoritmo clasificador *FilteredClassifier* pasa previamente los datos por un filtro arbitrario antes de ser clasificados.
- El algoritmo *RandomSubSpace* construye una clasificación basada en un árbol de decisión la cual logra mantener una alta precisión en el entrenamiento de datos, mejorando la precisión cuanto más crece la complejidad de los datos.
- El clasificador *Dl4jMlp* realiza clasificaciones y regresiones ignorando atributos irrelevantes.

Los resultados obtenidos de las tres combinaciones, se representaron en una matriz de confusión para cada par respectivo cabe aclarar que en los tres casos se obtuvo la misma matriz de confusión, esta se muestra en Tabla 4.

En la matriz de confusión del clasificador *AttributeSelectedClassifier+Dl4jMlp*, *FilteredClassifie+Dl4jMlp*, así como *RandomSpace+Dl4jMlp*, se puede apreciar que a pesar de que hubo una reducción en el número de instancias para clasificar no se obtuvieron instancias clasificadas erróneamente. Es decir, los valores 23 y 17 son los casos de hojas saludables y enfermas clasificadas correctamente, mientras que las casillas con el valor 0 indican que debido a la reducción de datos hay una reducción completa de la confusión en la clasificación de dichas instancias.

6. Conclusiones y trabajo futuro

Al combinar *AttributeSelectedClassifier+Dl4jMlp*, *FilteredClassifie+Dl4jMlp y RandomSpace+Dl4jMlp*, se obtuvo un porcentaje perfecto de clasificación en comparación con los métodos aplicados en las investigaciones del estado del arte en donde el porcentaje mejor obtenido fue de 99.35%. La obtención del 100% de instancias clasificadas correctamente se debió a la combinación de diferentes factores, desde el tamaño ligero de la base de datos, el procesamiento digital de las imágenes, la combinación de algoritmos de aprendizaje automático, el tratamiento de clasificación binaria y sobre todo la reducción de dimensionalidad, además del filtrado de datos aplicado a la base de datos por los clasificadores combinados.

Para la continuación futura de esta investigación se propone realizar las siguientes acciones:

- Aplicar la combinación de algoritmos obtenida de este estudio a una clasificación multiclase, agregando más enfermedades, además de características geométricas y de Haralick.
- Realizar una selección de mejores atributos, para encontrar la combinación ideal de algoritmos para realizar una comparación de resultados con todos los atributos y con los mejores.
- Implementar las técnicas usadas en esta investigación en una herramienta de detección para mejorar la precisión en la identificación de enfermedades.

Referencias

1. Camponez, M., Boasa, P., Miloria, D., Ferreiraa, E., Silva, M., Machadoc, M., Belleted, B., Fernandes-Da Silva, M.: Infrared spectroscopy: A potential tool in huanglongbing and citrus variegated chlorosis diagnosis. Talanta 91(1), pp. 1–6 (2012)
2. Cevallos, J., García-Torres, R., Etxeberria, E., Reyes-De Corcuera, J.: Gc-ms analysis of headspace and liquid extracts for metabolomic diferentiation of citrus huanglongbing and zinc defciency in leaves of Valencia sweet orange from commercial groves. Phytochemical Analysis, pp. 236–246 (2010)
3. Collobert, R., Bengio, S.: Links between perceptrons, MLPs and SVMs. In: Proceedings of the twenty-first international conference on Machine learning 23 (ACM) (2004)
4. Citrus variegated chlorosis (CVC): http://idtools.org/id/citrus/diseases/factsheet .php?name=Citrus+variegated+chlorosis+%28CVC%29 (2018)
5. Goodfeloow, I., Bengio, Y., Courville, A.: Deep Learning, pp. 167 (2016)
6. Witten, I., Frank, E., Hall, M.: Data Mining, pp. 234–582, Morgan Kaufmann (2011)
7. Kadir, A., Nugroho, L., Susanto, A., Santosa, P.: Leaf classification using shape, color, and texture features. International Journal of Computer Trends and Technology (2011)
8. Citrus variegated chlorosis (xylella fastidiosa): http://www.padil.gov.au/pests-and-diseases/pest/main/136652 (2018)
9. Mohanty, S., Hughes, D., Salathe, M.: Using deep learning for image-based plant disease detection. 4 Frontiers in Plant Science (2010)
10. Pourreza, A., Lee, W., Etxeberria, E., Banerjee, A.: An evaluation of a vision-based sensor performance in huanglongbing disease identfication. Biosystems Engineering, 130(1), pp. 13–22 (2015)

11. Pydipati, R., Burks, T., Lee, W.: Identification of citrus disease using color texture features and discriminant analysis. Computers and Electronics in Agriculture, 52(2), pp. 49–59 (2006)
12. Sankaran, S., Ehsani, R., Etxeberria, E.: Mid-infrared spectroscopy for detection of huanglongbing (greening) in citrus leaves. Talanta, 83(3), pp. 574–581 (2010)
13. Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA): Se consolida México como quinto productor mundial de naranja (2018)
14. Servicio Nacional de Sanidad Inocuidad y Calidad Agroalimentaria: Plagas reglamentadas de los cítricos (X. fastidiosa subsp. pauca, X. citri subsp. citri, G. citricarpa y CLV-c) (2018)
15. Da-Wen, S.: Computer Vision Technology for Food Quality Evaluation. Nikki Levy (2016)

ISSN 1870-4069

Estudio del desbalance de clases en bases de datos de microarrays de expresión genética mediante técnicas de Deep Learning

H. Cruz-Reyes[1], A. Reyes-Nava[2], E. Rendón-Lara[1], R. Alejo[1]

[1] Instituto Tecnológico de Toluca,
México

[2] Universidad Autónoma del Estado de México
México

{lugotache,adriananava0}@gmail.com, ralejoll@hotmail.com, erendonl@toluca.tecnm.mx

Resumen. Hoy en día, se ha incrementado significativamente el interés por desarrollar aplicaciones de Deep Learning enfocadas a atender problemas de aprendizaje automático, reconocimiento de patrones y minería de datos en el contexto del Big Data. Esto se debe principalmente a la alta capacidad de procesamiento y buen rendimiento en la selección de características, predicción y tareas de clasificación que presentan los algoritmos de Deep Learning, además, los algoritmos con este enfoque han mostrado tener un buen desempeño tratando con bases de datos utilizadas en el reconocimiento de imágenes y lenguaje natural, las cuales se caracterizan por ser bases de datos de muy alta dimensionalidad y una notable cantidad de datos o muestras. No obstante, al día de hoy uno de los principales retos abarca la clasificación de bases de datos de alta dimensionalidad, con muy pocas muestras y un alto desbalance de clases, un ejemplo de ello son las bases de datos de microarrays de expresión genética. La principal aportación de este trabajo es el estudio de bases de datos de expresión genética, utilizando los métodos convencionales de Sub y Sobre Muestreo para el balance de clases tales como RUS, ROS y SMOTE, las bases de datos fueron modificadas aplicando un incremento en su desbalance y en otro caso generando ruido artificial para realizar un estudio más detallado.

Palabras clave: Deep learning, perceptrón multicapa, microarrays de expresión genética.

Study of the Imbalance of Classes in Gene Expression Microarray Databases using Deep Learning

Abstract. Today, there has been a significant increase in the interest in developing Deep Learning applications focused on identify problems of automatic learning, pattern recognition and data mining in the Big Data context. This is mainly due to the high processing capacity and good performance in the

selec-tion of characteristics, prediction and classification tasks presented by the Deep Learning algorithms, in addition, the algorithms with this approach have shown to have a good performance with databases used in the recognition of images and natural language, which are characterized as very high dimensionality data-bases and a considerable amount of data or samples. However, today one of the main challenges involves the classification of high dimensionality databases, with very few samples and a high level of class imbalance, an example of which are the microarray databases of genetic expression. The main contribution of this work is the study of databases of genetic expression, using the conventional Over and Under Sampling methods for the balance of classes such as RUS, ROS and SMOTE, in addition the databases were modified applying an increase in its imbalance and generating artificial noise to perform a detailed study.

Keywords: Deep learning, multilayer perceptron, gene expression microarrays.

1. Introducción

Recientemente se ha popularizado el uso de Redes Neuronales Artificiales (ANN por sus siglas en inglés) para llevar a cabo tareas de clasificación enfocadas a problemas reales, una de las redes más usada es el Perceptrón Multicapa (MLP por sus siglas en inglés) entrenado con el método Back-Propagation [1,2], es una de las redes más populares debido a las ventajas que presenta, como son: rapidez, paralelismo inherente, no requiere conocimiento a priori de la distribución estadística de los datos y la tolerancia a fallos.

Tradicionalmente las ANN están conformadas de tres capas (capa de entrada, capa oculta y capa de salida), sin embargo, hoy en día cuando una ANN está compuesta por más de tres capas es conocida como una ANN Deep Learning (DL) [3]. La arquitectura de ANN-DL más representativa es el MLP con varias capas ocultas [3-5]. Las principales ventajas de este tipo de arquitectura son tres: alto desempeño, robustez al sobre-entrenamiento y alta capacidad de procesamiento.

Hoy en día los algoritmos de DL muestran buenos resultados al ser utilizados en la solución de distintos problemas [6-11], los cuales tienen características similares como la gran cantidad de datos y alta dimensionalidad. Sin embargo, uno de los principales retos que actualmente surge es la clasificación de bases de datos de alta dimensionalidad, con muy pocas muestras y alto desbalance de clases. Las bases de datos biomedicas de microarrays de expresión genética tienen las características antes mencionadas, a menudo presentan problemas de desbalance de clases, tienen pocas muestras y alta dimensionalidad. El problema del desbalance de clases surge cuando el conjunto de muestras perteneciente a una clase es mucho mayor al conjunto de muestras de la otra u otras clases [12], este problema se ha identificado como uno de los principales retos de los algoritmos aplicados en el contexto del Big Data [7,10, 11, 13].

Los clasificadores convencionales como el MLP fueron diseñados para trabajar con bases de datos balanceadas, es decir, que el número de muestras sea la misma para cada clase. Por esta razón, cuando se trabaja con una base de datos desbalanceada no se logran resultados óptimos, debido al alto porcentaje de muestras mal clasificadas en las clases menos representadas o minoritarias [14].

Aunado a ello, el método Back-Propagation (utilizado para entrenar el MLP) también se ve afectado por el desbalance, ya que ralentiza la convergencia de la red [15], lo que potencializa una de las desventajas de este método de entrenamiento.

El desbalance de clases ha sido ampliamente estudiando en problemas de dos clases [16], sin embargo, los problemas de múltiples clases [17, 18] y DL han sido poco abordados. Los trabajos enfocados al desbalance en múltiples clases comúnmente utilizan costos asociados a las diferentes clases en la etapa de entrenamiento, a pesar de ello, este enfoque solo es adecuado en el entrenamiento con Back-Propagation cuando el entrenamiento es por lotes o "batch mode" [19]. En la resolución de problemas reales el entrenamiento por lotes es menos usado que el entrenamiento estocástico, ya que este último es usualmente más rápido, alcanza mejores soluciones y puede ser usado para identificar cambios al pasar las muestras por el MLP.

Tradicionalmente los métodos utilizados para atacar el desbalance de clases se basan en duplicar o eliminar muestras hasta alcanzar un equilibrio en el número de muestras por clase, por ejemplo, Random Over-Sampling (ROS) y Random Under-Sampling (RUS) [20]. Uno de los métodos comúnmente usados es Synthetic Minority Over-sampling Technique (SMOTE), propuesto por Chawla et al. [21], esta técnica genera nuevas muestras sintéticas interpoladas en las muestras de la clase minoritaria. Este método ha servido como base de otros métodos de muestreo como Bordeline-SMOTE, Adaptative Synthetic Samplig (ADASYN), SMOTE editing nearest neighbor, entre otros [14, 16].

Por otro lado en las técnicas de sub-muestreo (under-sampling), RUS ha sido reportado como una de las técnicas más efectivas [22]. En estas técnicas se han propuesto métodos caracterizados por incluir un mecanismo heurístico en su funcionamiento, el cual tiene como objetivo eliminar o cambiar las etiquetas de las muestras, ya sean ruido, atípicos o redundantes [23], como ejemplo se puede mencionar a los métodos Neighborhood Clearing Rules y One-sided Selection.

Actualmente ha surgido el interés por desarrollar métodos de muestreo dinámico en el contexto de los MLP, donde el objetivo es usar el número apropiado de muestras o aumentar el tamaño de la clase minoritaria al momento de entrenar el perceptrón. Como ejemplo está el método SNOBALL [24], donde la clase mayoritaria se incrementa gradualmente, otro ejemplo es DyS [25], el cual atenúa el desbalance por medio de un mecanismo de sobre-muestreo e identificación de las muestras difíciles de aprender, es decir, pone más atención a las muestras más difíciles de clasificar.

En general el desbalance de clases afecta negativamente al desempeño de los algoritmos de aprendizaje automático (ML por sus siglas en inglés). El desbalance también está presente el contexto el Big Data donde el uso de Maquinas de Boltzman, redes de creencia, redes convolucionales y en general redes con un enfoque DL han mostrado buenos resultados [10,11], mientras que los algoritmos de aprendizaje automático han mostrado notables deficiencias en su desempeño.

En el contexto de Big Data son pocas las propuestas para atacar el problema del desbalance, actualmente se están adaptando los algoritmos de aprendizaje automático para desempeñarse en este enfoque. El objetivo de este trabajo es identificar el comportamiento del clasificador utilizando bases de datos de microarrays de expresión genética las cuales cuentan con pocas muestras, alta dimensionalidad y en algunos casos desbalance de clases.

2. Trabajos relacionados

Hoy en día en áreas como ML, el reconocimiento de patrones (PR por sus siglas en inglés), y la minería de datos (DM por sus siglas en inglés) se ha demostrado que el desbalance de clases es un problema crucial que afecta la eficiencia del algoritmo [21, 26], incluso en Big Data se considera como uno de los principales retos [11, 12, 14,15]. El desbalance se presenta cuando una o más clases tienen un menor número de muestras (clase minoritaria) que las demás (clase o clases mayoritarias) y afecta directamente a la capacidad de generalización del clasificador ya que este asume que los conjuntos de muestras por clase están balanceados.

El desbalance de clases surge cuando una o más clases se encuentran menos representadas en su número de muestras, en comparación con el número de muestras de otras clases. En las redes neuronales artificiales, en particular, en el MLP, el desbalance acentúa las debilidades de este clasificador, sobre todo cuando el entrenamiento se hace mediante el algoritmo del Back-Propagation [15].

Usualmente las bases de datos pueden estar agrupadas en dos clases o más, el dominio de dos clases ha sido ampliamente estudiado, sin embargo, trabajar con múltiples clases sigue siendo un reto del aprendizaje automático, la minería de datos y el reconocimiento de patrones [17, 18]. Actualmente, el problema del desbalance de clases se ha abordado de muchas maneras y enfoques diferentes, los más estudiados han sido los métodos de muestreo enfocados al desbalance entre clases, estos métodos suelen ser eficaces y son independientes del clasificador [22].

Los métodos de muestreo pueden ser simples y claros como el sobre o sub muestreo aleatorio (ROS o RUS) [14]. El primero replica aleatoriamente muestras existentes en la clase minoritaria, lo que en algunos casos podría dar pie a que ocurra un sobre ajuste [21], y el segundo quita un determinado número de muestras permitiendo así un balance entre el número de elementos por clases, aunque, en algunos escenarios, sería inapropiado debido a la enorme pérdida de información en la base de datos. Por lo tanto, se han desarrollado otros métodos de muestreo "inteligentes" que incluyen un mecanismo heurístico, como SMOTE, el cual crea muestras artificiales de la clase minoritaria mediante la interpolación de muestras existentes cerca de ellas [21] y de esta forma evitar la sobre especialización.

Una técnica propuesta para optimizar las deficiencias de las técnicas de muestreo como ROS o SMOTE es Borderline-SMOTE [27], la cual selecciona muestras de la clase minoritaria que están en el límite, realizando sólo SMOTE en esas muestras. Por otro lado el muestreo sintético adaptativo (ADASYN) es también una extensión de SMOTE, el cual crea en el límite de la región más muestras entre las dos clases que en el interior de la clase minoritaria. SMOTE Editing Nearest Neighbor (ENN) consiste en aplicar SMOTE y, a continuación, la regla ENN. Safe-Level-SMOTE genera muestras sintéticas de la clase minoritaria situadas más cerca del mayor nivel de seguridad, entonces todas las muestras sintéticas sólo se generan en regiones seguras [28]. SMOTE + Tomek Links (TL) es la combinación de SMOTE y TL [20], Neighborhood Cleaning Rule usa la regla ENN, pero sólo elimina las muestras de la clase mayoritaria. Condensed Nearest Neighbor rule (CNN) y One-sided Selection eliminan las muestras redundantes, pero esta última usa TL.

Se han presentado métodos más sofisticados para tratar el problema de desbalance de múltiples clases. Uno de ellos es el costo sensitivo (CS), el cual, es de los temas más relevantes en la investigación del aprendizaje de automático y es una buena solución para el problema de desbalance de clases [13]. El CS utiliza los costos asociados con la clasificación errónea de las muestras, emplea varias matrices de costos que definen los costos de clasificación errónea de cualquier muestra de datos [21]. Sin embargo, en estos métodos, el costo de clasificación errónea debe ser conocido de antemano, pero en un problema de clasificación real, el costo de clasificación errónea es a menudo desconocido. Zhi-Hua y Xu-Ying [29] proporcionan un marco unificado para el uso de CS para abordar el desbalance de clases.

Recientemente, se han propuesto métodos de muestreo dinámico para resolver el problema del desbalance de múltiples clases, los cuales establecen automáticamente la tasa de muestreo, por ejemplo, Fernández-Navarro et al. [30] combinan métodos a nivel de datos utilizando algoritmos genéticos para obtener la mejor relación de sobremuestreo. Alejo et al. [31] usa el error cuadrático medio con este propósito. Chawla et al. [32] proponen un paradigma Wrapper que descubre automáticamente la cantidad de sub-muestreo y tasa de sobre-muestreo para un conjunto de datos basado en la optimización de las funciones de evaluación. En [25] se propone un nuevo algoritmo para determinar el nivel de equilibrio de clases, y además incluyen un mecanismo de selección de patrones de entrenamiento difíciles de aprender, con el propósito de mejorar la capacidad de generalización del MLP entrenado con el algoritmo Back-Propagation. Un método más antiguo de muestreo dinámico (SNOWBALL) es propuesto por Wang y Jean [24] para entrenar redes del tipo MLP con datos desbalanceados, básicamente este método repite el entrenamiento de las muestras de las clases minoritarias hasta que el clasificador las identifica adecuadamente. Por otra parte, trabajos recientes muestran interés en encontrar las mejores muestras para construir el clasificador, por ejemplo eliminando las muestras que son difíciles de aprender o cercanas a la frontera de decisión, ya que podrían ser muestras con "ruido" o "solapadas".

El desbalance de clases es un problema muy frecuente en tareas de visión por computadora, por ejemplo, en el dominio de las redes convolucionales se han usado técnicas de re-muestreo o aprendizaje sensible a costos para hacer frente a este problema.

3. Microarrays de expresión genética

Los microarrays de expresión genética son conjuntos de datos de perfiles de expresión del mundo real que se utilizan en varios tipos de investigación del cáncer. Las bases de datos de este estudio serán Prostate, Ovarian y Breast, las cuales tienen muy pocas muestras, son de alta dimensionalidad y están agrupadas en dos clases, cabe mencionar que el número de elementos por clase no es balanceado. Estas se pueden obtener del Repositorio del conjunto de datos biomédicos de Kent Ridge (http://leo.ugr.es/elvira/DBCRepository/).

Las bases de datos de microarrays de expresión genética generalmente cuentan con un limitado número muestras, tienen una alta dimensionalidad y en algunos casos presentan desbalance de clases.

4. Perceptrón Multicapa Deep Learning (MLP DL)

Un MLP consiste en una red que contiene una capa de entrada, una de salida y una o más capas ocultas [5]. En la capa de entrada se ingresan los datos que serán analizados, estos pasan a la primera capa oculta, los resultados de esta capa pasan a la segunda capa oculta (si es el caso), y al final pasan a la capa de salida; al paso de cada una de las capas ocultas se van asignando pesos sinápticos. El número de nodos de cada capa oculta puede variar [3, 4].

La principal diferencia entre las arquitecturas de aprendizaje automático y aprendizaje profundo es el número de capas ocultas, convencionalmente en el aprendizaje automático las arquitecturas están compuestas de una capa de entrada, una oculta y una de salida, a diferencia del aprendizaje profundo donde se componen de más de 3 capas, debido a que tienen más de una capa oculta. Cuando la red tiene más de 3 capas en su arquitectura se clasifica como aprendizaje profundo [3].

Muchos de los avances del aprendizaje profundo dependen en gran medida de la tecnología que se usa para implementarse, algunas de las librerías más usadas para este enfoque son Theano, PyLearn2, Caffe, Tensorflow y la plataforma de trabajo Apache Spark.

En este trabajo se utilizó una red neuronal de enfoque DL, con una configuración muy similar para todas las bases de datos, donde la única diferencia es la capa de entrada debido a que las bases de datos no tienen el mismo número de atributos. Para ejecutar la red se buscó un entorno que pudiera realizar el procesamiento en paralelo, por lo que se utilizó la plataforma Spark, ya que ofrece un buen desempeño al procesar datos de gran tamaño y alta dimensionalidad, optimizando el tiempo de ejecución y generando resultados confiables.

5. Diseño experimental

El propósito de este trabajo es estudiar el comportamiento del clasificador MLP DL utilizando bases de datos de alta dimensionalidad, pocos patrones y alto desbalance de clases, como lo son las bases de datos de microarrays de expresión genética. Para este trabajo, se utilizaron bases de datos públicas sobre datos de cáncer disponibles en el repositorio Kent Ridge Biomedical Data Set. Los detalles de las bases de datos pueden ser consultados en la Tabla 1.

En la Tabla 1 se puede observar que además de las bases mencionadas se describen dos bases de datos más por cada una, las cuales se pueden identificar porque el nombre de la base de datos incluye los términos “–Ruido” y “–Disminuido”.

Tabla 1. Descripción de las bases de datos.

Base de Datos	Características	No. de Ejemplos	Clase 1	Clase 0
Ovarian	15154	253	162	91
Ovarian-Ruido	15154	253	172	81
Ovarian-Disminuido	15154	243	162	81
Prostate	12600	136	77	59
Prostate -Ruido	12600	136	87	49
Prostate -Disminuido	12600	126	77	49
Breast	24481	97	46	51
Breast -Ruido	24481	97	36	61
Breast -Disminuido	24481	87	36	51

Estas bases de datos se generaron para obtener un desbalance de clases aún más significativo. Para las bases identificadas como "Disminuido" se sustrajeron aleatoriamente 10 muestras, logrando que la clase minoritaria disminuyera su tamaño y así tener un desbalance más relevante. En las bases identificadas como "Ruido" se buscó la manera de generar Ruido Artificial, lo cual se logró seleccionando aleatoriamente diez muestras de la clase minoritaria para cambiar su clase, de este modo disminuir la clase minoritaria y aumentar la clase mayoritaria.

La configuración utilizada en la red neuronal es la establecida por defecto en el MLP de Spark, no obstante, se ajustaron algunos parámetros tales como la capa de entrada con 12600 nodos para la base de datos Prostate, 15154 para Ovarian y 24481 para Breast. Dos capas ocultas (la primera con 90 nodos, la segunda con 80) y finalmente una capa de salida de dos nodos. La configuración usada en las capas ocultas y la capa de salida fue exactamente la misma para todas las bases de datos.

Se aplicó el método Hold-Out [33] para la segmentación de los conjuntos de entrenamiento y test, quedando 60 y 40 por ciento respectivamente, se repitió el proceso de división 10 veces de forma aleatoria, donde, cada conjunto contenía diferentes muestras, es decir, las muestras contenidas en el conjunto de entrenamiento no se encontraban en el de test y viceversa.

Para evaluar la eficacia del modelo se usó el área bajo la curva (AUC por sus siglas en inglés), la cual es una medida ampliamente utilizada en investigaciones sobre desbalance de clases. Para la ejecución se optó por procesar diez veces cada una de las bases de datos y finalmente obtener el promedio de la métrica.

6. Resultados

La Tabla 2 muestra los resultados obtenidos por el MLP DL utilizando AUC para medir la eficacia del clasificador. Los resultados se muestran por cada una de las técnicas de muestreo usadas en todas las bases de datos, los valores en negritas señalan la técnica de muestreo (sub o sobre muestreo) con mejor desempeño en cada una de las bases de datos.

Tabla 2. Resultados obtenidos por el MLP DL utilizando la métrica AUC.

Base de Datos	ORIG	ROS	RUS	SMOTE
Ovarian	0.9634	**0.9792**	0.9624	0.9782
Ovarian-Ruido	0.8859	**0.9418**	0.8930	0.9361
Ovarian-Disminuido	0.9819	**0.9902**	0.9488	0.9896
Prostate	0.8347	0.8614	0.8380	**0.8956**
Prostate -Ruido	0.7121	0.8226	0.7743	**0.8264**
Prostate – Disminuido	0.8533	**0.8971**	0.7982	0.8783
Breast	0.5835	0.6609	0.5612	**0.6690**
Breast -Ruido	0.5851	0.6691	0.6704	**0.6980**
Breast -Disminuido	0.5791	**0.7124**	0.5801	0.6996

Para garantizar resultados fiables, se presenta el puntaje promedio de 10 ejecuciones en cada uno de los experimentos, se utilizaron cifras con cuatro decimales para dar una mayor información de los resultados.

Las bases de datos Ovarian, Prostate y Breast son las bases de datos originales, y en general tienen una buena clasificación, sin embargo, "Breast" presenta una exactitud de 0.5835 que es un nivel de eficacia muy bajo, cabe señalar que es la base con el menor número de muestras y mayor número de atributos, además de presentar el desbalance más pequeño con una diferencia de cinco muestras entre sus clases.

Al aplicar técnicas de muestreo a las bases antes mencionadas, se obtuvieron resultados similares a los de las bases de datos originales cuando se utilizó RUS, por otro lado, al aplicar las técnicas de sobre muestreo ROS y SMOTE se observó una notable mejora en la eficiencia del clasificador. Siendo ROS el que obtuvo la eficiencia más alta en Ovarian, mientras que SMOTE fue el mejor en Prostate y Breast, a pesar de ello, no hubo una diferencia significativa en la eficiencia del clasificador al utilizar ROS y SMOTE.

En las bases de datos Ovarian-Disminuido, Prostate-Disminuido y Breast-Disminuido, donde se incrementó el desbalance de clases sustrayendo diez muestras al conjunto de la clase minoritaria, se observó que elevaron su nivel de eficacia en comparación con las bases de datos originales, RUS volvió a ser el que tiene el más bajo nivel, incluso por debajo de las bases de datos originales. Cabe señalar que la base de datos Breast-Disminuido tuvo un nivel de eficacia considerablemente mayor que el de la base Breast.

En el caso de Ovarian-Ruido, Prostate-Ruido y Breast-Ruido, que son las bases de datos a las que se les aplicó ruido artificial, se observó que todos los métodos de muestreo alcanzaron mejor nivel que la base de datos original. Los resultados de RUS son muy cercanos a los de la base original, por otro lado SMOTE obtuvo la eficiencia más alta en Prostate-Ruido y Breast-Ruido mientras que ROS en Ovarian-Ruido, a pesar de ello, no hubo una diferencia significante en la eficiencia del clasificador al utilizar ROS y SMOTE.

Se puede observar que utilizando técnicas de sobre-muestreo, se obtienen mejores resultados en comparación a las bases de datos originales. En el caso del sub-muestreo aleatorio (RUS) se observa que el desempeño del clasificador es muy similar al de la base de datos original incluso, en algunos casos ligeramente menor, por lo tanto utilizando técnicas tradicionales de sobre muestreo tales como ROS y SMOTE el clasificador podrá obtener mejores resultados al trabajar con bases de datos de alta dimensionalidad, de pocas muestras y alto desbalance de clases.

7. Conclusiones

El desbalance de clases ha sido reconocido como uno de los principales retos a la hora de entrenar clasificadores supervisados, esto debido la mayoría de ellos fueron diseñados para trabajo con bases de datos relativamente balanceadas. Actualmente, muchas de las bases de datos que se están generando presentan problemas de des balance de clases, por ejemplo, las bases de datos de microarray de expresión genética, aunado a esto, características como alta dimensionalidad y escasas de muestras o patrones de entrenamiento caracterizan a estas bases de datos.

El deep learning, ha sido una excelente alternativa para tratar con bases de datos de gran tamaño y dimensionalidad, no obstante, ha mostrado notables deficiencias al trabajar con bases de datos desbalanceadas. En este trabajo se estudió la efectividad de métodos tradicionales para tratar el de balance de clases, en bases de datos de microarrays de expresión genética las cuales se caracterizan por tener pocas muestras o patrones de entrenamiento y un número excesivo de atributos o características.

Resultados presentados este trabajo muestran la efectividad de las técnicas de muestreo ROS y SMOTE para tratar el desbalace de clases en la clasificación de bases de datos de microarrays de expresión genética, sin embargo, se observa una tendencia en SMOTE a producir mejores resultados. Por otro lado, se muestra que es prohibitivo eliminar muestras en este tipo de bases de datos con la finalidad de tratar el desbalance de clases.

Es indudable se requiere profundizar en el tema no solo por la importancia del mismo sino por su relación con otras áreas del conocimiento como la biomedicina y el Big Data. Para trabajos futuros se tiene contemplado estudiar otros algoritmos clásicos para el tratamiento del desbalance de clases y en su momento proponer un nuevo método que ayude a superar las deficiencias de los métodos existentes en el estado del arte.

Referencias

1. Linderman, M., Liu, J., Qi, J., An, L., Ouyang, Z., Yang, J., Tan, Y.: Using artificial neural networks to map the spatial distribution of understorey bamboo from remote sensing data. International Journal of Remote Sensing, 25(9), pp. 1685–1700 (2004)
2. Pal, M.: Extreme learning machine for land cover classification. International Journal of Remote Sensing, 30(14), pp. 3835–3841 (2008)
3. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

4. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Networks, 61, pp. 85–117 (2015)
5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, 521, pp. 436—444 (2015)
6. Nene, S.: Deep learning for natural languaje processing. International Research Journal of Engineering Technology, 4, pp. 930–933 (2017)
7. Inur, A., Ritahani, A., Ahmad, A.: Convolutional Neural Networks and Deep Belief Networks for Analysing Imbalanced Class Issue in Handwritten Dataset. International Journal on Advanced Science Engineering Information Technology, 7, pp. 2302–2307 (2017)
8. Yan, Y., Chen, M., Shyu, M.L., Chen, S.S.: Deep Learning for Imbalanced Multimedia Data Classification. IEEE International Symposium on Multimedia, pp. 483–488 (2015)
9. Mahsereci, E., Ibrikci, T.: Discriminative deep belief networks for microarray based cancer classification. Biomedical Research, 28(3), pp. 1016–1024 (2017)
10. Heureux, A., Grolinger, K., Elyamany, H., Capretz, M.: Machine Learning With Big Data: Challenges and Approaches, In: IEEE Access, 5, pp. 7776–7797 (2017)
11. Zhou, L., Pan, S., Wang, J., Vasilakos, A.V.: Machine learning on big data: Opportunities and challenges. Neurocomputing, 237, pp. 350–361 (2017)
12. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recognition, 40(1), pp. 4–18 (2007)
13. Salman, H. K.: Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. IEEE Transactions on Neural Networks and Learning Systems (2017)
14. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf. Sci., 250, pp. 113–141 (2013)
15. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. IEEE Trans Neural Netw, 4, pp. 962–969 (1993)
16. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284 (2009)
17. Wang, S., Yao, X.: Multiclass imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics, 42(4), pp. 1119–1130 (2012)
18. Fernández, A., López, V., Galar, M., De Jesus, M.J., Herrera, F.: Analyzing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowledge-Based Systems, 42(8), pp. 97–110 (2013)
19. Kretzschmar, R., Karayiannis, N.B., Eggimann, F.: Feedforward neural network models for handling class overlap and class imbalance. Int. J. Neural Syst., 15(5), pp. 323–338 (2005)
20. Nguyen, A.B., Phung, S.L.: A supervised learning approach for imbalanced data sets. In: Proc. of the 19th International Conference on Pattern Recognition, pp. 1–4 (2008)
21. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, pp. 321–357 (2002)
22. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intell. Data Anal, 6(5), pp. 429–449 (2002)
23. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cyber., 2, pp. 408–420 (1972)
24. Wang, J., Jean, J.: Resolving multifont character confusion with neural networks. Pattern Recognit., 26, pp. 175–187 (1993)
25. Lin, M., Tang, K., Yao, X.: Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Trans. Neural Netw. Learning Syst., 24(4), pp. 647–660 (2013)

26. Debowski, B., Areibi, S., Gréwal, G., Tempelman, J.: A dynamic sampling framework for multi-class imbalanced data. In: ICMLA, (2), pp. 113–118 (2012)
27. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: Proceedings of the 2005 International Conference on Advances in Intelligent Computing, I. (ICIC'05), pp. 878–887 (2005)
28. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Proceedings of the 13th Pacific-Asia Conference (PAKDD 2009), 5476, pp. 475–482 (2009)
29. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. Computational Intelligence, 26(3), pp. 232–257 (2010)
30. Fernández-Navarro, F., Hervás-Martínez, C., Antonio Gutiérrez, P.: A dynamic over-sampling procedure based on sensitivity for multi-class problems. Pattern Recogn., 44(8), pp. 1821–1833 (2011)
31. Alejo, R., García, V., Pacheco-Sánchez, J.H.: An efficient over-sampling approach based on mean square error back-propagation for dealing with the multi-class imbalance problem. Neural Processing Letters, 42(3), pp. 603–617 (2014)
32. Chawla, N.V., Cieslak, D.A., Hall, L.O., Joshi, A.: Automatically countering imbalance and its empirical relationship to cost. Data Min. Knowl. Discov., 17, pp. 225–252 (2008)
33. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley-Interscience Publication (2001)

ISSN 1870-4069

Análsis, diseño y desarrollo de un sistema de recomendación basado en datos restauranteros de TripAdvisor y Foursquare

Saúl Pérez, Mary Carmen Cuecuecha, José Federico Ramírez,
José Crispín Hernández

Instituto Tecnológico de Apizaco, División de Estudios de Posgrado e Investigación,
Apizaco, Tlaxcala,
México
{saul.perez.tirzo, MaryCarmenCuemu, federico.ramirezc}@gmail.com,
josechh@yahoo.com

Resumen. Debido a la inmensa cantidad de información disponible en Internet, provoca que los usuarios se sientan abrumados con tanta información, haciendo difícil el proceso de búsqueda de productos y/o servicios que se adecuen a los gustos y necesidades de cada usuario. Por esta razón el desarrollo de herramientas inteligentes se ha vuelto indispensable como lo son los Sistemas de Recomendación, donde su objetivo primordial es ayudar a los usuarios a encontrar información de productos y/o servicios de mejor manera filtrando toda la información disponible logrando así un mejor uso de ella. En el presente trabajo de investigación se diseña y desarrolla un Algoritmo de Recomendación Hibrido para crear una lista de ítems (restaurantes) recomendables a los usuarios (consumidores), fusionando los algoritmos: Filtro Colaborativo y Basado en Contenido, utilizando un Clasificador Bayesiano con técnicas de Procesamiento de Lenguaje Natural. Además se mejora la experiencia del usuario aplicando la ubicación GPS del usuario como un filtro a las recomendaciones. Para medir el rendimiento del sistema propuesto se experimentó con un conjunto de datos extraídos de los Sitios Web Foursquare y TripAdvisor.

Palabras clave: Sistema de recomendación híbrido, filtro colaborativo, filtro basado en contenido, procesamiento de lenguaje natural, clasificador bayesiano.

Analysis, Design and Development of a Recommendation System Based on Tripadvisor and Foursquare Restaurant Data

Abstract. Due to the immense amount of information available on the Internet, it causes users to feel overwhelmed with so much information, making it difficult to search for products and/or services that suit the tastes and needs of each user. For this reason the development of intelligent

tools has become indispensable as are the Recommendation Systems, where its main objective is to help users find information of products and/or services in a better way filtering all the available information thus achieving a better use of it. In the present research work a Hybrid Recommendation Algorithm is designed and developed to create a list of recommended items (restaurants) for users (consumers), merging the algorithms: Collaborative and Content Based Filter, using a Bayesian Classifier with techniques of Natural Language Processing. Besides, the user experience is improved by applying your GPS location as a filter to the recommendations. To measure the performance of the proposed system, we experimented with a set of data extracted from the Foursquare and TripAdvisor Websites.

Keywords: Hybrid recommendation system, collaborative filtering, content-based filtering, natural language processing, Bayesian classifier.

1. Introducción

La industria restaurantera se encuentra en una fase completamente nueva, donde es necesario el uso de Tecnologías de Información y Comunicaciones (TICs) de vanguardia que el mundo globalizado actual demanda para ser competitivos dentro del mercado, representando uno de los principales sectores en pro de la economía [14], además de promover el turismo de la región, pretende que sus consumidores satisfagan una de sus necesidades básicas, pero de las más importantes, la alimentación.

En esta era de la web hay una sobrecarga de información excediendo la capacidad de una persona para poder procesarla en términos de lo que un usuario necesita encontrar; particularmente la búsqueda de restaurantes que tengan características o platillos peculiares para un determinado consumidor resulta una tarea tediosa y a menudo difícil, debido a que los consumidores buscan lugares apropiados a sus gustos personales, pero gracias a la gran cantidad de información que se puede encontrar en internet los resultados obtenidos no siempre son los mejores, lo que genera una satisfacción parcial, siendo una de las razones principales por las que los Sistemas de Recomendación (SR) juegan un rol importante en nuestra vida cotidiana [10], hasta cierto punto, el problema está siendo resuelto por los motores de búsqueda, pero no proporcionan la personalización de los datos. La personalización ha sido reconocida como un factor crítico para las industrias restauranteras exitosas y la utilización de sistemas de recomendación es el mejor enfoque para tratar con el problema de la personalización [21].

En un inicio los Sistemas de Recomendación emergen como una área de investigación individual cuando algunos investigadores iniciaron a trabajar en diferentes problemas de recomendación [23].

Los Sistemas de Recomendación (SRs) tienen una relación con los Sistemas de Búsqueda o Recuperación de Información, dado que ambos están diseñados

para que a partir de un conjunto de datos se obtenga información relevante para el usuario [1]. De acuerdo con Ricci [26] dice que un Sistema de Recomendación (SR) puede brindar información valiosa para asistir en el proceso de toma de decisiones del consumidor con el objetivo de proveer una recomendación de restaurantes con exactitud y de manera precisa [21], ya que las recomendaciones que se producen en un Sistema de Recomendación, se reducen a ayudar a los seres humanos a satisfacer sus gustos personales y descubrir nuevos elementos, con menos esfuerzo, que si realizaran la actividad de manera manual. Para lograr esto, computacionalmente, los SRs recolectan las preferencias de los usuarios, de forma explícita, por ejemplo: solicitando a los usuarios valoraciones sobre los elementos, o infiriéndolas a partir de las acciones de los usuarios. En la actualidad, los SRs han demostrado ser un medio valioso para lidiar con el problema del exceso de información [21]

Este trabajo presenta las bases para el desarrollo de un Sistema de Recomendación Hibrído de acuerdo a la clasificación propuesta por Burke [4], además de implementar el uso de técnicas de Inteligencia Artificial (IA) que permitan sugerir recomendaciones efectivas de restaurantes, orientados a mejorar la toma de decisiones de una manera sencilla y en un tiempo considerable, de acuerdo a los gustos de cada consumidor. La motivación de este trabajo se concentra en establecer la investigación, diseño y desarrollo de un Sistema de Recomendación para la industria restaurantera. La estructura del trabajo se conforma de la siguiente forma: sección 2 Trabajos relacionados, sección 3 Método propuesto, sección 4 Evaluación experimental y sección 5 Conclusiones y Trabajos a futuro.

2. Trabajos relacionados

Actualmente ha crecido el área de los SRs debido a la evolución constante de plataformas digitales que requieren de tecnologías inteligentes para generar conocimiento a partir de las grandes cantidades de datos que se generan. Jain, Grover, Thakur y Choudhary [16] ejemplificaron a Youtube.com, LinkedIn.com y Amazon.com como los principales sitios de internet que operan con un sistema de recomendación. Es importante remarcar que los sistemas de recomendación se han convertido en una herramienta de vital importancia para los usuarios, ya que engloban una infinidad de soluciones que tienen como objetivo ayudar en la selección de productos y/o servicios con el fin de facilitar la decisión de compra y consumo, y reducir el tiempo empleado mejorando la experiencia del usuario.

Hoy en día, los sistemas de recomendación juegan un papel importante en todos los campos, especialmente en el restaurante, centros de comida o el turismo [29]. Da Costa, Suyoto y Joko [5] propusieron us sistema de recomendación híbrido (Filtro Colaborativo y Basado en Contenido), con la implementación del algoritmo de vecinos cercanos (k-NN) y servicios de ubicación para brindar recomendaciones a los turistas y a su vez guiarlos a las mejores atracciones turísticas en Timor Oriental.

Las atracciones turísticas se basan en varias categorías: Comida/Gastronomía, Cultural, Historia y Religioso, por mencionar algunas. Los resultados obtenidos

revelan un interesante aporte a la investigación ayudando de manera eficiente a los turistas a planificar su destino en Timor Oriental.

Por otro parte Ananta, Suyoto, Joko [29] presentaron un sistema de recomendación utilizando el enfoque colaborativo con distancia euclidiana como el orden de la similitud de la calificación de los usuarios, para recomendar restaurantes a los turistas que llegan a Buleleng Regency (Indonesia), también se utiliza la tecnología de servicio basado en la ubicación, que utiliza el posicionamiento global (GPS) para que los usuarios puedan conocer su posición, definir y buscar ubicaciones específicas, ya sea lejos o cerca, con el propósito de encontrar el centro de comida adecuado en Buleleng Regency. Es importante mencionar que este sistema se ejecuta en dispositivos móviles Android. Por último los autores mencionan que para una mejor recomendación es necesario implementar otros enfoques que tomen en consideración los interesés del usuario. Martínez, Rodríguez y Espinilla [20] desarrollaron un sistema de recomendación híbrido para restaurantes de la región de Jaén (España), empleando el filtro colaborativo y basado en conocimiento, evitando de esta forma el problema de arranque en frío. Además introdujeron un módulo para sus usuarios que consiste en la información geográfica que proporciona Google Maps sobre los restaurantes recomendados, siendo capaz de proporcionar recomendaciones en cualquier situación requerida por los usuarios.

Otro trabajo de gran interés fue el propuesto por Habib, Rakib y Abul [13], quienes remarcaron que las redes sociales basadas en la ubicación, representan una plataforma para comprender las preferencias de los usuarios, abriendo un campo prometedor dentro de la investigación. Estudiaron la amalgamación (fusión) de registros de GPS en tiempo real, además de los datos históricos de check-in en el desarrollo de un sistema de recomendación para restaurantes, su método de recomendación toma en consideración la hora del día, la ubicación actual del usuario, sus preferencias y el historial de registro del usuario, el cual se analiza individualmente para descubrir tendencias de visitas de los usuarios, tendencias de preferencia de los alimentos y la popularidad general de los restaurantes, donde los resultados experimentales confirmaron la efectividad del método propuesto. Por otro lado Ketmaneechairat, Kongketwanich y Naijit [17] diseñaron y desarrollaron un sistema de recomendación para la cocina de comida tailandesa en teléfonos inteligentes. El desarrollo de la aplicación fue bajo la plataforma Android.

El objetivo perseguido con este sistema fue el brindar a los usuarios interesados información detallada y certera acerca de la cocina tailandesa. Algo interesante de este trabajo es que los resultados se muestran en dos idiomas: tailandés e inglés, además de detalles adicionales de los alimentos como el nombre del platillo, su imagen, los ingredientes que lo componen y su método de cocción. La aplicación consiste de dos funciones de búsqueda: buscar usando la categoría de la comida y buscar usando una palabra clave. Los resultados de las recomendaciones generadas por el sistema mostraron la relevancia del trabajo, ya que hubo una versión disponible en PlayStore para usuarios interesados en probar su funcionamiento.

Los SRs se clasifican en diferentes categorías de acuerdo al tipo de información que se utiliza para recomendar productos y/o servicios a los usuarios [18]. En la literatura se puede encontrar una diversidad de técnicas para implementar los SRs por ejemplo en la Fig. 1 se muestran las técnicas más usadas [11].

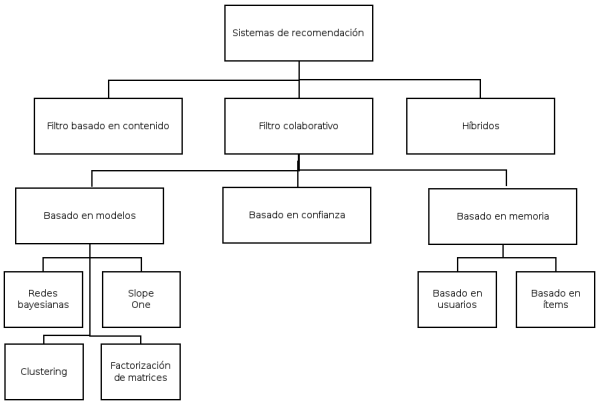


Fig. 1. Clasificación de los SRs.

Los principales enfoques que se emplean es el Filtro Colaborativo y el Filtro Basado en Contenido, pero ambos tienen algunas limitaciones y problemas. Mansur y Patel [18] en su trabajo de investigación dan un panorama general de los Sistemas de Recomendación, donde describieron los principales enfoques, problemas y limitaciones existentes. Así que en la mayoría de las aplicaciones que ocupan un SR se utiliza un enfoque híbrido, que combina diferentes técnicas para mejorar el funcionamiento del sistema, donde su idea principal es generar recomendaciones con una mejor exactitud y eficiencia; en cambio cuando se aplica un solo algoritmo es posible obtener resultados inexactos y poco confiables.

Como se mencionó anteriormente los sistemas de recomendación están presentes en muchos sectores productivos, además del restaurantero. En la literatura se encuentran diversas aplicaciones de un SR Híbrido por ejemplo, Pal, Parhi y Aggarwal [22], desarrollaron un algoritmo híbrido que donde fusionaron las ventajas del Filtro Colaborativo y el Filtro Basado en Contenido para generar recomendaciones de películas, el método presentado se enfoca en encontrar la correlación entre dos características usando la intersección de conjuntos en el filtro basado en contenido, y encontrar la similitud entre dos elementos (películas) y predecir las recomendaciones. El algoritmo fue probado y comparado con el filtro colaborativo puro (PCF) y SVD (Singular value Descomposition). Los valores MAE que se generaron después de la evaluación proporcionaron comparaciones exitosas, siendo el sistema híbrido el que produjo mejores valores MAE, mejorando la dispersión del conjunto de datos entre 1 %-2 %.

El análisis que presentaron Bogers y Van den Bosch [3] en su estudio incorporaron dos enfoques diferentes: filtro basado en contenido y filtro colaborativo

para recomendar artículos en los sitios web de marcado social de libros, incorporarando las etiquetas y metadatos (títulos, descripciones, etc.) que se generan en los sitios web de marcadores sociales, los cuales permiten a los usuarios almacenar, organizar y buscar marcadores de páginas web. El objetivo primordial de esta investigación fue el predecir qué marcadores ocultos pueden gustarle al usuario en base a su perfil. Generaron recomendaciones directamente basadas en la divergencia de Kullback-Leibler de los modelos de lenguaje de metadatos, exploraron el uso de esos metadatos para calcular la similitud entre usuarios y elementos. Los experimentos realizados fueron con tres conjuntos de datos de dos dominios diferentes: Delicious, CiteULike y BibSonomy.

Por otro lado, Renjith y Anjali [24] mostraron un SR usando un algoritmo híbrido para generar recomendaciones personalizadas a los usuarios durante su viaje, compras y otras rutinas que realice transformandolo en un compañero esencial. La aplicación es un recomendador de viajes híbrido que reúne las características de las técnicas de filtro basadas en contenido, colaborativo y demográfico, para extraer el patrón de comportamiento del usuario, permitiendo hacer recomendaciones adecuadas. El patrón de comportamiento aprende de las huellas de viaje del usuario, por lo tanto, no se necesita la intervención del usuario para generar la predicción y las recomendaciones personalizadas. Cabe mencionar que este trabajo de investigación resultó muy interesante, además los autores mencionaron las posibles mejoras que podrían ser implementadas en el sistema para afinar las recomendaciones como: medios de transporte para llegar a la atracción recomendada, que además de económico sea confiable. Instalaciones de alojamiento con ofertas interesantes y parámetros climáticos externos.

Además Ashraf, Rabbani, Martinez y Muhammad [2] propusieron un sistema de recomendación para los servicios en la nube de manera que el usuario pueda contratar el servicio que cumpla con sus propios requerimientos, lo que hace que sea fácil para un usuario decidir. El objetivo perseguido fue probar los servicios de acuerdo con las necesidades del usuario y algunos parámetros establecidos como: la seguridad, la privacidad, el almacenamiento de datos, etc. y así evaluar los servicios. Además de cumplir con los requisitos máximos del usuario, por lo que el tiempo de navegación disminuye y el buscador de servicio puede aprovechar el mejor de todos los servicios en la nube disponibles.

Los autores creen conveniente que las opiniones de los usuarios también son importantes de acuerdo a sus fechas para que se defina el concepto y el funcionamiento de un servicio en un momento determinado. García, Tello, De la Rosa y Sánchez [10], presentaron un Sistema de Recomendación para Música que aplica técnicas de aprendizaje semi-supervisado, que es capaz de aprender y adaptarse a los gustos de sus usuarios sin la necesidad de tener información previa del perfil de usuario y determinar cuándo una canción es probablemente atractiva para un usuario en particular. Los algoritmos de aprendizaje utilizados fueron Naive Bayes y J48. Para sus pruebas experimentales utilizaron un subconjunto de datos extraídos de la base de datos musical Gracenote, donde sus resultados obtenidos mostraron que con un conjunto reducido de características es psible construir de forma efectiva un modelo de recomendación. Para evaluar el método

que propusieron utilizaron las medidas tradicionales de evaluación: precisión, recall y medida F. Concluyeron que los árboles de decisión (J48) obtuvieron mejores resultados de clasificación y al mismo tiempo decir que el atributo de género es el más importantes en el proceso de clasificación.

Guzmán y Torres [12] propusieron las bases para un sistema de recomendación móvil que se basa en técnicas de Inteligencia Artificial, permitiendo sugerir a un usuario rutinas de ejercicio, orientadas a fortalecer la calidad de vida del usuario basándose en su perfil antropomórfico y patológico. Además la arquitectura de cómputo en la nube, permite el acceso libre a la aplicación y visualización de videos en el dispositivo móvil. El sistema de recomendación trabaja en función de la interpretación de la información suministrada por el usuario, ubicando al individuo dentro de un grupo poblacional identificado por patologías comúnes y así asignarle una rutina adecuada. Cada rutina se generó con la información que el usuario proporciona, construyendo de esta manera un sistema de reglas que permita razonar y recomendar rutinas de ejercicios idóneas, siendo un aporte interesante para la los campos de la investigación y la salud.

Y finalmente Otero, Gonzales y Franco [21] esbozaron un sistema de recomendación colaborativo usando un SIG (Sistema de Información Geográfica) para sugerir productos turísticos en Cuba, el cual consta de una interfaz gráfica de usuario y un motor de recomendación, donde los datos son procesados para generar las recomendaciones. Para pruebas experimentales utilizaron una base de datos con 4000 usuarios y 1000 productos turísticos. Analizando la información que brindó la herramienta se emplearon indicadores para la calidad de las recomendaciones obtenidas, como lo son la probabilidad de recomendaciones relevantes (precisión) y la probabilidad de que los ítems relevantes sean recomendados (recall). Finalmente los autores concluyen que las pruebas realizadas demostraron que para el contexto turístico descrito se obtivieron resultados positivos en la exactitud y el uso de las predicciones de valoración, calculadas mientras se incide positivamente en la cobertura del sistema implementado, lo que permite afirmar que la aplicación de la técnica logró obtener de manera eficaz el contenido que debe ser mostrado prominentemente a los usuarios de las organizaciones de turismo en Cuba.

Con lo descrito anteriormente se concluye que el campo de investigación de los sistemas de recomendación es amplio y diverso, ya que no solo se basan en los servicios que existen en internet sino su popularidad de esta clase de sistemas ha generado que su implementación sea en diferentes áreas de la vida real lo que conlleva a desafíos interesantes para estudios futuros.

3. Método propuesto

El SR propuesto utiliza un enfoque híbrido que toma como componentes principales: los usuarios que desean obtener sugerencias de establecimientos para comer y los ítems que son representados por los restaurantes.

En [15] se describen siete técnicas para implementar un SR Híbrido: *Weighted*, *Switching*, *Mixed*, *Feature*, *Feature augmentation*, *Cascade* y *Meta level*;

en este método se utiliza la técnica Switching, debido a que el sistema utiliza la situación actual del usuario. Por ejemplo, si el usuario no cumple con las condiciones para emplear el Filtro Colaborativo, entonces se aplica el Filtro Basado en Contenido. El objetivo de este Sistema de Recomendación Híbrido es cubrir las necesidades que tengan los diferentes usuarios.

3.1. Colección de datos

Debido a la falta de los datos idóneos a los propósitos planteados, fue necesaria la construcción de un conjunto de datos que permitirá evaluar la pertinencia del método propuesto. Para esto se obtuvieron datos del Sitio Web Foursquare, una compañía de tecnología que utiliza la ubicación inteligente para construir una experiencia significativa para el consumidor [7]. Otro Sitio Web que también fue utilizado es TripAdvisor, un sitio de viajes que permite a los viajeros aprovechar al máximo el potencial de cada viaje proporcionando reseñas de contenido relacionado con los viajes [27].

El sitio Foursquare proporciona una API[1] (Interfaz de Programación de Aplicaciones, abreviada como *API* del inglés: *Application Programming Interface*) que permite consumir un servicio web para extraer datos sobre lugares (restaurantes, cafeterías, bares, etcétera)[2]. Para este trabajo se obtuvieron los lugares de las catergorías *food*, *drinks* y *coffee* [8] tomando de muestra los que se encuentran en el Estado de Tlaxcala. Como resultado se concentraron 685 lugares seleccionando solo algunos datos, además se obtuvieron 4764 comentarios relacionados con los lugares y 65274 calificaciones registradas por 53051 usuarios. En la Tabla 1 se describen los datos que fueron seleccionados para concentrar la información.

Tabla 1. Datos seleccionados de la API de Foursquare sobre los lugares.

Dato	Descripción
id	Un identificador único en formato de cadena.
name	Nombre establecido.
lat	Valor de la latitud para indicar la ubicación.
lon	Valor de la longitud para indicar la ubicación.
type	Categoría.
text	Texto del comentario.
user_id	Un identificador único para el usuario.
rating	Calificación otorgada al lugar por el usuario.

[1] La API es un conjunto procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

[2] Para mayor información puede visitar este sitio web `https://developer.foursquare.com`

Debido a la escasez de los comentarios obtenidos en Foursquare se opta por utilizar la técnica Web Scraping que sirve para extraer información de páginas web de forma automatizada [19]. Para implementar la técnica, se utilizó la herramienta scrapy[3] que funciona con el lenguaje de programación Python. Dicha técnica se aplicó al sitio TripAdvisor para conseguir un total de 38107 comentarios de los cuales se etiquetaron de acuerdo a la calificación otorgada por el usuario quedando como resultado 14876 comentarios y con estos se categorizaron en dos clases: 7438 positivos y 7438 negativos.

Los datos se pueden descargar desde este enlace `https://drive.google.com/drive/folders/1XLyn-of_Y9zch9jmf81nCkbKvFAarAl9`.

3.2. Descripción del sistema de recomendación híbrido

La idea principal del sistema de recomendación híbrido es implementar las técnicas más populares que existen, el Filtro Colaborativo y Basado en Contenido. Además se integra la ubicación del usuario con el sistema de posicionamiento global más conocido por sus siglas en inglés, GPS (siglas de Global Positioning System)[9] para generar sugerencias de relevancia, porque no resultan relevantes aquellos lugares que se encuentren lejos del usuario.

En la Fig. 2 se observan los componentes principales del sistema de recomendación híbrido.

La propuesta inicia con el usuario, cuando registra información personal como es la calificación que le otorga a cada restaurante y los principales gustos que tiene, esto se almacena en una base de datos para su uso posterior. Después se obtiene el historial del usuario con los datos registrados previamente, evidentemente que a través del tiempo todos estos datos empezarán a generar recomendaciones significativas. Entonces en este punto es cuando empieza el trabajo de los diferentes algoritmos que integran al SR Híbrido. Como primer paso se valida la información que ha generado el usuario; si el usuario tiene calificaciones registradas, entonces se obtienen todas las que fueron otorgadas por él a los restaurantes y se utilizan en el Algoritmo del Filtro Colaborativo.

En el trabajo propuesto por Yang, Wu, Zheng, Wang y Lei [30] mostraron los principales Algoritmos de Filtro Colaborativo: Basado en el Usuario y Basado en el Ítem. El Sistema de Recomendación propuesto, utiliza el Algoritmo Basado en el Ítem debido a que el número de usuarios es más largo que el número de ítems y los ítems no tienen cambios frecuentemente. La Fig. 3a se presenta el proceso que emplea el algoritmo del Filtro Colaborativo para generar una lista previa de recomendaciones.

El primer paso es calcular la similitud que existe entre los ítems mediante la matriz de calificaciones que se crea con la información de la base de datos. Como ejemplo de la matriz de calificaciones se presenta en la Tabla 2.

Para el cálculo de la similitud se pueden utilizar las siguientes medidas: puro coseno, coseno ajustado o coeficiente de correlación Pearson [30]. En este trabajo

[3] Es una herramienta open source para extraer datos des sitios web de forma rápida, simple y completa. `https://scrapy.org`

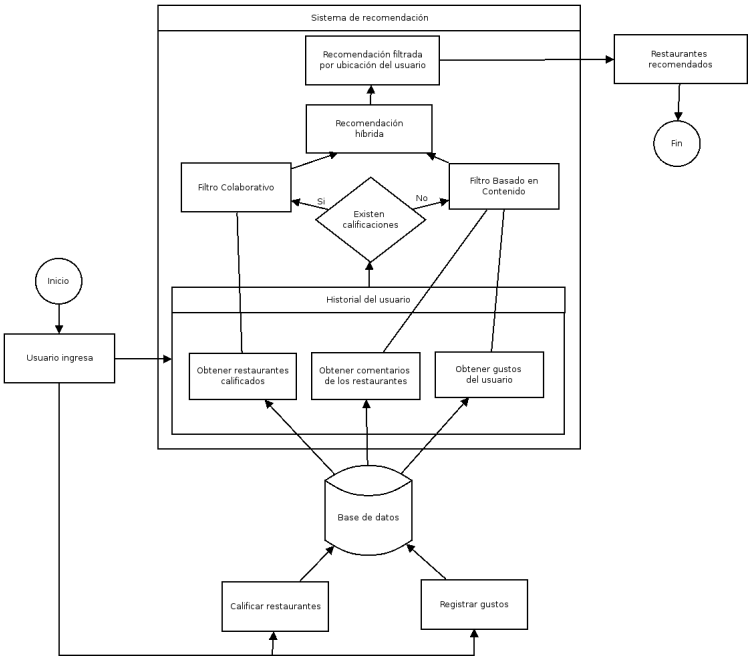


Fig. 2. Diagrama de bloques del sistema de recomendación híbrido.

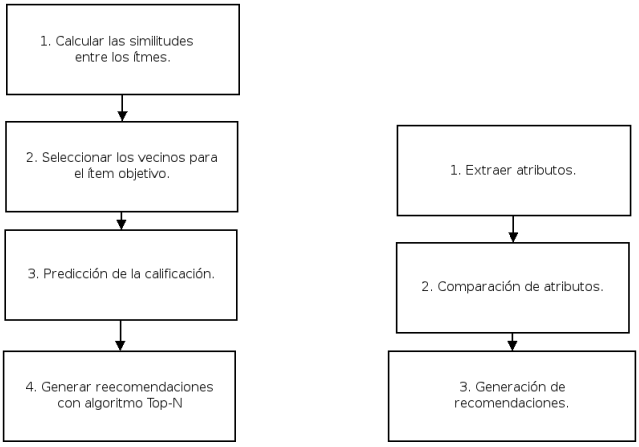
Tabla 2. Ejemplo de matriz de calificaciones (en una escala de 1-5).

	Ítem 1	**Ítem 2**	**Ítem 3**	**Ítem 4**
Usuario A	4	?	3	5
Usuario B	?	5	4	?
Usuario C	5	4	2	?
Usuario D	2	4	?	1

se aplicó el Coeficiente de Pearson, debido a que se utiliza el enfoque basado en el ítem del Filtro Colaborativo. Después se eligen los vecinos más cercanos para el ítem objetivo, con todos los vecinos (k) se predice la calificación del ítem y por último se genera una lista previa de las recomendaciones con el algoritmo Top-N [6].

En caso de que el usuario no tenga aún restaurantes calificados entonces se emplea el Algoritmo del Filtro Basado en Contenido. En la Fig. 3b se presenta el proceso del algoritmo que genera una lista previa de recomendaciones.

Como objetivo, el algortimo Basado en Contenido debe seleccionar y determinar los ítems basados en la correlación y relación entre el contenido de cada ítem (en este trabajo son los restaurantes) y los gustos de los usuarios. Este algoritmo en particular utiliza técnicas de Minería de Texto: Recuperación de Información, Extracción de Información, Categorización y PLN (Procesamiento de Lenguaje Natural) [28].



(a) Filtro colaborativo. **(b)** Filtro Basado en Contenido.

Fig. 3. Procesos de cada algoritmo del sistema de recomendación.

Para iniciar el proceso del Filtro Basado en Contenido, se obtienen los comentarios positivos que los usuarios han generado para cada restaurante y los gustos del usuario que desea obtener recomendaciones. La polaridad de los comentarios se mide con un Clasificador Bayesiano [25]. Después mediante técnicas de minería de texto, se extraen los atributos de los ítems. Posteriormente se comparan los atributos con los gustos del usuario y finalmente se generan las recomendaciones.

Este Sistema de Recomendación aplica un filtro más a las recomendaciones generadas. Con la ubicación del usuario se eligen los restaurantes mas cercanos al usuario. Con esto termina el flujo del sistema, generando una lista de restaurantes recomendados para el usuario.

4. Evaluación experimental

La evaluación de los algoritmos integrados en el Sistema de Recomendación Híbrido que genera recomendaciones de restauranes para los usuarios, hace uso de una base de datos que fue poblada con datos generados en TripAdvisor y Foursquare. En la Tabla 3 se describen los datos usados en las pruebas.

4.1. Precisión y *recall*

Para el análsis de la información resultante del SR Híbrido, se utilizaron dos métricas de evaluación sobre las recomendaciones obtenidas como lo son: Precisión y Recall que se enlistan a continuación en las ecuaciones 1 y 2.

La Precisión es la fracción de elementos recomendados que son realmente relevantes para el usuario [15]:

$$Precisión = \frac{Items\ correctamente\ recomendados}{Total\ de\ items\ recomendados}. \quad (1)$$

Tabla 3. Datos que se utilizan en las pruebas.

Dato
685 lugares como restauranes, bares, cafeterías, entre otros.
4764 comentarios relacionados a los lugares.
65274 calificaciones otorgadas a los lugares.
53051 usuarios que interactúan con los lugares.
14876 comentarios usados en el clasificador Bayesiano.

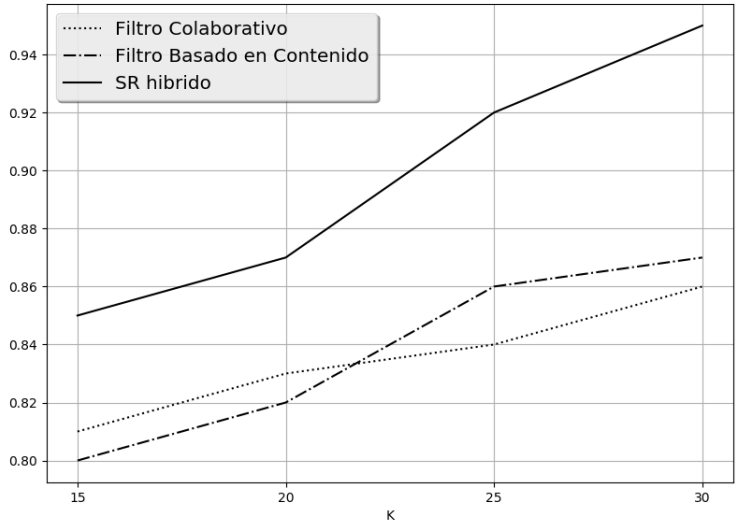


Fig. 4. Comparación de la Precisión entre Filtro Colaborativo, Filtro Basado en Contenido y SR Híbrido.

Recall se puede definir como la fracción de elementos relevantes que también forman parte del conjunto de elementos recomendados [15]:

$$Recall = \frac{Items\ correctamente\ recomendados}{Total\ de\ items\ recomendados\ útiles}. \quad (2)$$

4.2. Resultados de la evaluación

Después calcular los valores de Precisión y Recall para cada usuario, se obtuvo la media de estas métricas para el sistema empleando solo el Filtro Colaborativo, posteriormente se aplicó lo mismo al Filtro Basado en Contenido y al final el cálculo se realizó para el Sistema de Recomendación Híbrido. Las Fig. 4 y 5 muestran una comparativa de los resultados para Precisión y Recall de las diferentes pruebas que se realizaron con el Sistema de Recomendación.

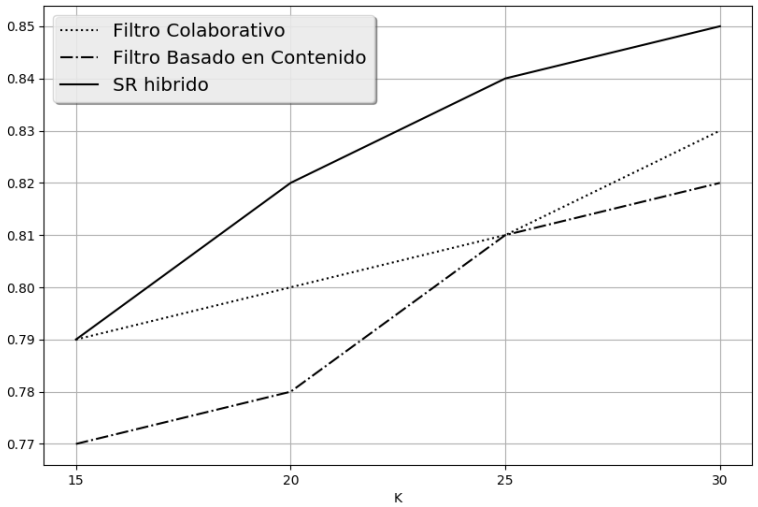


Fig. 5. Comparación de Recall entre Filtro Colaborativo, Filtro Basado en Contenido y SR Híbrido.

Como se puede apreciar, los resultados presentados en la gráfica de la Precisión aumentan según el incremento del valor de los vecinos (k) para todos los casos. La probabilidad de que un ítem recomendado por el sistema sea relevante para el usuario, supera el 80 % para las 4 variantes probadas en el caso del Filtro Colaborativo y Basado en Contenido y se mantiene sobre el 85 % cuando se utiliza el SR Híbrido. En el caso de Recall, la probabilidad de que un ítem relevante sea recomendado, también crece en relación directa con el valor de k y esta probabilidad se mantiene por encima del 77 % cuando se evalúa el Filtro Colaborativo y Basado en Contenido, y sobre el 79 % para la evaluación del SR Híbrido.

Con los resultados presentados se observa que el Sistema de Recomendación Híbrido mejora cuando exista la combinación de algoritmos, ya que encontramos algunas deficiencias en el uso particular de los algoritmos.

5. Conclusiones y trabajos a futuro

En este trabajo se ha presentado un Sistema de Recomendación que genera una lista de restaurantes recomendados para los consumidores basándose en las preferencias y localización del consumidor. La problemática se resolvió aplicando técnicas de Minería de Texto y Minería de Datos, usando los algoritmos de Filtro Colaborativo y Basado en Contenido.

Entre los objetivos que se plantearon al inicio del trabajo fue por un lado, obtener y construir un conjunto de datos con la temática de restaurantes. Y por otro lado, evaluar la efectividad del Sistema de Recomendación Híbrido con

los diferentes casos que se puedan presentar al atender las necesidades de los consumidores.

Los resultados obtenidos mostraron que la técnica híbrida brinda mejores resultados en comparación si los algoritmos de Filtro Colaborativo y Basado en Contenido trabajaran individualmente. Las desventajas que presentan estos algoritmos se mitigan al funsionarlos y optar por la técnica híbrida. Con esto se comprueba que la combinación de algoritmos proporciona una lista de recomendaciones acorde al usuario, donde es importante enfatizar que la Minería de Texto y la Minería de Datos son áreas de investigación que tienen un amplio sector de aplicación.

Como trabajo futuro se propone aplicar una técnica de desarrollo de software para implementar el Sistema de Recomendación Híbrido. Para tal situación es conveniente desarrollar un servicio de web para conectar el sistema con los dispositivos móviles, y así tener una ventaja más en el sistema, ya que con esto se tiene como ventaja que el consumidor tenga las recomendaciones de los restaurantes en su dispositivo móvil.

Referencias

1. Amatriain, X.: Recommender systems (machine learning summer school 2014 @ cmu) (Febrero 2018), https://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu/
2. Ashraf, A., Rabbani, I.M., Martinez-Enriquez, A.M., Muhammad, A.: A user-centered approach for cloud service selection and recommendation. Research in Computing Science 130, 99–110 (Diciembre 2016)
3. Bogers, T., Van den Bosch, A.: Collaborative and content-based filtering for item recommendation on social bookmarking websites. Submitted to CIKM 9 (2009)
4. Burke, R.: The adaptive web. chap. Hybrid Web Recommender Systems, pp. 377–408. Springer-Verlag, Berlin, Heidelberg (2007), http://dl.acm.org/citation.cfm?id=1768197.1768211
5. d. C. L. Soares, J., Suyoto, Santoso, A.J.: M-guide: Hybrid recommender system tourism in east-timor. In: 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT). pp. 303–309 (Sept 2017)
6. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
7. Foursquare, Inc.: Acerca de nosotros (Febrero 2018), https://es.foursquare.com/about/
8. Foursquare, Inc.: Venue categories (Febrero 2018), https://developer.foursquare.com/docs/resources/categories
9. Fundación Wikimedia, Inc.: Sistema de posicionamiento global (Febrero 2018), https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global
10. García, J.R.A., García, J.V.H., Tello, E.V., de la Rosa, G.R., Sánchez, C.S.: Sistema de recomendación de música basado en aprendizaje semi-supervisado. Research in Computing Science 94, 97–109 (Mayo 2015)
11. González, O.E., Jacques, S.M.: Estado del arte en los sistemas de recomendación. Research in Computing Science 135, 25–40 (Noviembre 2017)
12. Guzmán-Luna, J., Torres, I.D., Vallejo, S.: Un sistema recomendador móvil de rutinas de ejercicio basado en el perfil del usuario. Research in Computing Science 94, 137–149 (Mayo 2015)

13. Habib, M.A., Rakib, M.A., Hasan, M.A.: Location, time, and preference aware restaurant recommendation method. In: 2016 19th International Conference on Computer and Information Technology (ICCIT). pp. 315–320 (Dec 2016)
14. Instituto Nacional de Estadística y Geografía: Censos económicos 2014. resultados definitivos (Febrero 2018), http://www.inegi.org.mx/est/contenidos/proyectos/ce/ce2014/default.aspx
15. Isinkaye, F., Folajimi, Y., Ojokoh, B.: Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal 16(3), 261–273 (2015), http://www.sciencedirect.com/science/article/pii/S1110866515000341
16. Jain, S., Grover, A., Thakur, P.S., Choudhary, S.K.: Trends, problems and solutions of recommender system. In: International Conference on Computing, Communication Automation. pp. 955–958 (Mayo 2015)
17. Ketmaneechairat, H., Kongketwanich, C., Naijit, T.: Recommender system for thai food cooking on smartphone. In: 2017 Twelfth International Conference on Digital Information Management (ICDIM). pp. 169–178 (Sept 2017)
18. Mansur, F., Patel, V., Patel, M.: A review on recommender systems. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). pp. 1–6 (Marzo 2017)
19. Martí, M.: Qué es el web scraping? introducción y herramientas (Febrero 2018), https://sitelabs.es/web-scraping-introduccion-y-herramientas
20. Martinez, L., Rodriguez, R.M., Espinilla, M.: Reja: A georeferenced hybrid recommender system for restaurants. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. vol. 3, pp. 187–190 (Sept 2009)
21. Otero, J.D.C., Gonzalez1, M.J.R., Franco, M.T.M.: Esbozo de una técnica para la recomendación de productos turísticos en cuba usando gis. Research in Computing Science 79, 21–36 (Octubre 2014)
22. Pal, A., Parhi, P., Aggarwal, M.: An improved content based collaborative filtering algorithm for movie recommendations. In: 2017 Tenth International Conference on Contemporary Computing (IC3). pp. 1–3 (Agosto 2017)
23. Patel, B., Desai, P., Panchal, U.: Methods of recommender system: A review. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). pp. 1–4 (Marzo 2017)
24. Renjith, S., Anjali, C.: A personalized mobile travel recommender system using hybrid algorithm. In: 2014 First International Conference on Computational Systems and Communications (ICCSC). pp. 12–17 (Diciembre 2014)
25. Sanzón, Y.M., Vilariño, D., Somodevilla, M.J., Zepeda, C., Tovar, M.: Modelos para detectar la polaridad de los mensajes en redes sociales. In: Research in Computing Science
26. Stabb, S., Werther, H., Ricci, F., Zipf, A., Gretzel, U., Fesenmaier, D.R., Paris, C., Knoblock, C.: Intelligent systems for tourism. IEEE Intelligent Systems 17(6), 53–66 (Noviembre 2002)
27. TripAdvisor, Inc.: Acerca de tripadvisor (Febrero 2018), https://tripadvisor.mediaroom.com/mx-about-us
28. Vijayarani, S., Ilamathi, M.J., Nithya, M.: Preprocessing techniques for text mining-an overview. International Journal of Computer Science & Communication Networks 5(1), 7–16 (2015)
29. Wijaya, K.A., Suyoto, Santoso, A.J.: M-guide: Recommending systems of food centre in buleleng regency. In: 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT). pp. 310–314 (Sept 2017)

30. Yang, Z., Wu, B., Zheng, K., Wang, X., Lei, L.: A survey of collaborative filtering-based recommender systems for mobile internet applications. IEEE Access 4, 3273–3287 (Mayo 2016)

ISSN 1870-4069

Entornos de trabajo para procesamiento de datos masivos y aprendizaje automático

Angélica Guzmán Ponce[1], Rosa María Valdovinos Rosas[1], José Raymundo Marcial Romero[1], Roberto Alejo Eleuterio[2]

[1] Universidad Autónoma del Estado de México, Facultad de Ingeniería, Toluca, Estado de México, México

[2] Instituto Tecnológico de Toluca, Estado de México, México

aguzmanp@alumni.uaemex.mx, {rvaldovinosr,jrmarcialr}@uaemex.mx, ralejoll@hotmail.com

Resumen. *Big Data*, ha surgido como un concepto tanto en ámbitos públicos como privados por la creciente demanda de información, ya sea desde la web en redes sociales hasta por medio de los sensores de un teléfono móvil. La creciente demanda de datos ha implicado el buscar alternativas en técnicas de aprendizaje clásico que aborden los desafíos que *Big Data* presenta en sus características, el éxito depende de las metodologías de análisis y rapidez computacional que se utilicen. Para la aplicación de *Big Data*, recientemente han surgido entornos de trabajo que prometen un adecuado funcionamiento en contexto de grandes volúmenes de datos. Para identificar las bondades que cada uno ofrece, en este artículo se presenta un análisis desde tres perspectivas, sus características generales, los algoritmos de aprendizaje automático con los que disponen y la complejidad de su implementación y operación, de este modo determinar el mejor entorno de trabajo para *Big Data*.

Palabras clave: Big data, hadoop, spark, flink, aprendizaje automático, entornos de trabajo.

Frameworks for Big Data and Machine Learning

Abstract. *Big Data* has emerged as a concept in public and private spheres due to the growing demand for information of sources like web on social networks or through the sensors of a mobile phone. The growing demand for data has involved looking for alternatives in classical learning techniques that try to solve the challenges that *Big Data* has in its characteristics, the success depends on the methodologies of analysis and computational speed that are used. To apply *Big Data*, there are frameworks that have recently emerged and promise an adequate functioning in context of large volumes of data. To identify the benefits that each

one offers, in this paper, we present an analysis from three perspectives, the first one in general characteristics, the second one are algorithms of Machine Learning are implemented and the last one, the complexity of their implementation and operation, in this way to determine the best environment to *Big Data*.

Keywords: Big data, hadoop, spark, flink, machine learning, frameworks.

1. Introducción

Con la creciente demanda de información, el término *Big Data* ha tomado importancia en diversos ámbitos ya sean públicos o privados. De acuerdo a Morabito [19], Edd Dumbill define a *Big Data* como *el dato que excede la capacidad de procesamiento de sistemas de base de datos convencionales, siendo tan grande que los movimientos de datos son rápidos*. Es decir que, el volumen de los datos es tal que para tecnologías de bases de datos actuales, el procesamiento de estos se vuelve complejo y en ocasiones imposible de procesar. Derivado de esto, las principales características de *Big Data* de acuerdo a la literatura impactan en tres conceptos fundamentales [18]:

- *Volumen*: Se refiere a la cantidad de datos disponibles para su procesamiento. Por ejemplo, la creciente interacción de usuarios en redes sociales como Twitter, con el uso de hashtags (etiquetas) en reacción a eventos causan una enorme producción de datos en cortos periodos de tiempo.
- *Variedad*: Se refiere a la diversidad de fuentes de datos o también en los formatos de datos. Por ejemplo social media, tweets, correos, información de sensores, producen diferentes formatos así como también son diversas maneras de obtener información.
- *Velocidad*: Se refiere a la rapidez del flujo de datos que tienen que ser manejados por el sistema, es decir, el proceso de generar datos en cortos periodos de tiempo implica una agilidad del proceso en captarlos y procesarlos.

Adicionalmente a estas características, algunas investigaciones consideran dos más, las cuales impactan más en los resultados de un análisis que en la generación de datos son la *Veracidad* y *Valor*:

- *Veracidad*: Se refiere a la calidad y confianza de los datos disponibles en un grado incomparable de volumen, velocidad y variedad [19]. Es decir, los datos que se están utilizando son los adecuados para analizar un problema, en pocas palabras se habla de calidad de los datos.
- *Valor*: Se refiere al proceso de extraer información oculta de datos emergentes [18]. En pocas palabras, se hace referencia al resultado deseado del proceso de análisis de *Big Data*.

Como se muestra en la Tabla 1, para enfrentar los grandes retos de *Big Data* algunas estrategias son: 1. Diseño de ambientes escalables. 2. Proveer Tolerancia a fallos. 3. Diseño de soluciones eficientes. [8].

Tabla 1. Retos de Big Data [21].

Característica	**Reto**
Volumen	Gran escala: El volumen de datos a procesar puede ser tal que los recursos disponibles para ello no lo permitan.
Variedad	Alta dimensionalidad con tipos de datos mezclados con distribución lineal.
Velocidad	Tiempo de llegada de datos: real data streams y alta velocidad.
Veracidad	El contenido de los datos puede ser incierto o incompleto.
Valor	Bajo valor de densidad, Significado diverso de datos.

Debido a la gran cantidad de datos que se manejan en tiempo real, el éxito en *Big Data* depende de metodologías de análisis y rapidez computacional que se utilicen. Para ello, se buscan algoritmos de optimización que no sólo trabajen rápido sino que también reduzcan el uso de memoria [21].

De igual modo, la mayoría de las herramientas utilizadas en minería de datos para almacenamiento, procesamiento y análisis de datos se han vuelto poco eficientes para el tratamiento masivo de datos heterogéneos. En consecuencia se tiene la necesidad de encontrar o proponer alternativas que solventen la demanda de *Big Data*.

Algunas de estas propuestas son consideradas por algunos de los principales entornos de trabajo disponibles para *Big Data* en los que estudios como en [8,12,14,22] hace uso de Spark [4], Hadoop [3], Flink [2], Google Cloud [6], Amazon [1], entre otros, considerando la tolerancia a fallos, así como también el rendimiento de la solución hasta el diseño de la arquitectura. De estos entornos de trabajo resulta de interés identificar las bondades que ofrecen en la implementación, uso y manejo de algoritmos de aprendizaje automático.

En este artículo, se presenta un estudio comparativo de tres de los entornos de trabajo más usados en el área científica para el tratamiento de grandes volúmenes de información con características propias de *Big Data* y la integración de algoritmos de aprendizaje automático. El resto del artículo está organizado como sigue: En la Sección 1.1 se introduce la importancia de aprendizaje automático en *Big Data*, mientras que en la Sección 3 se describen tres de los entornos de trabajo que ayudan en tareas propias de *Big Data*. En la Sección 4 se describen características de cada entorno de trabajo, así como también los algoritmos de aprendizaje automático implementados, por último las conclusiones se presentan en la Sección 5.

1.1. Aprendizaje automático

Como ya se mencionó, una de las áreas que están apoyando fuertemente la realización de *Big Data* es el aprendizaje automático. En este sentido los algoritmos de aprendizaje automático se utilizan para encontrar patrones o analizar datos, por medio de la toma inteligente de decisiones, de manera similar *Big Data* maneja una gran cantidad de datos que pueden contener patrones a descubrir por medio del análisis, la situación que ha popularizado su uso.

Los campos de investigación y desarrollo de aprendizaje automático son diversos, no obstante en la literatura se consideran básicamente tres enfoques de proceso de aprendizaje [13]: aprendizaje supervisado, aprendizaje no supervisado y semi-supervisado. En el aprendizaje supervisado o inductivo, se realiza el entrenamiento con datos de los cuales se conoce a priori la salida deseada o esperada por el clasificador, es decir, los conjuntos de patrones utilizados se presentan al sistema ya etiquetados por un experto humano en el área de estudio [17].

Por otro lado, el aprendizaje no supervisado o deductivo, permite la construcción libre del agrupamiento de los patrones que carecen de las etiquetas de clases, donde en ocasiones no hay información a priori sobre la cantidad de etiquetas, el enfoque no supervisado intenta encontrar una hipótesis útil, basada en únicamente relaciones de similitud en el espacio de representación de los datos [10].

Por último, el aprendizaje semi-supervisado, el cual utiliza una cantidad mínima de objetos etiquetados que son representativos del espacio de representación de los datos combinando con la mayor parte de objetos no etiquetados realiza la clasificación de todos, de está manera se reduce el costo de etiquetado de patrones por un experto humano en el área, mejorando la exactitud del aprendizaje [16].

Para el logro de su cometido, en el aprendizaje automático se encuentra una amplia gama de algoritmos, algunos orientados a tareas de clasificación, regresión o agrupamiento. El contexto de la investigación se centra en los algoritmos de clasificación con paradigma de aprendizaje supervisado. Dentro de estos algoritmos se pueden distinguir árboles de decisión, memorias asociativas, Máquinas de vector soporte, redes neuronales, métodos Bayesianos y muchos más [11].

2. Trabajos relacionados

Existen trabajos comparativos de entornos de trabajo para *Big Data* en los cuales se muestran en resumen características de cada entorno de trabajo, por ejemplo, Inoubil et al. [9] presentan una revisión de los entornos de trabajo más populares y ampliamente usados en *Big Data* (Hadoop y Spark), en el cual se realizó una comparativa considerando el rendimiento para el procesamiento por lotes y el procesamiento en linea.

Por otro lado, Singh et al. [22] proporcionan un análisis de las plataformas Hadoop y Spark para realizar análisis de *Big Data* evaluando las ventajas e inconvenientes de cada plataforma en base a diversas medidas tales como escalabilidad, velocidad de datos, tolerancia a fallas, procesamiento en linea, tamaño

de datos y tareas iterativas. De las plataformas describen los puntos fuertes y desventajas de cada uno.

Landset et al. [12] realizan un trabajo de evaluación de herramientas tales como Spark, Hadoop, Flink, Storm y H_2O para *Big Data* bajo criterios de ventajas y desventajas de cada herramienta, así como también realizan una comparativa de los paradigmas de procesamiento y las librerías de aprendizaje automático que se pueden implementar. Sin embargo, este análisis no incluye un enfoque de instalación e implementación de algoritmos propios.

Otro estudio de tecnologías de código abierto como Spark, Hadoop, Kafka, Scribe, S4, HStreaming, All-RiTE e Impala, para el procesamiento de grandes volúmenes de datos en tiempo real, pero bajo el contexto del algoritmo *MapReduce* [15].

Para cada entorno o herramienta de trabajo bajo el contexto de *Big Data* el cambio de paradigma de programación y manejo de recursos computacionales que cada uno tiene han generado dificultades tales como el incremento de manejo de archivos como es el caso de *MapReduce*, hasta el escalamiento de algoritmos de aprendizaje automático en algún entorno de trabajo.

3. Entornos de trabajo para datos masivos

En esta sección se presenta un resumen de los entornos de trabajo más usados, destacando propiedad clave tales como el modelo de programación, los lenguajes de programación que se pueden emplear, así como el tipo de fuente de datos.

3.1. Apache Hadoop

Desde el año 2008, Hadoop ha sido una tecnología destacada en el área de *Big Data*, fue de los primeros entornos de trabajo que dieron solución a las cuatro $v's$. Es de código libre, desarrollado para ser usado de manera distribuida y escalable, así como también la administración de grandes cantidades de datos [20]. Se divide en dos componentes principales 1. HDFS (Hadoop Distributed File System). 2. Cómputo distribuido, basado en la idea *MapReduce*.

3.1.1. HDFS

HDFS es una implementación de código libre del sistema de archivos distribuido de Google (*Google File System, GFS*), este sistema de archivos se encarga de almacenar los archivos a lo largo del clúster, diseñado para obtener un acceso rápido para grandes archivos o conjuntos de datos grandes, es escalable y tolerante a fallas.

A pesar de las deficiencias detectadas a lo largo de los diferentes estudios [8], el sistema de archivos que maneja es el predilecto para otros entornos de trabajo debido a su funcionamiento y de implementación de código libre.

Como se muestra en la Figura 1 el sistema de archivos HDFS está estructurado por bloques, es decir que HDFS divide los archivos en bloques de tamaño fijo (128 MB), de ser posible HDFS difunde los bloques de un archivo por las

diferentes máquinas que se encuentran en clúster, de este modo se paraleliza la lectura y escritura del archivo, contribuyendo a la rápidez de solo leer o escribir en un sólo disco, sin embargo se aumenta el riesgo de no tener disponible un archivo en caso de falla, HDFS tiene en consideración esto, debido a que mitiga el riesgo al replicar cada bloque de archivos en varias computadoras, siendo un principio básico de la arquitectura [7].

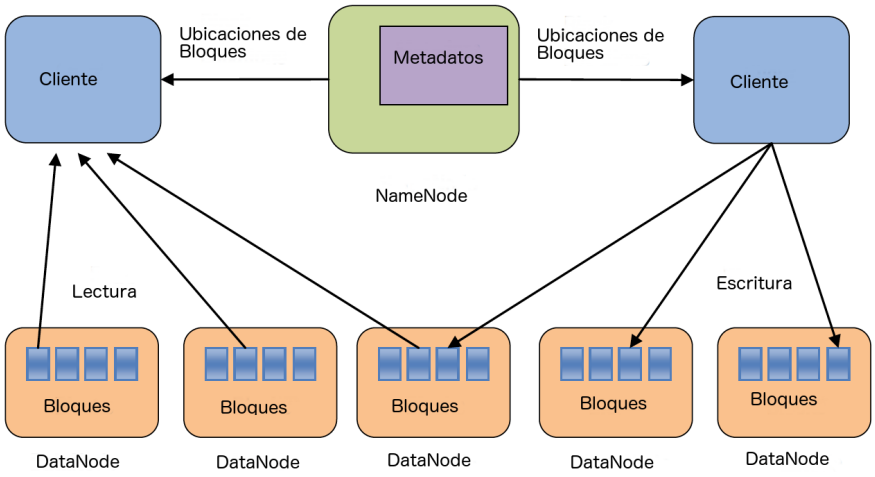


Fig. 1. Arquitectura HDFS.

HDFS introduce dos conceptos, por una parte *NameNode* el cual gestiona el espacio de nombres del sistema de archivos, almacenando en memoria los metadatos para un acceso rápido a estos. Por otro lado, *DataNode* se encarga de almacenar el contenido de un archivo completo en bloques de archivos.

3.1.2. MapReduce MapReduce es un algoritmo de cómputo distribuido, funciona para el procesamiento en paralelo de grandes volúmenes de datos, permitiendo realizar códigos de manera distribuida o paralela. La idea principal es dividir los datos masivos en pequeñas partes y éstas de manera paralela y distribuida generen resultados parciales, los cuales de forma conjunta den una solución global [5]. En general analiza cada registro al leer una entrada [8], cuenta con dos funciones importantes, *Map y Reduce*. La función *Map* considera un par $clave - valor$ de entrada y produce una lista de pares intermedios $clave - valor$, los valores intermedios están agrupados bajo la misma *clave* y son enviados a la función *Reduce*, la cual además de las claves intermedias, recibe el conjunto de valores por clave, estos son mezclados.

3.2. Apache Spark

Cuenta con diversas ventajas sobre el resto de los entornos de trabajo, hoy en día es utilizado por empresas como Yahoo, Baidu, entre otras [8]. De igual manera que Hadoop, es de código libre, para procesos de manera distribuida, basado en el mejoramiento del rendimiento de memoria.

La arquitectura que implementa Spark se basa en el uso de *RDD* (*Resilient Distributed Dataset*) (Figura 2), básicamente es una inmutable colección de objetos por todo el clúster de Spark. Es importante destacar que en Spark se tienen dos tipos de operaciones sobre los RDD, las *transformaciones* que consisten en crear nuevos RDD de uno ya existente aplicando funciones como *map, filter, union y join*, es decir, recorrido de los datos, filtrado de datos, union conjuntos de datos, union a pares respectivamente. Por otro lado, las *acciones*, las cuales son el resultado final de realizar transformaciones a los RDD.

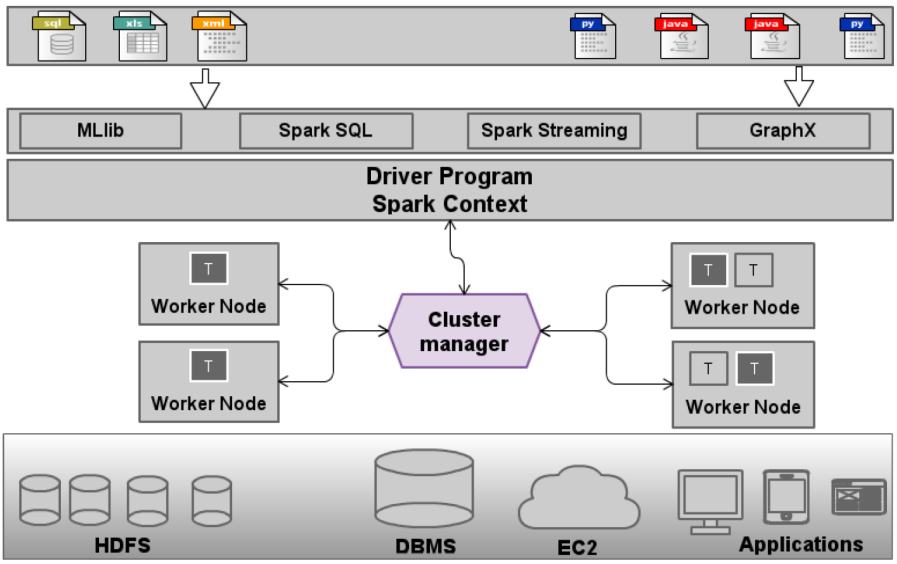


Fig. 2. Arquitectura Spark [8].

Como se muestra en la Figura 2 la arquitectura que presenta Spark está basada en cliente-servidor (maestro-exclavo). El apartado de *Driver Program* es el nodo cliente o esclavo, el cual tiene un objeto de tipo SparkContext, el cual administra la ejecución de las aplicaciones, en tanto que, el *Cluster Manager* gestiona el flujo de la aplicación en cuanto a recursos disponibles. Durante la ejecución de una aplicación en Spark, mantienen contenedores de operaciones denominados *Worker Nodes*. Una de las características que hace atractivo el uso

de Spark es que provee de API's que facilitan la programación e interacción con el entorno de trabajo [8], entre ellas están:

- *Spark Core*: Proporciona la gestión de memoria, así como también un modelo generalizado de ejecución que soporta una variedad de aplicaciones, así como también códigos de Java, Scala y Python.
- *Spark Stream*: Proporciona aplicaciones interactivas y analíticas en transmisión de datos, tolerante a fallas y puede ser usado para diferentes fuentes de datos.
- *Spark SQL*: Permite el manejo de SQL en datos estructurados de diversas fuentes.
- *Spark MLLib*: Proporciona un conjunto de algoritmos de aprendizaje automático, de mayor velocidad en comparación que MapReduce.
- *GraphX*: Es una librería para computo paralelo de gráficas, soporta operaciones sobre gráficas tales como unión de vértices, obtención de subgráficas, entre otras.

3.3. Apache Flink

Flink [2] es un entorno de trabajo de código libre, utilizado para el procesamiento de datos tanto en tiempo real como por lotes, capaz de ser aprovechado por las características de ser distribuido, alto rendimiento, alta disponibilidad y preciso. Una ventaja competitiva es que las aplicaciones pueden mantener una agregación o resumen de los datos procesados, asegurando el estado de una aplicación en caso de falla.

De acuerdo a Inoubil et. al. [8], el modelo de programación empleado en Flink es similar a *MapReduce*, sin embargo se tienen propiedades adicionales como el alto nivel de funciones tales como *join, filter* y *agregation*.

Como se observa en la Figura 3, Flink ofrece una arquitectura compuesta por modos de despliegue ya sean de manera local, en cluster o en la nube (*cloud*), el *core* de Flink es la manera distribuida procesando los datos como un evento a la vez en lugar de una serie de lotes, siendo esta de suma importancia y distintiva con respecto al rendimiento. Por último, en un nivel superior y abstracto se cuentan con las API's y librerías que permiten a los usuarios el computo distribuido y de manera transparente el uso.

Para las API's a nivel de procesamiento en línea, se implementan las transformaciones de flujo de datos (por ejemplo, filtrado, actualización de estado, definición de ventanas, agregación), mientras que para el procesamiento por lotes se aplican transformaciones en conjuntos de datos (por ejemplo, filtrado, mapeo, unión, agrupamiento). La API *Table* indica el uso del lenguaje SQL.

Como se muestra en la Figura 3, se cuenta con una librería de aprendizaje automático denominada *FlinkML* así como también una librería para el procesamiento de gráficas denominada *Gelly*.

De manera similar que Hadoop y Spark, Flink hace uso de sistema de archivos *HDFS*, así como también de archivos locales.

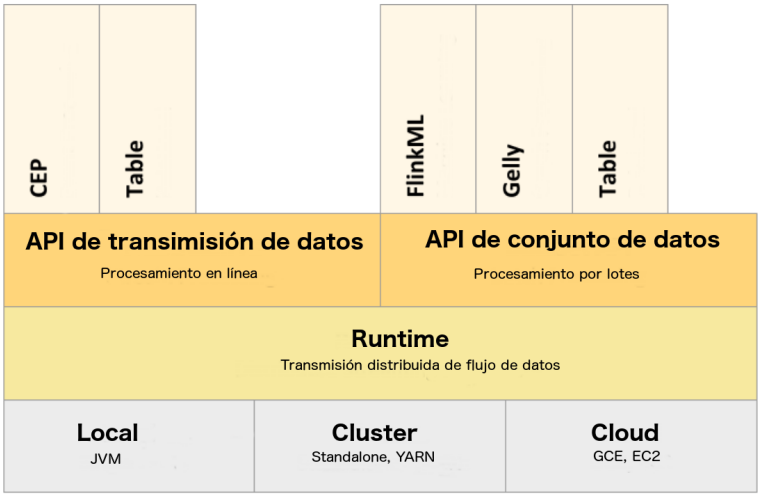


Fig. 3. Arquitectura Flink [2].

4. Análisis de entornos de trabajo

El objetivo de este artículo es presentar una comparativa de características que los entornos de trabajo más usados para *Big Data* tienen en la implementación de algoritmos de aprendizaje automático.

4.1. Características generales

La Tabla 2 muestra las características generales a considerar que tienen los entornos de trabajo en cuanto a lenguajes de programación, formato de datos, procesamiento de datos, entre otros.

Como se observa en la Tabla 2 Spark tiene un mayor abanico de posibilidades en términos de lenguajes de programación, y la inclusión de librerías de aprendizaje automático facilitan la implementación. Debido a que Hadoop, no cuenta con una integración de algoritmos de aprendizaje automático se tiene que incorporar la API necesaria para poder realizar una implementación en Java. Mientras que para Spark, se puede elegir el lenguaje de programación con el que se esté mayormente familiarizado.

Por otro lado, Flink, puede ser una alternativa viable de uso, debido a que cuenta con algoritmos de aprendizaje automático, así como también el modelo de programación aumentado de MapReduce a través de Transformaciones, sin embargo, en estudios como el de Inoubli [8] indica que el rendimiento depende del estado de la red, debido a que los resultados de las funciones MapReduce son enviados a través del cluster.

Tabla 2. Comparativa de entornos de trabajo.

	Hadoop	**Spark**	**Flink**
Formato de datos	clave-valor	RDD	clave-valor
Procesamiento de datos	Por lotes	Por lotes y en línea	Por lotes y en línea
Fuentes de datos	HDFS	HDFS, DBMS y KAFKA	KAFKA, mensajes, archivos
Modelo de Programación	MapReduce	Transformaciones y Acciones	Transformaciones
Lenguajes de Programación	Java	Java, Python y Scala	Java
Librerías de aprendizaje automático	No	Si	Si

4.2. Aprendizaje automático

Como anteriormente se mencionó la implementación de algoritmos propios de aprendizaje automático han sido escalados para dar solución a las características de *Big Data*, en consecuencia, es importante iniciar a considerar los entornos de trabajo que faciliten el uso.

Como se observa en la Tabla 2, Hadoop, Spark y Flink proporcionan una librería de código libre para el uso de algoritmos de aprendizaje automático, no obstante el resto de características hacen atractivo el uso de Hadoop, ya que aún cuando no incluye librerías propias del aprendizaje automático, es posible realizar una integración de código.

En la Tabla 3 se muestra los algoritmos de aprendizaje automático que Spark y Flink implementan. Como puede verse la cantidad de algoritmos de Spark es amplia y variada. Spark en comparación de Flink, cuenta con una mayor gama de algoritmos para implementar aprendizaje automático, adicionalmente, las bondades de distribuir el trabajo, hace que Spark sea una solución viable como herramienta para *Big Data*, permitiendo la implementación de algoritmos propios del entorno de trabajo, así como también el desarrollo e implementación de código.

Es importante resaltar la importancia de la ausencia de librerías de aprendizaje automático en Hadoop, ya que como se menciona anteriormente es un área que apoya fuertemente a *Big Data* en el análisis de datos para toma de decisiones.

Por otro lado, para el pre-procesamiento de datos Flink ofrece las siguientes funciones: Transformador de características polinomiales, el cual mapea un vector en el espacio de característica polinomial de grado d; Estándar de escalas, el cual escala el conjunto de datos, tal que todas las características tengan una medida y varianza especificadas; Escalas *MinMax*, el cual escala el conjunto de datos en un rango especificado por el usuario y *Validación cruzada* con las siguientes estrategias: 1. K-Fold 2. Train-Test 3. Multi-Random

Mientras que Spark, cuenta con una amplia gama de funciones para realizar extracción, transformación y selección de características como por ejemplo TF-

Tabla 3. Aprendizaje automático en entornos de trabajo.

Flink (*FlinkML*)	**Spark (*MLib*)**
▪ Aprendizaje Supervisado • SVM implementando el algoritmo de ascenso de coordenadas duales distribuidas. • Regresión lineal múltiple. ▪ Aprendizaje No Supervisado • KNN	▪ Clasificación: • Regresión logistica • Árbol de decisión. • Random forest. • Perceptron multicapa. • SVM lineal. • Naive Bayes ▪ Regresión: • Regresión lineal. • Árbol de decisión. • Random forest. ▪ Clustering: • K-means • LDA (*Latent Dirichlet allocation*).

IDF, Word2Vec, entre otras, para extracción; Tokenizer, StopWordsRemover, VectorIndexer, Normalizer, entre otras, para transformación y para selección esta VectorSlicer, RFormula.

4.3. Implementación de algoritmos

Parte importante del estudio aquí presentado es el análisis de funcionalidad que los entornos de trabajo ofrecen al momento de su instalación y al incorporar algoritmos desarrollados con código propio para su operación. Para ello, se utilizaron los equipos con las siguientes características: MacBook Pro, procesador Intel Core i5 a 2.6 GHz con memoria RAM de 8 GB DDR3 de 1600 MHz con sistema operativo OS X El capitán y MacBook Pro, procesador Inter Core i5 a 2.4 GHz con memoria RAM de 4 GB DDR3 con sistema operativo OS X Sierra.

La instalación de los entornos de trabajo no son engorrosas, para el sistema operativo de Mac OS, existen manuales y gestores de paquetes que lo facilitan. Sin embargo, para Hadoop en la versión más reciente 3.0.0, los puertos de acceso cambiaron, esto se puede consultar en `https://issues.apache.org/jira/browse/HDFS-9427`, es de suma importancia considerar esto para lograr tener acceso visual al sistema de archivos HDFS. Por otro lado, para desarrollar código, existen consideraciones importantes, debido a que la API que permite realizar trabajos para Hadoop cambió en el uso del objeto *Job*, así como también para el acceso de archivos, los cuales se encuentran en estado *deprecated*.

En cuanto a Spark, la implementación de algoritmos en el lenguaje de programación Python hace que sea entendible, la carga del conjunto de datos es por medio de una instancia del contexto de trabajo de Spark. Sin embargo hay que tener presente el formato del archivo de datos a cargar, debido a que

algoritmos como un árbol de decisión o una red neuronal pueden cargar datos en formato libsvm, pero no limita el uso de otros formatos de archivos de datos. La modificación de ejemplos incluidos en la instalación de Spark es sencilla en la carga de datos, sin embargo para modificar el algoritmo ya no es intuitivo, debido a que se puede integrar el trabajo de Java y Python.

5. Conclusión

El objetivo del estudio aquí presentado es exponer las bondades y facilidades que los entornos de trabajo tienen en la implementación de algoritmos de aprendizaje automático.

Como se mencionó anteriormente, los beneficios que ofrece Spark, hacen que este sea el entorno de trabajo de preferencia, debido a que cuenta con un mayor número de algoritmos de aprendizaje automático ya programados y disponibles para ser usados, así como también el uso de lenguajes de programación como Java, Scala o Python, amplían los beneficios, sin olvidar que por sus características el modelo de programación logra la optimización de recursos en cuanto a memoria, así como también el tipo de procesamiento de datos que ofrece en no sólo realizarlo por lotes, sino que también acepta el procesamiento en línea.

Sin embargo no se puede dejar atrás a Flink, debido a que también cuenta con algoritmos de aprendizaje automático que si bien la gama no es tan amplia, se tienen los más usados en el área. Al igual que Spark, permite el procesamiento de datos por lotes y en línea, siendo parte importante para tratar de dar solución a la velocidad de llegada en los datos. Como linea abierta de estudio, se encuentra la implementación de más algoritmos con los que no cuente Spark, entre ellos mezcla de expertos, reglas de decisión e inclusive el algoritmo KNN.

Otra linea abierta de estudio es realizar pruebas con grandes volúmenes de datos en cada entorno de trabajo realizando un estudio comparativo de optimización de recursos tanto de memoria como de tiempos de ejecución.

Para cada uno de estos entornos de trabajo se necesita conocimiento de las API's que logran la realización del trabajo distribuido de tareas, para lograr un aprovechamiento de recursos y cubrir con las características de *Big Data*, por lo que se requiere un tiempo de estudio para lograr resultados favorables en implementaciones propias de código.

Referencias

1. Amazon: Amazon aws. Obtenido de: https://aws.amazon.com/es/machine-learning/, Último acceso el 27 de Marzo del 2018
2. Apache: Apache flink. Obtenido de: https://flink.apache.org/, Último acceso el 20 de Marzo del 2018
3. Apache: Apache hadoop. Obtenido de: http://hadoop.apache.org/, Último acceso el 12 de Marzo del 2018
4. Apache: Apache spark. Obtenido de: https://spark.apache.org/, Último acceso el 11 de Febrero del 2018

5. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
6. Google: Google cloud. Obtenido de: https://cloud.google.com/?hl=es, Último acceso el 27 de Marzo del 2018
7. Guller, M.: Big data analytics with Spark: A practitioner's guide to using Spark for large scale data analysis. Springer (2015)
8. Inoubli, W., Aridhi, S., Mezni, H., Jung, A.: Big data frameworks: A comparative study. CoRR abs/1610.09962 (2016), http://arxiv.org/abs/1610.09962
9. Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., Nguifo, E.M.: An experimental survey on big data frameworks. Future Generation Computer Systems (2018), http://www.sciencedirect.com/science/article/pii/S0167739X17327450
10. Jebara, T.: Machine learning: discriminative and generative, vol. 755. Springer Science & Business Media (2012)
11. Kuncheva, L.I.: Using diversity measures for generating error-correcting output codes in classifier ensembles. Pattern Recognition Letters 26(1), 83–90 (2005)
12. Landset, S., Khoshgoftaar, T.M., Richter, A.N., Hasanin, T.: A survey of open source tools for machine learning with big data in the hadoop ecosystem. Journal of Big Data 2(1), 24 (Nov 2015), https://doi.org/10.1186/s40537-015-0032-1
13. L'Heureux, A., Grolinger, K., ElYamany, H.F., Capretz, M.: Machine learning with big data: Challenges and approaches. IEEE Access (2017)
14. Liu, X., Iftikhar, N., Xie, X.: Survey of real-time processing systems for big data. In: Proceedings of the 18th International Database Engineering & Applications Symposium. pp. 356–361. IDEAS '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2628194.2628251
15. Liu, X., Iftikhar, N., Xie, X.: Survey of real-time processing systems for big data. In: Proceedings of the 18th International Database Engineering & Applications Symposium. pp. 356–361. IDEAS '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2628194.2628251
16. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine learning: An artificial intelligence approach. Springer Science & Business Media (2013)
17. Mitchell, T.M., Carbonell, J.G., Michalski, R.S.: Machine learning: a guide to current research, vol. 12. Springer Science & Business Media (1986)
18. Mohanty, H., Bhuyan, P., Chenthati, D.: Big Data: A Primer. Springer India, 1 edn. (2015)
19. Morabito, V.: Big Data and Analytics: Strategic and Organizational Impacts. Springer International Publishing, 1 edn. (2015)
20. Prasad, B.R., Agarwal, S.: Comparative study of big data computing and storage tools: a review. International Journal of Database Theory and Application 9(1), 45–66 (2016)
21. Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S.: A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing 2016(1), 67 (May 2016), https://doi.org/10.1186/s13634-016-0355-x
22. Singh, D., Reddy, C.K.: A survey on platforms for big data analytics. Journal of Big Data 2(1), 8 (Oct 2014), https://doi.org/10.1186/s40537-014-0008-6

ISSN 1870-4069

Implementación de *kNN* sobre un *GPU* para predicción de la velocidad del viento

Hector Rodriguez Rangel[1], Glenn Della Rocca[1], Juan J. Flores[2], Luis A. Morales Rosales[3], Nora E. Cancela García[1]

[1]Instituto Tecnológico de Culiacán,
División de Estudios de Posgrado e Investigación,
Mexico

[2] Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Eléctrica,
División de Estudios de Postgrado, Michoacán,
Mexico

[3] CONACYT-Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Civil, Morelia, Michoacán,
Mexico

{hrodriguez, glenndellarocca}@itculiacan.edu.mx, juanf@umich.mx,
amorales@conacyt.mx, noracancela@gmail.com

Resumen. El algoritmo *kNN* se encuentra entre los primeros 10 mejores algoritmos de minería de datos. Puede ser utilizado en muy diversas áreas como regresor o pronosticador. Entre las desventajas que posee el algoritmo se encuentran los tiempos de ejecución largos y la dependencia a un valor óptimo de *k*. En este trabajo se propone la implementación del algoritmo *kNN* sobre un *GPU*, para dar solución a los problemas inherentes al mismo. Se hace uso del lenguaje de programación *Python* en conjunto con la librerías *PyCUDA* y *NumPy*, además del framework *CUDA* de *Nvidia* para la programación paralela. Se utilizó *kNN* como pronosticador y se evaluó con series de tiempo de la velocidad del viento. Los resultados experimentales demuestran grandes mejoras en los tiempos de ejecución de la implementación paralela con respecto a la versión secuencial, manteniendo la misma calidad en los resultados finales (i.e., error de predicción).

Palabras clave: Predicción, velocidad del viento, series de tiempo, KNN, GPU, paralelización.

Parallel Implementation of *kNN* on a *GPU* for Wind Speed Forecasting

Abstract. The *kNN* algorithm is amongst the top ten algorithms in data mining. It can be used on a wide range of areas for classification

or forecasting. Some of the disadvantages the algorithm has are the long execution times and the need for an optimum k value. This research proposes an implementation of the algorithm running on a graphics card (GPU), to fix the algorithm's problems. The implementation uses the *Python* programming language, the libraries *PyCUDA* and *NumPy* and the parallel part of the code is programmed with *Nvidia's CUDA* framework. The algorithm was used as a forecaster and evaluated with wind speed time series. The experimental results shows big improvements on execution times for the parallel implementation in contrast with the sequential one, keeping up the quality of the final results (i.e., error prediction).

Keywords: Forecasting, wind speed, time series, KNN, GPU, parallel.

1. Introducción

Los pronósticos son estimaciones sobre eventos futuros. La acción de pronosticar conlleva una predicción sobre algo que aún no existe. Son utilizados en diferentes campos y/o áreas (procesos de planeación, facturación, mantenimiento, etc.). Esto hace que sean útiles en campos como los negocios, la industria, la economía, las ciencias medio ambientales, ciencias sociales, política, medicina y finanzas, entre otros.

Existen dos enfoques para los métodos de pronóstico, el enfoque cualitativo y el enfoque cuantitativo [6]. El enfoque cualitativo hace uso de técnicas no estadísticas como la intuición y la experiencia del usuario para realizar estimaciones. Por otro lado, los métodos cuantitativos utilizan técnicas estadísticas y métodos de inteligencia artificial (IA), entre otros.

Dentro de los métodos de IA utilizados para hacer pronósticos se encuentran las redes neuronales artificiales (RNA), los k vecinos más cercanos (*kNN*, por sus siglas en inglés), máquinas de soporte vectorial (SVM en inglés), etc.

El algoritmo de k vecinos más cercanos se encuentra entre los diez mejores algoritmos de minería de datos de acuerdo con un estudio hecho por la IEEE en 2006 dentro de la Conferencia Internacional de Minería de Datos (ICDM) [9]. Pero este es poco utilizado para tareas de pronóstico, siendo descartado por algoritmos más complejos debido a problemas inherentes al mismo.

Entre los problemas que presenta *kNN* está la dependencia de un valor óptimo de k, la influencia nociva de atributos irrelevantes, el problema del ruido en los datos y los tiempos de respuesta largos.

En este trabajo se propone un algoritmo que realiza de manera paralela el método de vecinos más cercanos sobre una tarjeta aceleradora gráfica, aplicado al área de predicción. La implementación del algoritmo está basado en el framework *CUDA* de *Nvidia*.

El trabajo está estructurado de la siguiente manera: en la Sección 2 se encuentra un breve repaso de los pronósticos, las técnicas modernas de pronóstico, y trabajos que hacen uso de computación paralela. En la Sección 3 tenemos la descripción y funcionamiento del algoritmo *kNN* como pronosticador.

El modelo de cómputo distribuido de *Nvidia* (*CUDA*) se encuentra en la Sección 4. La descripción de la implementación del algoritmo de manera distribuida está en la Sección 5. Por último, en las Secciones 6 y 7 tenemos los resultados de los experimentos y conclusiones, respectivamente.

2. Trabajos relacionados

Pronosticar es una actividad que se realiza desde la antigüedad. Pero fue poco antes del siglo *XX* que se inició el análisis estadístico de las series de tiempo [11]. La teoría y métodos del análisis de series de tiempo son importantes bases y herramientas para los pronósticos. Ruey S. Tsay en [8] dice que los pronósticos son la razón por la que existen las series de tiempo y el análisis de éstas.

En el área de la industria y con mayor énfasis en el área de la energía, se han realizado trabajos de pronóstico desde hace años. Se utilizan los pronósticos para la planeación de la cantidad de materia prima disponible para las operaciones de las centrales eólicas, para la transmisión de la electricidad, la estabilidad y confiabilidad, entre muchos otros fines [10].

Las técnicas de pronóstico utilizadas para la predicción de la velocidad del viento son muy variadas. Dentro de las utilizadas, de acuerdo al trabajo de Zhao *et al.* [10], se encuentran: predicción numérica del clima (*NWP*, en inglés), métodos estadísticos, métodos de inteligencia artificial (IA), y métodos híbridos.

Entre los métodos de IA más utilizados en la literatura para el pronóstico del viento se encuentran las redes neuronales (*ANN*, en inglés), máquinas de soporte vectorial (*SVM*, en inglés), redes neuronales recurrentes (*RNN* en inglés), búsqueda de vecinos cercanos (*kNN*, en inglés), entre otros [12].

Se puede decir que el algoritmo de *kNN* es ampliamente utilizado para tareas de clasificación y regresión. Sin embargo este algoritmo cuenta con ciertos inconvenientes; uno de sus más grandes problemas es sus largos tiempos de ejecución.

En la literatura se encuentran trabajos que enfrentan este problema mediante la distribución de procesos del algoritmo. Entre otros, encontramos los trabajos de Garcia V. *et al.* [1], Kuang Q. y Zhao L. [2] y Liang S. *et al.* [3], que hacen uso de versiones tempranas del framework *CUDA*. Sin embargo, estas implementaciones tienen como limitantes principales el uso de funciones recursivas y la creación dinámica de hilos en el *GPU*, creando sobrecarga en las soluciones para algoritmos complejos.

El framework *CUDA* de la corporación *Nvidia* [7] es su propuesta para el desarrollo de aplicaciones de cómputo general sobre tarjetas aceleradoras gráficas (*GPU*, en inglés). Los *GPUs* son dispositivos masivamente paralelos, poseen miles de unidades de procesamiento aritmético-lógicas y grandes buses de datos para transferencias rápidas de información.

En la Figura 1 se muestra un diagrama que representa a grandes rasgos las diferencias entre las arquitecturas del *CPU* versus *GPU*. Se aprecia que la arquitectura de un *GPU* se enfoca enteramente en el procesamiento de grandes cantidades de operaciones.

Esto se debe a la gran cantidad de unidades de procesamiento Aritmético-lógicas (*ALU*, por sus siglas en inglés) que posee un solo chip gráfico.

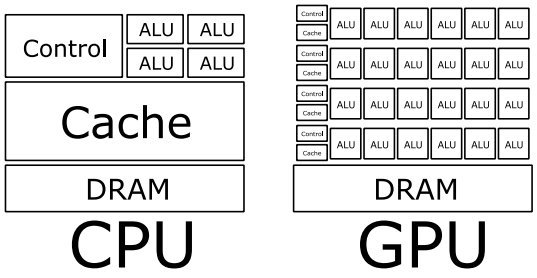


Fig. 1. Arquitectura de un *CPU* común en comparación con un *GPU*. Se observa que el *CPU* posee pocas unidades ALU, mientras que la arquitectura del *GPU* es diseñada pensando en el procesamiento de muchas operaciones simultáneas, con un pequeño *control* y *cache* a diferencia del *CPU* [7].

Teniendo en cuenta las complicaciones para la programación paralela, *Nvidia* desarrolla desde el año 2007 su framework *CUDA* (*Compute Unified Device Architecture*, en inglés). El framework es una extensión del lenguaje de programación C, que permite la utilización de los *GPUs* de la marca para tareas de cómputo general [4].

CUDA esta basado en la arquitectura paralela *SIMD* (*Single Instruction Multiple Data)*, pero haciendo uso de hilos es que la renombra como *SIMT* (*Single Instruction Multiple Threads*) [4].

Estos tipos de arquitectura hardware basan su funcionamiento en la ejecución de una única instrucción sobre un conjunto considerablemente grande de datos. Por lo que con una única unidad de *control* administra una cantidad grande de unidades de procesamiento (*ALUs*). Cada una de estas *ALUs* realiza de manera independiente operaciones en su porción específica de datos, completando la tarea de manera paralela.

3. K vecinos más cercanos

K Vecinos más Cercanos es un algoritmo de aprendizaje máquina (*ML* por sus siglas en inglés) de tipo no paramétrico y de aprendizaje perezoso (*Lazy Learning*). Lo que significa que no hace suposiciones sobre la distribución del conjunto de datos y no realiza ningún trabajo de pre-procesamiento del tipo *entrenamiento*.

Es un algoritmo de clasificación de objetos desconocidos. Utiliza un conjunto de datos etiquetados, que poseen atributos cuantificables. La clasificación se realiza midiendo la semejanza de un objeto a clasificar con el conjunto completo de datos, obteniendo los k objetos más similares, para definir una etiqueta a partir de estos.

En la Figura 2 se muestra un ejemplo de clasificación, en donde se aprecia que existen dos tipos de clase (Pentágonos y Cuadrados). Con un número k arbitrario se clasifica el objeto desconocido o hipótesis de acuerdo a los individuos más cercanos. Se puede observar si se utiliza la Moda, para $k = 3$ el objeto se clasificaría como pentágono y para $k = 7$ sería cuadrado. Mostrando la dependencia del algoritmo hacia un valor óptimo de k.

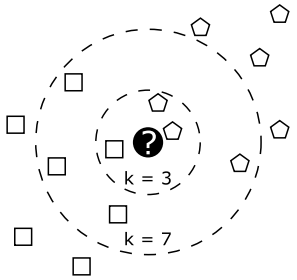


Fig. 2. Algoritmo de clasificación *kNN*. Se observa en el diagrama la dependencia hacia un valor óptimo de k del algoritmo. En este ejemplo en concreto, siendo $k = 3$ el objeto a clasificar sería un pentágono, mientras que con $k = 7$ sería un cuadrado.

La simplicidad del algoritmo permite su utilización para realizar regresión (la tarea de pronóstico se puede plantear como un problema de regresión). Además, esta simplicidad es la que hace que sea utilizado en un amplio rango de campos que van desde: visión computacional, geometría computacional, grafos, entre muchos otros.

El algoritmo asume que los datos se encuentran en un espacio de características y que los puntos de datos se pueden ubicar en un espacio métrico. Los datos pueden ser escalares o vectores multidimensionales, pero deben tener una noción de *distancia*; la métrica de la distancia *Euclidiana* es la más comúnmente utilizada.

Usando la regresión con *kNN*, el pronosticador de series de tiempo puede resolverse de manera similar a la clasificación. Al pronosticar se hace uso de valores continuos, por lo tanto, todos los valores calculados poseen un margen de error que es medible por distintas funciones.

Para utilizar *kNN* como pronosticador, en este trabajo se crea una base de datos. Esta base de datos (*BD*) generada en el ciclo de las Líneas 1 a 3 del Algoritmo 1 se hace a partir del recorrido de una ventana de w observaciones de la serie de tiempo. Cada una de estas ventanas genera una instancia nueva para la *BD*.

El siguiente paso en el pronóstico es el cálculo de las distancias. En el ciclo de las Líneas 4 a 6 del algoritmo se observa que el cálculo de la distancia se realiza entre la hipótesis y cada una de las instancias de la *BD*.

Como tercer paso, el pronosticador realiza el ordenamiento de las distancias, para la obtención de las k instancias más cercanas (Líneas 7 y 8 del algoritmo

respectivamente). Una vez con las distancias más cortas, se realiza un promedio de las etiquetas de las instancias seleccionadas, como se aprecia en la Línea 9. Este promedio es el resultado del algoritmo.

Algoritmo 1 pronosticador_kNN(serie_tiempo, hipotesis)

```
1: para cada w_observaciones de serie_tiempo hacer
2:     BD ← instancia //Creación de nueva instancia
3: fin para
4: para cada instancia de la BD hacer
5:     distancias ← calculo_distancia(instancia, hipotesis)
6: fin para
7: ordenamiento(distancias) //Llamada a función de ordenamiento
8: instancias_seleccionadas ← k_menores(distancias)
9: pronostico ← promedio(instancias_seleccionadas)
10: regresar pronostico
```

Entre los problemas del algoritmo se encuentran los tiempos largos de ejecución para conjuntos de datos de grandes dimensiones, su dependencia hacia un valor óptimo de k y problemas inherentes a las series de tiempo como los *outliers* o los valores nulos.

Los tiempos de ejecución largos del algoritmo al utilizar conjuntos grandes de datos hacen que el proceso no sea apto para soluciones en tiempo real. Esto es causado por la particularidad del aprendizaje perezoso del algoritmo, ya que no realiza trabajo alguno de pre-procesamiento o modelado de datos. Por lo que para cada pronóstico o clasificación es necesario realizar una gran cantidad de operaciones.

La selección de un valor óptimo de k es otra de las desventajas del método. Un número adecuado de k cambia el resultado obtenido por el algoritmo. Por lo general se selecciona un número arbitrario. Aunque también se puede obtener mediante técnicas de optimización, causando una sobrecarga aún mayor en los tiempos de ejecución.

4. Modelo *cuda* de *nvidia*

La programación paralela no es un invento reciente. Tiene ya bastantes años, incluso desde el surgimiento de los primeros computadores se crearon arquitecturas y modelos paralelos.

Pero fue el auge de la industria de los videojuegos que hizo que las tarjetas aceleradoras gráficas se convirtieran en más que un dispositivo de salida de vídeo. La necesidad de mayor poder de cómputo que requirieron los videojuegos impulsó a la fabricación de *GPUs* cada vez más potentes, abaratando los costos de las tarjetas y haciendo asequible utilizarlas como co-procesador del *CPU*.

Nvidia Corp., la más grande compañía productora de *GPUs*, decide comenzar el desarrollo de una arquitectura unificada para la programación de aplicaciones

de cómputo general [4]; con la generación G80 (o Tesla por su nombre comercial), *Nvidia* crea la primera versión de su framework *CUDA*.

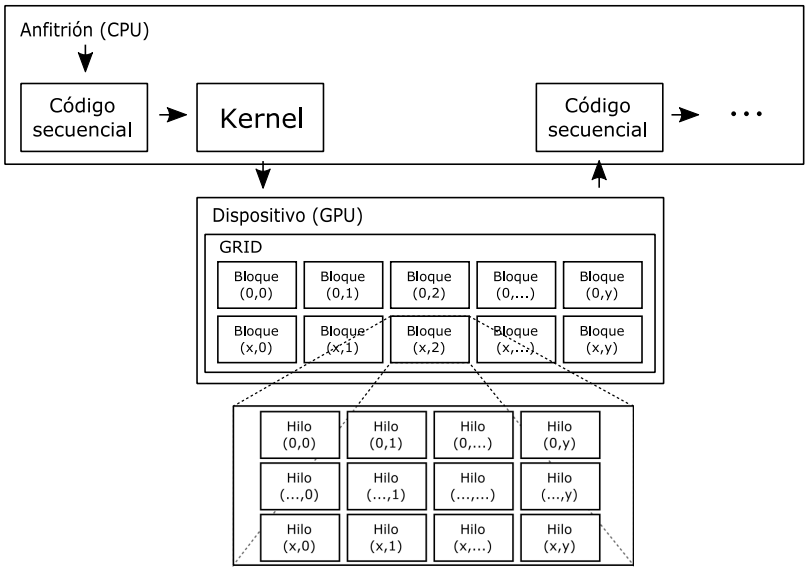


Fig. 3. Proceso de ejecución paralela sobre un *GPU*. El anfitrión es el encargado de las transferencias del código *Kernel* y de los datos hacia la memoria del dispositivo gráfico. Hecho esto, el control pasa al *GPU*, que realiza las operaciones indicadas en la función *Kernel* y regresa el control al *CPU*. Como último paso el anfitrión copia a la memoria principal los resultados dentro de la memoria del *GPU*.

El modelo *CUDA* de *Nvidia* es una colección de hilos ejecutando instrucciones en paralelo. La ejecución de estos hilos es completamente planificada por las unidades de control dentro del *GPU Nvidia*. El proceso de ejecución paralela se basa en la ejecución de las llamadas funciones *Kernel* que no son más que funciones ejecutadas paralelamente por conjuntos de hilos dentro del *GPU*. Estos *Kernels* son programados con el lenguaje de programación C y un conjunto de directivas especificadas por *CUDA*.

En la Figura 3 se aprecia el proceso de ejecución de un *Kernel* sobre un *GPU*. En el diagrama se muestra el modelo paralelo *CUDA*. A nivel software se hace referencia a una cuadrícula o *GRID*, la cual se dispone de acuerdo a las configuraciones del usuario. Se genera una serie de bloques de hilos (*Thread Blocks*, en inglés) y una configuración de bloque (*Block Dimension*, en inglés). Estos dos componentes constituyen la configuración del *Kernel* y son los que determinan el número de hilos a utilizar en la ejecución del código paralelo.

El proceso de ejecución paralela sobre un *GPU* es el siguiente: las funciones *Kernel* llamadas por el anfitrión (*CPU*) y los datos a utilizar son transferidos a través del bus *PCI-Express* hacia la tarjeta gráfica. El planificador de la tarjeta

administra y despacha la ejecución de hilos y datos para la ejecución de manera no secuencial. Los resultados obtenidos de los *Kernels* residen en la memoria del dispositivo y son transferidos como último paso hacia la memoria principal.

5. K vecinos cercanos paralelo

A partir de la programación del algoritmo secuencial se detectaron los cuellos de botella en su ejecución. Con esto realizamos la selección de la función del cálculo de la distancia y del ordenamiento, para su implementación sobre el *GPU*. En esta sección se describirá la implementación distribuida y las principales diferencias con la versión secuencial del algoritmo.

La comparación entre el elemento a clasificar o pronosticar con uno de los elementos del conjunto de *entrenamiento* es una tarea sencilla. Sin embargo con grandes conjuntos de datos el algoritmo realiza una gran cantidad de cálculos, requiriendo más poder computacional y tiempo de ejecución a medida que el número de datos crece o la dimensionalidad de los objetos aumenta.

En las siguientes subsecciones se definirá la arquitectura propuesta para el cálculo de las distancias, así como también para la función de obtención de resultados y de ordenamiento.

5.1. Función de cálculo de la distancia en paralelo

El cálculo de la distancia es la base central del algoritmo *kNN* puesto que evalúa de manera individual cada uno de los objetos dentro del conjunto de *entrenamiento* con cada uno de los objetos a clasificar o hipótesis a predecir.

Para el cálculo de las distancias se puede realizar de distintas maneras. En este trabajo se utilizó la distancia Euclidiana como medida de semejanza entre objetos. Esta distancia se encuentra definida en la ecuación (1), donde o es una instancia particular y h, la hipótesis a comparar:

$$D(o,h) = \sqrt{\sum_{j=1}^{v}(o_j - h_j)^2}, \tag{1}$$

donde v es el tamaño de los objetos. Para el cálculo de la distancia distribuida se generó, como se muestra en la Figura 4, una cantidad de hilos (T_1 a T_n) igual a la cantidad de instancias dentro del conjunto de entrenamiento. Cada hilo calculó la distancia Euclidiana de su respectiva instancia con cada una de las hipótesis (h_1 a h_m) de validación.

Los resultados de estos cálculos son almacenados en la memoria del dispositivo y utilizados como entrada para la función de ordenamiento.

El Algoritmo 2 muestra el proceso del cómputo de la distancia. Las entradas del algoritmo son n (Número de instancias), v (Número de atributos de las instancias), m (Número de hipótesis), $*E$ (Apuntador al vector de entrenamiento), $*V$ (Apuntador al vector de validación), $*D$ (Apuntador al vector de distancias).

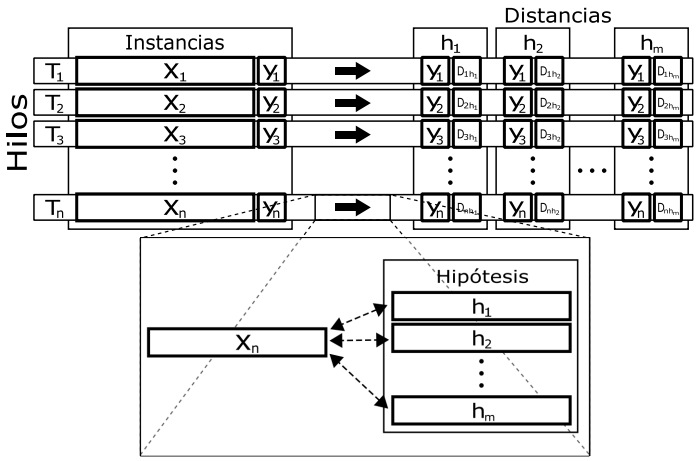


Fig. 4. Cálculo distancia distribuida. El diagrama muestra que se generó una cantidad de hilos igual a la cantidad de instancias en el conjunto de entrenamiento. Cada hilo computó la distancia entre su instancia y todas las hipótesis. Como resultado del cálculo se obtiene una matriz en memoria del *GPU*, de tuplas etiqueta-distancia.

Algoritmo 2 distancia_euclidiana_distribuida(n, v, m, *E, *V, *D)

1: $indice\_hilo \leftarrow (id\_bloque * tamaño\_bloque) + id\_hilo$ //Definición de ID global
2: //Ciclo GRID-STRIDE
3: **para** $indice\_hilo$ hasta n **hacer**
4: $fila\_entrenamiento \leftarrow v * indice\_hilo$
5: $fila\_resultado \leftarrow (m * 2) * indice\_hilo$
6: $etiqueta \leftarrow E[fila\_entrenamiento + v - 1]$
7: //Ciclo de cálculo de la distancia
8: **para** $i \leftarrow 0$ hasta m **hacer**
9: $suma \leftarrow 0.0$
10: $fila\_validacion \leftarrow v * i$
11: //Cálculo de la distancia euclidiana
12: **para** $j \leftarrow 0$ hasta $v - 1$ **hacer**
13: $suma \leftarrow (E[fila\_entrenamiento + j] - V[fila\_validacion + j])^2$
14: **fin para**
15: $indice\_resultado \leftarrow fila\_resultado + (i * 2)$
16: $D[indice\_resultado] \leftarrow etiqueta$
17: $D[indice\_resultado + 1] \leftarrow \sqrt{suma}$
18: **fin para**
19: **fin para**

La Línea 1 calcula el índice global de hilo. Este sirve para determinar la sección de datos específica con la que trabajará el hilo.

El ciclo externo de la función (Línea 3) sirve para trabajar con cualquier número de hilos. Es llamado ciclo *GRID-STRIDE* y permite utilizar cantidades de datos de un tamaño mayor al número total de hilos del *GPU* utilizado.

Cuando el hilo comienza su trabajo se realiza el cálculo de la posición de la fila de entrenamiento y de la de resultado (Líneas 4 y 5). Esto debido a que se utilizó un enfoque de vectores lineales.

En la Línea 6 se realiza la copia de la etiqueta de la instancia utilizada por el hilo. Con los *índices* de posición calculados, el algoritmo recorre todos y cada uno de los objetos del conjunto de validación (Ciclo en Líneas 8 a 18).

En la Línea 10 se calcula la posición del objeto de validación o hipótesis. Un último ciclo (Línea 12) obtiene la distancia euclidiana.

Para guardar los resultados se calcula la posición en el vector de distancias (Línea 15). Como paso final se escriben en el vector de distancias la etiqueta y distancia calculada (Líneas 16 y 17, respectivamente).

5.2. Función de obtención de resultados

Una vez concluido el cálculo de todas las distancias, se procede a obtener los resultados. El proceso de obtención de resultados se realiza a través de dos funciones *Kernel*. La función de obtención de resultados y la función de ordenamiento.

Algoritmo 3 obtener_resultados_distribuido(n, k, l, r, *D, *R)

1: $indice\_hilo \leftarrow (id\_bloque * tamaño\_bloque) + id\_hilo$ //Definición de ID global
2: //Ciclo GRID-STRIDE
3: **para** $indice\_hilo$ hasta n **hacer**
4: $columna \leftarrow indice\_hilo * 2$
5: $ancho\_matriz \leftarrow n * 2$
6: //Cálculo de la posición del pivote y ordenamiento parcial
7: $posicion\_pivote \leftarrow particion(l, r, c, w, D)$
8: $quicksort\_parcial\_distribuido(l, posicion\_pivote - 1, c, w, k, *D)$
9: $resultado \leftarrow 0.0$
10: //Ciclo de promedio
11: **para** $i \leftarrow 0$ hasta k **hacer**
12: $resultado \leftarrow D[(ancho\_matriz * i) + columna]$
13: **fin para**
14: $R[indice\_hilo] \leftarrow resultado/k$
15: **fin para**

La obtención de resultados como se aprecia en la Línea 1 del Algoritmo 3, inicia con el cálculo del índice global, explicado en la sección anterior, para generar un ciclo *GRID-STRIDE* como se detalló anteriormente.

El cálculo de la columna a ordenar por el hilo se realiza en la Línea 4 y el ancho de la matriz para la ubicación de la columna por parte de la función de ordenamiento se realiza en la Línea 5. Se procede a continuación al ordenamiento por medio de la llamada a la selección del elemento pivote de la columna seleccionada (Línea 7).

En la Línea 8 se observa la llamada recursiva hacia la parte izquierda o inferior del vector a ordenar. Esta función será descrita en la sección siguiente. Con el trabajo de ordenamiento concluido de la función *Quick Sort Parcial* [5], el hilo calcula la etiqueta a través del promedio de las etiquetas de los k objetos más cercanos (Línea 11).

Y por último el hilo en ejecución guarda en la posición adecuada del vector de resultados la etiqueta generada (Línea 14 del algoritmo).

5.3. Función de ordenamiento

Para la función de ordenamiento se utilizó la algoritmo *Quick Sort*, un algoritmo rápido y bastante utilizado en la literatura para operaciones de ordenamiento. Pero no se utilizó el algoritmo original, se utilizó una variante llamada *Quick Sort Parcial* [5].

El algoritmo *Quick Sort Parcial* al igual que su contraparte secuencial hace uso de la premisa, divide y conquista. Se seleccionó la variante parcial porque solo son necesarios los primeros k resultados ordenados correctamente. Con esto se evitan movimientos y comparaciones innecesarias, reduciendo aún más el tiempo de ejecución. Aún y cuando el orden de magnitud no cambia ($O(n\ log\ n)$ en el peor de los casos), el tiempo de ordenamiento se reduce en el caso promedio.

Como se aprecia en la Figura 5, para el modelo distribuido del ordenamiento se crearon una cantidad de hilos igual al número de hipótesis del conjunto de *validación*. Esto para el ordenamiento en paralelo de cada una de las columnas de tuplas etiqueta-distancia resultado del *Kernel* de la distancia distribuida.

En versiones iniciales, *CUDA* no permitía la ejecución de funciones recursivas, ni la invocación de hilos hijo que realizaran trabajo de manera más eficiente. Para realizar trabajos complejos era necesario regresar el control al *CPU* frecuentemente, con la consecuente sobrecarga en llamadas y manejo de memoria. Así, en operaciones complejas era difícil hacer un uso óptimo del paralelismo del *GPU*.

El *Kernel* de ordenamiento se encuentra definido en el Algoritmo 4 y como entradas del mismo tenemos: l (Posición izquierda del vector), r (Posición derecha del vector), c (Índice de la columna de datos a ordenar),w (Ancho de la matriz), k (Tamaño de k), $*D$ (Apuntador al vector de distancias).

El algoritmo utiliza ordenamiento por selección cuando el tamaño de datos a ordenar cae por debajo de un límite arbitrario de 32 datos (Línea 2). Esto para evitar una recursión más profunda y la consecuente sobrecarga que esto conlleva.

Si se utiliza *Quick Sort* (Línea 5), el algoritmo hace uso de invocaciones a *Kernels* hijo. La función de partición divide a la mitad el vector y utiliza de pivote el elemento central. Realiza el intercambio de valores de un extremo a otro y regresa la posición del elemento central, todo esto en la línea 7 del algoritmo con la invocación a la función de partición.

Una vez terminado este proceso, se procede recursivamente a la mitad más pequeña de la columna a ordenar (Línea 10). Se continua así hasta que el pivote divido entre 2 sea menor que k (Línea 9). Esto nos permite ordenar correctamente los primeros k elementos y obtener un resultado correcto.

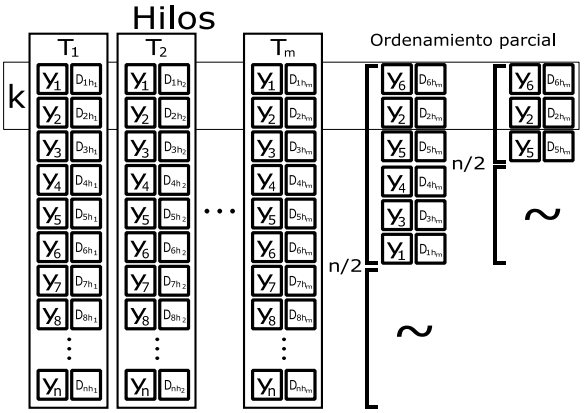


Fig. 5. *Quick Sort Parcial*. En el diagrama se muestra una cantidad de hilos generados igual a la cantidad de columnas dentro de la matriz de tuplas del paso anterior. Cada hilo ejecuta el algoritmo de *Quick Sort Parcial* para el ordenamiento de los datos, resultando en un ordenamiento óptimo de las k instancias más cercanas de cada una de las hipótesis. $\sim$ significa que los datos en ese segmento son mayores que los k menores, por consiguiente ya nos nos interesa ordenarlos.

Algoritmo 4 quicksort_parcial(l, r, c, w, k, *D)

```
1: //Comprobación del tamaño mínimo para el ordenamiento por selección
2: si (r − l) <= 32 entonces
3:    ordenamiento_seleccion(l, r, c, w, D)
4:    regresar
5: fin si
6: si l < r entonces
7:    posicion_pivote ← particion(l, r, c, w, D) //Cálculo del pivote y ordenamiento
8:    //Llamadas recursivas a las particiones del vector
9:    si r/2 > k entonces
10:      quicksort_parcial(l, posicion_pivote − 1, c, w, k, ∗D)
11:   si no
12:      quicksort_parcial(l, posicion_pivote − 1, c, w, k, ∗D)
13:      quicksort_parcial(posicion_pivote + 1, r, c, w, k, ∗D)
14:   fin si
15: fin si
```

6. Resultados

Los datos utilizados durante la experimentación fueron obtenidos de distintos puntos del estado de Michoacán, México. Estos se recopilaron en estaciones meteorológicas con el uso de dispositivos llamados anemómetros. Los cuales son dispositivos que sirven para la medición de la velocidad del viento en intervalos.

Se utilizaron 4 series de tiempo de diferentes tamaños: 8 mil, 22 mil, 32 mil y 43 mil datos (El Fresno, Malpaís, Melchor Ocampo y Markazuza, respectiva-

mente). Para probar la eficiencia, y velocidad de la propuesta presentada en este documento, 210 diferentes configuraciones por serie de tiempo fueron utilizadas para cada serie de tiempo descrita anteriormente.

Las configuraciones utilizadas fueron las siguientes, cabe destacar que fueron tomadas de manera arbitraria, en base a experiencia propia de los autores:

- Para el tamaño del vector de atributos w (ventana de predicción) se utilizaron los valores de : [2, 3, 5, 10, 15, 20, 25]
- Para la división entre el conjunto de entrenamiento y de validación, se utilizó : [70-30 %, 80-20 %, 90-10 %, 95-5 %, 99-1 %]
- Para el número vecinos k se utilizó : [1, 3, 5, 10, 15, 20]

Los resultados de los experimentos realizados se observa en la Tabla 1. La cual muestra el promedio de la aptitud (MSE) de cada serie de tiempo descrita anteriormente. Las diferencias menores entre el *knn* sobre *CPU* vs *GPU* observados en los resultados de la Tabla 1 se deben la diferencia de arquitectura hardware que poseen el *CPU* y el *GPU*. *Nvidia* fabricante de la tarjeta define una velocidad para cálculos de 32 bits y otra velocidad para el cálculo de valores de 64 bits, siendo este último significativamente más lento que el primero. Y aunque el *GPU* sea capaz de procesar información de 64 bits, se prefirió utilizar direcciones de memoria de 32 bits para no mermar el rendimiento de la tarjeta.

Tabla 1. Comparación entre *CPU* y *GPU*. Diferencias en la precisión entre la arquitectura secuencial y paralela.

MSE			
Serie	**Tamaño**	**CPU**	**GPU**
El Fresno	8 mil datos	**0.00636108**	0.007125687
Malpais	22 mil datos	**0.003438114**	0.003823412
Melchor Ocampo	32 mil datos	**0.002532585**	0.002834963
Markazuza	43 mil datos	**0.00024559**	0.000282856

Por otro lado en cuestión de tiempo de ejecución, la Tabla 2 muestra los tiempos promedio de ejecución de los 210 diferentes experimentos de cada serie de tiempo. De la Tabla 2 se observa que a medida que el número de datos aumenta, el tiempo de ejecución tanto en la implementación secuencial como en la paralela aumenta, aunque el incremento de la implementación secuencial (*CPU*) es considerablemente mayor. Además, se observa que los experimentos ejecutados sobre el *GPU* realiza de manera casi instantánea; para el experimento de Markazuza sobre *CPU* el tiempo de ejecución es de 45 minutos aproximandamente.

Tabla 2. Comparación entre *CPU* y *GPU*. Tiempo en segundos de la arquitectura secuencial y paralela.

Tiempos			
Serie	**Tamaño**	**CPU**	**GPU**
El Fresno	8 mil datos	130.614 s	**0.150 s**
Malpais	22 mil datos	601.143 s	**0.478 s**
Melchor Ocampo	32 mil datos	1510.716 s	**1.550 s**
Markazuza	43 mil datos	2732.124 s	**1.922 s**

Al comparar ambas implementaciones de kNN (*CPU* vs *GPU*), se observa que se reduce considerablemente el tiempo de ejecución, pero se tiene perdida en la precisión del modelo esto debido a las características del hardware.

Finalmente cabe señalar que los experimentos secuenciales fueron ejecutados sobre un servidor HP con un procesador Intel Xeon E5-2603V4 a 1.7GHz con 8 GB DDR4. Mientras que para la parte no secuencial se utilizó sobre el mismo servidor una tarjeta gráfica EVGA GTX 1080 TI.

7. Conclusiones

La utilización de valores de 32 bits con respecto a los de 64 bits no altera significativamente los resultados obtenidos, como se aprecia en la Tabla 1. Por otro lado, el cambio más significativo se dio en los tiempos de ejecución del algoritmo. En la Tabla 2 se aprecia claramente que en todos los casos la implementación sobre el *GPU* venció con un gran margen a la implementación secuencial.

Otro resultado a tener en cuenta es la reducción del error con el aumento de datos registrado con las distintas series de tiempo. Esto se demuestra que a mayor número de datos de entrenamiento, la precisión de los resultados mejoran con creces. Pero cabe mencionar que el tiempo de ejecución de ambas arquitecturas aumenta.

Las configuraciones del modelo influyen en la precisión de éste. Pero sin importar el modelo, con un mayor número de datos es posible obtener mejores resultados. La mejor opción entonces, es la arquitectura paralela debido a que con los tiempos de ejecución reducidos es posible utilizar una mayor cantidad de datos.

Los resultados demuestran que el algoritmo es capaz de obtener excelentes resultados en la predicción de variables del viento. No por ello limitado al pronóstico del viento, claro está. Sin embargo, se tiene que mencionar que la sobrecarga en tiempo de desarrollo del algoritmo sobre la arquitectura *CUDA*,

es solo un pequeño precio a pagar con respecto a los grandes beneficios que aportan en rendimiento los *GPUs*.

Referencias

1. Garcia, V., Debreuve, E., Barlaud, M.: Fast k nearest neighbor search using gpu. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on. pp. 1–6. IEEE (2008)
2. Kuang, Q., Zhao, L.: A practical gpu based knn algorithm. In: International symposium on computer science and computational technology (ISCSCT). pp. 151–155. Citeseer (2009)
3. Liang, S., Wang, C., Liu, Y., Jian, L.: Cuknn: A parallel implementation of k-nearest neighbor on cuda-enabled gpu. In: Information, Computing and Telecommunication, 2009. YC-ICT'09. IEEE Youth Conference on. pp. 415–418. IEEE (2009)
4. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: Nvidia tesla: A unified graphics and computing architecture. IEEE micro 28(2) (2008)
5. Martınez, C.: Partial quicksort. In: Proc. 6th ACMSIAM Workshop on Algorithm Engineering and Experiments and 1st ACM-SIAM Workshop on Analytic Algorithmics and Combinatorics. pp. 224–228 (2004)
6. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to time series analysis and forecasting. John Wiley & Sons, New Jersey, USA, second edn. (2015)
7. Nvidia: Cuda 9.0 compute unified device architecture programming guide. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html, last accessed: 2018-02-15
8. Tsay, R.S.: Time series and forecasting: Brief history and future research. Journal of the American Statistical Association 95(450), 638–643 (2000)
9. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al.: Top 10 algorithms in data mining. Knowledge and information systems 14(1), 1–37 (2008)
10. Wu, Y.K., Hong, J.S.: A literature review of wind forecasting technology in the world. In: Power Tech, 2007 IEEE Lausanne. pp. 504–509. IEEE (2007)
11. Yule, G.U., et al.: Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. Phil. Trans. R. Soc. Lond. A 226(636-646), 267–298 (1927)
12. Zhao, X., Wang, S., Li, T.: Review of evaluation criteria and main methods of wind power forecasting. Energy Procedia 12, 761–769 (2011)

ISSN 1870-4069

Ensamble de clasificadores para determinar el perfil académico del estudiante usando árboles de decisión y redes neuronales

Maricela Quintana López, José Martín Flores Albino, Saúl Lazcano Salas, Víctor Manuel Landassuri Moreno

Centro Universitario UAEM, Valle de México, México

{mquintanal, jmfloresa, slazcanos, vmlandassurim}@uaemex.mx

Resumen. En este artículo, se propone el uso de un ensamble de clasificadores para determinar el perfil académico del estudiante, basado en su promedio general y en datos relacionados a los factores educativos: actividades de estudio, formas de aprendizaje y hábitos de estudio. Los datos usados se obtuvieron del cuestionario socioeconómico aplicado a los estudiantes del Centro Universitario UAEM Valle de México, asignándole la clase correspondiente de acuerdo con su promedio general. Las clases se definieron como excelente, bueno y regular. Para cada grupo de factores, se utilizó el algoritmo C4.5 para generar el clasificador correspondiente. El ensamble de clasificadores fue entonces diseñado utilizando una red neuronal artificial. La red neuronal recibe como entrada la clasificación asignada por los tres clasificadores y es entrenada para asignar la clase correcta usando un subconjunto de los datos. Se observa en los resultados que el ensamble propuesto tiene mejor desempeño comparado con los clasificadores independientes.

Palabras clave: Ensamble de clasificadores, árboles de decisión, redes neuronales artificiales.

Ensemble of Classifiers to Determine Student Academic Profile Using Decision Trees and Neural Networks

Abstract. In this paper, an ensemble of classifiers is proposed to determine student academic profile using decision trees and neural networks, based on his grade point average and data related to educative factors: study activities, learning forms and study habits. The data came from a socioeconomic questionnaire applied to the students of the University Center UAEM Valley of Mexico, and the corresponding class was assigned based on his grade point average. The classes were defined as: excellent, good, and regular. For each group of factors, the C4.5 algorithm was applied to build the corresponding

classifier. The ensemble of classifiers was designed using a neural network which receives the classes assigned by the three classifiers as input and it was trained to assign the corresponding class using a data subset. It is observed that the performance of the ensemble is better than the performance obtained for each independent classifier.

Keywords: Ensemble of classifiers, decision trees, artificial neural network.

1. Introducción

Una de las preocupaciones en las universidades es la deserción y el bajo desempeño académico de los estudiantes, es sabido que en esto influyen muchos factores, entre estos los sociales, económicos, educativos, académicos e institucionales, por citar algunos. En este trabajo se busca generar un ensamble de clasificadores para determinar el perfil académico de un estudiante, con base en su promedio general y usando como discriminantes los factores educativos como son: actividades de estudio, formas de aprendizaje y hábitos de estudio. La base de datos consiste en los registros de los estudiantes del Centro Universitario UAEM Valle de México que ingresaron en los años 2008, 2009 y 2010 en las diferentes carreras que se imparten.

Una estrategia popular es probar diferentes algoritmos para generar los modelos, evaluarlos y elegir el que proporcione los mejores resultados, esto es el que tenga el menor error en la predicción de la clase de una instancia desconocida. En contraste, un método de ensamble construye o utiliza varios clasificadores y los combina [1]. Usualmente se utilizan clasificadores construidos a partir de algoritmos diferentes sobre los mismos datos. Sin embargo, debido a que al hacerlo de esta manera, en este trabajo no llevó a buenos resultados, se optó por dividir los datos por grupos de factores y se aplicó el mismo algoritmo C4.5 construyéndose 3 clasificadores y generando el ensamble con ellos.

El documento contiene la siguiente estructura, en la sección 2, se presenta la metodología empleada para realizar el presente trabajo, mientras que en la sección 3 se muestra de manera muy general los algoritmos de minería de datos y redes neuronales empleados, así como el método de ensamble utilizado. En la sección 4, se presentan los experimentos y resultados obtenidos; finalmente en la sección 5, se muestran las conclusiones y trabajo futuro.

2. Metodología

Para conducir la investigación se realizaron etapas en las que la metodología utilizada fue la del proceso de extracción del conocimiento, conocido como KDD (Knowledge Discovery in Databases) [2], ver Figura 1 con esto:

a. Se creó y probó un clasificador considerando todos los factores educativos.

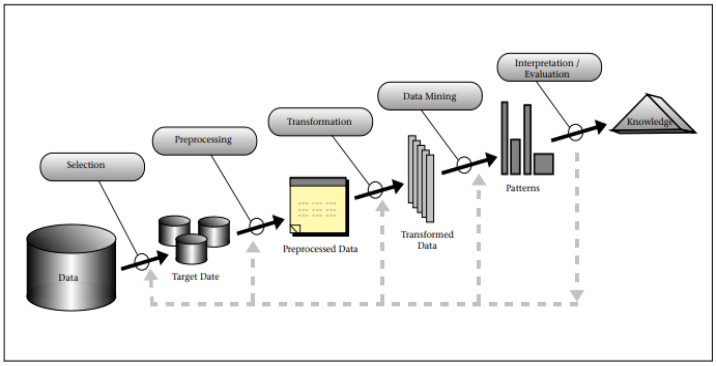


Fig. 1. Etapas del Proceso de Extracción del Conocimiento [2]

Tabla 1. Actividades de estudio.

Me reúno con mis compañeros para preparar un examen
Me reúno con mis compañeros para elaborar una tarea o un trabajo en grupo
Al iniciar, identifico lo que necesito estudiar y elaboro un plan de trabajo
Reviso qué es lo que recuerdo de lo que estudié
Identifico los conceptos que aún no he comprendido
Cuando no entiendo algo busco información esclarecedora
Estudio principalmente con monografías
Estudio principalmente con mis apuntes de clase
Estudio principalmente con el libro de texto de la asignatura
Estudio principalmente con los apuntes de mis compañeros
Uso enciclopedias, diccionarios o atlas
Uso computadora o Internet para estudiar, hacer tarea o resolver un examen

b. Se generaron clasificadores con bloques de factores educativos, es decir, actividades de estudio, formas de aprendizaje y hábitos de estudio y se probaron los modelos.
c. Se generó el ensamble utilizando redes neuronales artificiales.
d. Se analizaron y compararon los resultados.

2.1. Selección de los datos

Los datos fueron obtenidos del estudio socioeconómico perteneciente a los estudiantes de las generaciones 2008-2013, 2009-2014 y 2010-2015 de todas las carreras del Centro Universitario UAEM Valle de México. Los datos seleccionados corresponden a la sección de factores educativos, que se aplica a los estudiantes, específicamente a las actividades de estudio, ver tabla 1, formas de aprendizaje, ver tabla 2 y a los hábitos de estudio, ver tabla 3. Estos datos son relacionados con los datos de control escolar, específicamente con el promedio general del estudiante.

La escala de respuesta a las preguntas se divide en: *Nunca*, *Normal*, *Siempre*, excepto en aquellas donde se solicita un número de horas o de libros, que son las primeras 3 preguntas de los hábitos de estudio.

Tabla 2. Formas de aprendizaje.

Aprendo más cuando trabajo con otros compañeros
Es de gran ayuda que todos aporten ideas cuando trabajo en grupo
Estudio para asegurar económicamente mi futuro
Estudio para obtener un buen trabajo
Estudio para aprender más
Estudio para vivir mejor
Confío en que puedo entender lo que estudio inclusive los textos más difíciles
Confío en que puedo realizar un excelente trabajo en mis tareas y en exámenes
Tengo seguridad en que domino las habilidades que me enseñaron
Aprendo rápidamente en la mayoría de las asignaturas
Soy competente en la mayoría de las asignaturas
Resuelvo bien los exámenes en la mayoría de las asignaturas
Me gusta trabajar con otros compañeros
Solamente leo cuando tengo la obligación de hacerlo
La lectura es uno de mis pasatiempos favoritos
Me gusta comentar los libros con otras personas
Me cuesta trabajo terminar de leer un libro
Me gusta que me regalen libros
La lectura me parece una pérdida de tiempo
Disfruto el visitar librerías o bibliotecas
Solamente leo para obtener la información que necesito
Me cuesta trabajo sentarme a leer por mucho tiempo

Tabla 3. Hábitos de estudio.

¿Horas a la semana que estudia o hace tarea fuera del horario escolar?
¿Horas a la semana se dedica a leer sobre lo que le gusta o interesa?
Indique cuántos libros completos ha leído en los últimos 12 meses sin tomar en cuenta sus libros de texto
Libros de literatura (novela, teatro, poesía)
Libros de otros temas (ciencia, tecnología, economía, etc.)
Revistas
Periódicos
Historietas
Páginas de Internet

2.2. Preprocesamiento

En el archivo original la cantidad de instancias era de 3600, sin embargo, debido a que algunos estudiantes no contestaban una gran cantidad de preguntas, las instancias con información incompleta fueron eliminadas, quedando únicamente 1021 instancias.

Por otro lado, dentro de los datos, existen atributos que actúan únicamente como identificadores de las instancias, pero que no se utilizan para generar los clasificadores como el nombre del alumno, la edad, y el sexo.

2.3. Transformación

El perfil académico del estudiante se dividió en 3 clases, de acuerdo con el promedio general obtenido desde su ingreso y hasta el último semestre cursado. La tabla 4 presenta esta información, así como la cantidad de estudiantes que tienen el perfil.

Tabla 4. Distribución de alumnos por perfil académico.

Perfil Académico	Promedio General	Alumnos
Regular	[0.0,7.3]	168
Bueno	[7.4,8.7]	691
Excelente	[8.8,10]	162
	Total	**1021**

Tabla 5. Proporción de datos por perfil para los conjuntos de entrenamiento y prueba.

Conjunto	Perfil Académico			Total
	Regular	Bueno	Excelente	
Entrenamiento	136	553	129	**818**
Prueba	32	138	33	**203**
Total	**168**	**691**	**162**	**1021**

El conjunto de datos seleccionados, 1021 instancias o ejemplares, se dividió en dos conjuntos: el de entrenamiento con 818 instancias, y el de prueba con 203 instancias; con el fin de que todas las clases estuvieran representadas, se tomó aproximadamente el 20% de cada una para el conjunto de prueba y el resto de las instancias en el conjunto de entrenamiento (≈80%). La división de los datos se presenta en la tabla 5.

3. Minería de datos, redes neuronales y ensambles

A continuación se describe, de manera sucinta, lo referente a la minería de datos, y en especial, al algoritmo empleado en este trabajo; también se presentan los conocimientos básicos acerca de las redes neuronales y los ensambles.

3.1. Minería de datos

La minería de datos aparece como la tercera etapa del proceso de extracción del conocimiento, con ello se pueden realizar diferentes tipos de tareas que tiene como metas principales la predicción y la descripción [2]. De acuerdo con [3], la minería de datos se define como el proceso de encontrar información novedosa y comprensible a partir de los datos. En este trabajo, la información que se busca es la que indica que un alumno es excelente, regular o malo, y poder emplear, en un futuro, esta información tanto para clasificar a nuevos alumnos, como para entender qué actividades, hábitos y formas de aprendizaje definen a las diferentes clases, y con esta información crear

estrategias que permitan fomentarlas. El algoritmo de clasificación empleado en este trabajo es el C4.5., el cual es una mejora del algoritmo ID3 desarrollado por Ross Quinlan, permite trabajar con atributos numéricos y considera la información de la partición y el radio de ganancia, entre otras mejoras, para elegir los atributos ganadores; la salida es un árbol de decisión; Este algoritmo está implementado en el software libre WEKA de la Universidad de Waikato bajo el nombre de J48 [4] y es el que se utiliza en este trabajo.

3.2. Redes neuronales

Las redes neuronales artificiales son estructuras de cálculo que se basan en unidades relativamente simples de procesamiento, llamadas neuronas. Al interconectar las neuronas artificiales se multiplica su capacidad para representar conocimiento, entendido este último como una relación funcional compleja entre datos de entrada y datos de salida. La adquisición del conocimiento en una red neuronal depende del modelo de aprendizaje utilizado. Para problemas de clasificación se hace uso de un conjunto de aprendizaje que consiste en datos de entrada con su descriptor de clase conocido, a través de este conjunto se ajustan los parámetros de la red para que señalen la clase a la que se sabe que pertenecen. El proceso de actualización de los parámetros de la red neuronal es lo que se considera el entrenamiento. El entrenamiento hace uso de una medida del error de clasificación de la red (índice de desempeño) y por medio de una medida de la sensibilidad de los parámetros de la red al error, se realiza su adaptación, este suele basarse en el algoritmo llamado *backpropagation*.

Como se puede ver en [5], se presenta un panorama del uso de las redes neuronales en el análisis de grandes bases de datos. Las redes neuronales pueden construir modelos estadísticos no lineales a través los datos disponibles. El modelo va ajustándose para cumplir con el nivel del índice de desempeño establecido. Las tareas de clasificación consisten en asociar un grupo de patrones de entrada a la categoría que pertenecen y que los identifica o describe.

3.3. Métodos de ensamble

Los métodos de ensamble o combinación de modelos, surgen con el propósito de mejorar la precisión de las predicciones. Un ensamble contiene un número de aprendices (modelos base) que, cuando son del mismo tipo son llamados homogéneos y si son de diferentes, heterogéneos. La característica es que estos aprendices no tienen un buen desempeño. El ensamble se genera utilizando otro algoritmo que combina los aprendices, ejemplo de éstos son el voto mayoritario, la tabla de decisión y las redes neuronales [1].

La precisión obtenida por el ensamble o combinación de modelos, generalmente supera la precisión de cada componente. En [6] se presenta el esquema de combinación de modelos que tienen árboles de decisión como base, ver Figura 2. Se observa que los datos son dados a los árboles de decisión y cada uno entrega su clasificación, se realiza entonces la combinación y se entrega la predicción combinada.

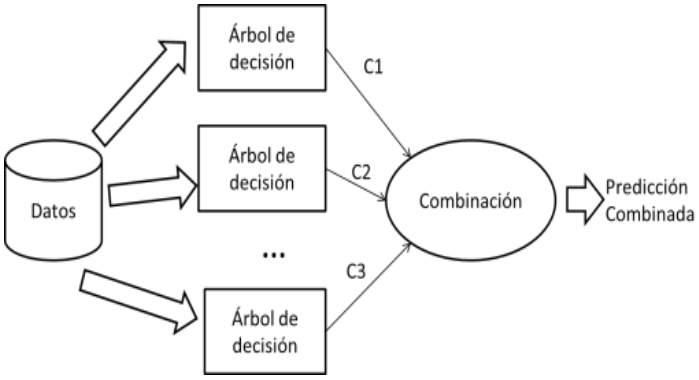


Fig. 2. Combinación de modelos usando árboles de decisión como base [6]

Los ensambles de clasificadores se han utilizado en ámbitos muy diversos, por ejemplo en [7], analizan diversos clasificadores detectores de malware para sistemas Android que basan su análisis en el uso de los recursos de hardware de los diversos programas en ejecución para detectar patrones sospechosos que sean característicos de programas malware. Realizan un ensamble final de dichos clasificadores, logrando que la detección exitosa de malware se incremente de manera significativa, comparado contra los clasificadores individuales.

Otro ejemplo de uso de ensambles de clasificadores es el trabajo de [8], en el cual analizan mensajes en la red social Twitter para clasificar los mismos en cuatro categorías. Los ensambles que proponen logran una clasificación de los mensajes con un grado de éxito más elevado comparado con otros clasificadores usados en esta tarea. Además, los ensambles que proponen tienen la característica de trabajar muy bien en situaciones donde el conjunto de entrenamiento no es lo suficientemente grande.

El trabajo presentado en [9], se enfoca en la detección de leucemia analizando imágenes de las muestras sanguíneas; dichas imágenes las pasan a través de una red neuronal convolutiva para la extracción de características, reducen el número de las mismas y seleccionan las más significativas y finalmente, las analizan a través de un ensamble de 3 clasificadores *(Support vector machine, multilayer perceptron, random forest*) logrando una detección exitosa prácticamente del 100% de los casos analizados, destacando la reducción del tiempo de análisis comparado con técnicas de detección de leucemia tradicionales.

En este trabajo se utiliza un ensamble de clasificadores para mejorar el desempeño de los clasificadores individuales; el ensamble se realizó utilizando Redes Neuronales.

4. Experimentos y resultados

A continuación se presentan los experimentos realizados y los resultados obtenidos, iniciando con la generación del clasificador único que utiliza todos los datos, prosiguiendo con los clasificadores individuales y terminando con el ensamble.

4.1. Generación y prueba de un clasificador considerando todos los factores educativos

El primer clasificador se generó utilizando el algoritmo C4.5. El modelo se generó a partir del conjunto de entrenamiento (818 instancias). Los resultados son presentados en la tabla 6, en términos de porcentajes de acierto y error en la clasificación de las instancias, de igual manera se presenta la matriz de confusión.

Tabla 6. Resultados del clasificador generado con el algoritmo C4.5 (J48 WEKA).

Clases	**Matriz de Confusión**			**Acierto**	**Error**
	A	**B**	**C**	89.12%	10.88%
a = Bueno	**538**	9	6	818 instancias	
b = Regular	29	**105**	2	729 clasificadas correctamente	
c = Excelente	35	8	**86**	89 clasificadas incorrectamente	

Tabla 7. Resultados del clasificador en el conjunto de prueba.

Clases	**Matriz de Confusión**			**Acierto**	**Error**
	A	**B**	**C**	58.62%	41.38%
a = Bueno	**102**	17	19	203 instancias	
b = Regular	19	**11**	3	119 clasificadas correctamente	
c = Excelente	24	2	**6**	84 clasificadas incorrectamente	

Al utilizar el modelo generado para clasificar las instancias en el conjunto de prueba, se puede observar que baja la eficiencia del clasificador de 89.12% a 58.62%. Los resultados del conjunto de prueba son presentados en la tabla 7.

4.2. Generación y prueba de clasificadores usando los factores educativos por separado: actividades de estudio, formas de aprendizaje y hábitos de estudio

Los datos fueron separados en actividades de estudio, formas de aprendizaje y hábitos de estudio. Con cada grupo se generó un clasificador usando el algoritmo C4.5 sobre el conjunto de entrenamiento y se calculó su desempeño con el conjunto de prueba. En la tabla 8 se presentan los resultados, es posible notar que, si bien los clasificadores mostraron un buen desempeño con el conjunto de entrenamiento, al probarlos la eficiencia al clasificar descendía.

Tabla 8. Resumen del desempeño de los clasificadores.

Factor Educativo	Entrenamiento		Prueba	
	Acierto	**Error**	**Acierto**	**Error**
Actividad de estudio	71.64	28.36	65.52	34.48
Formas de aprendizaje	75.18	24.82	64.53	35.47
Hábitos de estudio	78.24	21.76	62.07	37.93
Todos los factores	89.12	10.88	58.62	41.38

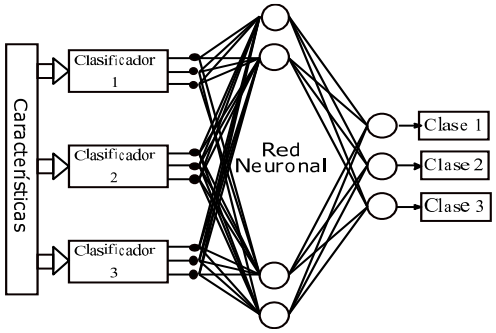


Fig. 3. Arquitectura de la Red Neuronal: 9 entradas y bias, 10 neuronas con función de activación *Logsig* y 3 neuronas de salida con función de activación *Logsig.*

A pesar de que en todos los factores educativos, los clasificadores individuales bajan el porcentaje de aciertos al utilizarse en el conjunto de prueba, estos tienen mejor desempeño que el clasificador único que contempla a todos los atributos (último renglón de la tabla 8).

4.3. Ensamble de clasificadores usando una red neuronal artificial

Arquitectura de la red neuronal (RN). La salida de los clasificadores se representa por un vector 3x1. P= *[Clase 1, Clase 2, Clase 3]*^{T}, de manera que la RN tiene nueve entradas. Las nueve entradas pasan a la primera capa de 10 neuronas, con de una entrada independiente de compensación (bias). La función de activación es de tipo logsig, (1). La capa final tiene tres neuronas con bias y función de activación logsig. El objetivo es que la salida represente la clase por medio del vector S= *[Clase1, Clase 2, Clase3]*^{T}, ver Figura 3:

$$Logsig(x) = \frac{1}{1-e^{-x}}. \quad (1)$$

Algoritmo de entrenamiento. Se utiliza el algoritmo de entrenamiento "El Gradiente Escalado Conjugado" [10]. El índice de desempeño es:

$$H_{y'}(y) = \sum_i y_i' \log(y_i), \quad (2)$$

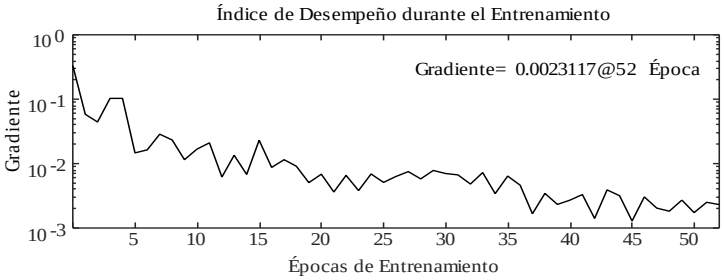


Fig. 4. Comportamiento del Índice de desempeño.

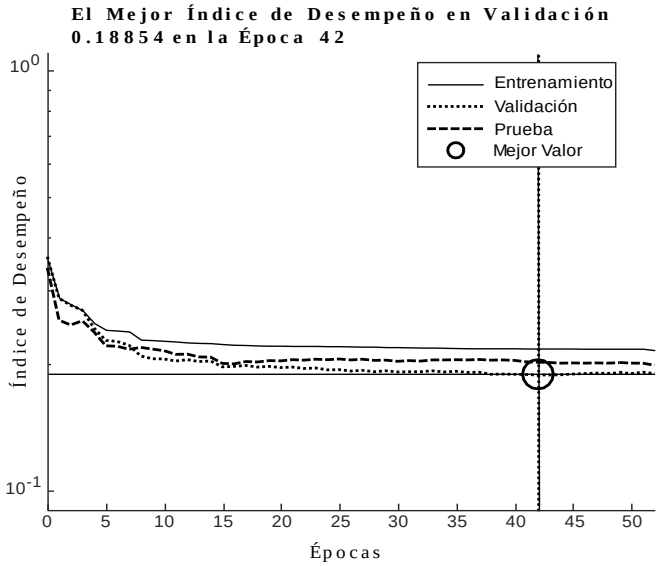


Fig. 5. Comportamiento del índice de desempeño con respecto a las fases de Entrenamiento, Validación y Prueba de la Red Neuronal.

donde:

y_i'es la probabilidad de la clase i que se espera,

y_i'es la probabilidad estimada.

Resultados experimentales. Al realizar el entrenamiento durante 52 épocas se obtuvieron los siguientes resultados. En la figura 4, se presenta el índice de desempeño de cada época de entrenamiento. El cálculo de este índice es a través de (2), que en la última época tiene un valor de 0.00123117.

En la Figura 5, se muestran tres series de valores de índice de desempeño para el grupo de datos tomados para entrenamiento, validación y prueba. Destaca que en la época 42 se obtuvo el mínimo del índice de desempeño igual a *0.18854* para el conjunto

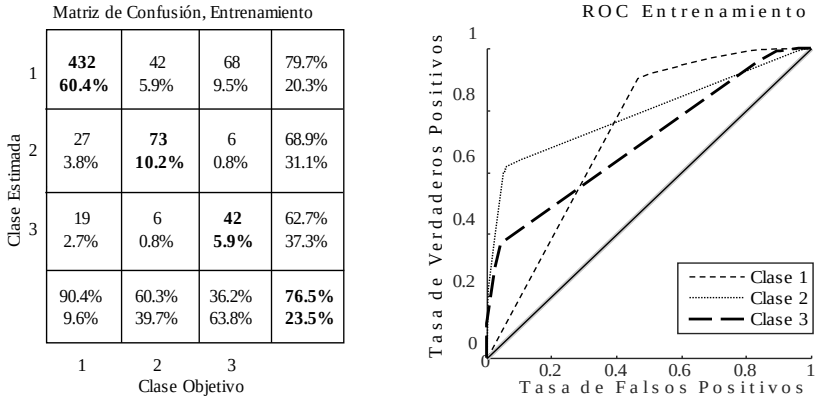


Fig. 6. (a) Matriz de confusión en la última etapa de entrenamiento. Porcentaje de predicción correcta del 76.5% para las tres clases. (b) Gráfico de "Receiver operating characteristic" Observar como en todos los resultados las curvas de sensibilidad están en la parte superior de la línea de no discriminación (diagonal).

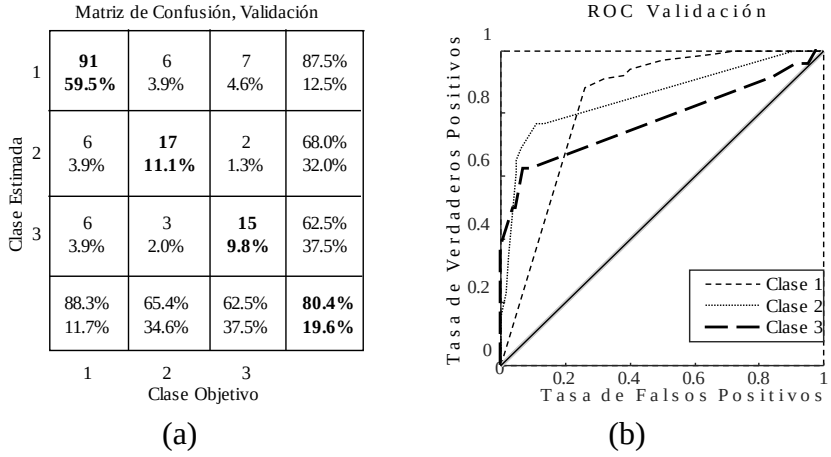


Fig. 7. (a) Matriz de confusión en la Validación. Porcentaje de predicción correcta del 80.4% para las tres clases. (b) Gráfico de "Receiver operating characteristic".

de datos de validación, esta información es tomada como medio de paro del algoritmo de entrenamiento.

Para evaluar la calidad del clasificador en las Figuras 6, 7 y 8 están las matrices de confusión y las gráficas ROC "Receiver operating characteristic" que permiten valorar la calidad de la estimación de clase. Durante el entrenamiento en la Figura 6, se observa un porcentaje de clasificación correcta para la clase 1 de 79.7%, para la clase 2 de 68.9% y de la clase 3 del 62.7%, dando un porcentaje promedio del 76.5%.

Al observar estos valores para los conjuntos de validación y prueba se observa, un porcentaje de estimación correcta del 80.4%, ver Figuras 7 y 8.

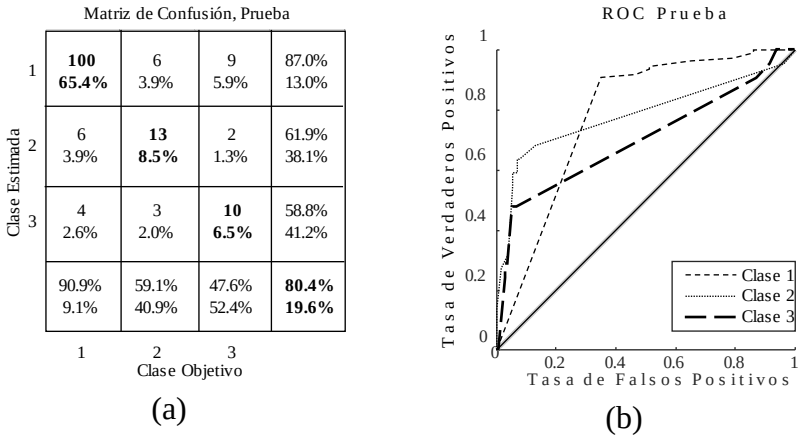


Fig. 8. Matriz de confusión en la fase de prueba. Porcentaje de predicción igual al de validación. (b) Gráfico de "Receiver operating characteristic".

5. Conclusiones y trabajo futuro

El ensamble de clasificadores tuvo un mejor desempeño al evaluarse sobre el conjunto de prueba, 80.4%, que los clasificadores individuales, que oscilan entre 62% y 65.5% o del clasificador que consideraba a todos los factores, 58.62%. Por lo que a través de estos resultados se puede concluir que el ensamble de clasificadores construido con la red neuronal resultó mejor durante la fase de prueba.

Se toman los resultados en la fase de prueba porque es cuando el clasificador se somete a datos considerados durante la fase de entrenamiento, siendo un mejor reflejo de su capacidad para clasificar.

Como trabajo futuro queda evaluar con otros métodos de ensamblaje, como son: el voto mayoritario y la tabla de decisión, entre otros, para evaluar si se mejora el desempeño.

También analizar la ponderación que le otorga la red neuronal del ensamble para identificar al clasificador que más se acercó a la clase correcta, y así examinar la estructura del árbol correspondiente, con el fin de detectar los factores educativos para la clase excelente y generar estrategias para fortalecerlos.

Referencias

1. Zhi-Hua, Z.: Ensemble methods: Foundations and Algorithms. CRC Press, Taylor & Francis Group (2012)

2. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), pp. 37–54 (1996)
3. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd ed. Morgan Kaufmann Publishers (2005)
4. WEKA 3: Data Mining Software in Java Homepage. https://www.cs.waikato.ac.nz/ml/weka/ (2016)
5. Singh, Y., Chanuhan, A.: Neural Networks in Data Mining. Journal of Theorical & Applied Information Technology, 5(1), pp.37–42 (2009)
6. Orallo, J., Ramírez, M., Ferri, C.: Introducción a la Minería de Datos. Pearson Education, (2008)
7. Khasawneh, K., Ozsoy, M., Ghazaleh, N., Ponomarev, D.: EnsembleHMD: Accurate Hardware Malware Detectors with Specialized Ensemble Classifiers. IEEE Transactions on Dependable and Secure Computing, pp. 10 (2018)
8. Yan, Y., Yang, H., Wang, H.: Two simple and effective ensemble classifiers for twitter sentiment analysis. Computing Conference 2017, pp. 1386–1393 (2017)
9. Vogado, L., Veras, R., Andrade, A., Araujo, F., Silva, R., Aires, K.: Diagnosing Leukemia in Blood Smear Images Using an Ensemble of Classifiers and Pre-Trained Convolutional Neural Networks. 30th (SIBGRAPI) Conference on Graphics, Patterns and Images, pp. 367–373, Niteroi (2017)
10. Hestenes, M., Stiefel, E.: Methods of Conjugate Gradients for Solving Linear Systems. Journal of Research of the National Bureau of Standards, 49(6), pp 409–436 (1952)

ISSN 1870-4069

Segmentación de placas vehiculares usando Haar-AdaBoost y Clustering

José Hernández Santiago[1,2], José Sergio Ruiz Castilla[2], Carlos Hiram Moreno Montiel[1], Beatriz Hernández Santiago[2]

[1] Tecnológico de Estudios Superiores de Chimalhuacán, México

[2] Universidad Autónoma del Estado de México, Posgrado e Investigación, Texcoco, Estado de México, México

{josehernandez, carlosmoreno}@teschi.edu.mx, jhernandezs@uaemex.mx,jsergioruizc@gmail.com,betty_hsbt@hotmail.com

Resumen. Actualmente los sistemas de seguridad vial cuentan con cámaras IP para monitorear las carreteras, sin embargo el análisis de estos representa un problema muy importante ya que es el personal a cargo quienes realizan esta labor. Un primer paso para desarrollar un sistema de visión que implemente técnicas de inteligencia artificial para llevar el registro de los automóviles que circulan es poder extraer la imagen de la placa del auto a partir del video de las cámaras de seguridad, después segmentarla para extraer las imágenes de los caracteres para finalmente poder realizar la clasificación y obtener el número de serie del automóvil. En este artículo, se presenta un nuevo enfoque para generalizar el problema de la segmentación de las placas de automóviles usando primero AdaBoost para ubicar la placa dentro de la imagen, después el clasificador SVM para descartar falsos positivos y Clustering para ubicar los caracteres dentro de la placa. El método propuesto permite segmentar las placas a pesar de los diversos diseños que existen en México, su ubicación en el auto y la vista frontal o lateral de la cámara.

Palabras clave: Características de Haar, AdaBoost, clustering, máquinas de vectores soporte, segmentación de placas vehiculares.

Licence Plate Segmentation by Haar-AdaBoost and Clustering

Abstract. In the Mexican Republic, vial safety systems currently have IP cameras to monitor roads. However, the analysis of monitoring system represents a several problem because it works in a personal human. Through three steps can develop a vision system to generate a record video of the cars that circulate. With artificial intelligence techniques the image of the plate and the car could be extracted from the video of the surveillance cameras, the first step. As a second step, the image must be

segmented to extract the most important characteristics and generate a comparative model of the plates. Finally, as a third step to perform identification, a classification is needed to extract the car's serial number. In the present article a novel approach is shown to generalize the problem of segmentation of automobile license plates. Using in the first instance the AdaBoost method to localize the plate inside the image. Then an SVM classifier is used to detect false positives and clustering to locate the characters inside the plate. The generated method the plates to be segmented regardless of the different designs that exist in the Mexican Republic and location of the auto, tilt, the front or side view of the camera.

Keywords: Haar features, AdaBoost, clustering, support vector machines, licence plate segmentation.

1. Introducción

Actualmente en el Estado de México se reportan 12 mil autos robados, mientras que la cifra asciende a 86 mil durante el último año, representando un problema importante de seguridad. Las instituciones de seguridad pública han implementado sistemas de video vigilancia con cámaras IP, Fig. 1.b, sin embargo, el análisis de los videos almacenados se realiza con el personal disponible, Fig. 1 a, siendo insuficiente para atender todos los incidentes reportados y buscar los autos robados.

a) Monitoreo de la cámara IP b) Cámara IP

Fig. 1. Sistema de Seguridad.

La detección de placas vehiculares a partir de los videos de seguridad es un problema que se ha abordado de diferentes formas. En [1] el método de segmentación de placas se basa en buscar regiones candidatas donde puede haber caracteres usando Maximally Stable Extremal Region (MSER), obteniendo una precisión de 95% para videos viales y con diferentes niveles de iluminación, sin embargo sus imágenes tienen alta definición y la norma en China estandariza sus placas en fondo azul con texto blanco o fondo amarillo con texto negro, reduciendo la complejidad del problema.

En [2] proponen un algoritmo híbrido aplicando segmentación y extracción de contornos, mapeando cada región candidata a un plano euclidiano y determinando su relevancia a través de un una función de costo; las imágenes usadas tienen diferentes iluminaciones, sin embargo la tarea se facilita al usar solo fotos frontales y placas con fondo negro con texto blanco.

En [3] utilizan las líneas como características para ubicar las placas chinas, usando el espacio de color de Munsell y distancia NBS, así como un tratamiento a la imagen con clustering antes de aplicar los filtros; después otro filtro morfológico es empleado para seleccionar las placas candidatas de acuerdo a su forma.

Un RFID es usado en [4] para detectar cuando el automóvil esta frente a la cámara Full HD, autorizando su paso si es que el número de la placa tiene acceso. La extracción de la placa se realiza primero usando un umbral adaptativo, dilatación y búsqueda de contornos cerrados, que al aplicar condiciones como regiones con 20 a 25% de pixeles negros con fondo blanco y rectangularidad de 90%, detecta el contorno como placa más probable con 97% de precisión.

El método presentado en [5] utiliza el seguimiento del vehículo en el video para ocupar la secuencia de frames y crear clusters con las placas parecidas que ayudaran en la clasificación del número de la placa con OpenALPR, mejorando la precisión de 7 a 32% comparado con un sistema que no usa tracking.

En [6] se comparan las técnicas de Redes Neuronales Artificiales Backpropagation (BPNN), Redes de Función de Base Radial (RBF) y Ensamble de Redes Neuronales (ENN) para reconocer placas de automóviles de Malasia respecto a su método propuesto, que compara el histograma de los objetos encontrados en la imagen binarizada para ubicar la placa y clasifica los caracteres usando una ENN que promedia la salida de una BPNN con una RBF; sin embargo las fotos provienen de un estacionamiento, no presentan inclinación o diferente vista y las placas están rotuladas en color blanco sobre fondo negro.

Una Red Neuronal Convolutiva es usada en [7] para clasificar placas chinas, reportando 98.95% de precisión para ubicar la placa, 96.58% de precisión en la segmentación de los caracteres y 98.09% en la clasificación. Para localizar la placa usaron un filtro de color, detección de contornos y un análisis morfológico, sin embargo las fotos probadas son frontales y provienen de estacionamientos. De forma similar en [8] se obtiene 98.42% de precisión en la localización de la placa.

En los artículos anteriormente citados se aplican filtros para poder ubicar la placa dentro de las imágenes, mientras que otros parten de imágenes de placas previamente cortadas para centrarse en la clasificación de los caracteres, además de que las placas incluyen poco ruido por desgaste.

Las características de Haar inicialmente fueron usadas para el tracking de los rostros [9, 10] y extraer características como los ojos [11], pero se ha demostrado que pueden emplearse en otras aplicaciones como en [12] donde las características de Haar y AdaBoost son usados para detectar y seguir los vehículos completos en videos de seguridad con vista superior logrando 93% de precisión mientras que en [13] una técnica similar permite reconocer los logotipos de algunas marcas de vehículos.

Actualmente se han implementado técnicas para el seguimiento de objetos usando boosting, como en [14] donde mejoran el algoritmo AdaBoost con un cumulo de partículas para detectar autos completos a partir de características rectangulares y puntos de control; mientras que en [15] de forma similar logran detectar vehículos con 93.2% de precisión usando Gentle AdaBoost y Patrones Binarios Locales (LBP).

En este artículo se presenta un nuevo enfoque para extraer la placa y los caracteres a partir de las imágenes de video. En la metodología se explica cómo el método propuesto aplica AdaBoost para clasificar usando características de Haar y poder ubicar la placa dentro

de la imagen del video; posteriormente se eliminan los falsos positivos usando una Maquina de Vectores Soporte (SVM); después de extraer la imagen de la placa se detectan las líneas para corregir la inclinación y finalmente se aplica un filtro de umbral por rango de color de forma iterativa hasta encontrar el cluster con todos los caracteres.

En la sección de resultados se puede observar que el método de segmentación propuesto mantiene una buena precisión para ubicar la placa y los caracteres, aunque las placas vehiculares tengan ruido por inclinación, escala, vista de la cámara e imágenes de fondo.

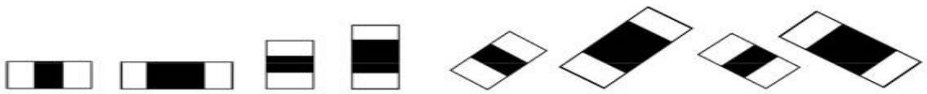
2. Preliminares

a. Características de Haar

Las características de Haar indican la diferencia de intensidades de los pixeles en regiones rectangulares locales permitiendo la detección de bordes (Fig. 2), líneas (Fig. 3) y el centro (Fig. 4). Estas características serán usadas como entradas de un clasificador básico que permitirá detectar objetos de forma rápida.

a) b) c) d)

Fig. 2. Características para detectar bordes.



a) b) c) d) e) f) g) h)

Fig. 3. Características para detectar líneas.

a) b)

Fig. 4. Características para detectar el centro.

b. AdaBoost

Este método está diseñado para combinar reglas, se basa en la combinación lineal de muchas reglas débiles pero muy precisas, para crear un clasificador muy robusto con un error arbitrariamente bajo en el conjunto de entrenamiento. Estas reglas débiles se aprenden secuencialmente manteniendo una distribución de pesos D_t sobre los ejemplos de entrenamiento. Estos pesos se van actualizando a medida que se adquieren nuevas reglas.

AdaBoost es un algoritmo que pretende obtener una regla de clasificación muy precisa combinando muchos clasificadores débiles, cada uno de los cuales obtiene una precisión

moderada. Este algoritmo trabaja eficientemente con espacios de atributos muy grandes y ha sido aplicado con éxito a muchos problemas prácticos.

Las hipótesis débiles $h_t: X \rightarrow \{-1, +1\}$, se aprenden secuencialmente en cada iteración, sesgándola para clasificar los ejemplos con más dificultad de acuerdo al conjunto de reglas con las que se cuentan en esa iteración, manteniendo un error moderadamente bajo respecto a la distribución de pesos D_t. Inicialmente, la distribución de pesos D_1es uniforme, y en cada iteración, el algoritmo de boosting incrementa (o decrementa) exponencialmente los pesos $D_t\,(i)$ en función de si $h_t\,(x_i)$ realiza una buena (o mala) predicción. La combinación final de hipótesis, $h_t : X \rightarrow \{-1, +1\}$, calcula sus predicciones ponderando con pesos los votos de las diferentes hipótesis débiles como se muestra en la ecuación (1):

$$f(x) = \sum_{t=1}^{T} \propto_t \cdot h_t\,(x). \tag{1}$$

Para cada nuevo ejemplo x, el signo de $f(x)$ se interpreta como la clase predicha (–1 o +1), y la magnitud $|f\,(x)|$ como una medida de la confianza de la predicción. En 2001 Viola y Jones inventaron un clasificador basado en las características de Haar que permite detectar objetos en tiempo real empleando cámaras con resoluciones VGA y obteniendo 95% de precisión para la vista frontal [9].

c. Máquinas de Vectores Soporte (SVM)

Las SVM fueron inspiradas en los resultados de la teoría de aprendizaje estadístico desarrollado por Vapnik en los 70's [16]. Este clasificador permite encontrar un hiperplano capaz de separar linealmente dos clases, proyectando el espacio de entrada original a un espacio de características altamente dimensional donde maximiza el margen entre clases.

Las SVM permiten estimar una función de clasificación óptima empleando datos de entrenamiento etiquetados como X_{tr}, de esta forma, la función f clasificará correctamente datos no vistos antes por el clasificador (datos de prueba). Considerando el caso más simple de clasificación binaria, asumimos que el conjunto X_{tr} es dado como en la ecuación (2):

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \tag{2}$$

i.e. $X_{tr} = \{x_i, y_i\}_{i=1}^{n}$ donde $x_i \in R^d$ y $y_i \in R(+1, -1)$ corresponde a la etiqueta de clasificación de la muestra x_i. La función de clasificación se expresa en la ecuación (3):

$$y_i = sign\left(\sum_{i=1}^{n} \alpha_i y_i K\langle x_i \cdot x_j\rangle + b\right), \tag{3}$$

donde $x = [x_1, x_2, \ldots, x_n]$ son los datos de entrada. Un nuevo objeto *x* puede ser clasificado usando (3). El vector x_i es mostrado en la forma de producto punto. Las α'_i son multiplicadores de Lagrange y b es el bias obtenido al entrenar la SVM.

3. Metodología

a. Segmentación de la placa dentro de la imagen usando AdaBoost

El primer paso del método propuesto consiste en ubicar la placa dentro de la imagen extraída del video. Para realizar esta tarea, se entrenó el algoritmo AdaBoost con las

a) b)

Fig. 5. Detección de placas en imágenes con vista frontal.

a) b)

Fig. 6. Detección de placas en imágenes con vista lateral izquierda.

características de Haar provenientes de 300 imágenes positivas (en las que se marcó la placa) y 1000 imágenes negativas (donde no hay placas de vehículos), provenientes de videos con tres diferentes vistas: frontal, lateral izquierda y derecha. Asimismo, se definió una proporción para la placa de 40 pixeles de ancho y 18 pixeles de alto para reducir la complejidad del análisis.

En la Fig. 5 se puede notar que las placas detectadas presentan poca inclinación debido a que son frontales, sin embargo en los videos con vista lateral izquierda (Fig. 6) y derecha (Fig. 7) es necesario aplicar una corrección a la inclinación. En la Fig. 6 se muestra que la vista además de ser lateral izquierda, también es trasera, ofreciendo evidencias de la generalización de la posición de la placa dentro de la imagen y aunque existen algunos falsos positivos, estos serán eliminados en una etapa posterior por la SVM.

b. Corrección de la inclinación de la placa

Debido a que la imagen proviene de los videos grabados con cámaras IP con diferentes vistas, se debe aplicar una rotación a la placa para alinearla horizontalmente y facilitar la extracción de los caracteres. En la Fig. 8 se muestran las líneas detectadas, agrupadas en horizontales, verticales, sesgadas a la izquierda y sesgadas a la derecha de acuerdo a su pendiente; después una rotación es aplicada en sentido contrario al sesgo mayoritario.

c. Detección de falsos positivos en las placas usando SVM

Con el fin de mejorar la detección de las placas vehiculares, se usó el clasificador SVM, entrenado con un conjunto de muestras positivas formadas por los histogramas de las placas correctamente detectadas por AdaBoost, mientras que las muestras negativas las integraron

a) b)

Fig. 7. Detección de placas en imágenes con vista lateral derecha.

a) Placa con sesgo a la izquierda b) Placa con corrección del sesgo

Fig. 8. Corrección de la inclinación.

los histogramas de los falsos positivos. El entrenamiento con SVM empleó una función de base radial (RBF), definida en la ecuación (4):

$$K(x_i - x_j) = exp\left(-\gamma\|x_i - x_j\|^2\right), \gamma > 0. \quad (4)$$

El parámetro *C* regula el punto medio entre error de entrenamiento y complejidad, mientras que γ es un parámetro del *kernel*. La obtención de buenos parámetros se logró usando una búsqueda en malla.

Para disminuir la complejidad del entrenamiento de la SVM, se usó el algoritmo Sequential Minimal Optimization (SMO) desarrollado por Platt [17].

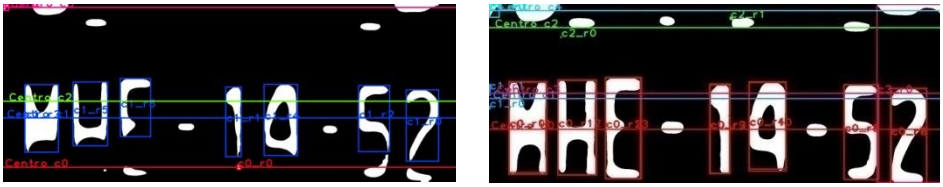
d. Segmentación de los caracteres dentro de la placa usando Clustering

Una vez que la imagen de la placa ha sido extraída y alineada, se aplica un filtro de rango de color para dejar solo los pixeles negros y quitar las imágenes de fondo. Para facilitar la separación de los caracteres se quitan las filas con 80% de pixeles negros continuos que forman el marco superior e inferior como se muestra en la Fig. 9.

Después se buscan los pixeles negros continuos que formen un contorno cerrado y se agrupan de acuerdo a su altura y su posición en el eje "Y" para descartar los caracteres correspondientes al nombre del estado y manchas cerradas que pudieran confundirse como caracteres. El algoritmo se aplica máximo diez veces de forma iterativa, como se muestra

a) Segmentación con umbral original b) Marco superior e inferior eliminados

Fig. 9. Eliminación del marco superior e inferior de la placa.



a) Iteración inicial b) Iteración final

Fig. 10. Segmentación aplicando clustering y rango de color.

en la Fig. 10, hasta encontrar el cluster con la mayor cantidad de contornos en forma de caracteres.

4. Resultados

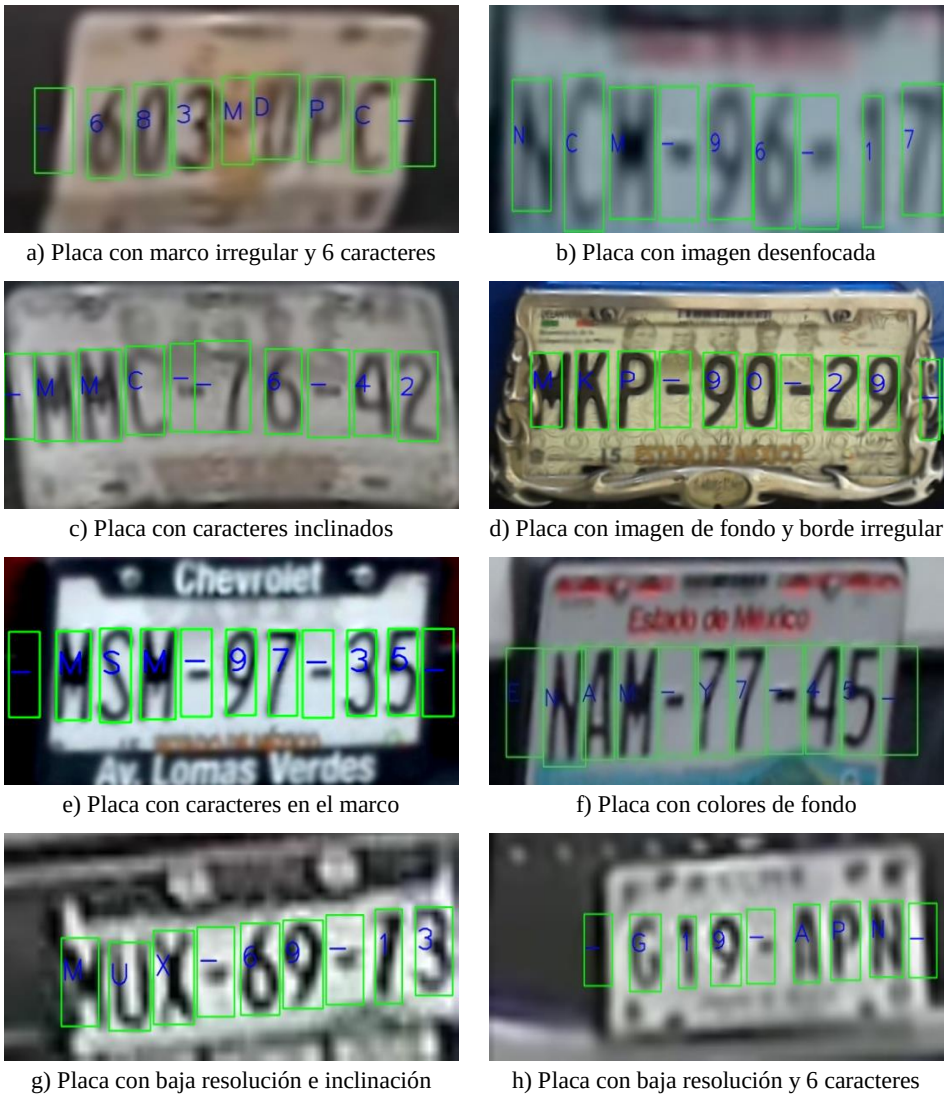
En las pruebas realizadas se emplearon videos de cámaras IP colocadas para la investigación a un metro de altura en los semáforos del municipio de Texcoco debido a que estas no pueden enfocar el vehículo a más de 4 kilómetros por hora.

Para realizar los experimentos se grabaron nuevos videos y se prepararon los conjuntos de prueba de acuerdo a la vista de la cámara, destinando 149 fotos frontales, 42 para la vista lateral derecha y 50 para la izquierda. En la Fig. 11 se pueden ver algunos ejemplos de la detección de los caracteres para las placas con ruido por desenfoque, placas dobladas, con imagen de fondo, adornos en el marco, color de fondo y diferente longitud de caracteres.

En la prueba para ubicar la placa dentro de la imagen, el desempeño de AdaBoost con SVM fue de 95.97%, detectando 143 de 149 para la vista frontal; mientras que para la vista lateral derecha obtuvo una precisión de 85.71%, logrando ubicar 36 de 42 placas. En la prueba para ubicar los caracteres dentro de la imagen de la placa recortada, el desempeño

Tabla 1. Precisión para el método propuesto.

Conjunto de Datos	Segmentación de la placa		Segmentación de los caracteres
	Fotos	ACC	ACC
Frontal	149	0.95	0.83
Lateral izquierda	50	0.86	0.70
Lateral derecha	42	0.85	0.72



a) Placa con marco irregular y 6 caracteres
b) Placa con imagen desenfocada
c) Placa con caracteres inclinados
d) Placa con imagen de fondo y borde irregular
e) Placa con caracteres en el marco
f) Placa con colores de fondo
g) Placa con baja resolución e inclinación
h) Placa con baja resolución y 6 caracteres

Fig. 11. Caracteres encontrados en la placa

de Clustering con el filtro por rango de color fue de 83.27% para la vista frontal, mientras que para la vista lateral derecha se obtuvo 72.53% como se muestra en la Tabla 1.

5. Conclusión

El análisis de los videos de seguridad es un campo de aplicación importante para la inteligencia artificial, tareas como la búsqueda de vehículos robados, aparcamiento y acceso a edificios requieren poder ubicar la placa dentro de la imagen antes de poder clasificar los

caracteres y obtener su número de serie. Las técnicas clásicas usan filtros para detectar los colores de la placa, su forma rectangular o una rejilla con las posiciones a partir de la norma del país y estado al que pertenece el vehículo, sin embargo, su precisión disminuye cuando la posición de la placa cambia debido a la altura del vehículo como camiones de carga, trailers y autobuses; cuando la vista ya no es frontal y las placas están muy desgastadas o tienen adornos en el marco.

Algunos autores se han centrado en mejorar la precisión de la clasificación de los caracteres, recortando las placas manualmente para poder entrenar los algoritmos, sin embargo en este artículo, se presentó un nuevo método que mejora la segmentación de las placas vehiculares; que a diferencia de otros reportados en la literatura, usa AdaBoost y las características de Haar para ubicar la placa a pesar de que pudiera localizarse en cualquier parte de la imagen, con diferente escala e inclinación.

La extracción de los caracteres también está generalizada ya que no depende de una ubicación fija, sino que agrupa los caracteres candidatos de acuerdo a su posición dentro de la placa, descartando los caracteres alrededor que pudieran corresponder al nombre del estado al que pertenece el vehículo, la marca o caracteres ajenos al diseño.

Otro problema es que en México cada estado usa imágenes de fondo conmemorativas, dificultando la segmentación, sin embargo, el método propuesto lo resuelve filtrando por rango de color hasta encontrar el cluster con todos los caracteres como se muestran en las figuras para las pruebas.

De acuerdo con los resultados, el método propuesto presenta una buena precisión, de 95.97% para segmentar las placas con vista frontal del automóvil, mientras que llego a 85% para la vista lateral derecha e izquierda, presentando un problema para clasificar aquellas con un sesgo de más de 45 grados. El método propuesto permite detectar las placas aun si son de otro tipo de vehículos como motocicletas o camiones de carga.

La segmentación de los caracteres fue 83.27% para la vista frontal, mientras que para la vista lateral derecha e izquierda obtuvo 70%, afectando la segmentación cuando las imágenes provienen de cámaras con resolución VGA y presentan mucho ruido.

Se propone comparar diversas técnicas para clasificar los caracteres de las placas, desarrollar un sistema distribuido para obtener los videos de diferentes cámaras IP HD ubicadas en lugares estratégicos como casetas, entradas de edificios, aeropuertos y semáforos para buscar autos con reporte de robo. También se requerirá usar algún algoritmo de optimización para encontrar los parámetros que mejoren la precisión de todo el sistema.

Referencias

1. Gu, Q., Yang, J., Kong, L., Cui, G.: Multi-scaled license plate detection based on the label-moveable maximal MSER clique. pp. 669–678 (2015)
2. Tedjojuwono, S.M.: Fast Performance Indonesian Automated License Plate Recognition Algorithm Using Interconnected Image Segmentation. In: Intan, R., Chi, CH., Palit, H., Santoso, L.(eds.) Intelligence in the Era of Big Data (ICSII) Communications in Computer and Information Science, 516, pp. 289–300, Springer, Berlin, Heidelberg (2015)
3. Xie, J., Zhou, H., Wu, X., Zhou, Y.: A Method of License Character Recognition Based on Fast Nearest Feature Line. In: Pan, Z., Cheok, A., Mueller, W., Zhang, M.(eds.) Transactions on Edutainment XI, (LNCS), 8971, pp. 52–60, Springer-Verlag, Berlin Heidelberg (2015)

4. Mohandes, M., Deriche, M., Ahmadi, H., Kousa, M., Balghonaim, A.: An Intelligent System for Vehicle Access Control using RFID and ALPR Technologies. Arab J SciEng, pp. 3521–3530, Springer, Berlin Heidelberg (2016)
5. Kluwak, K., Segen, J., Kulbacki, M., Drabik, A., Wojciechowski, K.: ALPR - Extension to Traditional Plate Recognition Methods. In: Nguyen, N.T., Trawiński, B., Fujita, H., Hong, TP.(eds.) Intelligent Information and Database Systems, ACIIDS 2016, Part II, Lecture Notes in Computer Science, 9622, pp. 755–764, Springer, Berlin, Heidelberg (2016)
6. On, C.K., Yao, T.K., Alfred, R., Ibrahim, A.A.A., Cheng, W., Guan, T. T.: A Comparison of BPNN, RBF, and ENN in Number Plate Recognition. In: Berry, M., Hj. Mohamed, A., Yap, B.(eds.) Soft Computing in Data Science, SCDS, Communications in Computer and Information Science, 652, pp. 37–47, Springer, Singapore (2016)
7. Liu, Y., Huang, H., Cao, J., Huang, T.: Convolutional neural networks-based intelligent recognition of Chinese license plates. Soft Comput. Springer-Verlag, Berlin Heidelberg (2017)
8. Fu, Q., Shen, Y., Guo, Z: License Plate Detection Using Deep Cascaded Convolutional Neural Networks in Complex Scenes. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy ES., (eds.) Neural Information Processing, (ICONIP), Lecture Notes in Computer Science, 10635, Springer, Cham (2017)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. Conf. on Computer Vision and Pattern Recognition (CVPR) Kauai, HI, pp. 511–518 (2001)
10. Rezaei, M., Ziaei-Nafchi, H., Morales, S.: Global Haar-Like Features: A New Extension of Classic Haar Features for Efficient Face Detection in Noisy Images. In: Klette, R., Rivera, M., Satoh, S. (eds.) Image and Video Technology, (PSIVT´13), Lecture Notes in Computer Science, 8333, Springer, Berlin, Heidelberg (2014)
11. Prasanna, D., Prabhakar, M.: An efficient human tracking system using Haar-like and hog feature extraction. Cluster Comput, pp. 1–8, Springer US (2018)
12. Elkerdawi, S.M., Sayed, R., ElHelw, M.: Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking. In: Armada, M., Sanfeliu, A., Ferre, M. (eds.) ROBOT2013, First Iberian Robotics Conference, Advances in Intelligent Systems and Computing, 252, Springer, Cham (2014)
13. Sotheeswaran, S., Ramanan, A.: A Coarse-to-Fine Strategy for Vehicle Logo Recognition from Frontal-View Car Images. Pattern Recognition and Image Analysis, vol. 28, pp. 142–154. Pleiades Publishing (2018)
14. Benabderrahmane, S.: Combining boosting machine learning and swarm intelligence for real time object detection and tracking: towards new meta-heuristics boosting classifiers. International Journal of Intelligent Robotics and Applications, vol. 1, pp. 410–428. Springer Singapore (2017)
15. Jiang, T., Cai, M., Zhang, Y., Zhao, X.: A Fast Video Vehicle Detection Approach Based on Improved Adaboost Classifier. In: Tan, Y., Takagi, H., Shi, Y., Niu, B. (eds.) Advances in Swarm Intelligence, ICSI, Lecture Notes in Computer Science, vol. 10386, Springer, Cham (2017)
16. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag (1995)
17. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14 (1998)

ISSN 1870-4069

Comparación del nivel de precisión de los clasificadores Support Vector Machines, k Nearest Neighbors, Random Forests, Extra Trees y Gradient Boosting en el reconocimiento de actividades infantiles utilizando sonido ambiental

Diego M. Blanco-Murillo[1], Antonio García-Domínguez[1], Carlos E. Galván-Tejada[1], José M. Celaya-Padilla[2]

[1] Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, México

[2] CONACyT - Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas, México

{diegomurillo,antonio.garcia,ericgalvan}@uaz.edu.mx,jose.celaya@uaz.edu.mx

Resumen. La información de audio desempeña un papel importante en el creciente contenido digital disponible hoy en día, de tal manera que esto, da como resultado la necesidad de desarrollar sistemas o aplicaciones que analicen de forma automática dicho contenido. Algunas de las aplicaciones más comunes en esta área son: reconocimiento de eventos de audio para domótica y sistemas de vigilancia automática, reconocimiento de voz, recuperación de información musical, análisis multimodal, reconocimiento de actividades humanas, entre otras más en el área de la Inteligencia Ambiental y la Inteligencia Artificial. Sin embargo, los estudios en algunas de las áreas anteriormente mencionadas son escasos, principalmente en el área de reconocimiento de actividades humanas mediante el sonido ambiental. Es por ello, que en este trabajo se realiza una evaluación y comparación del rendimiento de los clasificadores Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET) y Gradient boosting (GB) aplicados al reconocimiento de actividades realizadas por infantes en el rango de edad de 12 a 36 meses.

Palabras clave: Reconocimiento de actividades infantiles,sonido ambiental, support vector machines, K nearest neighbors, random forests, extra trees, gradient boosting.

Precision Level Comparison of the Support Vector Machines, K Nearest Neighbors, Random Forests, Extra Trees and Gradient Boosting

Diego M. Blanco-Murillo, Antonio García-Domínguez, Carlos E. Galván-Tejada, et al.

Classifiers in the Recognition of Children's Activities Using Environmental Sound

Abstract. Audio information plays an important role in the growing digital content that is available nowadays, in this manner, this leads to an outcome that shows the necessity of developing systems or applications that automatically analyze this content. Some of the most common applications in this area are: recognition of audio events for home automation and automatic surveillance systems, voice recognition, music information retrieval, multimodal analysis, recognition of human activities, among others in the area of Environmental Intelligence and Artificial Intelligence. Nevertheless, studies in some of the areas mentioned above are scarce, mainly in the area of recognition of activities through environmental sound. It is therefore, in this paper, an evaluation and comparison of the performance of the Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET) and Gradient boosting (GB) classifiers applied to the recognition of activities carried out by infants in the age range of 12 to 36 months is made.

Keywords: Child activity recognition, environmental sound, support vector machines, K nearest neighbors, random forests, extra trees, gradient boosting.

1. Introducción

El análisis, clasificación y predicción automática de actividades humanas es un tema de gran interés en diferentes áreas de la inteligencia artificial, tanto por su dificultad, como por sus aplicaciones. El reconocimiento de actividades puede ser la base para que sistemas realicen tareas complejas, por ejemplo, sistemas para dar seguimiento a pacientes, cuidado de ancianos, como el presentado por Jalal, Kamal y Kim en [1], cuidado de infantes, sistemas de rehabilitación, entrenamiento físico, vigilancia inteligente, como el presentado por Mun Sim, Lee y Ohbyung Kwon en [2], robots inteligentes, entre muchas otras.

Existen muchos trabajos orientados al cuidado de infantes que se inclinan por utilizar aparatos o dispositivos colocados directamente sobre el cuerpo de la persona, entre estos se encuentran sistemas portátiles, acelerómetros, dispositivos de radiofrecuencia, sensores barométricos, como los presentados por Onkar y Agrawal en [3], Anusha, Belagali, Maheendrachari y Prashant en [4], Boughorbel, Breebaart, Bruekers, Flinsenberg y Kate en [5] y Nam y Park en [6], que se enfocan en el monitoreo de grupos de niños con la finalidad de evitar un accidente y salvaguardar la integridad física de los mismos. Por el contrario, existen pocos trabajos que se basan en utilizar el sonido ambiental como fuente de datos para reconocer actividades de niños, como el presentado por García-Domínguez y Galván-Tejada en [7].

La orientación de los trabajos desarrollados hasta hoy en día sobre reconocimiento de actividades en niños resultan ser completamente invasivos, ya que se

centran en utilizar sensores colocados directamente sobre el cuerpo de la persona o en alguna prenda que lleve puesta, provocando que éstos puedan interferir con las actividades que los niños realizan, obteniendo datos imprecisos que no permitan realizar un análisis coherente y detallado de la actividad a analizar.

El enfoque de utilizar sonido como fuente de datos para reconocer y clasificar las actividades resulta no invasivo y más cómodo para los infantes, ya que el dispositivo de grabación puede estar a una distancia considerablemente alejada para captar un audio de buena calidad y sin intervenir en las actividades realizadas por el niño.

En la actualidad, existen varios clasificadores de datos que son aplicados al reconocimiento de actividades, entre los cuales se destacan: Naive Bayes (NB), Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET), Gradient boosting (GB) y Neural Networks (NN). Un ejemplo de estos clasificadores, es la utilización de Random Forest (RF) y Neural Networks (NN), como el presentado por Carlos Galván-Tejada, Jorge Galván-Tejada, Celaya-Padilla, Delgado-Contreras, Magallanes-Quintanar, Martinez-Fierro, Garza-Veloz, López-Hernández y Gamboa-Rosales en [8], con el objetivo de realizar un análisis de las características de los audios para desarrollar un modelo de reconocimiento de actividades humanas.

El objetivo de este estudio es evaluar el nivel precisión de los clasificadores SVM, kNN, RF, ET, GB, a la hora de catalogar grabaciones de audio de actividades comunes en niños de 12 a 36 meses de edad, tales como caminar, correr, jugar y llorar.

Tabla 1. Descripción de las actividades.

Actividad	**Descripción**
Caminar	Recorrer a pie determinada distancia a velocidad media
Correr	Recorrer a pie con rapidez una distancia determinada
Jugar	Manipular bloques de plástico de manera que éstos produzcan ruido al golpearse
Llorar	Producir un sonido de llanto como reacción a algún suceso

2. Materiales y métodos

En la presente sección se describe la extracción de características que se realizó con *pyAudioAnalysis* para clasificar audios de actividades infantiles, así como los métodos y clasificadores utilizados para llevar a cabo el presente estudio.

2.1. Descripción del data-set

El Data-Set que se utiliza para este trabajo, está conformado por grabaciones de audio de cuatro actividades realizadas por niños en el rango de edad de 12

a 36 meses, tales como, caminar, correr, jugar y llorar. La Tabla 1 muestra la descripción de cada una de las actividades que fueron consideradas en el análisis. **Fuente de los archivos de audio.** Para establecer el Data-Set, los audios de las diferentes actividades, fueron recolectados directamente del estudio realizado sobre el Reconocimiento de actividades infantiles utilizando sonido ambiental [7], y descargados directamente de una página web [9].

2.2. Procesamiento de audio

Para realizar procesamiento de audio *pyAudioAnalysis* [10] es una biblioteca abierta de Python que proporciona una amplia gama de funcionalidades relacionadas con el audio que se centran en problemas de extracción, clasificación, segmentación y visualización de características.

2.3. Extracción de características (análisis a corto plazo)

La señal de audio primero se divide en ventanas de corto plazo (cuadros) y para cada cuadro se calculan las 34 características mencionadas en la Tabla 2. Esto da como resultado una secuencia de vectores de características a corto plazo de 34 elementos cada uno. Los tamaños de ventana de corto plazo ampliamente aceptados son de 20 a 100 milisegundos. En *pyAudioAnalysis*, el proceso a corto plazo se puede llevar a cabo utilizando superposición de encuadre, es decir, el paso del cuadro es más corto que la longitud del cuadro, o no superposición de encuadre, es decir, el paso del cuadro es igual a la longitud del cuadro.

2.4. Modelo de clasificación

Una parte importante en el reconocimiento y clasificación de actividades humanas es el clasificador que se utiliza para catalogar dichas actividades. Cuando se analizan señales de audio, como es el caso del presente trabajo, la utilización de un clasificador adecuado para el análisis de las muestras ayuda a obtener un procesamiento adecuado, preciso y confiable.

Los modelos de clasificación utilizados para este trabajo implementan un procedimiento de validación cruzada para estimar el parámetro clasificador óptimo, la elección de estos modelos se hizo tomando en cuenta la forma en que estos modelos dividen la señal de audio en ventanas de corto plazo (cuadros) y después calculan una serie de características para cada cuadro. Este proceso conduce a una secuencia de vectores de características a corto plazo para toda la señal.

2.5. Clasificadores

En este trabajo, se utilizaron cinco clasificadores diferentes, Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET), Gradient boosting (GB). Una descripción detallada de los clasificadores utilizados se puede encontrar en esta sección.

Tabla 2. Características de audio.

Índice	**Nombre**	**Descripción**
1	Tasa de cruce cero	La tasa de cambio de signo de la señal durante la duración de un cuadro particular.
2	Energía	La suma de cuadrados de los valores de señal, normalizados por la longitud respectiva.
3	Entropía de energía	La entropía de las energías normalizadas de los subfotogramas. Se puede interpretar como una medida de cambios abruptos.
4	Centroide espectral	El centro de gravedad del espectro.
5	Extensión espectral	El segundo momento central del espectro.
6	Entropía espectral	Entropía de las energías espectrales normalizadas para un conjunto de subfotogramas.
7	Flujo espectral	La diferencia cuadrada entre las magnitudes normalizadas de los espectros de los dos cuadros sucesivos.
8	Desplazamiento espectral	La frecuencia por debajo de la cual se concentra el 90 % de la distribución de la magnitud del espectro.
9-21	CCFM	Los coeficientes cepstrales de la frecuencia Mel forman una representación cepstral donde las bandas de frecuencia no son lineales sino que se distribuyen de acuerdo con la escala de mel.
22-33	Vector de Croma	Una representación de 12 elementos de la energía espectral en la que los contenedores representan las 12 clases de tono temperamental de música de tipo occidental (espaciado de semitonos).
34	Desviación cromática	La desviación estándar de los 12 coeficientes de croma.

Support Vector Machines (SVM). El clasificador SVM es un algoritmo de aprendizaje supervisado basado en kernel que clasifica los datos en dos o más clases. SVM está especialmente diseñado para la clasificación binaria. Durante la fase de capacitación, SVM construye un modelo, mapea el límite de decisión para cada clase y especifica el hiperplano que separa las diferentes clases. Aumenta la distancia entre las clases, incrementando el margen del hiperplano con la finalidad de ayudar a la precisión de la clasificación. SVM se puede utilizar para realizar de manera efectiva la clasificación no lineal.

Como se mencionó anteriormente, el clasificador SVM es un clasificador basado en kernel. Una función Kernel es un procedimiento de mapeo realizado en el conjunto de entrenamiento para mejorar su semejanza con un conjunto de datos linealmente separable. El objetivo del mapeo es aumentar la dimensionalidad del conjunto de datos y se realiza de manera eficiente utilizando una función kernel. Algunas de las funciones de kernel comúnmente utilizadas son lineales, RBF (del inglés radial basis function), cuadráticas, kernel Perceptron multicapa y kernel polinomial.

k Nearest Neighbors (k-NN). En el reconocimiento de patrones, el algoritmo K-NN es un método de aprendizaje basado en instancias utilizado para clasificar objetos en función de sus ejemplos de entrenamiento más cercanos en el espacio de características. Un objeto se clasifica por el voto mayoritario de sus vecinos, es decir, el objeto se asigna a la clase que es más común entre sus k vecinos más cercanos, donde k es un número entero positivo. En el algoritmo k-nn, la clasificación de un nuevo vector de características de prueba está determinada por las clases de sus k-vecinos más cercanos.

Random forests (RF). El clasificador Random forests (RF) es un meta estimador que se adapta a una serie de clasificadores de árbol de decisiones en varias submuestras del conjunto de datos y utiliza un promedio para mejorar la precisión predictiva y controlar el ajuste excesivo. El tamaño de submuestra siempre es el mismo que el tamaño de muestra de entrada original.

En Random Forests, cada árbol en el conjunto se construye a partir de una muestra extraída con reemplazo, es decir, una muestra de arranque del conjunto de entrenamiento. Además, al dividir un nodo durante la construcción del árbol, la división que se elige ya no es la mejor división entre todas las características. En cambio, la división que se selecciona es la mejor división entre un subconjunto aleatorio de las características. Como resultado de esta aleatoriedad, el sesgo del bosque generalmente aumenta ligeramente (con respecto al sesgo de un solo árbol no aleatorio) pero, debido al promedio, su varianza también disminuye, generalmente más que compensando el aumento en el sesgo, lo que arroja un modelo global mejor.

Extra trees (ET). Extra trees, implementa un meta estimador (igual que Random forests) que se adapta a una serie de árboles de decisión aleatoria en varias submuestras del conjunto de datos y utiliza promedios para mejorar la precisión predictiva y controlar el ajuste excesivo.

El módulo *sklearn.ensemble* incluye dos algoritmos de promedio basados en árboles de decisión aleatoria: el algoritmo RandomForest y el algoritmo Extra-Trees. Ambos algoritmos son técnicas de perturbación y combinación diseñadas específicamente para árboles. Esto significa que se crea un conjunto diverso de clasificadores introduciendo aleatoriedad en la construcción del clasificador. La predicción del conjunto se da como la predicción promedio de los clasificadores individuales.

Gradient boosting (GB). Gradient boosting (GB) construye un modelo aditivo de forma progresiva; permite la optimización de funciones de pérdida diferenciables arbitrarias. En cada etapa, los árboles de n_classes_regression se ajustan al gradiente negativo de la función de pérdida de desviación binomial o multinomial. La clasificación binaria es un caso especial en el que solo se induce un solo árbol de regresión.

Gradient Tree Boosting o Gradient Boosted Regression Trees (GBRT) es una generalización de impulsar a las funciones de pérdida diferenciables arbitrarias.

GBRT es un procedimiento comercial preciso y efectivo que se puede usar tanto para problemas de regresión como de clasificación. Los modelos de Gradient Tree Boosting se utilizan en una variedad de áreas, incluidas la clasificación de búsqueda web y la ecología.
Las ventajas de GBRT son:

- Manejo natural de datos de tipo mixto (características heterogéneas)
- Poder predictivo
- Robustez a valores atípicos en el espacio de salida (a través de funciones de pérdida robustas)

Las desventajas de GBRT son:

- Escalabilidad, debido a la naturaleza secuencial de impulsar, difícilmente se puede paralelizar.

El módulo *sklearn.ensemble* proporciona métodos para la clasificación y la regresión a través de árboles de regresión potenciados por el gradiente.

3. Experimentación

Se utilizaron un total de 70 grabaciones, de las cuales 43 se utilizaron para realizar el entrenamiento del modelo, con el fin de obtener un entrenamiento robusto y preciso, y las restantes 27, para realizar pruebas de precisión a cada clasificador, dicha cantidad se consideró aceptable para no estresar al modelo, probandolo con una cantidad menor de grabaciones en comparación con la cantidad de grabaciones con las que se había creado. Las cantidades de archivos de audio que se tiene para cada actividad, y que se utilizan en cada una de las etapas (entrenamiento y pruebas) se muestran en las Tablas 3 y 4 respectivamente.

Tabla 3. Archivos de audio por actividad para entrenamiento.

Actividad	**Archivos de audio**
Caminar	11
Correr	11
Jugar	10
Llorar	11

El proceso de entrenamiento y pruebas fue llevado a cabo con la ayuda del lenguaje de programación Python, que es un lenguaje de programación de alto nivel que ha estado atrayendo un interés creciente, especialmente en la comunidad académica y científica durante los últimos años y en el cual se encuentra implementada la librería *pyAudioAnalysis*, que cubre una amplia gama de tareas de análisis de audio, tales como extraer características de audio, entrenar y aplicar clasificadores de audio, segmentar una secuencia de audio utilizando metodologías supervisadas o no supervisadas y visualizar relaciones

Tabla 4. Archivos de audio por actividad para pruebas.

Actividad	Archivos de audio
Caminar	7
Correr	7
Jugar	6
Llorar	7

de contenido. Para el análisis de los archivos, se trabajó con los clasificadores Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET) y Gradient boosting (GB) de la librerías de Python.

Con el propósito de crear un modelo preciso, cada clasificador fue entrenado aisladamente a partir de los datos de la Tabla 3, es decir, el conjunto de archivos de audio de dicha tabla fue colocado en carpetas separadas con el nombre de cada clasificador.

Los pasos a realizar en el proceso de entrenamiento son los siguientes:

- La extracción de características de cada audio (para este trabajo sería la extracción de características a corto plazo).
- La evaluación del clasificador y selección de parámetros.
- Obtener el parámetro clasificador óptimo.
- Almacenar el entrenamiento del modelo.

Además, también se crea un archivo ARFF (con el mismo nombre que el modelo), donde se almacena todo el conjunto de vectores de características y etiquetas de clase respectivas.

La obtención de este archivo se realizo con la funcion *featureAndTrain()* incluida en la librería *pyAudioAnalysis*.

La fase de pruebas para cada clasificador se realizó de manera separada para evitar una posible confusión en los resultados. La forma de comenzar el proceso de prueba para cada clasificador inicia cargando el clasificador, es decir el archivo que se generó en la etapa de entrenamiento, en seguida se leen los archivos de audio y se convierten en sonido mono, los dos últimos pasos consisten en realizar la extracción de características a corto plazo y la clasificación de la actividad de cada audio.

4. Resultados

Fueron un total de cinco pruebas las que se realizaron, una prueba por cada clasificador con el conjunto de datos de la Tabla 4.

En la Tabla 5 se pueden observar los datos que arrojaron las pruebas que se le aplicaron a cada modelo a cerca de la precisión con la que se llevó a cabo la clasificación de las actividades analizadas en el presente trabajo.

Observando la Tabla 5, se puede comprobar que los clasificadores k Nearest Neighbors (kNN) y Extra trees (ET) fueron los más acertados a la hora de

Tabla 5. Resultados.

	Clasificadores				
	SVM	**kNN**	**RF**	**ET**	**GB**
Actividad	**Porcentaje de precisión**	**Porcentaje de precisión**	**Porcentaje de precisión**	**Porcentaje de precisión**	**Porcentaje de precisión**
Caminar	40.31 %	100 %	76.50 %	100 %	99.93 %
Correr	55.53 %	100 %	71.50 %	100 %	99.95 %
Jugar	74.94 %	100 %	84.00 %	100 %	99.95 %
Llorar	41.93 %	100 %	73.50 %	100 %	99.95 %

clasificar un archivo de audio, obteniendo un 100 % de precisión, seguido por el clasificador Gradient boosting (GB) que obtuvo una precisión del 99.94 % en promedio. Por el contrario, se puede observar que los clasificadores Support Vector Machines (SVM) y Random forests (RF) clasifican a los archivos de audio con una precisión del 53.17 % y 76.37 % en promedio respectivamente.

5. Discusión y conclusiones

El enfoque principal de este trabajo de investigación es realizar un estudio comparativo sobre distintos clasificadores utilizados en el área de reconocimiento y clasificación de actividades humanas mediante sonido ambiental, específicamente en actividades realizadas por infantes de 12 a 36 meses (considerados biológicamente como bebés). Los clasificadores tomados en cuenta en el presente trabajo son: Support Vector Machines (SVM), k Nearest Neighbors (kNN), Random forests (RF), Extra trees (ET) y Gradient boosting (GB). De este análisis se puede discutir y concluir lo siguiente:

- Actividades como caminar y correr pudieron ser clasificadas de forma correcta por los clasificadores ya que contienen suficiente información para diferenciar actividades que generan un sonido ambiental similar.
- k Nearest Neighbors (kNN) y Extra trees (ET) resultaron ser los clasificadores más precisos a la hora de evaluar el archivo de audio de las diferentes actividades.
- En contraste con lo anterior, Support Vector Machines (SVM) y Random forests (RF) que resultaron ser los clasificadores menos acertados en esta experimentación.

No obstante, se tiene la inquietud de que un Data-Set más grande sería de bastante ayuda para el clasificador a la hora de su entrenamiento, permitiendo así obtener un modelo de estimación de actividades infantiles altamente preciso.

6. Trabajo futuro

Como trabajo futuro, más actividades habitualmente realizadas por infantes de 12 a 36 meses serán agregadas al Data-Set y se realizarán más pruebas con

los clasificadores presentados con anterioridad a dichas actividades, de manera adicional se propone los siguientes puntos específicos:

- Aplicar otros clasificadores de datos para hacer una comparación y lograr obtener el mejor clasificador enfocado al reconocimiento y clasificación de actividades realizadas por infantes.
- Agregar más actividades realizadas por infantes.
- Optimizar la clasificación de las actividades realizadas por infantes.

Referencias

1. Jalal, A., Kamal, S., Kim, D.: A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments. Sensors (14248220), 11735–11759 (2014)
2. Mun Sim, J., Lee, Y., Kwon O.: Acoustic Sensor Based Recognition of Human Activity in Everyday Life for Smart Home Services. International Journal of Distributed Sensor Networks, 1–11 (2015)
3. Onkar Nehete, J., Agrawal, D.G.: Real time Recognition and monitoring a Child Activity based on smart embedded sensor fusion and GSM technology. The International Journal Of Engineering, 35–40 (2015)
4. Anusha, A.M., Belagali R., Mahendrachari, Prashant: Child Activity Recognition Using Accelerometers. In: NCRIET-2015 & Indian, 007–012 (2015)
5. Boughorbel, Sabri, Breebaart, Jeroen, Bruekers, Fons, Flinsenberg, Warner Ten, K.: Child-activity recognition from multi-sensor data. In: Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research - MB 10, (2010)
6. Nam, Y., Wook Park, J.: Physical activity recognition using a single triaxial accelerometer and a barometric sensor for baby and child care in a home environment. Ambient Intelligence & Smart Environments, 381–402 (2013)
7. García-Domínguez, A., Galván-Tejada, C.E.: Reconocimiento de actividades infantiles utilizando sonido ambiental: Un enfoque preliminar. Research in Computing Science 139 (2017), 71–79 (2017)
8. Galván-Tejada, C.E., Galván-Tejada, J.I., Celaya-Padilla, J.M., Delgado-Contreras, J.R., Magallanes-Quintanar, R., Martinez-Fierro, M.L., Garza-Veloz, I., López-Hernández, Y., Gamboa-Rosales, H.: An Analysis of Audio Features to Develop a Human Activity Recognition Model Using Genetic Algorithms, Random Forests, and Neural Networks. Mobile Information Systems, 1–10 (2016)
9. Freesound, `https://freesound.org/`
10. Giannakopoulos, T.: pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. PloS one (2015)

ISSN 1870-4069

Método de compresión de electrocardiogramas basado en muestreo compresivo

Rodolfo Moreno-Alvarado[1], Héctor Pérez-Meana[2], Mariko Nakano-Miyatake[2], Daniel Robles-Camarillo[1]

[1] Universidad Politécnica de Pachuca, Zempoala, Hidalgo, México,

[2] Instituto Politécnico Nacional, Esime Culhuacan, México

rg.moreno@lasallistas.org.mx., {hmperezm, mnakano}@ipn.mx, danielrc@upp.edu.mx

Resumen. El muestreo compresivo (MC) es un enfoque prometedor que ayuda a comprimir y recuperar señales de electrocardiograma (ECG). Esta técnica explota la dispersión y la compresibilidad de las señales de ECG con una representación dispersa en cierto dominio Ψ usando la matriz de medición Φ, que satisface la propiedad de isometría restringida. El método propuesto combina primero la dispersión y la compresibilidad de las señales de ECG en el dominio de la Transformada Wavelet Discreta (DWT) para lograr una señal dispersa y compresible. En segundo lugar, utilizando una matriz de medición ortogonal (MMO) basada en Transformada de Coseno Discreta (DCT) tomamos medidas incoherentes y finalmente se mejora el proceso de recuperación usando la traspuesta de (MMO) en vez de los algoritmos de reconstrucción de MC. Los resultados experimentales utilizan la base de datos de arritmias MIT-BIH mostrando que pueden obtenerse ganancias significativas, en términos de la tasa de compresión y calidad de reconstrucción, mediante el algoritmo propuesto en comparación con los métodos de MC actuales.

Palabras clave: Electrocardiograma, muestreo compresivo, coherencia, dispersión.

Electrocardiogram Compression Method Based on Compressive Sampling

Abstract. Compressive sampling (CS) is a promising approach which aids to compress and recover electrocardiogram (ECG) signals. This technique exploits the sparseness and compressibility of ECG signals with sparse representation in

certain domain Ψ using measurement matrix Φ which satisfies the restricted isometry property. The proposed method first combines the sparseness and compressibility of ECG signals in Discrete Wavelet Transform (DWT) domain to achieve a sparse and compressible signal, secondly using an orthogonal measurement matrix (OMM) based on Discrete Cosine Transform (DCT) we take incoherent measurements and finally improve the recovery process, with the OMM transpose instead the use of typical reconstruction algorithms for CS. Experimental results utilizing the MIT-BIH Arrhythmia Database show that significant performance gains, in terms of compression rate and reconstruction quality, can be obtained by the proposed algorithm compared to current CS methods.

Keywords: Electrocardiogram, compressive sampling, coherence, sparsity.

1. Introducción

Estudios de la OMS reportan a las Enfermedades cardiovasculares (ECV) como la principal causa de muerte en todo el mundo. Se calcula que en 2012 murieron por esta causa 17,5 millones de personas, lo cual representa un 31% de todas las muertes registradas en el mundo. Para las personas con ECV o con alto riesgo cardiovascular (debido a la presencia de factores de riesgo, como la hipertensión arterial, la diabetes o la hiperlipidemia) son fundamentales la detección y el tratamiento temprano [1]. Así, a fin de detectar y realizar un diagnóstico clínico temprano de forma no invasiva de las ECV se usa el ECG, el cual, nos revela información importante sobre el estado del corazón por lo que es considerado un estándar en el diagnóstico de arritmias.

Debido a que el registro del ECG se lleva a cabo mediante sistemas de monitoreo, estos, deben de ser capaces de manejar una gran cantidad de datos, almacenarlos, procesarlos y en algunos casos transmitirlos, ya sea a otro sistema o al profesional de la salud encargado para su evaluación, por tal motivo, es de gran importancia la compresión del registro del ECG generado.

Actualmente, aunque se tiene un auge en los métodos de compresión y son diversos, es necesario desarrollar algoritmos de compresión de ECG que produzcan mejores relaciones de compresión y una mayor calidad en los datos que son reconstruidos. El ya conocido enfoque tradicional de compresión y reconstrucción de señales o imágenes a partir de datos medidos sigue al teorema de muestreo de Shannon-Nyquist Whittaker-Kotelnikov, el cual establece que la frecuencia de muestreo debe ser al menos del doble de la frecuencia más alta presente en la señal [8]. Del mismo modo, el teorema fundamental del álgebra lineal sugiere que el número de muestras recolectadas (mediciones) presentes en una señal con dimensión finita debe ser al menos tan grande como su longitud (dimensión) para garantizar la reconstrucción.

En los últimos años se ha venido gestando una nueva teoría del muestreo, la teoría del MC [3], que, por su parte, permite la reconstrucción de señales a partir de menos muestras (mediciones) o datos; que las sugeridas por el muestreo convencional. Esta novedosa teoría propuesta por Donoho [8] y Candes [3] proporciona un enfoque

fundamentalmente nuevo para la adquisición y compresión de datos simultáneamente [4].

En comparación con los algoritmos de compresión de ECG convencionales, el MC tiene algunas ventajas importantes: entre las cuales se encuentra la transferencia comprimida y codificada de la señal de interés [12] y la fácil integración de codificadores en hardware por mencionar algunas. Así, el MC va teniendo un gran auge en la rama de ECG, existen diferentes trabajos que implementan de manera formidable cada vez más esta técnica: Desde la implementación de sistemas dedicados [5, 14], compresión de datos [5], transmisión [7], arquitectura de codificadores [16], matrices de medición [16, 17], medidas contra el ruido [18], y la reconstrucción eficiente de los coeficientes de baja magnitud [19, 21], entre otros. Sin embargo, a pesar de los avances, todos los trabajos mencionados anteriormente estos, utilizan algoritmos de reconstrucción que incrementan el costo computacional en su implementación.

En este trabajo se propone un método de compresión basado en muestreo compresivo, el cual tiene como objetivo la recuperación de la señal de ECG a partir de la traspuesta de la matriz de medición generada, sin realizar iteraciones como en ciertos algoritmos de reconstrucción de MC. A lo largo de este proceso se usa la Transformada de Wavelet Discreta (DWT) así como la Transformada Inversa de Wavelet Discreta (IDWT) con el fin de dispersar, comprimir y recuperar nuestras señales de ECG. El rendimiento de este método de compresión propuesto se mide en términos del radio de compresión (CR) y la calidad de la señal reconstruida usando el porcentaje de diferencia cuadrático medio (PRD), así como el porcentaje de diferencia cuadrático medio normalizado (PRDN) los cuales se evalúan utilizando los registros extraídos de la base de datos de arritmias del MIT-BIH [15].

Los resultados de PSNR y NPSNR validan el algoritmo propuesto en comparación con el algoritmo presentado en [20]. El resto del documento está organizado de la siguiente manera. La Sección 2 presenta el Marco Teórico del MC. La Sección 3 detalla la dispersión de la señal de ECG, la construcción de la matriz de muestreo y las medidas utilizadas para la evaluación de los resultados. La Sección 4 presenta la metodología desarrollada. La Sección 5 presentan los resultados de la simulación. La Sección 6 presenta la Discusión y las principales conclusiones respectivamente.

2. Muestreo compresivo

Esta sección comprende los conceptos básicos del muestreo compresivo. Como ya se ha mencionado anteriormente, el MC es una técnica que permite reconstruir una señal a partir de pocas mediciones, es importante establecer que esta teoría se basa en dos principios: La dispersión y la incoherencia [3]. Dado que una señal puede ser representada bajo diferentes bases, se dice que está cumple con la condición adecuada de dispersión si en esta representación cuenta con un número pequeño de elementos k distintos de cero, siendo esta condición fundamental para las señales compresibles. Por otro lado, la propiedad de incoherencia está inversamente relacionada con la coherencia, medida que indica la correlación que se tiene entre los elementos de la matriz de medición y la matriz o base en la cual se desarrolló la dispersión de la señal.

Por lo anterior, es posible expresar el MC como un problema inverso, ver ecuación (1), donde en el contexto de los principios antes mencionados $x \in R^N$ es el vector $N \times 1$ que representa a la señal de ECG en el dominio disperso y a su vez Φ representa la matriz de medición (el sensor que adquiere la información), así podemos denotar a las muestras de la señal comprimida como $z \in R^{MxN}$ donde $M < N$:

$$z = \Phi \mathrm{x}. \tag{1}$$

Una vez que se tiene la señal comprimida z, esta se envía al receptor donde se recupera y decodifica por medio de la matriz Φ y los algoritmos de recuperación propuestos. Para recuperar con éxito la señal de ECG se requiere que x sea dispersa, en el caso en el que x no sea dispersa se busca una matriz de dispersión Ψ de tal forma que x pueda ser representada con pocos elementos. En general, $x = \Psi\alpha$ se construye usando distintas bases, en este trabajo, se considera como bases de dispersión a funciones wavelet.

Entonces, un algoritmo de MC puede recuperar α usando las mediciones disponibles z y la matriz $\Theta = \Phi\Psi$, modelo que ha sido ampliamente utilizado para señales de ECG [12,14]. Por lo tanto, reescribiendo la ecuación (1) esta queda en términos de los coeficientes de señal dispersos α como:

$$z = \Phi \mathrm{x} = \Phi\Psi\alpha = \Theta\alpha. \tag{2}$$

Cabe mencionar que el proceso de medición no es adaptativo, lo que significa que Φ es fijo y no depende de la señal x, como se puede apreciar, es necesario construir una matriz Φ de tal forma que a pesar de reducir la dimensión de $x \in R^N$ a $z \in R^M$ la información esencial contenida en la señal x no sea distorsionada o corrompida por este proceso. Una forma en la cual la matriz Φ pueda realizar este proceso es cumpliendo la propiedad de isometría reservada [4] ecuación (3). Donde se muestra que la constante de isometría reservada δ_s de la matriz $\Phi \in R^{mxN}$ se define como la más pequeña si se cumple para toda $x \in R^N$ en el espacio disperso S:

$$(1 - \delta_s)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s)\|x\|_2^2. \tag{3}$$

Así, una matriz Φ satisface la propiedad de isometría reservada si δ_s es pequeña, otra condición relacionada se denomina coherencia, y fue introducida al MC en [3]. Para ser formal, la coherencia está definida por el máximo valor absoluto de la correlación cruzada entre las columnas de la matriz Φ y la matriz Ψ como se muestra en la ecuación (4). En particular, se usa como una medida de calidad para la matriz Φ, donde se busca que la coherencia sea muy pequeña lo que significa que su incoherencia será mayor y cumplirá esta propiedad:

$$\mu(\Phi, \Psi) = \max|\Phi_k \Psi_k|. \tag{4}$$

Las matrices más usadas y que cumplen estas propiedades son: las matrices gaussianas y binarias, aunque existen una gran cantidad de trabajos relacionados con matrices deterministas, no deterministas, adaptativas, no adaptativas, en las que se cumple con las propiedades citadas [4].

Por otra parte, es necesario desarrollar a la par un algoritmo de reconstrucción capaz de recuperar x a partir de las mediciones z dadas por la ecuación (2). Por lo tanto, un sistema de MC queda determinado por la toma de muestras z de una señal x a partir de una matriz Φ y una reconstrucción de x a partir de las muestras M tal que $M < N$.

En el caso especial de $M = N$, x se recupera a partir de la inversa de Φ, sin embargo, para el caso $M < N$, el sistema se vuelve indeterminado al existir un numero infinito de soluciones para x; sin embargo, si x cumple con las propiedades de dispersión e incoherencia la solución más dispersa es la solución correcta.

Una vez realizado el proceso de adquisición, se obtiene una estimación de la señal mediante un algoritmo de reconstrucción. Un enfoque práctico utilizado para determinar la solución dispersa es resolver este problema como un problema optimización convexa. Los trabajos originales versan sobre la minimización de la norma l_1 y la programación lineal, como se aprecia en la ecuación (5), la señal se reconstruye usando el problema de optimización [3,4]:

$$min\|\hat{x}\|_1 \;\; sujeto\ a\ \; z = \Phi\Psi\alpha, \tag{5}$$

donde: $\|\cdot\|_1$ denota la norma l_1 de un vector disperso, Ψ es la matriz de dispersión, Φ es la matriz de muestreo y α son los coeficientes dispersos, de tal forma que al recuperar la señal se tiene $z = \Phi\hat{x}$ donde $\hat{x}$ es la solución óptima.

En el caso de los algoritmos de reconstrucción podemos mencionar una amplia gama de técnicas que incluyen: Búsqueda de Correspondencia Ortogonal (Orthogonal Matching Pursuit), Búsqueda de Base (Basis Pursuit) [4] entre otros, el método se selecciona dependiendo de la aplicación, el rendimiento, el tiempo y las necesidades de velocidad es importante mencionar que cada uno de estos métodos debe realizar un numero de iteraciones determinado para recuperar la señal de interés.

3. Propiedades del muestreo compresivo aplicadas al electrocardiograma

Esta sección comprende los conceptos que intervienen y están relacionados con las señales de ECG y el MC. Como se mencionó en las secciones anteriores el MC se fundamenta en dos propiedades: la dispersión, propiedad que tiene que ver directamente con la señal de interés y la incoherencia propiedad que hace referencia a la matriz de muestreo.

Dispersión de la señal de ECG. El grado de dispersión es proporcional al número de componentes cero presentes en la señal, de tal forma que a mayor número de componentes cero, mayor dispersión y viceversa, así, si observamos detenidamente una señal de ECG, Figura 1, podemos apreciar que, en el dominio del tiempo, esta presenta muy poca dispersión, en general, una señal de ECG por sí sola no es dispersa, y si mencionamos que el éxito del MC está directamente relacionado con la forma en la cual se representa nuestra señal, es de suma importancia representarla en un dominio distinto de manera que esta sea dispersa.

Debido a lo anterior, la mayoría de los trabajos sobre MC y ECG representan a la señal de ECG a partir de la Transformada Discreta de Fourier (FFT), la Transformada

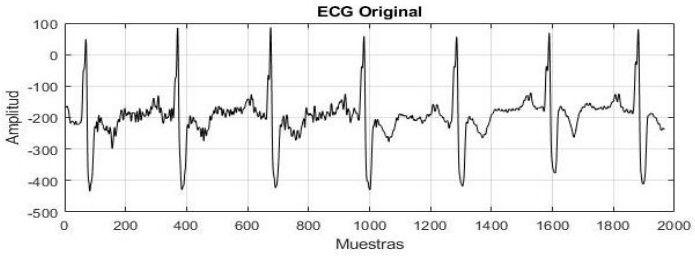


Fig. 1. Señal de Electrocardiograma.

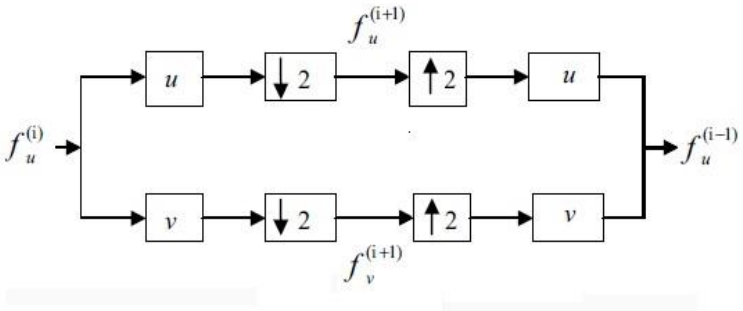


Fig. 2. Diagrama de Bloques de la DWT diádica.

Discreta de Coseno (DCT) o la Transformada Discreta de Wavelet (DWT) entre otros Finalmente esta última, la DWT es la que tiene el mayor auge dentro del MC debido a las propiedades intrínsecas de multi-resolución y compactación de energía [6]. Así, la DWT usada en nuestro método consiste en la descomposición de una señal con longitud N en $\log_2 N$ niveles; llevada a través de un proceso de filtrado en serie y decimado para su compresión, por lo que para la reconstrucción de la señal original se interpola, se filtra en serie y se añaden sub-bandas, la Figura 2 muestra el diagrama de bloques para ambos procesos [10].

Cabe mencionar que la señal de entrada es filtrada usando un filtro paso-bajas (u) para obtener los coeficientes de aproximación y un filtro paso-altas (v) para los coeficientes de detalle, el proceso de filtrado se consigue por medio de la convolución entre los coeficientes del filtro y la señal de entrada, para conseguir una reconstrucción perfecta estos filtros deben de ser ortogonales [10]. Así los coeficientes de detalle nos dan información sobre la frecuencia baja y los de aproximación sobre la frecuencia alta, comúnmente para dispersión la frecuencia baja es de mayor importancia que la frecuencia alta por lo que los coeficientes de aproximación son siempre la entrada de los siguientes niveles de descomposición, lo que nos permite descartarlos.

De esta forma dada una señal x de longitud N y los filtros paso bajas (u) y paso altas (v), los niveles de descomposición quedan representados por las ecuaciones (6) y (7), mientras que el proceso de reconstrucción se obtiene por medio de la ecuación (8). [10]:

$$f_u^{(i)}(j) = \sum_{k=1}^{2^{n-i+1}} u(k-2j) f_u^{(i-1)}(k) \quad j = 1,2,\dots,2^{n-i}, \tag{6}$$

$$f_v^{(i)}(j) = \sum_{k=1}^{2^{n-i+1}} v(k-2j) f_v^{(i-1)}(k) \quad j = 1,2,\dots,2^{n-i}, \tag{7}$$

$$f_u^{(i-1)}(j) = \sum_{k=1}^{2^{n-i+1}} u(j-2k) f_u^{(i)}(k) + \sum_{k=1}^{2^{n-i+1}} v(j-2k) f_v^{(i)}(k). \tag{8}$$

Matriz de muestreo. La matriz de muestreo tiene un papel importante, debido a que es la encargada de llevar a cabo las mediciones correctas para que sea posible la reconstrucción de la señal de ECG en el proceso siguiente, por lo que varios autores con el fin de cumplir con los criterios de incoherencia e isometría reservada han propuesto una serie de matrices para ECG como lo son las ampliamente utilizadas matrices aleatorias (Gaussianas o Binarias) [13], mientras que otros autores proponen matrices estructuradas como Toeplitz [2], circulares, o determinísticas [9].

Matriz transformada de Coseno. Esta matriz de NxN es ampliamente utilizada en el procesamiento de imágenes en bloques pequeños [17], y puede expresarse de la siguiente forma:

$$\Phi = \begin{cases} \frac{1}{\sqrt{M}} & p = 0, \quad 0 \le q \le M-1, \\ \sqrt{\frac{2}{M}} cos \frac{\pi(2q+1)p}{2M} & 1 \le p \le M-1, \quad 0 \le q \le M-1. \end{cases} \tag{9}$$

De esta forma para una matriz A de NxN la operación $\Phi * A$ da como resultado una matriz de NxN en la que cada columna contiene la DCT de una dimensión de las columnas de A. Cabe mencionar que al ser Φ ortogonal su inversa es la misma que su traspuesta.

Medidas utilizadas. Cualquier algoritmo de compresión debe tener la habilidad de preservar la calidad de la señal una vez que esta es descomprimida, aunado a esto, se debe contar con un buen radio de compresión, aunque lo único realmente importante es la calidad, pues es posible tener compresiones muy grandes con una pérdida de información proporcional. Así, en la compresión de ECG se debe considerar dos factores:

1. Eficiencia de compresión la cual puede ser medida con el radio de compresión (CR) por medio de la ecuación (10); donde b_o hace referencia a los bits presentes en la señal de ECG original y b_c a los bits de la señal comprimida, para nuestro caso se usa $b_o = 2000$ y $b_c = 312$, siendo $CR = 6.4$.

$$CR = \frac{b_o}{b_c}. \tag{10}$$

2. Calidad del esquema de compresión, que para esquemas con pérdida está determinada por la comparación entre los datos descomprimidos y los datos

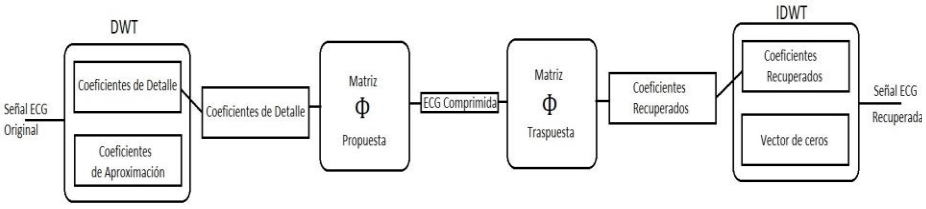


Fig. 3. Diagrama de bloques del algoritmo de compresión propuesto.

originales. Si no existen diferencias la compresión es sin pérdida, mientras que las medidas convencionales están basadas en distorsión, como lo son: el porcentaje de diferencia cuadrático medio (PRD), ecuación (11); y el porcentaje de diferencia cuadrático medio normalizado (PRDN) ecuación (12) [20]:

$$PRD = \frac{\|x-\hat{x}\|_2}{\|x\|_2}, \tag{11}$$

$$PRDN = \frac{\|x-\hat{x}\|_2}{\|x-\bar{x}e\|_2}, \tag{12}$$

donde para ambas ecuaciones x y $\hat{x}$ representan a la señal original y a la señal recuperada respectivamente, en el caso de la ecuación (12) e representa un vector n-dimensional de unos y $\bar{x}$ representa el valor promedio de x.

4. Algoritmo propuesto

El diagrama de bloques del método de compresión propuesto es presentado en la Figura 3, El proceso inicia con una transformación lineal seguida del descarte de los coeficientes de aproximación y nuevamente una transformación lineal con la matriz de coseno para obtener los datos comprimidos del ECG. Los detalles de su implementación se detallan a continuación.

Dispersión propuesta de la señal de ECG. El proceso inicia con el procesado de la señal de ECG en tramas de N muestras sin traslape, denotando x como el vector de ECG N dimensional que corresponde a la trama seleccionada y con $x \in R^N$, se dispersa la señal a través de una transformación lineal, en la cual se utiliza la DWT con una base ortonormal de Symlet, $\Psi = [\Psi_1, \Psi_2 \cdots \Psi_N]$. Donde Ψ es la representación matemática de la estructura de filtros mencionados en las ecuaciones (6) y (7) obteniéndose una representación conjunta de la señal en términos de los coeficientes de detalle u, y los coeficientes de aproximación v , así podemos denotar $\alpha = \Psi x$ donde $\alpha = \begin{bmatrix} u \\ v \end{bmatrix}$ es la señal dispersa ECG con ambos coeficientes [11]

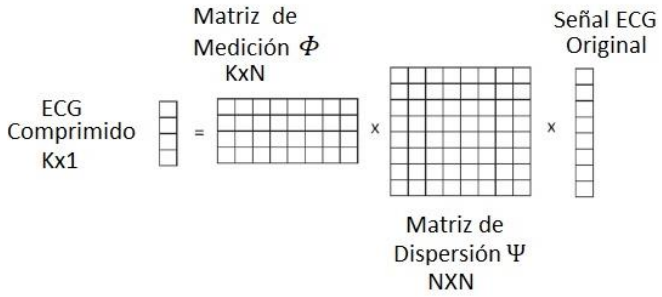


Fig. 4. Estructura matricial del esquema propuesto.

Tabla 1. Contraste de PRD Y PRDN.

	Método Propuesto			**Método de Polania et al 2015[20]**			
	MMO			**MMB-IHT**		**MMB-CoSaMP**	
Señal	CR	PRD	PRDN	PRD	PRDN	PRD	PRDN
115	6.4	0.6816	0.9876	2.74	6.62	2.57	6.19
118	6.4	0.5879	1.0054	2.54	4.61	2.86	5.2
119	6.4	0.5531	0.8109	2.29	4.5	2.61	5.12

Tabla 2. Relación de Coherencia entre la señal y la matriz de medición.

Señal	**Coherencia**
115	0.0012
118	0.0009
119	0.0042

Matriz de muestreo. Una vez que se ha obtenido nuestra señal dispersa, $\boldsymbol{\alpha = \Psi x}$ se procede a solo tomar los coeficientes de detalle y descartar los coeficientes de aproximación, los cuales presentan una dimensión de $\boldsymbol{N/4}$ siendo $\boldsymbol{N}$ la dimensión de nuestra señal original de ECG.

De notando a α_d como los coeficientes de detalle de nuestra señal de ECG se procede a construir la matriz de muestreo en base a la matriz de DCT de la ecuación (9).

$$\Phi = \begin{cases} \frac{1}{\sqrt{L}} & S = 0, \quad 0 \le W \le L-1 \\ \sqrt{\frac{2}{L}} cos\frac{\pi(2W+1)S}{2L} & 1 \le S \le L-1, \quad 0 \le W \le L-1 \end{cases}, \tag{13}$$

donde L es la dimensión dada por los coeficientes de detalle α_d , S y W son matrices compuestas de la siguiente forma:

$$S = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ L & \cdots & L \end{bmatrix} ; \; W = \begin{bmatrix} 0 & \cdots & L \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L \end{bmatrix}.$$

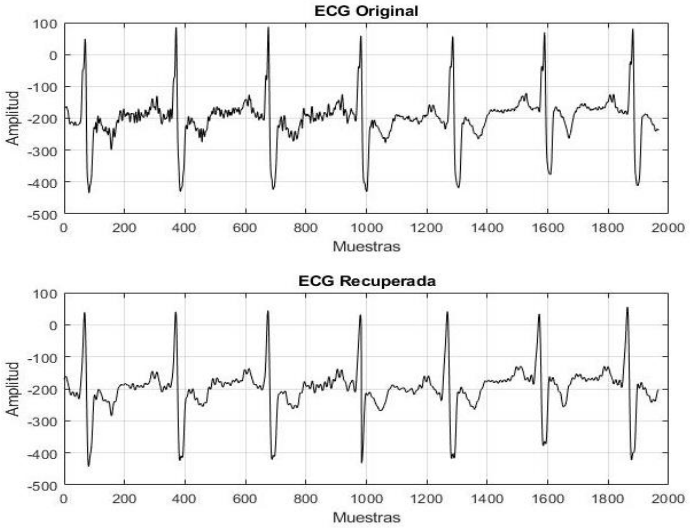


Fig. 5. Señal 118 de 2000 muestras original y recuperada.

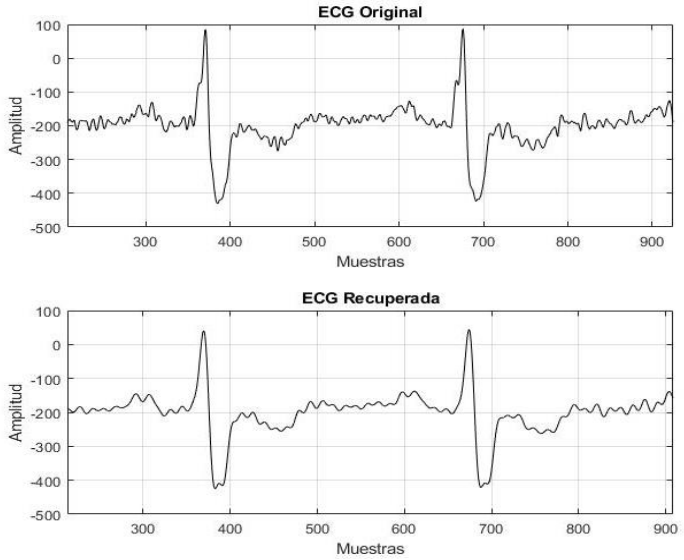


Fig. 6. Segmentos de la señal 118 de 2000 muestras original y recuperada.

Una vez construida la matriz mostrada en la ecuación (13), se procede nuevamente a realizar una transformación lineal de la forma $z = \Phi\alpha_d$ donde los renglones de la matriz Φ serán modificados de acuerdo con el nivel de compresión que se requiera. El proceso de transformación se muestra en la Figura 4. Debido a que una de las propiedades implícitas en la matriz DCT es la ortogonalidad, sabemos que la inversa de Φ es la traspuesta de Φ.

Por lo que el proceso de reconstrucción se da a través de la aplicación de la matriz traspuesta Φ a las mediciones y ,que contienen la señal de ECG comprimida, después

se aplica la Transformada Inversa de Wavelet (IDWT) con el vector de los coeficientes de aproximación en cero y finalmente se recupera nuestra señal de ECG.

5. Resultados y discusión

Para evaluar el algoritmo propuesto se usa la base de datos de MIT-BIH-Arrhythmia [15] como nuestras señales de ECG, cada señal de ECG tiene 11-bits de resolución sobre 10mV y esta muestreada a una frecuencia de 360Hz, los experimentos son llevados a cabo sobre los registros extraídos con una longitud de 10 min, tomando una longitud de 2000 muestras, para una tasa de compresión de 6.4, los registros utilizados son 115, 118, 119.

La forma de procesar cada ECG es por medio del algoritmo descrito en la sección previa, para la evaluación se utilizan las métricas descritas en la sección 3.3. cabe mencionar que estas mediciones no son acordes para los ECG ya que solo reflejan la distorsión presente en la señal desde un criterio estadístico.

En la Tabla 1 se muestra los valores del PRD y PRDN para las tres señales presentadas a una tasa de compresión de 6.4, las cuales son contrastadas contra los valores obtenidos por [15] cabe mencionar que, aunque se tiene un mejor desempeño esta no es una medida óptima para evaluar el ECG debido a que este es para uso médico por lo que es necesaria la evaluación de un profesional de la salud.

En la Tabla 2 se muestra la coherencia entre la señal original de ECG y la matriz de medición, como se vio en las secciones anteriores: a menor coherencia se llega a una mejor reconstrucción pues cumple con la propiedad de isometría reservada.

Las Figuras 5,6 y 7 nos muestran las señales originales, así como las recuperadas, específicamente en los todos los casos se encuentra una señal recuperada inteligible, donde el médico es capaz de realizar un diagnóstico.

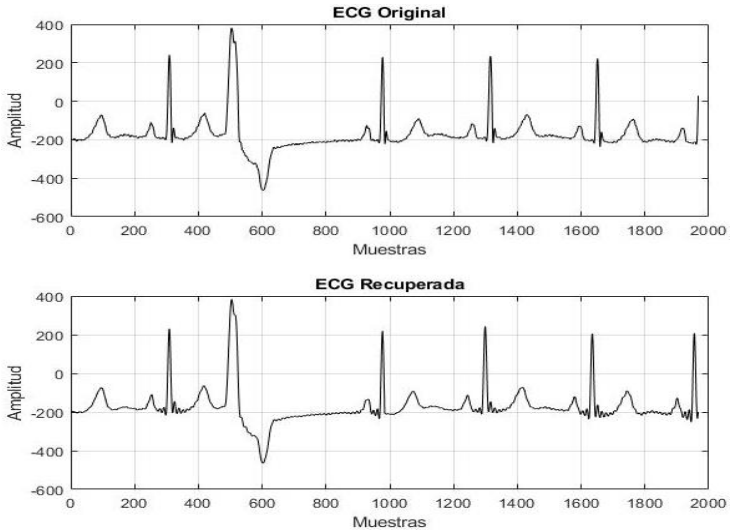


Fig. 7. Señal 119 de 2000 muestras original y recuperada.

Con el fin de evaluar el desempeño del algoritmo se realiza un acercamiento para el caso de la señal 118 donde podemos ver que a pesar de no tener una etapa de reconstrucción con pocas muestras es posible recuperar inteligiblemente la señal de ECG.

Para señales más complejas el algoritmo muestra, de la misma forma que tiene un buen desempeño ya que es posible recuperarlas; en algunas partes la señal presenta una ligera distorsión, sin embargo, siguen siendo inteligibles para el profesional de la salud.

6. Conclusiones

El Algoritmo Propuesto cumple con las condiciones base de la teoría del muestreo compresivo, se efectúa la dispersión de la señal por medio de la función wavelet symlet, cumpliendo con la propiedad de dispersión, se cumplen con las propiedades de isometría reservada e incoherencia, al ser la coherencia de nuestra matriz cercana a cero como se aprecia en la Tabla 2. La matriz de medición es ortogonal lo que hace que el proceso de recuperación sea alcanzable con tan solo trasponer la matriz sin necesidad de utilizar algoritmos de reconstrucción los cuales tienen una mayor complejidad computacional y algunos de ellos son basados en iteraciones. El algoritmo propuesto es capaz de comprimir la señal hasta el punto de solo enviar pocos datos los cuales son codificados y recuperados por la misma matriz con una calidad inteligible para el profesional de la salud.

Agradecimientos. Este trabajo ha sido realizado con el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT).

Referencias

1. Alwan, A.: Global Status Report on Noncommunicable Diseases 2010. World Health Organization, Geneva, Switzerland (2011)
2. Ahmed, N., Natarajan, T., Rao, K.R.: On image processing and a discrete cosine transform. IEEE Transactions on Computers, 23(1), pp. 90–93 (1974)
3. Candes, E., Romberg, J., Tao, T.: Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. IEEE Transactions on Information Theory, 52, pp. 489–509 (2006)
4. Candes, E.J., Wakin, M.B.: An Introduction to Compressive Sampling. IEEE Signal Processing Magazine, 25, pp. 21–30 (2008)
5. Chen, F., Chandrakasan, A.P., Stojanovic, V.M.: Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors. IEEE Journal of Solid-State Circuits, 47, pp. 744–756 (2012)
6. Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. IEEE transactions on information theory, 36(5), pp. 961–1005 (1990)
7. Dixon, A.M.R., Allstot, E.G., Gangopadhyay, D., Allstot, D.J.: Compressed Sensing System Considerations for ECG and EMG Wireless Biosensors. IEEE Transactions on Biomedical Circuits and Systems, 6, pp. 156–166 (2012)
8. Donoho, D.L.: Compressed Sensing. IEEE Transactions on Information Theory, 52, pp. 1289–1306 (2006)

9. Haupt, J., Bajwa, W.U., Raz, G., Nowak, R.: Toeplitz compressed sensing matrices with applications to sparse channel estimation. IEEE Transactions on Information Theory, 56, pp. 5862–5875 (2010)
10. Iwen, M. A.: Compressed sensing with sparse binary matrices: Instance optimal error guarantees in near optimal time. Journal of Complexity, 30(1), pp. 1–15 (2014)
11. Yan, J.: Wavelet Matrix. http://www.ece.uvic.ca/~jyan/ (2018)
12. Kumar, A., Kumar, G., Singh, K.: Electrocardiogram signal compression using singular coefficient truncation and wavelet coefficient coding. IET Science, Measurement & Technology, 10(4), pp. 266–274 (2016)
13. Lorne, A., Stephen, D. H., Stephen, S., Calderbank, R.: Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. Applied and Computational Harmonic Analysis, 26, pp. 283–290 (2009)
14. Mamaghanian, H., Khaled, N., Atienza, D., Vandergheynst, P.: Compressed Sensing for Real-Time Energy- Efficient ECG Compression on Wireless Body Sensor Nodes. IEEE Transactions on Biomedical Engineering, 58, pp. 456–2466 (2011)
15. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH Arrhythmia Database. IEEE Eng. Med. Biol. Mag. 20, pp. 45–50 (2001)
16. Polania, L.F., Carrillo, R.E., Blanco-Velasco, M., Barner, K.E.: Compressive Sensing Exploiting Wavelet Domain Dependencies for ECG Compression. In: Proceedings of the SPIE Defense, Security, and Sensing, Baltimore, pp. 23–27 (2012)
17. Polania, L.F., Carrillo, R.E., Blanco-Velasco, M., Barner, K.E.: On Exploiting Interbeat Correlation in Compressive Sensing-Based ECG Compression. In: Proceedings of the SPIE Defense, Security, and Sensing, Baltimore, pp. 23–27 (2012)
18. Polania, L.F., Carrillo, R.E., Blanco-Velasco, M., Barner, K.E.: Compressive Sensing for ECG Signals in the Presence of Electromyography Noise. In: Proceedings of the 38th Annual Northeast Bioengineering Conference (NEBEC), Philadelphia, pp. 16–18 (2012)
19. Polania, L.F., Carrillo, R.E., Blanco-Velasco, M., Barner, K. E.: Matrix completion-based ECG compression. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, pp. 1757–1760 (2011)
20. Polanía, L, Carrillo, R.E., Blanco-Velasco, M., Barner, K. E.: Exploiting prior knowledge in compressed sensing wireless ECG systems. IEEE Journal of Biomedical Health Informatics, 19(2), pp. 508–519 (2015)
21. Zhang, Z., Jung, T., Makeig, S., Rao, B.D.: Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Non-Invasive Fetal ECG via Block Sparse Bayesian Learning. IEEE Transactions on Biomedical Engineering, 60(2), pp. 300–309 (2013)

ISSN 1870-4069

Detección de comunidades en redes sociales por medio de un algoritmo de agrupamiento dinámico en alta definición

Christian Iván Ledesma Bermúdez[1], Abel García Najera[2]

[1] Universidad Autónoma Metropolitana Unidad Cuajimalpa,
Posgrado en Ciencias Naturales e Ingeniería,
México

[2] Universidad Autónoma Metropolitana Unidad Cuajimalpa,
Departamento de Matemáticas Aplicadas y Sistemas,
México

2163807450@alumnos.cua.uam.mx, agarcian@correo.cua.uam.mx

Resumen. Las redes sociales han sido foco de un creciente interés por el amplio espectro de aplicaciones que han demostrado tener, desde marketing hasta política. Por este motivo todos los enfoques de estudio relacionados a éstas tienen una gran importancia. Existen trabajos relacionados con la detección de comunidades que estudian el problema desde varios enfoques: algunos lo tratan como un problema estático, mientras que los más recientes lo tratan como un problema dinámico. Ambos enfoques presentan ventajas y desventajas que hacen que la selección del enfoque se decida en parte por la naturaleza del problema. En este trabajo se sigue el enfoque de AI-NSGA-II y, para mejorar la forma en que se detectan las comunidades, se considera una función objetivo adicional para lograr definir mejor el agrupamiento. Posteriormente, el desempeño de la propuesta HD-AI-NSGA-II se compara con AI-NSGA-II usando conjuntos de datos artificiales y un conjunto de datos real bien conocido que contiene ID de videos de YouTube. Finalmente se demuestra que nuestro enfoque HD-AI-NSGA-II destaca al lograr una mejor definición de los grupos y, como resultado directo, una detección sobresaliente de comunidades gracias a la forma en que guía la búsqueda del frente de Pareto.

Palabras clave: Algoritmos evolutivos multiobjetivo, agrupamiento multiobjetivo, detección de comunidades, redes sociales.

Community Detection in Social Networks by a High-Definition Dynamic Clustering Algorithm

Abstract. Social networks have been focus of growing interest for the wide spectrum of applications they have, from marketing to politics. For this reason all approaches related to them have a great importance. There are several studies

releated to community detection that tackle the problema from several perspectives: some of them study it as a static problem, while the more recent ones consider it as a dynamic problem. In this work the AI-NSGA-II approach is followed and, in order to improve the way in which communities are detected, an additional objetive function is considered, which manages to better define the clustering. Subsequently, the permformance of our proposed HD-AI-NSGA-II is compared against AI-NSGA-II using artifical data sets and a well-known real instance containing YouTube videos ID's. Finally, it is demonstrated that the proposed HD-AI-NSGA-II improves the other approach by achieving a better definition of the clusters and, as a direct result, a better community detection thanks to the way in which it guides the search of the Pareto front.

Keywords: Multi-objective evolutionary algorithms, multi-objective clustering, community detection, social networks.

1. Introducción

En la actualidad, la detección de comunidades es uno de los tópicos con mayor importancia por las aplicaciones que puede tener en redes sociales, economía, marketing, política, búsqueda de información, minería de datos, adquisición de conocimiento, análisis de datos, informática, física, sociología, etc. [1-8]. Algunas de las aplicaciones se han combinado para desarrollar un papel principal en los últimos años, como el caso de las redes sociales y la política. Por estos motivos, la detección de comunidades es un problema ampliamente investigado, abierto y de relevancia [1, 9-12]. Este problema puede ser modelado como uno de optimización multiobjetivo, ya que comúnmente se plantean varios objetivos que están en conflicto para lograr un buen conjunto de soluciones, en particular, se modela como un problema de partición de un grafo. En este sentido, los algoritmos genéticos son una de las herramientas más importantes para la optimización de problemas multiobjetivo NP-difíciles [2, 9]. Tomando en cuenta la estructura temporal del grafo, hay dos enfoques que abordan el tema, el primero considera el problema como un caso estático, en donde el estado del grafo no cambia con el tiempo [7, 11, 13-17], mientras que el segundo considera al problema como dinámico e implementa algún mecanismo que considera los cambios del grafo a través del tiempo [2, 4, 18-23].

El problema de agrupamiento trata de encontrar la mejor partición C de un grafo $G(V, A)$ con nodos V y aristas A. Algunos algoritmos hacen suposiciones *a priori* de la estructura del grafo y esto puede afectar el rendimiento y la calidad de las soluciones. Por otro lado, agrega un mejor rendimiento [10]. En este trabajo se adopta el enfoque evolutivo de AI-NSGA-II [24] para formar las soluciones. HD-AI-NSGA-II, que es el algoritmo propuesto, puede trabajar tanto de manera estática como temporal, ya que no está enfocado a la calidad del agrupamiento conforme el paso del tiempo. En su lugar, crea soluciones de alta definición para cada estado en el tiempo. HD-AI-NSGA-II considera un objetivo adicional para aumentar la calidad y rendimiento a comparación del enfoque adoptado. Este objetivo trata de maximizar el número de grupos y así poder diferenciar dos soluciones, que en teoría podrían ser medidas como iguales por los otros dos objetivos, logrando una mayor diversificación del frente de Pareto.

El resto del trabajo está estructurado de la siguiente forma. En la Sección 2 se hace una comparativa con algunos trabajos relacionados. En la Sección 3 se plantea el problema y la nuestra propuesta para mejorar el agrupamiento, posteriormente, en la Sección 4 se presentan los resultados de comparar el enfoque adoptado contra nuestra propuesta. Finalmente, en la Sección 5 se presentan algunas conclusiones y se describe el trabajo futuro.

2. Trabajo relacionado

Uno de los trabajos relacionados al enfoque adoptado es el de GrasSC [25], el cual hace agrupación de usuarios para realizar recomendación de contenido. Para lograrlo utiliza el historial de páginas visitadas por cada usuario como nodo y la similitud de un historial a otro como arista. En este sentido, generan un grafo no dirigido y ponderado, dando como resultado un problema de partición de grafo, el cual es NP-difícil. Utilizan las mismas funciones objetivo que AI-NSGA-II, las cuales son $1/(1 + Mcut)$ y $Gobal\ Silhouette$, pero a diferencia AI-NSGA-II, GrasSC utiliza una codificación basada en etiquetas, además de estar basado en SPEA2 [26] y opera sobre tipos de datos categóricos y matrices de distancia.

Otro trabajo que está muy relacionado con AI-NSGA-II es DYN-MOGA [27], ya que ambos utilizan representación basada en grafo de adyacencia, adoptan el enfoque de NSGA-II [28] y operan sobre tipos de datos que representan grafos. Sin embargo, tiene una diferencia significativa y es que una de sus funciones objetivo prioriza suavizar la diferencia de los agrupamientos de dos momentos consecutivos en lugar de maximizar la precisión del agrupamiento. DYN-MOGA utiliza $Community\ Score$ y $Normalized\ Mutual\ Information$, como funciones objetivo. El primer objetivo de DYN-MOGA trata de maximizar la calidad del agrupamiento en el estado actual de tiempo, mientras que el segundo objetivo considera el cambio que sufre una solución que pasa del tiempo t al tiempo $t + 1$. Estos dos objetivos entran en conflicto cuando el estado del grafo es dinámico y no se mantiene idéntico para cada estado de tiempo.

3. Detección de comunidades de alta definición

El problema de detección de comunidades en redes sociales lo abordamos desde una estrategia multiobjetivo por la naturaleza de las redes sociales, ya que es difícil encontrar una red social representada por grafos disjuntos, lo común es tener una red social representada por un grafo conexo que contenga la mayoría de las relaciones entre todos los nodos. En el caso de que pudiéramos garantizar que el grafo es disjunto, podría aplicarse un método de agrupamiento de un solo objetivo. Pero en el caso de un grafo conexo, es deseable tener más de una métrica para medir la calidad del agrupamiento. Y como se demuestra en esta sección, a veces es necesario agregar nuevos objetivos para elevar la calidad de los resultados.

Usualmente, el problema de detección de comunidades trata de encontrar, en un grafo $G(A, V)$, una partición C compuesta por k conjuntos disjuntos de nodos v tal que cada $v \in V$, que minimice el número de aristas a entre grupos y maximice el número

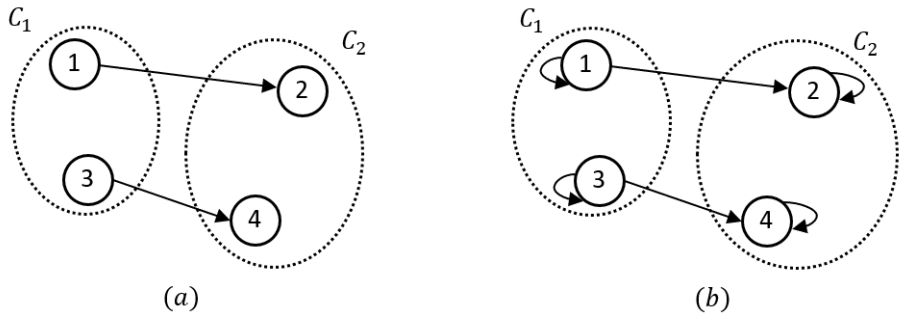


Fig. 1. En (a) se muestra el caso especial de un agrupamiento donde $W(C_1) = 0$ y $W(C_2) = 0$, ya que para C_1 y C_2 no hay relaciones internas. Para evitar la división entre cero en los términos de $Mcut_k$ se considera implícitamente el bucle $W_{uu} = 1$, para todos los nodos de G como se muestra en (b), de esta forma se tiene $W(C_1) = 2$ y $W(C_2) = 2$.

de aristas a dentro de cada grupo para cada $a \in A$. La ecuación (1) muestra el primer objetivo a maximizar, que es una modificación a la ecuación (2) $Mcut$ [29], ya que $Mcut$ por sí sola debe ser minimizada:

$$f_1 = \frac{1}{1+Mcut}, \tag{1}$$

en donde:

$$Mcut_k = \frac{cut(C_1, \bar{C_1})}{W(C_1)} + \frac{cut(C_2, \bar{C_2})}{W(C_2)} + \cdots + \frac{cut(C_k, \bar{C_k})}{W(C_k)}, \tag{2}$$

$$W(C_\mathrm{i}) = cut(C_\mathrm{i}, C_\mathrm{i}), \tag{3}$$

$$cut(C_i, \bar{C_i}) = \sum_{u \in C_1, v \in \bar{C_1}} W_{uv}, \tag{4}$$

para toda $i \in \{1, \ldots, k\}$ y $k > 1$. f_1 toma valores en el intervalo $(0,1]$, ya que $Mcut_k$ toma valores en el intervalo $[0, \infty)$. El mejor caso de $Mcut_k$ ocurre cuando la ecuación (4) de cada termino es igual a cero, es decir, $cut(C_i, \bar{C_i}) = 0$, para toda $i \in \{1, \ldots, k\}$. Esto quiere decir que no existe ninguna arista a que vaya desde algún nodo a otro nodo de otro grupo, lo cual puede verse como que ninguna arista es cortada por la agrupación. El caso neutral está dado cuando el número de aristas entre grupos es igual al número de aristas internas de cada grupo, con lo cual cada término $\frac{cut(C_i, \bar{C_i})}{W(C_\mathrm{i})} = 1$, teniendo como resultado $Mcut_k = m$, en donde m es el número de nodos de G. Finalmente, se tendría que $f_1 = \frac{1}{1+m}$. Esta observación es importante ya que f_1 toma valores en el intervalo $(0,1]$ y sería fácil pensar que el caso neutral estaría dado por $f_1 = 0.5$.

La última consideración con respecto a esta función se muestra en la Fig. 1 y hace evidente un caso especial en la ecuación (3) y por el cual se considera $W_{uu} = 1$, para toda $u \in \{1, \ldots, m\}$.

El segundo objetivo por maximizar es *Global Silhouette* [30], está definido en la ecuación (5):

$$f_2 = Global\ Silhouette\ = \frac{\sum_{i \in V} S(i)}{|V|}, \tag{5}$$

en donde:

$$S(i) = \begin{cases} \frac{a(i)-b(i)}{\max\{a(i),b(i)\}}, para\ |C_l| > 1 \\ 0, para\ |C_l| = 1 \end{cases}, \forall\ i\ \in \{1, \dots, |V|\}\ , \tag{6}$$

$$a(i) = \frac{\sum_{j \in C_l} W_{ij}}{|C_l|}, i\ \in C_l\ , \tag{7}$$

$$b(i) = \max\{d(i, C_m)\}, m\ \in \{1, \dots, |C|\}, m \neq l\ , \tag{8}$$

$$d(i, C_m) = \frac{\sum_{j \in C_m} W_{ij}}{|C_m|}\ , \tag{9}$$

f_2 toma valores en el intervalo $(-1,1]$, ya que cada término de la ecuación (5) también toma valores en el intervalo $(-1,1]$ y f_2 es el promedio los mismos. En la ecuación (7) se calcula el promedio de aristas dentro del grupo al que pertenece el nodo i. La ecuación (8) selecciona el promedio máximo de aristas que existen entre el nodo i y el resto de los grupos, cada promedio es calculado con la ecuación (9). El mejor caso de $Global\ Silhouette$ ocurre cuando la ecuación (6) toma el valor de uno y la ecuación (8) toma el valor de cero, es decir, $S(i) = 1$, dado $b(i) = 0$, para toda $i\ \in \{1, \dots, |V|\}$. Esto quiere decir que no existe ninguna arista a que vaya desde algún nodo a otro nodo de otro grupo. El caso neutral está dado cuando el número de aristas entre grupos es igual al número de aristas internas de cada grupo, con lo cual para cada término $S(i)$ de la ecuación (5), $a(i) = b(i)$, teniendo como resultado $Global\ Silhouette = 0$. Para el caso que el grupo C_l solo contará con un nodo se le asigna a $S(i)$ el valor cero.

La última consideración con respecto a esta función se muestra en la Fig. 2 y hace evidente un caso especial por el cual, al igual que el primer objetivo, se considera $W_{uu} = 1$, para toda $u \in \{1, \dots, m\}$.

3.1 HD-AI-NSGA-II

Los dos primeros objetivos deben maximizarse para encontrar el conjunto de soluciones que formen el frente de Pareto, pero por sí solos tienen una limitante ya que existen soluciones que representan diferentes agrupamientos con un número diferente de grupos, pero con mismos valores de aptitud, como se muestra en la Fig. 3.

Como se puede apreciar en la Fig. 3, la solución (b) es mejor que la (a) al estar mejor definida, por lo tanto, proponemos como tercer objetivo a maximizar la ecuación (10), que es el número de grupos k:

$$f_3 = k. \tag{10}$$

3.2 Mecanismos genéticos

Para poder hacer una comparación estricta entre AI-NSGA-II y HD-AI-NSGA-II, se utilizan los mismos mecanismos genéticos propuestos por el enfoque seguido. Para la codificación utilizamos *locus-based adjacency* [31] como forma de representar cada solución. En esta representación el genotipo es del tamaño del número de nodos que tiene el grafo y cada locus del genotipo indica una arista del nodo representado por el

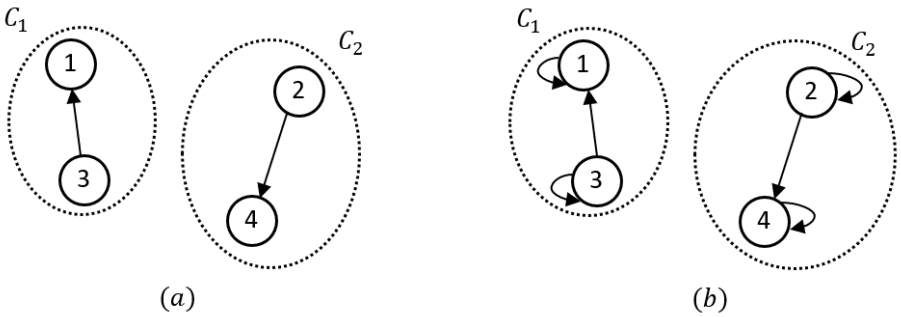


Fig. 2. En (a) se muestra el caso especial de un agrupamiento donde tenemos $a(1) = 0$ y $b(1) = 0$, es decir, que el nodo 1 no tiene relación con ningún otro, ya sea de su mismo grupo o de otro grupo. Para evitar la división entre cero en $S(i)$, se considera implícitamente el bucle $W_{uu} = 1$, para todos los nodos de G como se muestra en (b), en donde tenemos $a(1) = 0.5$ y $b(1) = 0$.

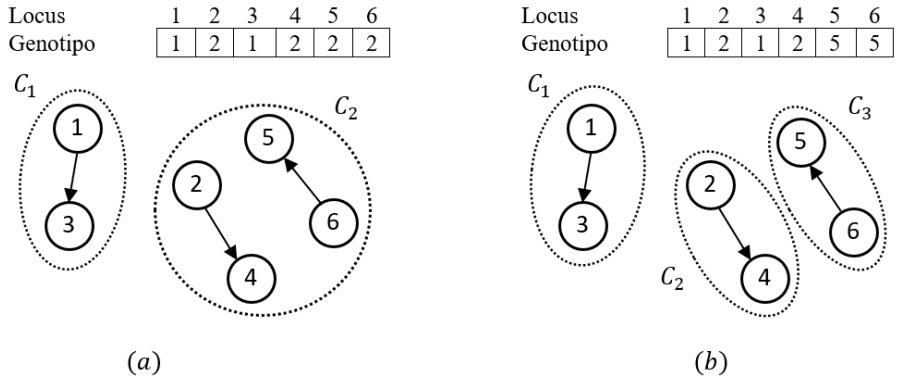


Fig. 3. En (a) se muestra la solución de un agrupamiento con valores de aptitud $f_1 = 1$, $f_2 = 1$, la cual forma dos grupos. En (b) se muestra la solución de un agrupamiento que también tiene los valores de aptitud $f_1 = 1$, $f_2 = 1$, pero formada por tres grupos.

locus al nodo representado por el valor del locus. La selección se hace por torneo binario [32], en donde se escogen al azar dos individuos de la población, su nivel en el frente de Pareto y la probabilidad para seleccionar uno de los dos. Se utiliza cruza uniforme por no ser sesgada [24] y poder generar cualquier solución posible.

4. Resultados

Para medir el desempeño de nuestra propuesta, usamos dos conjuntos de datos artificiales y el conjunto bien conocido de datos reales de Cheng et al. [33] obtenidos de YouTube. Los parámetros de ejecución (ver Tabla 1) son los mismos reportados por el enfoque seguido [24].

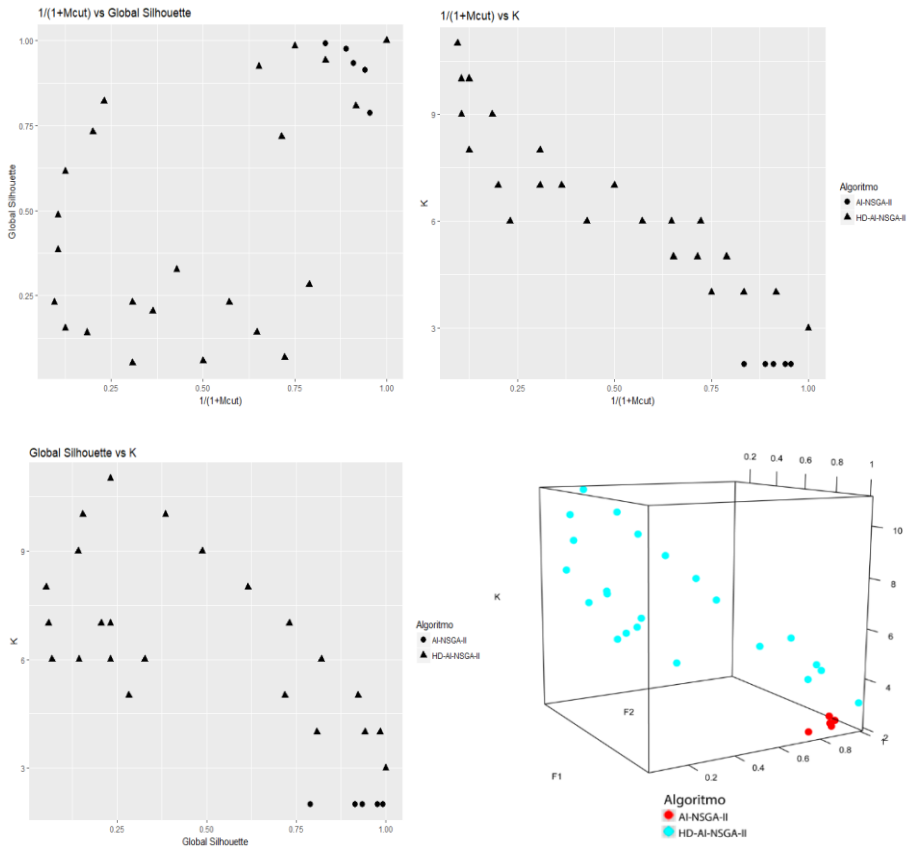


Fig. 4. Comparación, por pares de funciones objetivo y de los tres objetivos, de las soluciones de AI-NSGA-II y HD-AI-NSGA-II, para el primer conjunto de datos.

Tabla 1. Parámetros de ejecución.

	Número máximo de generaciones	**Población P**	**Población Q**	**Selección Ps**	**Cruza Pc**	**Mutación Pm**
AI-NSGA-II	200	30	60	0.6	1.0	0.01
HD-AI-NSGA-II	200	30	60	0.6	1.0	0.01

El primer conjunto de datos artificial consta de tres grafos disjuntos, por lo tanto, las soluciones óptimas son las que tengan $\frac{1}{1+Mcut} = 1$, $GS = 1$ y $k = 3$, es decir, las soluciones optimas deben representar tres grupos sin aristas entre grupos. En la Fig. 4 se puede ver cómo únicamente HD-AI-NSGA-II encuentra las soluciones óptimas para

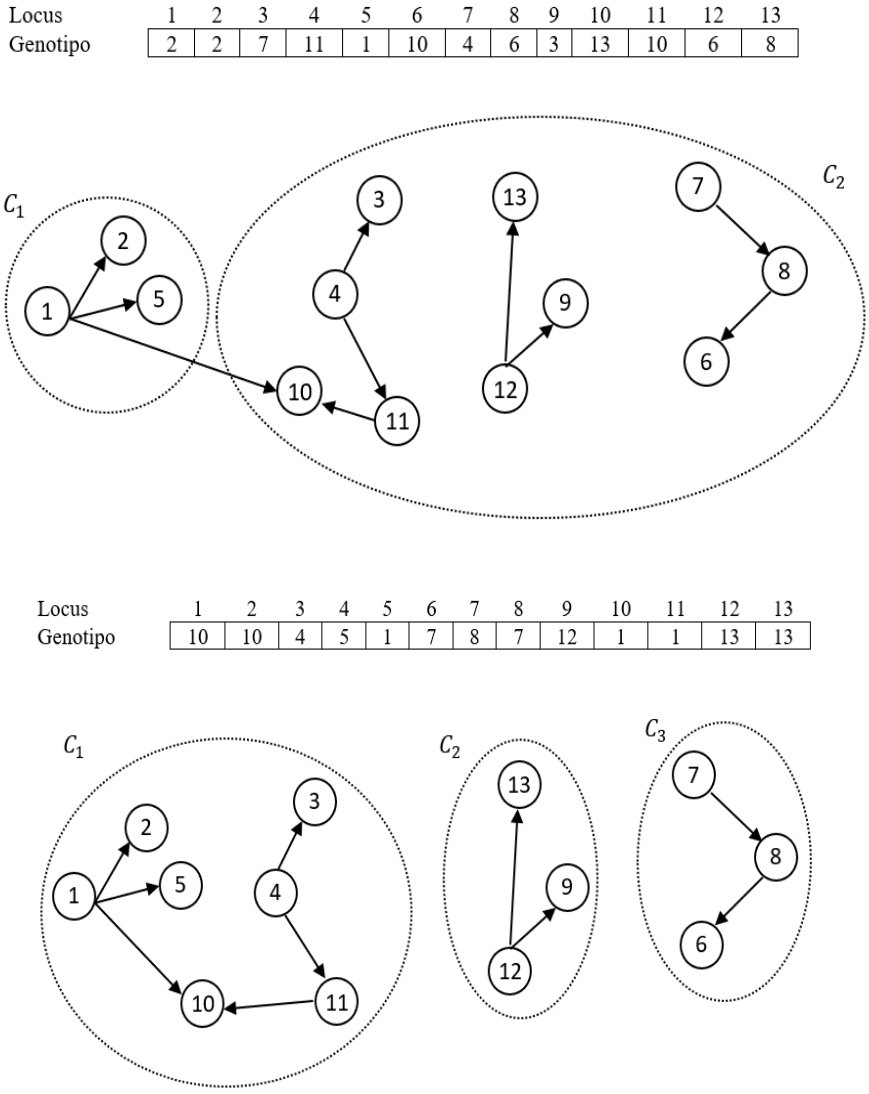


Fig. 5. Superior: la mejor solución de AI-NSGA-II. Inferior: la mejor solución de HD-AI-NSGA- II.

este conjunto de datos, las cuales son precisamente las soluciones $\left(\frac{1}{1+Mcut}, GS, k\right) = (1,1,3)$.

En la Fig. 4 se puede observar cómo las soluciones de AI-NSGA-II se acercan a valores de $\frac{1}{1+Mcut} = 1$, $GS = 1$, las cuales indudablemente alcanzaría aumentando el número de iteraciones, pero se estanca en valores de $k = 2$, motivo por el cual no deja que este enfoque obtenga agrupaciones óptimas de $k = 3$.

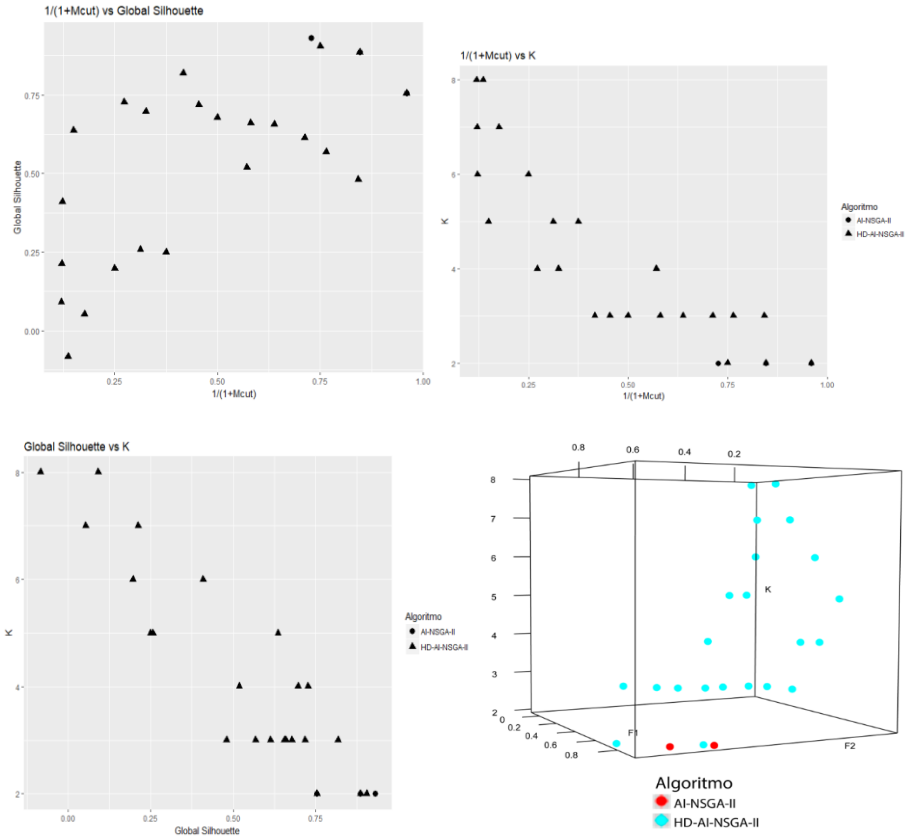


Fig. 6. Comparación, por pares de las funciones objetivo y de los tres objetivos, de las soluciones de AI-NSGA-II y HD-AI-NSGA-II, para el segundo conjunto de datos.

En la Fig. 5 se muestra la representación y comparación de la mejor solución del enfoque adoptado AI-NSGA-II y la mejor solución de nuestra propuesta HD-AI-NSGA-II, respectivamente. La mejor solución de AI-NSGA-II únicamente corta una arista, pero al no ser guiada la búsqueda en el sentido del numero de grupos k le es difícil definir los tres grupos que existen. En contraste, nuestra propuesta si logra identificar perfectamente los tres grupos.

El segundo conjunto de datos artificial consta de un grafo conexo que contiene tres comunidades, por lo tanto, las mejores soluciones son las que tengan $\frac{1}{1+Mcut} \approx 1$, $GS \approx 1$ y $k = 3$. En la Fig. 6 se puede ver que también únicamente HD-AI-NSGA-II encuentra las soluciones óptimas para este tipo de grafo.

En la Fig. 6 se puede observar cómo las soluciones de AI-NSGA-II se acercan a valores de $\frac{1}{1+Mcut} = 1$, $GS = 1$, pero se estanca en valores de $k = 2$ motivo por el cual no deja que este enfoque obtenga agrupaciones óptimas con $k = 3$.

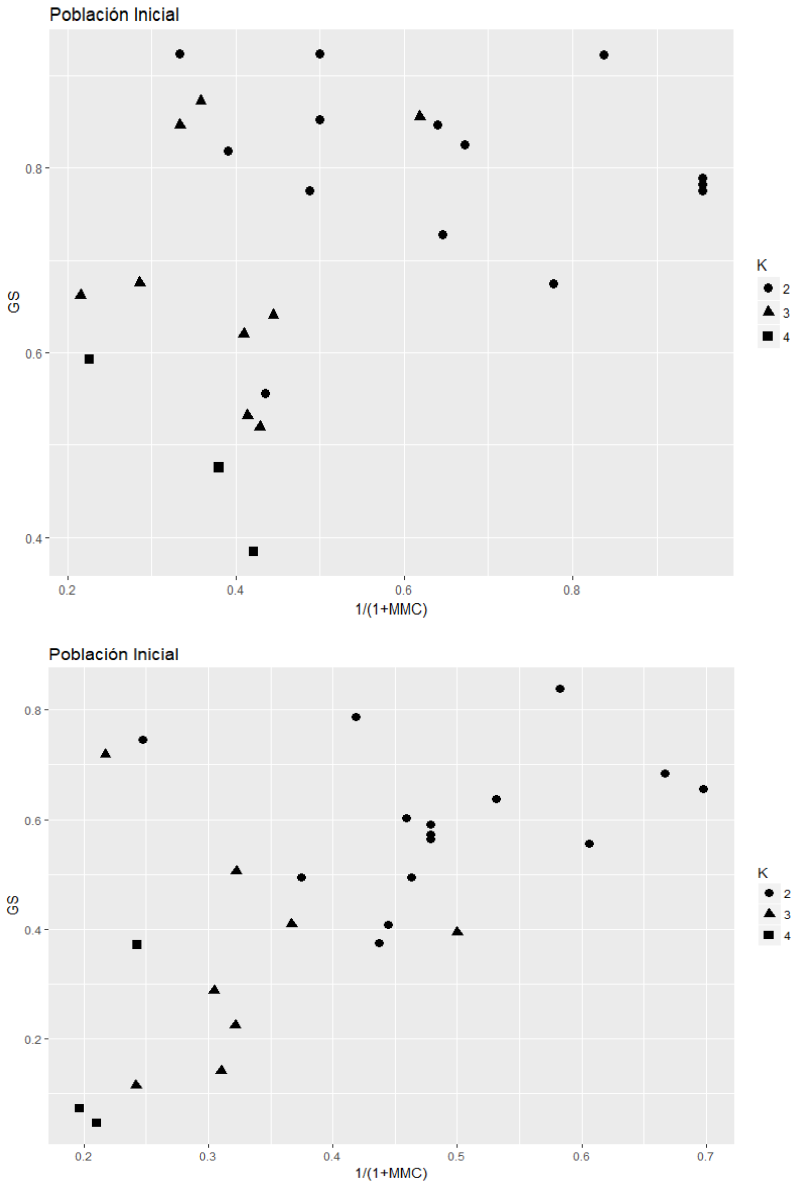


Fig. 7. Número de conjuntos en las poblaciones iniciales para el primer (superior) y segundo (inferior) conjunto artificial.

El motivo por el que sólo HD-AI-NSGA-II encuentre soluciones compuestas de tres grupos, se debe a que las soluciones iniciales (ver Fig. 7) con mejores valores de aptitud f_1 y f_2 generalmente son las de menor valor f_3 y, por lo tanto, AI-NSGA-II va remplazando las soluciones con altos valores de f_3 por soluciones con valores menores para dicha función.

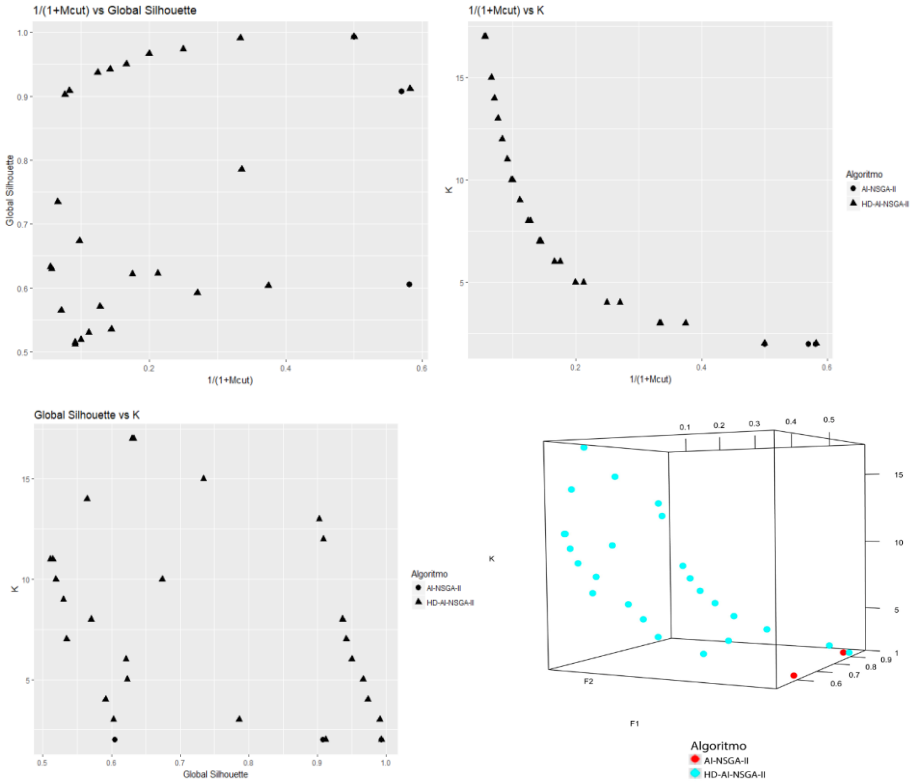


Fig. 8. Comparación, por pares de las funciones objetivo y de los tres objetivos, de las soluciones de AI-NSGA-II y HD-AI-NSGA-II para el conjunto de datos de YouTube que contiene tres grafos disjuntos.

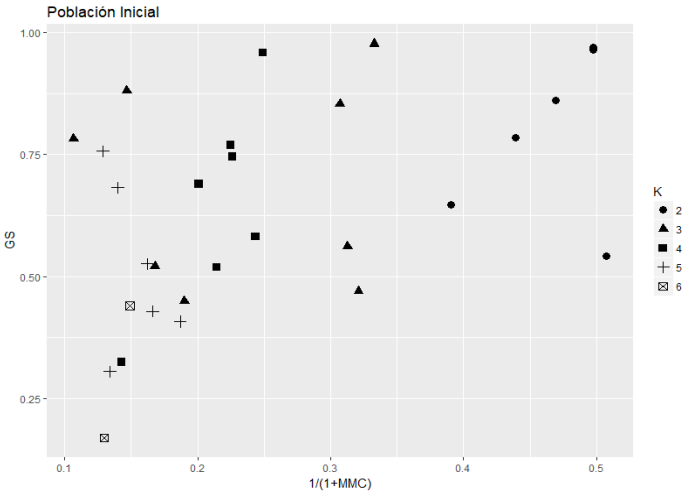


Fig. 9. Número de conjuntos en las poblaciones iniciales para el conjunto de datos de YouTube.

Por último, en la Fig. 8 se muestran las soluciones encontradas para el conjunto de datos que contienen los ID de videos de YouTube, en el cual se puede apreciar que HD-AI-NSGA-II es el único en encontrar soluciones con un mayor número de grupos.

El motivo por el que solo HD-AI-NSGA-II encuentre soluciones compuestas de tres grupos, se debe a que las soluciones iniciales (ver Fig. 9) con mejores valores de aptitud f_1 y f_2 generalmente son las de menor valor f_3 y, por lo tanto, AI-NSGA-II va remplazando las soluciones con altos valores de f_3 por soluciones con valores menores para dicha función.

5. Conclusiones

En el presenta trabajo presentamos una alternativa al enfoque propuesto por AI-NSGA-II para la detección de comunidades. Nuestra propuesta agrega una función objetivo que permite guiar a soluciones que definen mejor los grupos hacia el frente de Pareto. Este objetivo maximiza el número de grupos formados por cada solución, dando como resultado una mayor exploración y evitar estancarse en óptimos locales que contengan un bajo número de grupos. Ambos enfoques fueron probados y comparados con dos conjuntos de datos artificiales y uno bien conocido, llegando a la conclusión de que, tanto para los grafos disjuntos como para los conexos que contienen comunidades internamente, trabaja mejor nuestra propuesta logrando definir mejor los grupos y haciendo una mejor detección de comunidades.

Actualmente se está trabajando con un conjunto de datos compuesto por ID de perfiles de Facebook creado por la empresa tecnológica Cad&Lan, los cuales forman un grafo conexo al estar construido de forma recursiva. Se pretende demostrar de igual manera la efectividad del enfoque propuesto sobre otro conjunto de datos reales. Posteriormente, se pretende adaptar los demás objetivos de optimización e implementar métricas multiobjetivo para conocer el desempeño.

Agradecimientos. El presente trabajo está patrocinado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) a través la beca otorgada en el marco del Programa Nacional de Posgrado de Calidad. También agradecemos el apoyo prestado por la empresa tecnológica Cad&Lan que a través de su departamento de investigación y desarrollo nos facilitó su base de datos de Facebook IDs.

Referencias

1. Menendez, D.H., Llorente, J.L.: The Combination of Graph Theory and Unsupervised Learning Applied to Social Data Mining. In: Cavalcante, A.(eds.) Graph Theory: New Research, Nova Science Publishers Inc (2013)
2. Gong, M., Cai, Q., Ma, L., Wang, S., Lei, Y.: Introduction. In: Computational Intelligence for Network Structure Analytics. Springer (2017)
3. Leu, G., Abbass, H.: A multi-disciplinary review of knowledge acquisition methods: From human to autonomous eliciting agents. Knowledge-Based Systems, 105, pp: 1–22 (2016)

4. Rana, C., Jain, S.K.: An extended evolutionary clustering algorithm for an adaptive recommender system. Social Network Analysis and Mining, 4(1) pp. 1–13 (2014)
5. Mukhopadhyay, A., Maulik, U.: Survey of Multi-Objective Evolutionary Algorithms for Data Mining: Part-II. IEEE Transactions on Evolutionary Computation, 18(1), pp. 20–35 (2014)
6. Mukhopadhyay, A., Maulik, U.: A Survey of Multi-Objective Evolutionary Algorithms for Data Mining: Part-I. IEEE Transactions on Evolutionary Computation, 18(1), pp. 4–19 (2014)
7. Menendez, H.D., Barrero, D.F., Camacho, D.: A genetic graph-based approach for partitional clustering. International journal of neural systems, 24(03) (2014)
8. Fan, L., Wu, W., Lu, Z., Xu, W., Du, D.Z.: Influence diffusion, community detection, and link prediction in social network analysis. Springer Proceedings in Mathematics and Statistics, 51, pp. 305–325 (2013)
9. Cai, Q., Ma, L., Gong, M., Tian, D.: A survey on network community detection based on evolutionary computation. International Journal of Bio-Inspired Computation 8(2), pp. 84–98 (2016)
10. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: A Survey of Multiobjective Evolutionary Clustering. ACM Computing Surveys 47(4), pp. 1–46 (2015)
11. Delgado C.C.: Utilización de técnicas de clustering para mejorar la detección de metatopics en conjuntos de datos extraídos de Twitter. MS thesis (2015)
12. Gong, M, et al.: Big Network Analytics Based on Nonconvex Optimization. In: Emrouznejad A. (ed.), Big Data Optimization: Recent Developments and Challenges. Studies in Big Data, 18, pp. 345–373 (2016)
13. Bucur, D., Iacca, G., Marcelli, A., Squillero, G., Tonda, A.: Multi-objective Evolutionary Algorithms for Influence Maximization in Social Networks. In: Squillero G., Sim K. (eds.), Applications of Evolutionary Computation. Evo Applications, Lecture Notes in Computer Science, 10199, Springer (2017)
14. Kirkland, O.: Multi-objective evolutionary algorithms for data clustering. Diss., University of East Anglia (2014)
15. Delgado C.C.: Modelo de identificación de Meta-Topics a través de análisis semántico de conjuntos de datos extraídos de twitter. BS thesis (2014)
16. Bello-Orgaz, G., Menéndez, H.D., Camacho D.: Adaptive K-Means Algorithm for Overlapped Graph Clustering. International Journal of Neural Systems, 22(5), 1250018 (2012)
17. Menéndez, H.D.: A genetic approach to the graph and spectral clustering problem. MS thesis (2012)
18. Li, Q., Zou, J., Yang, S. et al.: A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. Soft Computing (2018)
19. Zou, J., Li, Q., Yang, S., Bai, H., Zheng, J.: A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization. Applied Soft Computing Journal, 61, pp. 806–818 (2017)
20. Rong, M., Gong, D., Zhang Y.: A Multi-direction Prediction Approach for Dynamic Multi-objective Optimization. In: Huang DS., Han K., Hussain A. (eds) Intelligent Computing Methodologies, (ICIC) Lecture Notes in Computer Science, 9773, Springer, Cham (2016)
21. Zadeh, P. M., Kobti, Z.: A multi-population cultural algorithm for community detection in social networks. Procedia Computer Science 52(1), pp. 342–349 (2015)
22. Hartmann T., Kappes A., Wagner, D.: Clustering Evolving Networks. In: Kliemann L., Sanders P. (eds), Algorithm Engineering, Lecture Notes in Computer Science, 9220 (LNCS), pp. 280–329 (2016)

23. Held, P., Dannies, K.: Clustering on dynamic social network data. Synergies of Soft Computing and Statistics for Intelligent Data Analysis. 190, pp. 563–571 (2013)
24. Kim, K., McKay, R.I., Moon, B.R.: Multiobjective evolutionary algorithms for dynamic social network clustering. In: Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, (GECCO), pp. 1179–1186 (2010)
25. Demir, G., Uyar, A., Oguducu, S.: Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO), pp.1943–1950 (2007)
26. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pp. 95–100 (2001)
27. Folino, F., Pizzuti, C.: A Multiobjective and Evolutionary Clustering Method for Dynamic Networks. In: Advances in Social Networks Analysis and Mining (ASONAM), International Conference, pp. 256–263 (2010)
28. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), pp. 182–197 (2002)
29. Ding, C.H., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: Data Mining (ICDM), Proceedings IEEE International Conference, pp. 107–114 (2001)
30. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20(C), pp. 53–65 (1987)
31. Park, Y., Song, M.: A genetic algorithm for clustering problems. In: Proceedings of the third annual conference on genetic programming, pp. 568–575 (1998)
32. Holland, J.H., Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley (1989)
33. Cheng, X., Dale, C., Liu, J.: Statistics and social network of youtube videos. In: Quality of Service, IWQoS 16th International Workshop, pp. 229–238 (2008)

\

ISSN 1870-4069

Sistema híbrido basado en redes neuronales artificiales y descomposición modal empírica para la evaluación de la interrelación entre la irradiancia solar total y el calentamiento global

Eric Alberto Suárez-Gallareta[1], Jorge Javier Hernández-Gómez[2], Gerardo Cetzal-Balam[1], Mauricio Gabriel Orozco-del-Castillo[1], Mario Renan Moreno-Sabido[1], Raúl Alberto Silva-Aguilera[3]

[1] Instituto Tecnológico de Mérida,
Departamento de Sistemas y Computación,
México

[2] Instituto Politécnico Nacional,
Centro de Desarrollo Aeroespacial,
México

[3] Universidad Nacional Autónoma de México,
Instituto de Ciencias del Mar y Limnología,
México

{ericsuarezgallareta1996,gerardoce23,xacdc12}@gmail.com,
mauricio.orozco@itmerida.edu.mx,
jjhernandezgo@ipn.mx, raul.s@ciencias.unam.mx

Resumen. El calentamiento global o cambio climático es probablemente el mayor reto científico actual de la humanidad. De todos los factores tanto naturales como antropogénicos involucrados en el calentamiento global, así como sus complejas interrelaciones, poca atención se ha centrado en los factores externos al sistema Tierra, como lo es la variabilidad solar. En este trabajo se presenta un sistema híbrido de inteligencia artificial basado en redes neuronales artificiales y descomposición modal empírica para determinar la interrelación entre la irradiancia solar total recibida en la Tierra en las últimas cuatro décadas con un índice clave en el cambio climático: la temperatura superficial del mar. Los resultados hasta el momento muestran una evidente interrelación entre ambos índices, sugiriendo que el principal motor de la variabilidad en la temperatura superficial del mar son las variaciones en la entrada de energía solar al sistema Tierra.

Palabras clave: Redes neuronales artificiales, descomposición modal empírica, cambio climático, calentamiento global, irradiancia solar, inteligencia artificial, reconocimiento de patrones, temperatura superficial del mar, IA, RNA, DME.

Eric Alberto Suárez-Gallareta, Jorge Javier Hernández-Gómez, Gerardo Cetzal-Balam, et al.

Hybrid System based on Artificial Neural Networks and Empiric Modal Decomposition for the Assessment of the Interrelation between Total Solar Irrandiance and Global Warming

Abstract. Global warming or climate change is probably the current greatest scientific challenge that mankind faces. From all both natural and anthropogenic factors involved in global warming, as well as their complex interrelations, few attention has focused on those factors external to Earth, as the solar variability. In this work we present a hybrid system in artificial intelligence based on artificial neural networks and empirical mode decomposition to determine the interrelation between total solar irradiance received by Earth in the last four decades with a key index in climate change: oceanic surface temperature. The results of this ongoing research show an evident interrelation between both indexes, and strongly suggest that the main external factor driving of the variations in the oceanic surface temperature are the variations in the input solar energy of Earth.

Keywords: Artificial neural networks, empirical mode decomposition, climate change, global warming, solar irradiance, artificial intelligence, pattern recognition, oceanic surface temperature, AI, ANN, EMD.

1. Introducción

Actualmente, es probable que no haya mayor reto científico en el mundo que el que representa el calentamiento global. Desde la mitad del siglo XIX se empezaron a notar anomalías en los registros de temperatura alrededor del mundo, que empezaron a marcar una tendencia clara de aumento a partir de la década de los 1950s. Estos registros junto con registros paleoclimáticos de los últimos miles de años constituyen una prueba fehaciente del calentamiento global [25]. A raíz de la preocupación de las tendencias del calentamiento global en los 1980s, países miembros de la Organización Meteorológica Mundial (OMM) y el Programa de las Naciones Unidas para el Medio Ambiente (PNUMA) establecieron en 1988 el Panel Intergubernamental del Cambio Climático, conocido por el acrónimo en inglés IPCC (Intergovernmental Panel on Climate Change) [24], mismo que fue posteriormente ratificado por la Asamblea General de la Organización de las Naciones Unidas (ONU) mediante la Resolución 43/53. El IPCC publica informes especiales de evaluación del cambio climático basados en información científica, técnica y socioeconómica actual sobre el riesgo que representa, sus potenciales consecuencias medioambientales y socioeconómicas, así como las posibles opciones de adaptabilidad o mitigación de sus efectos.

El calentamiento global se define como el aumento de la temperatura global media del aire sobre la Tierra y el océano [25], y en el último informe del IPCC se destaca que si bien en 2013 el aumento en la temperatura global media fue del orden de 0.4 C (respecto a la media de temperatura de 1961 a 1980), se advierte

que para el año 2100 la tendencia podría superar los 2.5 C con consecuencias catastróficas tanto para la humanidad como para el planeta y la vida en él (Fig. 1).

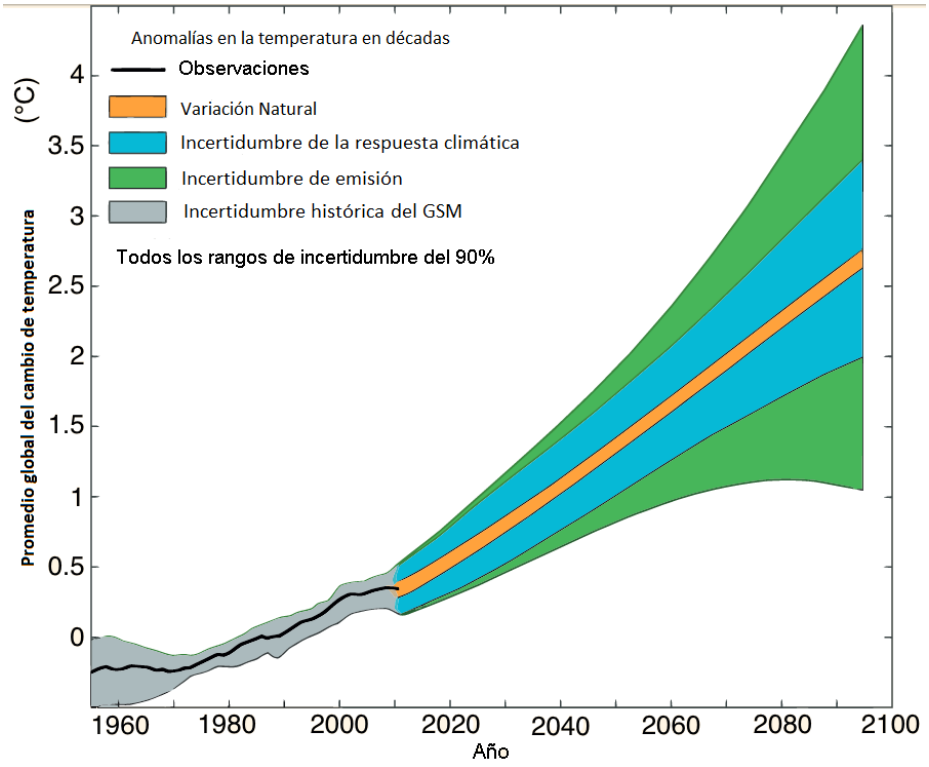


Fig. 1. Diagrama esquemático del cambio en la temperatura media superficial (C) por décadas de registros históricos (línea negra) con sus incertidumbres históricas (gris), junto con proyecciones futuras del clima y sus incertidumbres [25]. Los valores están normalizados con valores medios de 1961 a 1980. La variabilidad natural (naranja) se deriva del modelo de variabilidad interanual y se supone constante en el tiempo. La incertidumbre en las emisiones (verde) se estima como la diferencia media del modelo en proyecciones para diversos escenarios. La incertidumbre en la respuesta del clima (azul, sólida) se basa en la dispersión del modelo climático, con incertidumbres agregadas del ciclo del carbón, así como con estimados gruesos de incertidumbres adicionales de algunos procesos pobremente modelados [19, 23].

Aunque son incontables los procesos fisicoquímicos involucrados en el calentamiento global, así como complejas y altamente no lineales sus interrelaciones, es indudable que el aumento de la temperatura global media se debe a que el balance global de energía del sistema Tierra ha sufrido modificaciones en las

últimas décadas, de modo que el contenido de calor del planeta ha aumentado. Esto podría deberse a una mayor entrada de energía al planeta, o bien, a una disminución en su disipación energética. Sin embargo, la mayoría de la investigación respecto al tema del cambio climático se centra, o bien en el estudio de dichos procesos fisicoquímicos *per se*, en sus causas y consecuencias antropogénicas, o bien en la realización de proyecciones o predicciones del comportamiento de la temperatura global media en diversos escenarios. No obstante, y a pesar de la importancia de tener predicciones confiables, la gran mayoría de la investigación al respecto se basa en construir modelos, ya sean fenomenológicos o de primeros principios, acerca del clima regional o global.

Estos modelos se suelen dividir en Modelos de Circulación General Atmosférico-Oceánicos, Modelos del Sistema Tierra [12], Modelos del Sistema Tierra de Complejidad Intermedia [8,13,16,39,43,45,48,51] y Modelos del Clima Regionales [31, 44]. Para una revisión exhaustiva de los modelos climáticos, ver [25].

Claramente, el calentamiento global es un tema delicado que podría beneficiarse ampliamente del poder predictivo de las técnicas de la inteligencia artificial para el reconocimiento de patrones. Todos los procesos internos en el sistema Tierra, tanto naturales como antropogénicos que contribuyen al cambio climático, afectan principalmente a la salida de energía del planeta y además poseen complejas interrelaciones no lineales. De esta manera, vale la pena observar los procesos que afectan la entrada energética al planeta. Dentro de ellos puede estar la variabilidad solar, que puede tomar escalas centenarias o inclusive milenarias [21], variaciones en los alineamientos astronómicos entre el sol y la tierra (como los ciclos de Milankovitch), así como por ejemplo impactos de asteroides mayores [25]. De todos estos fenómenos, el mayor forzamiento térmico natural sobre la Tierra es indiscutiblemente la variabilidad solar, misma que ha probado ser multiperiódica [26]. Existen diferentes índices que pueden dar cuenta de la variabilidad solar como lo son mismos los ciclos solares [4, 38, 40], las manchas solares [17, 41], la presencia de *Ground Level Enhancements* en los rayos cósmicos solares [2,6,38,46], el viento solar y el índice geomagnético [14,34], el campo magnético solar [37], entre otros.

Sin embargo, de todos estos, probablemente el de mayor relevancia para el problema del calentamiento global sea la irradiancia solar total (IST) [25], que es la potencia solar recibida por unidad de área (energía solar recibida por unidad de tiempo y área), pues la IST es la mayor contribución de entrada de energía térmica al sistema Tierra. De esta forma, el tener predicciones confiables del comportamiento futuro de la IST es fundamental para proyectar los balances de energía térmica al futuro y así tener evaluaciones más precisas de las posibles tendencias de la temperatura global media en diversos escenarios a corto, mediano y largo plazo. Cabe destacar que todas las mediciones de la ITS coinciden en la existencia de periodicidad en este índice [29].

En este trabajo, actualmente en proceso, se propone la implementación de un método híbrido en inteligencia artificial constituido por una Red Neuronal Artificial (RNA) junto con el método de Descomposición Modal Empírica (DME) para el reconocimiento de patrones dentro de datos de IST obtenidos de Agencia

Nacional Oceánico-Atmosférica (NOAA por sus siglas en inglés) de los Estados Unidos de Norteamérica. Para este estudio, se utilizó la base de datos de valores compuestos de la IST en un periodo de 1978 al 30 de junio de 2017 [9] (Fig. 2). Este sistema híbrido se encarga de buscar relaciones entre la IST y otros indicadores fuertemente relacionados con el cambio climático, en este caso, la Temperatura Superficial del Mar (TSM).

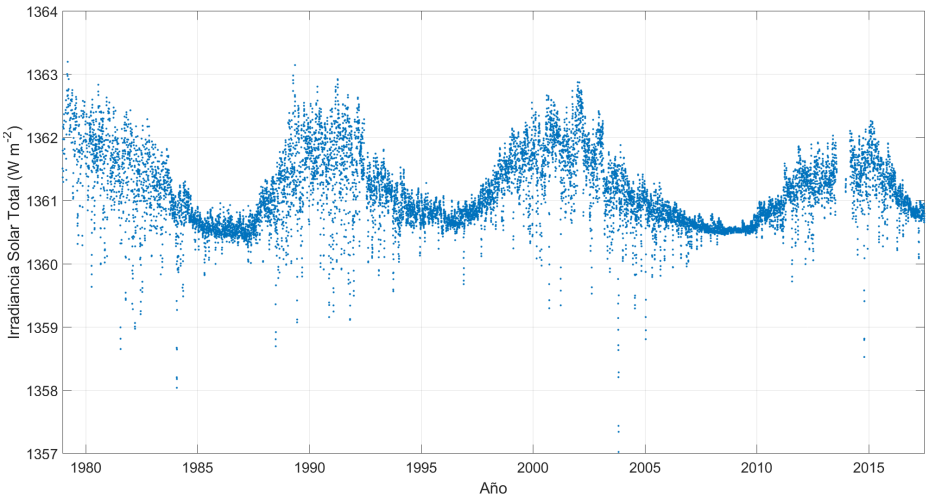


Fig. 2. Datos de irradiancia solar total obtenidos de la Agencia Nacional Oceánico-Atmosférica de los Estados Unidos de Norteamérica en un periodo de 1978 al 30 de junio de 2017 [9].

2. Metodología

A continuación, se presentan los métodos utilizados en este estudio. Debido a que la base de datos diaria de la IST [9] tiene valores faltantes, estos son interpolados con el método de kriging [30, 33].

2.1. Método de kriging

El método de kriging (krigeaje o krigeado) es un procedimiento geoestadístico de interpolación avanzado que genera una superficie estimada a partir de un conjunto de puntos dispersados. El algoritmo presupone que la distancia o la dirección entre los puntos de muestra refleja una correlación espacial que puede utilizarse para explicar la variación en la superficie. La herramienta ajusta una función matemática a una cantidad especificada de puntos o a todos los puntos

dentro de un radio específico para determinar el valor de salida para cada ubicación. El método consiste primeramente en el análisis estadístico exploratorio de los datos, el modelado de variogramas, la creación de la superficie y (opcionalmente) la exploración de la superficie de varianza. Este método es más adecuado cuando se sabe que hay una influencia direccional o de la distancia correlacionada espacialmente en los datos, como es el caso de los datos en la Fig. 2. El método kriging pondera los valores medidos circundantes para calcular una predicción de una ubicación sin mediciones mediante la ecuación (1):

$$\hat{Z}(s_0) = \sum_{i=1}^{N} \lambda_i Z(s_i) \ . \tag{1}$$

El método de kriging se ha usado extensivamente tanto en las ciencias naturales como directamente en las matemáticas aplicadas y ciencias de la computación [32,50]. Una vez interpolados los datos, se procedió a determinar las tendencias y componentes modales principales en la serie de datos (Fig. 2) mediante la DME.

2.2. Descomposición modal empírica

El algoritmo de DME, presentado por primera vez en 1998 [22], se basa en producir envolventes lisos definidos por máximos y mínimos locales de una secuencia y substracción subsiguiente de la media de estas envolventes a partir de la secuencia inicial. Esto requiere la identificación de todos los extremos locales que están conectados además por líneas spline cúbicas para producir los envolventes superior e inferior [28]. Es un método adaptivo de análisis adecuado para el procesamiento de series que son no estacionarias y no lineales. DME realiza operaciones que dividen una serie en modos, o Funciones Modales Intrínsecas (FMIs, o IMFs por las siglas en inglés de *Intrinsic Mode Functions*) sin salir del dominio del tiempo. Se puede comparar con otros métodos de análisis tiempo-espacio como la transformada de Fourier y la descomposición de ondas. La DME ha sido ampliamente aplicada en distintos campos de la ciencia con fines de reconocimiento [11], análisis [36], filtrado [1], predicción [10], etcétera.

2.3. Redes neuronales artificiales

Una RNA es una estructura que consiste en un número de nodos conectados a través de conexiones direccionales. Cada nodo representa una unidad de procesamiento, y las conexiones entre los nodos especifican la relación causal entre los nodos conectados. El objetivo principal de una RNA es el de imitar la sinapsis generada en las neuronas, las cuales son la unidad fundamental del sistema nervioso y se encuentran conformadas por un núcleo y un sistema de entradas y salidas denominadas dendritas y axones, respectivamente [5]. Una red neuronal se puede considerar como un procesador distribuido masivamente paralelo que tiene una propensión natural para almacenar conocimiento experimental y ponerlo a disposición para su uso [20]. Su principal similitud con el cerebro humano

se debe a que a través de un proceso de aprendizaje, la red es capaz de adquirir conocimiento mismo que almacena en los denominados pesos sinápticos. Algunas de las ventajas de las RNAs que las distinguen de los métodos computacionales convencionales son la manera directa en la que adquieren información acerca del problema mediante la etapa de entrenamiento, su capacidad para trabajar con datos tanto analógicos como discretos, su robustez, su tendencia a describir comportamientos no lineales, etc. [35]

3. Desarrollo y resultados

Utilizando los datos de IST mostrados en la Fig. 2, se aplicó DME para encontrar las distintas descomposiciones de los datos correspondientes a distintas aportaciones de frecuencia (Fig. 3). Los datos de IST originales se muestran indicados como puntos en color azul; las descomposiciones modales empíricas se muestran como líneas continuas. La línea negra indica la componente de menor frecuencia, $D_1(t)$, mientras que las siguientes dos líneas, roja y azul, indican las siguientes componentes de mayor frecuencia $D_2\,(t)$ y $D_3\,(t)$ respectivamente.

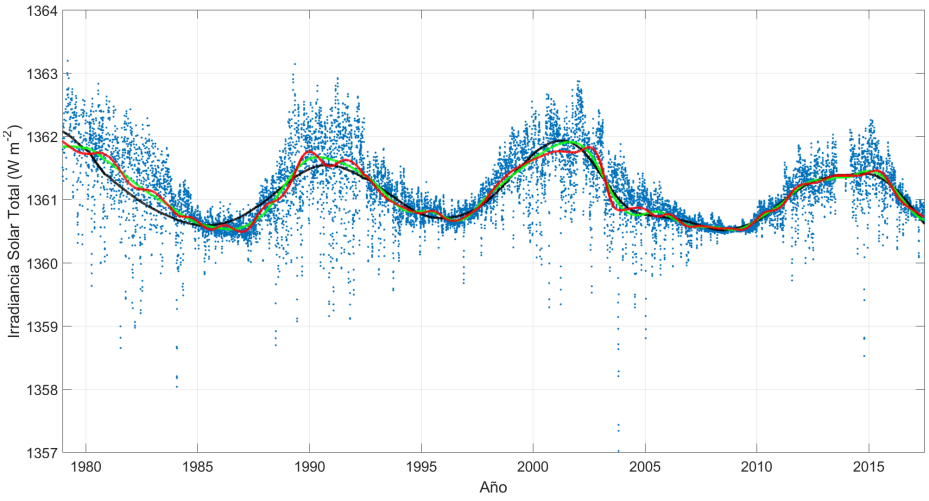


Fig. 3. Descomposición Modal Empírica de los datos de IST (Fig. 2). La línea negra indica la componente de menor frecuencia, $D_1\,(t)$, mientras que las siguientes dos líneas, roja y azul, indican las siguientes componentes de mayor frecuencia, $D_2\,(t)$ y $D_3\,(t)$ respectivamente

.

Una vez obtenidas las curvas representativas de la DME, partimos de la hipótesis de que el índice de TSM está relacionado con estas decomposiciones,

pero no de manera suficiente con los datos temporales, es decir, no está únicamente relacionado con su distribución en el tiempo, sino también con su comportamiento creciente/decreciente y su comportamiento cóncavo hacia arriba/abajo. En este sentido es entonces necesario calcular las derivadas y segundas derivadas numéricas de $D_1(t)$, $D_2(t)$ y $D_3(t)$. De esta manera, un vector de entrada para el entrenamiento de la red neuronal se construye al calcular $D_i(t_0)$, $D_i{}'(t_0)$ y $D_i{}''(t_0)$ para cada una de las tres descomposiciones (para $i = 1, 2, 3$), es decir, la RNA es entrenada utilizando vectores de entrada de 9 elementos. Cabe destacar que aunque pareciera limitado considerar únicamente tres entradas para determinar el vector de rasgos de cada curva representantiva obtenida mediante la descomposición modal, ésta última evidenció la naturaleza periódica (tipo sinusoidal) de las curvas, lo que garantiza que estos tres elementos contienen una parte muy importante de la información respectiva de cada curva. Adicionalmente, el desempeño de la RNA con vectores de rasgo de esta longitud resulta ser adecuado en cuanto a recursos computacionales bajos para su entrenamiento y ejecución.

Con respecto a las salidas deseadas de la RNA, relacionamos cada una de las entradas (formadas por los 9 elementos descritos anteriormente) para un tiempo t_0 con el dato correspondiente a la TSM en ese mismo tiempo (1 salida). Para entrenar la RNA utilizamos datos desde 1978 hasta 2005 (70 % de los datos). El resto de los datos, desde 2005 hasta 2016 (30 %), fueron separados del entrenamiento para ser utilizados como datos de validación. Al usar distintos conjuntos de entrenamiento y de validación, verificamos que la RNA está aprendiendo un patrón, y no únicamente memorizando las entradas de entrenamiento con sus respectivas salidas deseadas. La estructura básica de la RNA es una red neuronal prealimentada (*feedforward*) con tres tipos de capas: entrada, oculta y salida, como se muestra en la Fig. 4. Los parámetros de la RNA se muestran en la Tabla 1. La red entrenada se probó con los conjuntos de entrenamiento y de validación (desde 1978 hasta 2016); los resultados se muestran en la Fig. 4. En esta gráfica se muestran los datos correspondientes a las temperaturas superficiales del mar con la línea azul, y los datos obtenidos de la RNA se muestran con puntos, rojos correspondientes a los datos del conjunto de entrenamiento, y verdes correspondientes a los datos del conjunto de validación. El desempeño de la RNA durante el entrenamiento puede observarse en la Fig. 5.

Tabla 1. Parámetros utilizados en la red neuronal prealimentada.

Característica	Valor/parámetro
Capas de salida	1
Capas ocultas	10
Capas de entrada	1
Tasa de aprendizaje	0.05
Algoritmo de aprendizaje	Levenberg-Marquardt
Error	Error cuadrático medio

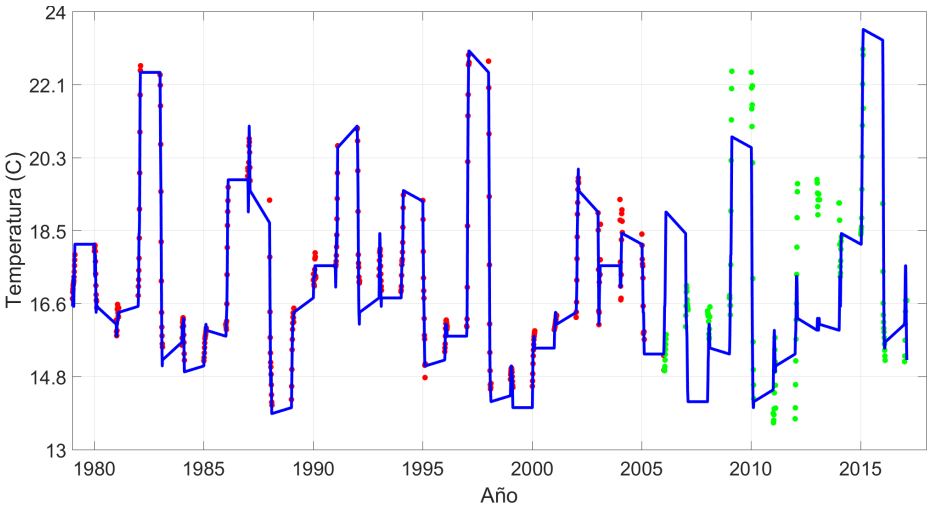


Fig. 4. Los datos de TSM (línea continua en azul) junto con la salida de la RNA para los mismos valores de tiempo. Las salidas correspondientes a datos utilizados para el entrenamiento de la red se muestran como puntos rojos, mientras que aquellos utilizados para la validación se muestran como puntos verdes.

De las Figs. 4 y 5, es evidente apreciar la efectividad de la red para relacionar los datos de IST a través de sus descomposiciones obtenidas mediante DME y sus respectivas derivadas con los datos de TSM, un indicador directamente relacionado con el calentamiento global. Sin embargo, es evidente también de ambas figuras que los datos de TSM y los datos de salida de validación de la RNA difieren en mayor medida en comparación con aquellos utilizados para su entrenamiento. En este sentido, es importante apreciar que el comportamiento general de ambos conjuntos de datos es muy similar, es decir, los datos de la RNA crecen y decrecen al mismo tiempo que lo hacen los datos de TSM, aunque en distinta magnitud. Como parte del trabajo, se considerarán distintas arquitecturas y configuraciones de la RNA para reducir la discrepancia entre los datos de TSM y los datos de salida de validación.

4. Conclusiones

En este trabajo se presenta el diseño y los resultados preliminares de un sistema híbrido basado en DME y en una RNA para relacionar datos de IST con el calentamiento global mediante el indicador de la TSM. Inicialmente, los datos de IST son completados mediante la técnica geoestadística de kriging, para posteriormente ser sujetos a un análisis de DME para obtener sus distintas descomposiciones modales, correspondientes a distintos valores de frecuencia. Estas descomposiciones, junto con sus primeras y segundas derivadas numéricas

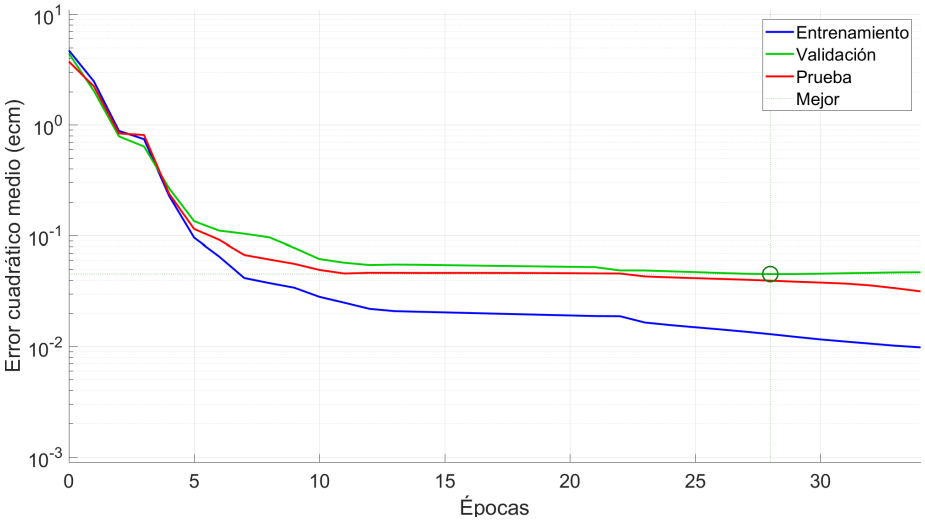


Fig. 5. Desempeño de la RNA durante 34 épocas de entrenamiento. Se muestra el error cuadrático medio para los conjuntos de entrenamiento, validación, prueba. Se indica también el mejor valor alcanzado para el conjunto de validación.

que representan periodos crecientes/decrecientes y con concavidad hacia arriba/abajo, respectivamente, forman al conjunto de entrenamiento de una RNA. Las salidas deseadas de la red fueron asignadas a los distintos valores de TSM. Después de asignar el 70 % del conjunto de datos de entrada al entrenamiento, y el 30 % a la validación, se entrena exitosamente a la RNA de tal manera que nos permite observar una evidente relación entre los datos de IST las variaciones en la TSM.

Con respecto al calentamiento global, los océanos son los mayores responsables de la retención de calor en el sistema Tierra, ya que el calor específico del agua es mucho mayor que el del aire. En este sentido, a pesar de la influencia de materia orgánica y gases de efecto invernadero en las capacidades térmicas del mar [25], este estudio demuestra que las principales variaciones en la TSM se deben a un factor externo como lo es la variabilidad solar.

Mas aún, con los resultados de este estudio, es posible suponer una posible relación entre la IST y el Fenómeno de El Niño. El Fenómeno de El Niño/Oscilación Meridional (ENOM) es uno de los fenómenos climáticos más intrigantes y socioeconómicamente dañinos cuyo origen es aún incierto. Investigación reciente ha apuntado ciertas correlaciones entre el fenómeno ENOM y el cambio climático [3, 15, 49].

Incluso, el IPCC ha sugerido que el cambio climático conllevará a episodios de ENOM cada vez más largos y profundos, los denominados mega-ENOMs [25, 27]. Y como uno de los indicadores fundamentales para determinar los periodos de ENOM es la TSM, será interesante explorar, mediante técnicas de inteligencia

artificial, la relación entre los indicadores aquí estudiados y otros indicadores fundamentales propios del ENOM.

5. Trabajo a futuro

A pesar de ser éste un trabajo aún en proceso, los resultados son claramente alentadores. La relación evidenciada entre los datos de IST y TSM hace de este trabajo el primer paso para la elaboración de un sistema de predicción altamente complejo sobre el calentamiento global con los datos de IST más susceptibles a ajustes y consecuentes predicciones [7, 18, 42, 47].

El próximo paso para el diseño de este sistema consiste en el ajuste de las descomposiciones modales mediante funciones analíticas, de tal forma que puedan utilizarse para extrapolar las entradas de la RNA, y sea posible obtener predicciones del comportamiento de la TSM en el futuro a corto, mediano y largo plazo, permitiendo evaluaciones más precisas sobre calentamiento global, y potencialmente del fenómeno ENOM.

Agradecimientos. Se agradece al Tecnológico Nacional de México/I.T. Mérida por el apoyo económico mediante los proyectos 6513.18-P y 6511.18-P. Los autores también agradecen el apoyo económico parcial de los proyectos 20181139, 20180472, 20181441, 20181028 y 20181141, así como al EDI, todos provistos por SIP/IPN.

Referencias

1. Andrade, A.O., Nasuto, S., Kyberd, P., Sweeney-Reed, C.M., Van Kanijn, F.: EMG signal filtering based on empirical mode decomposition. Biomedical Signal Processing and Control 1(1), 44–55 (2006)
2. Andriopoulou, M., Mavromichalaki, H., Preka-Papadema, P., Plainaki, C., Belov, A., Eroshenko, E.: Solar activity and the associated ground level enhancements of solar cosmic rays during solar cycle 23. Astrophys. Sp. Sci. Trans. 7, 439–443 (2011)
3. Ashok, K., Yamagata, T.: Climate change: The El Niño with a difference. Nature 461(7263), 481 (2009)
4. Attia, A.F., Ismail, H.A., Basurah, H.M.: A Neuro-Fuzzy modeling for prediction of solar cycles 24 and 25. Astrophysics and Space Science 344(1), 5–11 (2013)
5. Basogain Olabe, X.: Redes neuronales artificiales y sus aplicaciones. Med. Intensiva 29, 13–20 (2005)
6. Belov, A., Eroshenko, E., Kryakunova, O., Kurt, V., Yanke, V.: Ground level enhancements of solar cosmic rays during the last three solar cycles. Geomagnetism and Aeronomy 50(1), 21–33 (2010)
7. Calvo, R., Ceccato, H., Piacentini, R.: Neural network prediction of solar activity. The Astrophysical Journal 444, 916–921 (1995)
8. Claussen, M., Mysak, L., Weaver, A., Crucifix, M., Fichefet, T., Loutre, M.F., Weber, S., Alcamo, J., Alexeev, V., Berger, A., et al.: Earth system models of intermediate complexity: closing the gap in the spectrum of climate system models. Climate dynamics 18(7), 579–586 (2002)

9. Coddington, O., Lean, J., Lindholm, D., Pilewskie, P., Snow, M., NOAA CDR Program: NOAA Climate Data Record (CDR) of Total Solar Irradiance (TSI), NRLTSI Version 2. Daily record. https://www.ngdc.noaa.gov/docucomp/page?xml=NOAA/NESDIS/NCDC/Geoportal/iso/xml/C00828.xml&view=getDataView&header=none (2015), last accessed 2018/01/06
10. Drakakis, K.: Empirical mode decomposition of financial data. Int. Math. Forum 4, 1191–1202 (2008)
11. Du, H.k., Cao, J.x., Xue, Y.j., Wang, X.j.: Seismic facies analysis based on self-organizing map and empirical mode decomposition. Journal of Applied Geophysics 112, 52–61 (2015)
12. Flato, G.M.: Earth system models: an overview. Wiley Interdisciplinary Reviews: Climate Change 2(6), 783–800 (2011)
13. Ganopolski, A., Petoukhov, V., Rahmstorf, S., Brovkin, V., Claussen, M., Eliseev, A., Kubatzki, C.: CLIMBER-2: a climate system model of intermediate complexity. Part II: model sensitivity. Climate Dynamics 17(10), 735–751 (2001)
14. Gazis, P., Richardson, J., Paularena, K.: Long term periodicity in solar wind velocity during the last three solar cycles. Geophysical research letters 22(10), 1165–1168 (1995)
15. Gergis, J.L., Fowler, A.M.: A history of ENSO events since AD 1525: implications for future climate change. Climatic Change 92(3-4), 343–387 (2009)
16. Goosse, H., Brovkin, V., Fichefet, T., Haarsma, R., Huybrechts, P., Jongma, J., Mouchet, A., Selten, F., Barriat, P.Y., Campin, J.M., et al.: Description of the Earth system model of intermediate complexity LOVECLIM version 1.2. Geoscientific Model Development 3, 603–633 (2010)
17. Hale, G.E.: Sun-spots as magnets and the periodic reversal of their polarity. Nature 113(2829), 105 (1924)
18. Hathaway, D.H., Wilson, R.M., Reichmann, E.J.: A synthesis of solar cycle prediction techniques. Journal of Geophysical Research: Space Physics 104(A10), 22375–22388 (1999)
19. Hawkins, E., Sutton, R.: The potential to narrow uncertainty in regional climate predictions. Bulletin of the American Meteorological Society 90(8), 1095–1107 (2009)
20. Hayati, M., Mohebi, Z.: Application of artificial neural networks for temperature forecasting. World Academy of Science, Engineering and Technology 28(2), 275–279 (2007)
21. Helama, S., Fauria, M.M., Mielikäinen, K., Timonen, M., Eronen, M.: Sub-Milankovitch solar forcing of past climates: mid and late Holocene perspectives. Bulletin 122(11-12), 1981–1988 (2010)
22. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In: Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences. vol. 454, pp. 903–995. The Royal Society (1998)
23. Huntingford, C., Lowe, J., Booth, B.B., Jones, C.D., Harris, G., Gohar, L., Meir, P.: Contributions of carbon cycle uncertainty to future climate projection spread. Tellus B 61(2), 355–360 (2009)
24. Intergubernamental Panel on Climate Change: Homepage. http://www.ipcc.ch/home_languages_main_spanish.shtml (2014), last accessed 2018/04/01
25. Interngubernamental Panel on Climate Change: IPCC: Working Group I Contribution to the IPCC Fifth Assessment Report, Climate Change 2013: The Physical Science Basis. IPCC. AR5. Interngubernamental Panel on Climate Change (2014)

26. Kane, R.: Short-term periodicities in solar indices. Solar Physics 227(1), 155–175 (2005)
27. Kim, B.H., Ha, K.J.: Observed changes of global and western Pacific precipitation associated with global warming SST mode and mega-ENSO SST mode. Climate dynamics 45(11-12), 3067–3075 (2015)
28. Kim, D., Oh, H.S.: EMD: a package for empirical mode decomposition and Hilbert spectrum. The R Journal 1(1), 40–46 (2009)
29. Kopp, G., Lean, J.L.: A new, lower value of total solar irradiance: Evidence and climate significance. Geophysical Research Letters 38(1) (2011)
30. Krige, D.: A statistical approach to some basic mine valuation problems on the witwatersrand. J. Chem. Metall. Min. Soc. S. Afr. p. 201–215 (1952)
31. Laprise, R.: Regional climate modelling. Journal of Computational Physics 227(7), 3641–3666 (2008)
32. Liu, L., Cheng, Y., Wang, X.: Genetic algorithm optimized Taylor Kriging surrogate model for system reliability analysis of soil slopes. Landslides 14(2), 535–546 (2017)
33. Matheron, G.: Principles of geostatistics. Economic geology 58(8), 1246–1266 (1963)
34. Mursula, K., Zieger, B.: The 1.3-year variation in solar wind speed and geomagnetic activity. Advances in Space Research 25(9), 1939–1942 (2000)
35. Nannariello, J., Fricke, F.: Introduction to neural network analysis and its application to building services engineering. Building Services Engineering Research and Technology 22(1), 58–68 (2001)
36. Nunes, J.C., Bouaoune, Y., Delechelle, E., Niang, O., Bunel, P.: Image analysis by bidimensional empirical mode decomposition. Image and vision computing 21(12), 1019–1026 (2003)
37. Obridko, V., Shelting, B.: Occurrence of the 1.3-year periodicity in the large-scale solar magnetic field for 8 solar cycles. Advances in Space Research 40(7), 1006–1014 (2007)
38. Orozco-Del-Castillo, M., Ortiz-Alemán, J., Couder-Castañeda, C., Hernández-Gómez, J., Solís-Santomé, A.: High solar activity predictions through an artificial neural network. International Journal of Modern Physics C 28(06), 1750075 (2017)
39. Petoukhov, V., Claussen, M., Berger, A., Crucifix, M., Eby, M., Eliseev, A., Fichefet, T., Ganopolski, A., Goosse, H., Kamenkovich, I., et al.: EMIC Intercomparison Project (EMIP–CO 2): comparative analysis of EMIC simulations of climate, and of equilibrium and transient responses to atmospheric CO 2 doubling. Climate Dynamics 25(4), 363–385 (2005)
40. Petrovay, K.: Solar cycle prediction. Living reviews in solar physics 7(1), 6 (2010)
41. Prince, A.M., Thomas, S., Jon, R., Jayapandian, D.: A study on midrange periodicity of sunspot number during solar cycles 21, 22, 23 & 24. Int. J. Sci. Res. Publ. 3, 1–5 (2013)
42. Rigozo, N., Echer, M.S., Evangelista, H., Nordemann, D., Echer, E.: Prediction of sunspot number amplitude and solar cycle length for cycles 24 and 25. Journal of Atmospheric and Solar-Terrestrial Physics 73(11-12), 1294–1299 (2011)
43. Ritz, S.P., Stocker, T.F., Joos, F.: A coupled dynamical ocean–energy balance atmosphere model for paleoclimate studies. Journal of Climate 24(2), 349–375 (2011)
44. Rummukainen, M.: State-of-the-art with Regional Climate Models. Wiley Interdisciplinary Reviews: Climate Change 1(1), 82–96 (2010)

45. Shaffer, G., Malskær Olsen, S., Pepke Pedersen, J.: Presentation, calibration and validation of the low-order, DCESS Earth System Model (Version 1). Geoscientific Model Development 1(1), 17–51 (2008)
46. Shea, M., Smart, D.: A summary of major solar proton events. Solar Physics 127(2), 297–320 (1990)
47. Uwamahoro, J., McKinnell, L.A., Cilliers, P.J.: Forecasting solar cycle 24 using neural networks. Journal of Atmospheric and Solar-Terrestrial Physics 71(5), 569–574 (2009)
48. Weaver, A.J., Eby, M., Wiebe, E.C., Bitz, C.M., Duffy, P.B., Ewen, T.L., Fanning, A.F., Holland, M.M., MacFadyen, A., Matthews, H.D., et al.: The UVic Earth System Climate Model: Model description, climatology, and applications to past, present and future climates. Atmosphere-Ocean 39(4), 361–428 (2001)
49. Xie, S.P., Deser, C., Vecchi, G.A., Collins, M., Delworth, T.L., Hall, A., Hawkins, E., Johnson, N.C., Cassou, C., Giannini, A., et al.: Towards predictive understanding of regional climate change. Nature Climate Change 5(10), 921 (2015)
50. Zhang, J., Li, X., Yang, R., Liu, Q., Zhao, L., Dou, B.: An extended Kriging method to interpolate near-surface soil moisture data measured by wireless sensor networks. Sensors 17(6), 1390 (2017)
51. Zickfeld, K., Eby, M., Weaver, A.J., Alexander, K., Crespin, E., Edwards, N.R., Eliseev, A.V., Feulner, G., Fichefet, T., Forest, C.E., et al.: Long-term climate change commitment and reversibility: an EMIC intercomparison. Journal of Climate 26(16), 5782–5809 (2013)

ISSN 1870-4069

Neuronas artificiales con wavelets paramétricos

Oscar Herrera-Alcántara[1], Miguel González-Mendoza[2]

[1] Universidad Autónoma Metropolitana,
México

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey,
México

oha@azc.uam.mx, mgonza@itesm.mx

Resumen. Presentamos neuronas artificiales con wavelets paramétricos con las cuales es posible representar las compuertas lógicas AND, OR, NOT, NAND, NOR, y XOR con una única neurona. Esta es una ventaja respecto a otro tipo de neuronas, incluido el modelo de McCulloch-Pits y del perceptrón, para las cuales se requieren conectar más de una de ellas para expresar la compuerta XOR que representa un problema no linealmente separable. Los resultados experimentales muestran propiedades y ventajas de las neuronas con wavelets paramétricos en problemas de clasificación y aproximación de funciones. En particular, comparamos el desempeño de redes de neuronas con wavelets Haar y wavelets paramétricos. Las redes neuronales con wavelets paramétricos lograron clasificaciones del 100 % en un conjunto de datos tomados de la base de datos del UCI.

Palabras clave: Wavelets, neurona artificial, compuertas lógicas, aproximación de funciones.

Artificial Neurons with Parameterized Wavelets

Abstract. We present artificial neurons with parameterized wavelets from which is possible to represent the AND, OR, NOT, NAND, NOR, and XOR logic gates with a single neuron. This is an advantage with respect to other kind of neurons including the McCulloch-Pits and perceptron models, for which is required more than a single one to express the XOR gate that represents a not linearly separable problem. Experimental results show properties and advantages of parameterized wavelet neurons in classification and aproximation problems. In particular, we compare the performance of neural networks with the Haar wavelet and parametric wavelets. The neural networks with parameterized wavelets achieved 100 % when classifying data taken from the UCI database.

Keywords: Wavelets, artificial neurons, logic gates, function approximation.

1. Introducción

En [15] fueron presentadas redes neuronales con wavelets como una alternativa a las redes de retropropagación para aproximar funciones no lineales. Sin embargo, a más de dos décadas, no ha proliferado su uso, al menos en comparación con otros tipos de redes neuronales que involucran funciones crecientes, continuas, y acotadas [2].

Aunque existe evidencia de la eficiencia de aproximación con redes de wavelets [1,11], en muchas aplicaciones se ha preferido el uso de funciones con propiedades como la derivabilidad, que facilitan la aplicación de métodos de optimización basados en gradiente, como es el caso del algoritmo de retropropagación para perceptrones multicapa (MLPBP) con funciones tangente hiperbólico o sigmoidal.

En el caso de wavelets, también existen funciones como el "sombrero mexicano" que son derivables, y que permiten usar métodos de gradiente en el entrenamiento de redes que conectan varios "wavelons" o unidades de procesamiento en redes wavelet.

En varias arquitecturas de redes neuronales, incluidas las MLPBP, se plantea el número de neuronas y el número de capas ocultas como los factores primordiales en la mejora de la precisión de la aproximación. Sin embargo, no se considera la opción de adaptar las funciones de activación (funciones base). Si acaso, se limitan a elegir de entre un número limitado de funciones predeterminadas.

Las funciones sigmoidal, tangente hiperbólico, lineal, o escalón usadas ampliamente en MLPBP, o las funciones wavelet en los wavelons, tienen una "forma fija", que no se puede adaptar para ajustarse a los datos de entrada, así que la aproximación depende de otros parámetros.

Nuestra idea principal es promover el uso de funciones base adaptables, bajo la hipótesis de que el uso de funciones construidas *ad-hoc* permite minimizar el número de neuronas requeridas para aproximar la función que mejor se ajuste a los datos de entrada. La idea anterior está fundamentado en la teoría de aproximación con wavelets [3] y su relación con las parametrizaciones de filtros de reconstrucción perfecta [5,6,16,12,13].

Con fines explicativos, considérese que quisieramos rellenar un rectángulo con círculos, requeriríamos un número infinito de ellos aún cuando podamos colocar sus centros en diferentes posiciones y con diferentes radios. Por otro lado, si se tuviera la oportunidad de elegir cuadrados en lugar de círculos, es fácil pensar que con un menor número de ellos se pueda rellenar el rectángulo en cuestión.

Para mostrar lo anterior, presentamos la aproximación de funciones booleanas (compuertas lógicas) mediante neuronas con wavelets, mismas que se comparan con la arquitectura presentada para una MLP de la función XOR (ver [4]) que representa un problema no linealmente separable.

El artículo está organizado de la siguiente manera, en la Sección 2 presentamos la metodología seguida en el presente trabajo de investigación, en la Sección 3 presentamos la arquitectura de una neurona con wavelet paramétrico, en la Sección 4 presentamos varios experimentos y sus resultados, y finalmente, en la Sección 5 presentamos conclusiones y trabajos futuros.

2. Metodología

En el estudio de las neuronas con wavelets paramétricos seguimos los siguientes pasos:

1. Revisamos la arquitectura de una neurona artificial básica, y proponemos una que involucre funciones wavelet con paramétros.
2. Elegimos un conjunto de funciones a aproximar, que permitan medir la eficiencia de aproximación de las neuronas con wavelets.
3. Realizamos experimentos con un programa de computadora para medir la capacidad de aproximación de neuronas con wavelets.

3. Arquitectura de una neurona con wavelets paramétricos

El término "neurona" se atribuye a la inspiración en el modelo biológico. Se usa neurona artificial para hacer énfasis en que se trata de modelos matemáticos que asemejan el funcionamiento de la neurona biológica, al menos como fue planteado hace varias décadas en trabajos pioneros en este tema [9].

En ocasiones, se da poca importancia a la justificación del por qué las redes neuronales funcionan. En particular, para redes de perceptrones, en [2] se puede revisar uno de los trabajos que sustenta y da pauta a la elección de las funciones que pueden usarse en redes neuronales, y de hecho a la justificación de que una sola capa oculta es suficiente para aproximar cualquier función de energía finita.

En cuanto a los wavelets, la fórmula de reconstrucción de la transformada wavelet continua para una función $f(x)$ está dada por la ecuación (1) [3,10]:

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} < f(x), \psi(\frac{x-b}{a}) > \frac{1}{\sqrt{|a|}} \psi(\frac{x-b}{a}) \frac{1}{a^2} dadb, \qquad (1)$$

donde $\psi(x)$ es la función wavelet, con término de traslación $b \in \mathbb{R}$, y factor de escalamiento $a \neq 0$. C_ψ es una constante de admisibilidad, y $< f(x), \psi(\frac{x-b}{a}) >$ representa la transformada wavelet (coeficientes de transformación) dada por la ecuación (2):

$$< f(x), \psi(\frac{x-b}{a}) >= \int f(x) \frac{1}{\sqrt{|a|}} \overline{\psi}(\frac{x-b}{a}) dx, \qquad (2)$$

Las ecuaciones (1) y (2) representan la piedra angular de la arquitectura que define una red neuronal con "átomos" de descomposición en el espacio wavelet.

Entonces, en forma análoga al modelo matemático del perceptrón, se propone una neurona wavelet como se ilustra en la Figura 1.

Su descripción matemática esta dada por la ecuación (3):

$$Y = w_s \; g(u), \qquad (3)$$

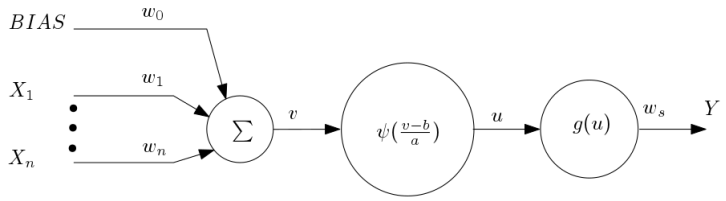


Fig. 1. Neurona con wavelet paramétrico.

donde w_s es el peso de la capa de salida. La función wavelet $\psi(\cdot)$ es trasladada por el término b y escalada por el factor $a \neq 0$, entonces se tiene la ecuación (4) para:

$$u = \psi(\frac{v-b}{a}), \tag{4}$$

con $g(u)$ como función de la capa de salida. El potencial de activación v está dado por la ecuación (5):

$$v = \sum_n w_i X_i, \tag{5}$$

donde X_i es el dato de un vector de dimensión n en la capa de entrada, X_0 es el *BIAS* unitario, y w_i son los pesos sinápticos.

Las funciones wavelet ψ son generadas mediante el algoritmo en cascada [8], y dependen de parámetros de filtros de reconstrucción perfecta que determinan su forma. Algunas parametrizaciones de filtros que se pueden usar para dar lugar a la aproximación de estas funciones wavelet se pueden revisar en [5]. En este trabajo, se implementaron las parametrizaciones con un parámetro α para filtros de longitud 4, y α, β para filtros de longitud 6.

Cabe mencionar que la forma del wavelet está determinada por los valores de los parámetros. En [5] pueden revisarse distintas combinaciones de parámetros asociados a diferentes wavelets. La longitud del filtro determina el soporte compacto del wavelet (el dominio del wavelet a partir del cual su valor, tanto a la izquierda o derecha del eje real, es cero). Así, por ejemplo, un valor de $\alpha = \frac{\pi}{4}$ de un filtro de longitud 4, genera un wavelet tipo Haar, con soporte en el intervalo $[0, \frac{3}{2}]$. En forma más general, un filtro de longitud L tiene un soporte compacto $L-1$, y en el caso anterior, se tiene que la mitad de ese intervalo (de $\frac{3}{2}$ a 3) es cero.

4. Experimentos y resultados

Acorde con la metodología de la Sección 2, realizamos varios experimentos por computadora con los siguientes pasos:

1. Elegir un conjunto de compuertas lógicas y datos de clasificación para verificar el modelo propuesto de neurona con wavelet paramétrico

2. Aplicar el modelo de neurona propuesto y optimizar los parámetros libres w_i, a, b, w_s, y α o β según corresponda a la longitud del filtro. Para ello se aplicó un algoritmo genético con población P, cruzamiento anular, selección directa con pares (0, $P-1$), (1, $P-2$),(2, $P-3$),..., probabilidad de cruzamiento Pc, y probabilidad de mutación P_m.
3. Medir el error de aproximación con el RMS, y el porcentaje de clasificación.

4.1. Experimento 1

En un primer experimento, se aprovechó que entre los pocos wavelets conocidos con expresión analítica, está el wavelet de Haar dado por la ecuación (6):

$$\psi(x) = \begin{cases} 1 & \text{Si} \quad 0 \leq x < \frac{1}{2}, \\ -1 & \text{Si} \quad \frac{1}{2} \leq x < 1, \\ 0 & \text{caso contrario}, \end{cases} \tag{6}$$

el cual, como ya se mencionó, representa un caso especial dentro de los wavelets paramétricos, que puede aproximarse con $\alpha = \frac{\pi}{4}$ y el algoritmo en cascada. El wavelet Haar de la ecuación (6) tiene un soporte compacto de $[0, 1)$.

Así que, para facilitar la evaluación de la función wavelet de Haar sin necesidad de ejecutar el algoritmo en cascada, se optó por fijar el parámetro $\alpha = \frac{\pi}{4}$ y usar la ecuación (6) con la debida adaptación al soporte compacto.

Las funciones de entrada seleccionados fueron las compuertas AND, OR, NAND, NOR y XOR, como se especifican en la Tabla 1, que representan problemas de clasificación binaria. También se probó con la función NOT que invierte las entradas -1 y 1.

Tabla 1. Tablas de verdad para las compuertas AND, OR, NAND, NOR y XOR.

X1	X2	AND	OR	NAND	NOR	XOR
-1	-1	-1	-1	1	1	-1
-1	1	-1	1	1	-1	1
1	-1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1

La función de salida utilizada fue la función signo, dada en la ecuación (7):

$$g(u) = \begin{cases} 1 & \text{Si} \quad 0 \leq u, \\ -1 & \text{Si} \quad u < 0, \end{cases} \tag{7}$$

Una de las ventajas de usar wavelets con soporte compacto, es que permiten acotar los valores de los parámetros a y b, toda vez que fuera del soporte el valor del wavelet es cero. En los experimentos, el valor absoluto máximo de a y b fue 32 que es casi 10 veces el valor del soporte.

Tabla 2. Parámetros para las compuertas AND, OR, NAND, NOR y XOR.

Función	w_0	w_1	w_2	a	b	Generaciones
AND	-8.547	1.574	1.255	-8.047	-0.141	4
OR	-17.070	5.063	4.773	-24.0472	-0.141	91
NAND	8.548	10.187	7.773	28.848	-16.554	6
NOR	8.548	10.187	7.773	28.848	-16.554	6
XOR	-1.0629	4.063	4.773	-8.047	-0.141	10
NOT	8.548	10.187	4.773	28.848	-16.554	2

Por simplicidad, se fijó $w_s = 1{,}0$. Los demás valores de los parámetros libres de la neurona wavelet se optimizaron evolutivamente y los resultados se muestran en la Tabla 2.

En la Tabla 2 se puede apreciar, en la última columna, el número de generaciones que requirió el algoritmo genético para alcanzar el 100 % de precisión, es decir, el error igual a cero. Se logró el empate idéntico entre los valores de entrada y la salida en parte debido a la apropiada elección de la función de salida descrita en la ecuación (7) que incluye los valores -1 y 1.

Nótese que la función que más generaciones requirió fue la función lógica OR.

Para la función XOR, que se sabe representa un problema no linealmente separable, se puede tener una solución usando una sola capa oculta con dos neuronas N_1 y N_2 del modelo McCulloch-Pitts, con entradas 0 y 1 (ver [4,14]). En tal caso, los parámetros solución son: $w_{11} = w_{12} = 1$, $b_1 = -\frac{3}{2}$, $w_{21} = w_{22} = 1$, $b_2 = -\frac{1}{2}$, $w_{31} = -2$, $w_{32} = 1$, y $b_3 = -\frac{1}{2}$ como se ilustra en la Figura **??**. La capa de salida contiene la neurona N_3.

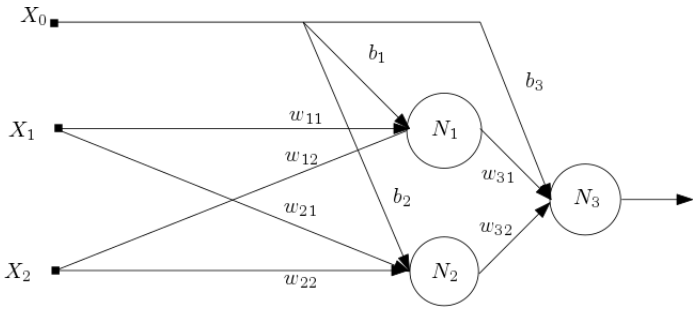


Fig. 2. Red de neuronas McCulloch-Pitts que resuelven el problema del XOR.

4.2. Experimento 2

En un segundo experimento se usaron 26 neuronas. Cada una de las 26 es una instancia de la neurona wavelet descrita en el Experimento 1. De esta forma,

se contruyó una red neuronal con arquitectura "tipo MLP" con una capa de entrada, una capa oculta con neuronas wavelets, y una capa de salida con función signo.

Los datos de prueba fueron los de clasificación de vinos, que considera 3 tipos de vinos [7]. Entonces, se dividieron en dos experimentos: Caso 1. Los de la primera clase con la tercera clase, y un Caso 2: Los de la primera clase con los de la segunda clase. En ambos casos se tiene una clasificación binaria, así que los valores de salida fueron etiquetados como -1 y 1. Ahora se describen los experimentos con ambos casos.

Experimento del caso 1. Con $G = 314$ generaciones, $P = 40$, $P_{cruz} = 0{,}97$ y $P_{mut} = 0{,}01$, se logró el máximo éxito de clasificación posible, es decir el 100 % para los 107 datos de entrada, con un valor de parámetro óptimo $\alpha = 4{,}517208$. En tanto que, con un filtro Haar (equivalente al caso especial de $\alpha = \frac{\pi}{4}$) apenas se logró, en las mismas $G = 314$ generaciones, una clasificación de 92,52 %. El tiempo promedio de ejecución en estos experimentos del Caso 1, fue de 173,2 segundos.

En la Figura 3 se puede observar que, durante la ejecución del algoritmo genético, la red neuronal con wavelets paramétricos tiene una convergencia competitiva con la red neuronal con wavelet Haar, pero a partir de la generación 49 la neurona con filtro wavelet paramétrico converge mucho más rápido a la solución óptima ($RMS = 0$, clasificación del 100 %) en 313 generaciones, pero no así la red neuronal con wavelet Haar para la cual con 2000 generaciones logra un $RMS = 0{,}193$.

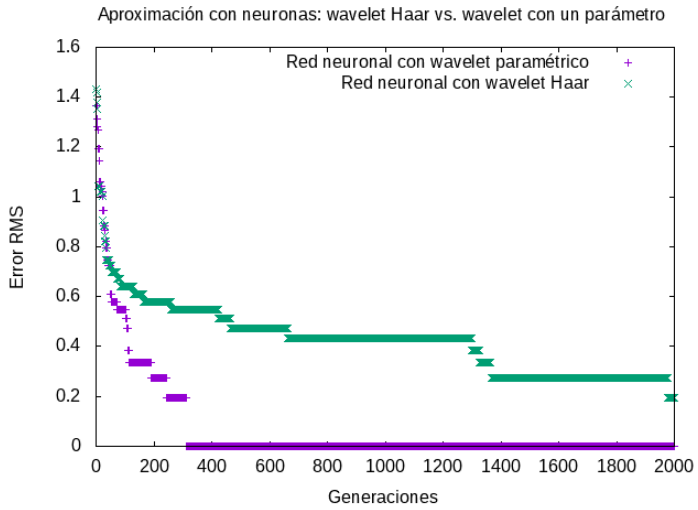


Fig. 3. Desempeño de dos redes neuronales en la clasificación de 107 muestras de dos tipos de vinos: i) red de neuronas wavelet Haar ii) red de neuronas wavelet con paramétro α.

Experimento del caso 2. Con $G = 500$ generaciones, $P = 40$, $P_{cruz} = 0{,}97$ y $P_{mut} = 0{,}02$, se logró una clasificación del 100 % ($RMS = 0$) para 130 datos de entrada con $\alpha = 4{,}0535$ y $\beta = 1{,}855$. Esto representa un desempeño superior al obtenido con $\alpha = \frac{\pi}{4}$ (filtro Haar) que logró una clasificación de 93,84 % y un $RMS = 0{,}4961$ en 2000 generaciones. En la Figura 4 se ilustra la ejecución del algoritmo genético en donde se aprecia que una red neuronal con wavelet Haar no logra reducir el error RMS aún durante un gran número de $G = 2000$ generaciones. En contraste, una red neuronal con wavelet de dos parámetros logra en $G = 500$ generaciones la aproximación perfecta para 130 datos de entrada.

El tiempo promedio de ejecución en estos experimentos del Caso 2, fue de 265,2 segundos.

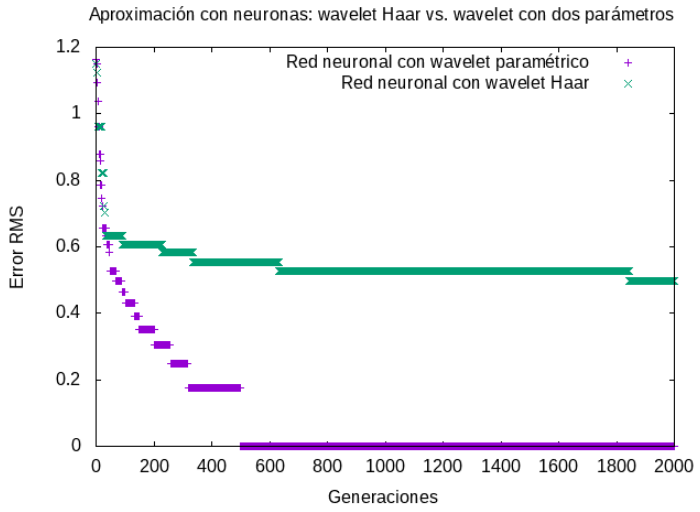


Fig. 4. Desempeño de dos redes neuronales en la clasificación de 130 muestras de dos tipos de vinos: i) red de neuronas wavelet Haar ii) red de neuronas wavelet con dos paramétros α y β.

5. Conclusiones y trabajo futuro

Se presentó un modelo de neurona artificial con una capa de entrada, una capa de procesamiento con función wavelet paramétrica de soporte compacto y una capa de salida. Para probar su validez, se usó la compuerta NOT y compuertas lógicas AND, OR, NAND, y NOR con dos entradas, con las cuales se logró un error de aproximación de cero, con una sola neurona de este tipo. También se aplicó a la función lógica XOR, que suele requerir más neuronas de otros modelos incluido el de McCulloch-Pitts y el perceptrón. En el caso de la neurona

con wavelet, fue posible resolver el problema del XOR con una única neurona y wavelet Haar visto como un caso especial de wavelet paramétrico.

Aunque no se conocen las funciones analíticas de los wavelets paramétricos, es posible aplicar métodos evolutivos para optimizar los parámetros y ajustar la red neuronal a los datos de entrada. Los experimentos complementan los conceptos teóricos que afirman que la transformada wavelet (y por ende las redes neuronales basadas en wavelets) requieren un menor número de coeficientes (neuronas) para concentrar la energía de una función y mejorar la aproximación.

Los resultados obtenidos son alentadores y sugieren proponer más arquitecturas basadas en funciones paramétricas, que provean alternativas a otros métodos que usan funciones "de forma fija" primordialmente derivables, cuyo comportamiento no permite capturar cambios abruptos en las funciones a aproximar. Por otro lado, las funciones wavelet han mostrado una superioridad ya que permiten detectar fenómenos transitorios, y en nuestro caso al usar neuronas con wavelets Haar en problemas de clasificación se lograron desempeños cercanos al 100 %, y al optimizar parámetros con un algoritmo genético se lograron clasificaciones del 100 % en un menor número de generaciones.

Como trabajo futuro se plantea proponer técnicas heurísticas para mejorar los tiempos de ejecución, y aplicarlas a problemas de clasificación y aproximación de funciones más complejos.

Referencias

1. Candès, E.J., Donoho, D.L.: Ridgelets: a key to higher-dimensional intermittency? Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 357(1760), 2495–2509 (1999)
2. Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function. Math. Control Signal Systems 24 (1989), https://doi.org/10.1007/BF02551274
3. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1998)
5. Herrera-Alcántara, O., González-Mendoza, M.: Inverse formulas of parameterized orthogonal wavelets. Computing (2018)
6. Lai, M.J., Roach, D.W.: Parameterizations of Univariate Orthogonal Wavelets With Short Support. Vanderbilt University Press (2002)
7. Lichman, M.: UCI Machine Learning Repository (2013)
8. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press (1999)
9. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 51(115) (1943)
10. Navarro, J., Elizarraraz, D.: Introducción a la Transformada de Wavelet Continua. Reverte, Mexico (2010)
11. Pati, Y.C., Krishnaprasad, P.S.: Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. IEEE Transactions on Neural Networks 4(1), 73–85 (Jan 1993)
12. Roach, D.W.: Frequency selective parameterized wavelets of length ten. Journal of Concrete and Applicable Mathematics 8(1), 1675–179 (2010)
13. Roach, D.: The Parameterization of the Length Eight Orthogonal Wavelets with No Parameter Constraints (2008)

14. Touretsky, D.S., Pomerleau, D.A.: What is hidden in the hidden layers? Byte 14, 227–233 (1989)
15. Zhang, Q., Benveniste, A.: Wavelet networks. IEEE Transactions on Neural Networks 3(6), 889–898 (Nov 1992)
16. Zou, H., Tewfik, A.: Parametrization of compactly supported orthonormal wavelets. IEEE Trans. Signal Process. 41(3), 1428–1431 (1993)

ISSN 1870-4069

Clasificación de jugadores de futbol soccer basada en sus habilidades deportivas, físicas y mentales

Enrique Antonio Pedroza Santiago, Maricela Quintana López,
Héctor Rafael Orozco Aguirre, Víctor Manuel Landassuri Moreno

Centro Universitario UAEM Valle de México,
México

enriquepedroza2012@gmail.com, {mquintanal, hrorozcoa, vmlandassurim}@uaemex.mx

Resumen. Regularmente, para determinar la posición adecuada en la que un jugador de futbol soccer debe jugar, el director técnico realiza diversos movimientos a fin de conocer cómo se desarrolla un jugador en cada posición en la cancha, esto con el objetivo de conocer las habilidades que cada jugador debe desarrollar. Este artículo presenta la propuesta de un clasificador de jugadores de futbol soccer, determinando su posición con base en las habilidades deportivas, físicas y mentales de éstos. Para ello, se utilizará un comparativo de dos algoritmos, uno de generación de árboles de decisión (J48) y otro de generación de reglas (PART), esto con el objetivo de analizar las diferentes habilidades que conllevan a una clasificación, así como determinar cuál de éstos algoritmos tiene una mejor precisión. De los experimentos, se puede concluir que los mejores resultados con datos numéricos lo tiene J48, mientras que PART trabaja mejor con datos nominales. De igual forma, es posible concluir que hay habilidades muy bien definidas en algunas posiciones, como es el caso del portero, mientras que en otras como la del mediocampista no lo están del todo.

Palabras clave: Habilidad, clasificación, minería de datos, jugador, futbol.

Classification of Soccer Players Based on Their Sport, Physical and Mental Skills

Abstract. Regularly, to determine the appropriate position in which a soccer player should play, the technical director performs various movements in order to know how a player develops in each position on the court, this with the objective of knowing the skills that each player must develop. This article presents the proposal of a classifier of soccer players, determining their position based on their sport, physical and mental skills. To do this, a comparison of two algorithms will be used, one for the generation of decision trees (J48) and other for the generation of rules (PART), with the aim of analyzing the different skills that lead to a classification, as well as how to determine which of these algorithms has a better precision. From the experiments, it can be concluded that the best results with numerical data is J48, while PART works better with nominal data.

Similarly, it is possible to conclude that there are very well defined skills in some positions, as in the case of the goalkeeper, while others such as the midfielder are not at all.

Keywords: Skill, classification, data mining, player, soccer.

1. Introducción

El futbol soccer es un deporte táctico donde participan diversos actores tales como los jugadores, el director técnico y el árbitro, este último es quien se encarga de dirigir los encuentros. Las habilidades personales de los jugadores se consideran fundamentales para un buen rendimiento deportivo y junto con las estrategias del equipo, influyen o determinan el resultado del encuentro a favor o en contra.

Cuando se realizan entrenamientos a nivel profesional o semiprofesional, los jugadores no sólo adquieren y desarrollan como factores sus habilidades físicas y deportivas, sino que también fortalecen sus habilidades mentales, todos estos factores influyen para lograr un buen desempeño y resultado a favor en un encuentro.

Es necesario encontrar un equilibrio en las habilidades de los jugadores, esto con el objetivo de tener un mejor rendimiento tanto en el juego como en el aspecto personal.

De acuerdo con Jover [1], cuando un jugador fortalece sus habilidades mentales, puede mejorar sus capacidades de convivencia, disfruta del deporte que practica y encuentra un nivel adecuado como deportista.

El futbol es un deporte de conjunto donde cada equipo se conforma de un total de once jugadores en la cancha, cada jugador tiene una posición y función definida, la cual le es asignada por el director técnico, dependiendo de las habilidades mostradas como fortalezas en los entrenamientos. Estas posiciones se dividen principalmente en cuatro: portero, defensa, medio y delantero, con las cuales se trabajaron; aunque dependiendo de qué tan alejado este un jugador de la portería o la dirección donde se encuentre ubicado, estas pueden sub dividirse en otras que no son consideradas en este trabajo. Cada posición cuenta con un objetivo el cual se describe a continuación:

- Portero: su función es evitar que el equipo contrario anote. Es el único que puede tomar el balón con las manos. Sólo se permite un portero por equipo.
- Defensa: se encargan de evitar que los jugadores del equipo contrario lleguen a la portería. Una alineación clásica, incluye 4 defensas, aunque pueden variar dependiendo de la formación establecida por el director técnico.
- Medio: se encuentran en la mitad de la cancha. Apoyan tanto en la defensa como en el ataque. Su función principal es el de distribuir los balones.
- Delantero: debido a su posición, son quienes anotan principalmente los goles.

Regularmente, en los entrenamientos a los jugadores se le asignan posiciones a modo de probar cómo se desenvuelven en cada una de ellas, de manera que aquella donde cada jugador se sienta más cómodo y dé mejores resultados sea en la que juegue de forma constante o permanente en los encuentros. Las habilidades de un jugador determinan la posición en la que obtendrá un mejor desempeño.

Una manera de apoyar en la toma de decisiones respecto a cuál es la posición más adecuada para un jugador, basada en sus habilidades, es el tener un mecanismo

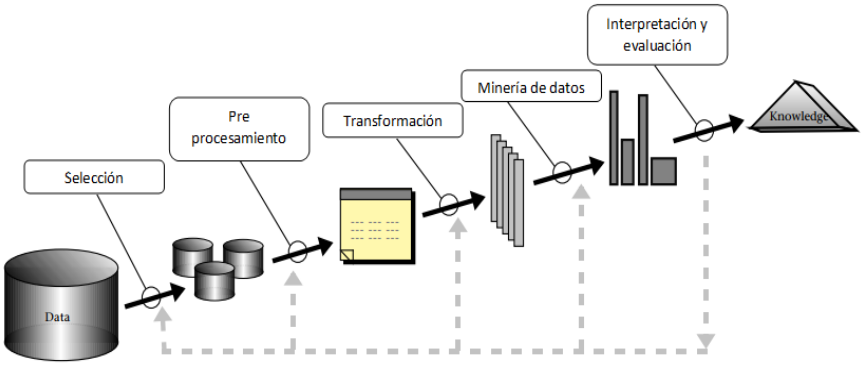


Fig. 1. Etapas de la extracción del conocimiento.

confiable de clasificación. Existen clasificadores de diferentes tipos, algunos de caja negra como las redes neuronales artificiales, que realizan la clasificación pero el conocimiento adquirido no es fácil de examinar [2] y los de caja blanca, como los generadores de árboles de decisión o reglas de clasificación, en los que se puede analizar la estructura que se forma al adquirir el conocimiento. Los algoritmos a utilizar son el C4.5, que genera árboles de decisión y permiten determinar qué atributos se seleccionan para obtener una clasificación y, el algoritmo PART de generación de reglas; esto con el objetivo de analizar las diferentes habilidades que conllevan a una clasificación, así como determinar cuál de éstos algoritmos tiene una mejor precisión. El software que se utilizará es Weka de la universidad de Waikato [3].

El resto de este trabajo está organizado como sigue: en la sección 2, se describe la metodología utilizada; mientras que en la sección 3 se incluyen los algoritmos empleados; posteriormente, en la sección 4, se describen los experimentos realizados y sus resultados; finalmente, en la sección 5, se presentan las conclusiones y el trabajo futuro.

2. Metodología

Para la realización de este trabajo, se utilizó la metodología KDD (Knowledge Discovery in Databases), conocida como el proceso de extracción del conocimiento, propuesta por Fayyad [4], ver figura 1. El proceso consiste de cinco etapas secuenciales e iterativas.

Todas las etapas de la extracción del conocimiento son iterativas, debido a que la salida de una de ellas puede retornar a una etapa anterior, esto con la finalidad de depurar los datos correctamente. A continuación se describe en que consiste cada una.

Preparación de los datos. Esta fase incluye las etapas de selección, procesamiento y transformación. Se determina qué datos se necesitan y se selecciona la información que sea relevante y útil, descartando la mayor cantidad de datos erróneos o que no tengan una aportación, este proceso es conocido como selección y limpieza. Posteriormente se

realiza un agrupamiento de todas las características con la finalidad de conocer si existen anomalías o datos redundantes. Finalmente, la transformación de datos se refiere a todo aquel proceso en el cual se modifiquen éstos.

Minería de datos. Una vez seleccionados los datos, se determina aquellos patrones que se desean obtener, el predecir algún comportamiento o el obtener una clase. Para ello se elige el algoritmo que mejor se adapte a los datos.

Interpretación y evaluación. Finalmente en esta etapa se obtienen los resultados y se determina si se logró una evaluación exitosa. Para ello es necesario una etapa de pruebas donde los datos estén fuera de aquellos con los que se entrenó, de manera que si los resultados tienen un porcentaje alto de acierto se puede determinar que son confiables.

En este trabajo, la preparación de los datos se describe en la sección 3, mientras que las de minería de datos y la de interpretación y evaluación se describen en las secciones 4 y 5 respectivamente.

3. Preparación de los datos

La Federation International de Football Association (FIFA) [5], es la encargada de reunir y gobernar a las diversas asociaciones de futbol existentes en el planeta, es además quien organiza los diversos campeonatos internacionales y, es quien tiene la facultad de crear o modificar las reglas del juego. Este organismo reconoce y almacena las distintas habilidades que un jugador profesional debe poseer siendo un total de 34. Estas habilidades incluyen las deportivas, las físicas y las mentales. A estas se les asignan un valor numérico del 0 al 100 dependiendo de la evaluación de cada jugador.

En la tabla 1, se pueden observar los valores de la media y desviación estándar de las habilidades de un jugador profesional reconocidas por la FIFA.

Tabla 1. Habilidades deportivas, físicas y mentales de los jugadores.

DEPORTIVAS			FÍSICAS	MENTALES
CONTROL DEL BALÓN $\bar{x}$= 59.70, σ=17.25	RETÉN DEL BALÓN $\bar{x}$= 17.87, σ=19.11	FUERZA DE TIRO $\bar{x}$= 58.83, σ=18.08	ACELERACIÓN $\bar{x}$= 66.30, σ=14.54	AGRESIVIDAD
REGATES $\bar{x}$= 56.62, σ=19.62	DESPEJE $\bar{x}$= 18.07, σ=18.60	REMATES $\bar{x}$= 47.26, σ=20.18	ENERGÍA $\bar{x}$= 65.54, σ=16.40	REACCIONES
CENTROS $\bar{x}$= 51.78, σ=18.77	REFLEJOS $\bar{x}$= 18.20, σ=20	TIROS LARGOS $\bar{x}$= 50.61, σ=20.33	FUERZA $\bar{x}$= 65.01, σ=12.24	POSICIÓN DE ATAQUE
PASE CORTO $\bar{x}$= 59.96, σ=14.27	MARCAJE $\bar{x}$= 45.76, σ=21.78	CURVA $\bar{x}$= 50.10, σ=18.76	BALANCE $\bar{x}$= 66.11, σ=12.42	INTERCEPCIÓN
PASE LARGO $\bar{x}$= 54.80, σ=15.52	BARRIDA $\bar{x}$= 46.60, σ=21.99	PRECISIÓN TIRO LIBRE $\bar{x}$= 44.71, σ=18.11	VELOCIDAD DE SPRINT $\bar{x}$= 66.20, σ=14.79	VISIÓN
COLOCACIÓN $\bar{x}$= 18.38, σ=19.39	ENTRADA LIMPIA $\bar{x}$= 47.81, σ=22.33	PENALES $\bar{x}$= 53.44, σ=16.59	AGILIDAD $\bar{x}$= 64.25, σ=14.20	COMPOSTURA
ESTIRADA $\bar{x}$= 18.15, σ=19.69	CABECEO $\bar{x}$= 54.07, σ=18.74	VOLEAS $\bar{x}$= 45.69, σ=18.74	SALTO $\bar{x}$= 67.63, σ=10.84	

Por otro lado, existen un total de 18 equipos en liga mexicana torneo clausura 2018 (ver tabla 2), cada uno tiene de 20 a 40 jugadores registrados ante la Federación Mexicana de Futbol (FMF) [6], la mayoría de ellos no tienen actividad de forma profesional ya que están en preparación constante para poder debutar. De aquellos que tienen actividad se buscaron sus habilidades en la página oficial de la FIFA.

Tabla 2. Equipos participantes en la liga mexicana torneo clausura 2018.

ATLÁS	MONTERREY	SANTOS
AMÉRICA	MORELIA	TIJUANA
CRUZ AZUL	NECAXA	TOLUCA
GUADALAJARA	PACHUCA	U.A.N.L
LEÓN	PUEBLA	U.N.A.M
LOBOS BUAP	QUERÉTARO	VERACRUZ

Se trabajó con un total de 513 jugadores de primera división en México. Con estos datos se formaron dos conjuntos: el de entrenamiento, y el de prueba. Para el primer conjunto se tomaron los datos de los titulares y suplentes de los equipos del torneo clausura 2018 con un total de 440 jugadores, teniendo 52 porteros, 101 delanteros, 130 defensas y 157 medios, mientras que para la prueba se tomó una muestra de 73 jugadores que militaban en la liga mexicana en el torneo apertura 2017 con la restricción de que no se encuentren en los datos del conjunto de entrenamiento. Para ello se trabajo con 8 porteros, 18 delanteros, 26 defensas y 21 medios; cada uno de ellos con 34 habilidades, las cuales fueron mencionadas en la tabla 1.

Los algoritmos usados admiten datos tanto numéricos como nominales, por ello se hicieron dos análisis, uno utilizando datos con escala de 0-100, mientras que en el segundo se utilizó una escala Likert de 3 rangos: bajo, medio y alto. Esta transformación de datos se realizó con el objetivo de tener una mejor comprensión de los mismos, donde es más fácil asimilar que un jugador es bueno por tener habilidades medias o altas, que en el caso donde se tiene una evaluación numérica.

4. Minería de datos

La minería de datos es el proceso mediante el cual es posible obtener información precisa y valiosa de aquellos datos que se encuentran desorganizados [7]. Existen investigaciones donde se han utilizado algoritmos de minería de datos en el deporte, ejemplo de ellos es [8], donde es posible evaluar el rendimiento de los jugadores de voleibol utilizando los algoritmos C4.5 y EM. Otro ejemplo es el encontrado en [9], donde utilizan el clasificador Naive Bayes para obtener un pronóstico en los encuentros de tenis. Otro enfoque que han utilizado es para el comportamiento de las apuestas deportivas, específicamente en quinielas como lo es el caso de [10].

Existen diversos algoritmos de clasificación, entre ellos se encuentran ID3 y C4.5, los cuales suelen ser precisos y dan una gran confiabilidad [11], este último es el que se utilizó para determinar las posiciones de cada jugador.

Otra opción para clasificar es la generación de reglas, para construirlas se encuentran relaciones entre diversos atributos, como es el caso de PART.

4.1. Algoritmo C4.5 (J48)

C4.5 también conocido como J48 es un algoritmo utilizado para resolver problemas de decisión o de clasificación [12]. Este surge en el año de 1993 por J. Quinlan, como

una mejora del algoritmo ID3, el cual solo admite valores numéricos. C4.5 genera árboles al encontrar un atributo que tenga la mayor ganancia y es utilizado como nodo raíz, posteriormente la división de datos se basa en la recursividad de estos. Una de las características principales de los arboles generados por C4.5 es el radio de ganancia, el cual considera el número de nodos que tiene el árbol y de esta manera el atributo raíz puede dividir los demás conjuntos sin importar la información de la clase. Otro de los aspectos es la poda de árbol que permite que no se expanda cuando los datos se repiten o no son relevantes.

4.2. PART

Es un algoritmo de clasificación basado en reglas, creado por Witten y Frank en el año de 1998 [13]. Este algoritmo podría considerarse como una mezcla de árboles de decisión y reglas de clasificación. PART adopta una estrategia similar a la de J48 al usar la técnica "divide y vencerás", con lo cual crea las reglas al tomar las ramas que tengan una mayor cobertura y elimina aquellas que no cumplan con las condiciones dadas. Estas se siguen creando de manera recursiva hasta que no queden atributos a considerar. Una de las ventajas que tiene PART sobre otros algoritmos como PRISM [14] es que sus reglas son muy cortas y toma sólo los atributos más relevantes.

5. Experimentos y resultados

El primer experimento con el conjunto de datos numéricos de entrenamiento se realizó aplicando el algoritmo C4.5. Los resultados son presentados en la tabla 3.

Tabla 3. Resultados del algoritmo C4.5 con el conjunto de entrenamiento de datos numéricos.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	64	1	13	23	101	63.4	36.6
PORTERO	4	41	4	3	52	79	21
DEFENSA	8	6	92	24	130	70.8	29.2
MEDIO	11	12	16	118	157	75.2	24.8
	Matriz de confusión 315 en la diágnonal				440	71.59	28.41

Se obtuvo un 71.59% de instancias clasificadas correctamente, lo que equivale a un total de 315 jugadores de 440. Interpretando la tabla anterior, la primera posición que aparece es la de delantero. Esta aparece con 64 jugadores clasificados correctamente, mientras que existe confusión en un caso con el portero, 13 con la defensa y 23 con la posición medio, esto sucede ya que existen jugadores que comparten habilidades que son muy parecidas en sus valores.

La confusión más grande ocurre en la clase medio, esto se debe a que la función de estos últimos es apoyar tanto en la defensa como en el ataque. La posición de portero es aquella que se encuentra mejor definida y que tiene poca confusión con otras.

El árbol de decisión generado se muestra en las figuras 2 y 3. En ellas, se pueden observar las cualidades del jugador que proporcionan mayor información para separar las clases. Los atributos se visualizan en un óvalo, mientras que las clasificaciones se encuentran en un rectángulo.

Tabla 4. Resultados del algoritmo C4.5 con el conjunto de prueba con datos numéricos.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	16	0	0	2	18	88.9	11.1
PORTERO	0	8	0	0	8	100	0
DEFENSA	2	0	20	4	26	76.9	23.1
MEDIO	4	1	2	14	21	66.7	33.3
	Matriz de confusión 58 en la diágnonal				73	79.45	20.55

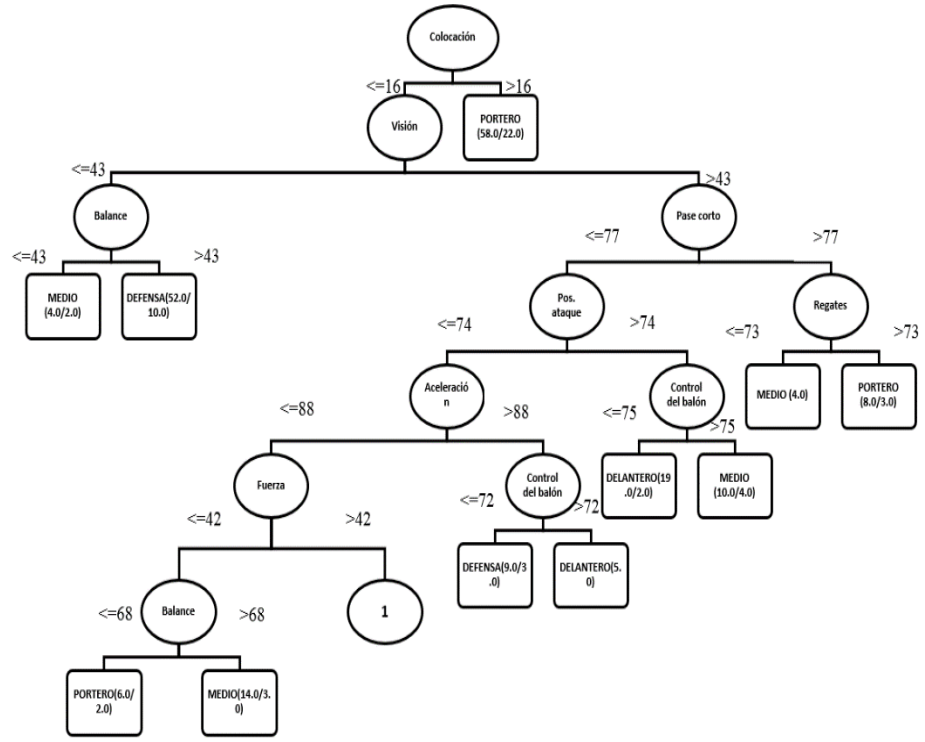


Fig. 2. Primera parte árbol de decisión con datos numéricos.

Analizando el árbol de las figuras 2 y 3, el cual contiene datos numéricos, es posible observar que en la raíz se encuentra la habilidad "colocación", la cual si es mayor a 16 se determina automáticamente que es un portero, mientras que si es menor se observan otros atributos hasta llegar a una clasificación.

En el proceso de prueba, 58 jugadores fueron clasificados correctamente, lo que corresponde a un 79.45%, mientras que 15 de ellos tuvieron un error, dando un total de 20.55%. Los resultados se pueden observar en la tabla 4.

El segundo experimento realizado fue mediante el algoritmo PART. Dando un 72.73% de acierto y un 27.27% de error. Se obtuvieron un total de 30 reglas de las cuales se observa que las habilidades que más destacan por cada posición son: colocación para el portero, retén del balón para el medio, visión y barrida en el caso de la defensa y aceleración y precisión de tiro en cuestión de los delanteros. Al comparar

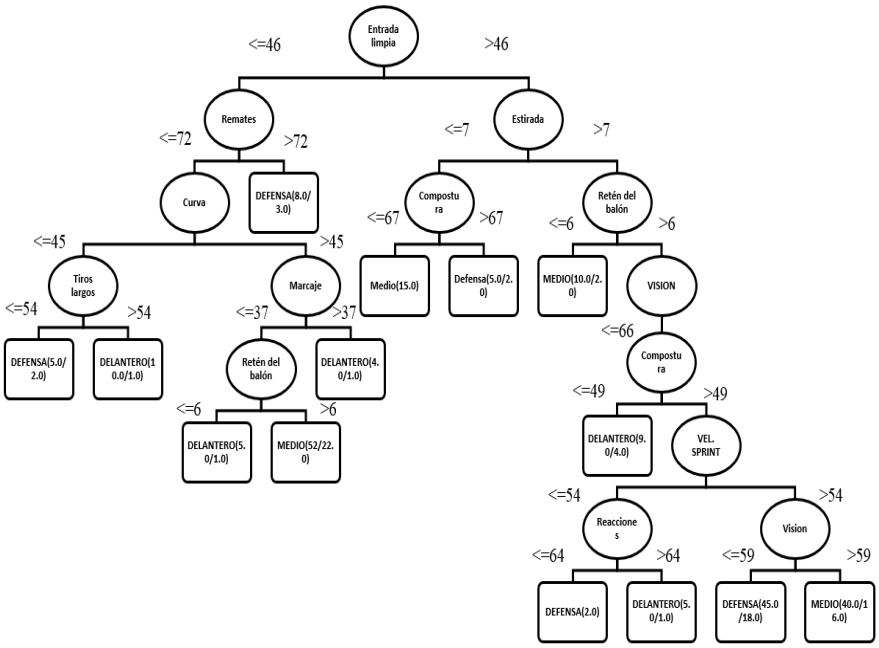


Fig. 3. Segunda parte árbol de decisión con datos numéricos.

Tabla 5. Resultado del algoritmo PART con el conjunto de entrenamiento con datos numéricos.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	58	1	14	28	101	57.4	42.6
PORTERO	4	41	4	3	52	79	21
DEFENSA	4	6	91	29	130	70.0	30.0
MEDIO	7	12	8	130	157	82.8	17.2
	Matriz de confusión 320 en la diágnonal				440	72.73	27.27

ambos algoritmos se puede determinar que coinciden en las características que debe tener un jugador, teniendo como regla principal la de la colocación, donde el 79% de estos caen en el apartado portero (ver en tabla 5). Del total de las reglas obtenidas, la tabla 6 muestra aquellas con una mayor cobertura y precisión.

Para el proceso de prueba se hizo uso de los mismos 73 jugadores que se usaron con C4.5. Del 100%, el 73.97% de los jugadores tuvieron una clasificación correcta, mientras que en el 26.03% hubo algún error. Los resultados se muestran en la tabla 7.

En la sección 2, se mencionó que los valores numéricos fueron transformados en nominales, para ello se encontraron los valores máximos y mínimos por cada habilidad. Una vez hecho esto, se calcularon los rangos de estos límites.

Posteriormente, se dividió el rango entre 3 valores, y cada resultado recae sobre una clasificación, los cuales son bajo, medio y alto, esto con la finalidad de agrupar valores cercanos y conocer que tan bueno es un jugador, diferente a los valores numéricos donde son específicos para cada uno de estos.

Tabla 6. Conjunto de reglas de clasificación datos numéricos.

Si colocación > 16 y centros > 10 y regates > 17 y remates <= 13.	Entonces = DEFENSA (12.0/4.0)
Si reten del balón <= 16 y visión <= 43 y balance > 43 y regates > 27.	Entonces= DEFENSA (46.0/6.0)
Si reten del balón > 16.	Entonces= PORTERO (46.0/14.0)
Pos. Ataque > 74 y pase largo <= 66.	Entonces: DELANTERO (21.0/3.0)
Si reacciones <= 45 y pase largo <= 52.	Entonces= MEDIO (6.0)
Si intercepción > 30 y visión > 65 y reacciones > 69.	Entonces= MEDIO (35.0/14.0)
Si intercepción > 30 y barrida > 68 y reflejos <= 14 y pase largo <= 72 y estirada <= 10.	Entonces= MEDIO (31.0/14.0)
Si intercepción > 30 y barrida <= 69 y reten > 6 y tiro libre > 59 y retén del balón > 7 y boleas > 45 y visión <= 72.	Entonces= MEDIO (27.0/2.0)
Si entrada > 51 y barrida > 69.	Entonces= DEFENSA (15.0/2.0)
Si aceleración > 81 y despeje > 10.	Entonces= DELANTERO (10.0/1.0)
Si reacciones <= 46 y control > 63.	Entonces= PORTERO (3.0/1.0)
Si colocación > 7 y entrada <= 51 y reten > 7 y centros <= 66 y control > 56 y estirada <= 15.	Entonces= MEDIO (20.0/7.0)
Si colocación > 6 y retén del balón > 6 y estirada <= 9 y compostura <= 63.	Entonces= MEDIO (13.0/2.0)
Si reten del balón > 6 y colocación > 6 y estirada > 9 y entrada > 66 y reflejos <= 13 y Aceleración > 53.	Entonces= DELANTERO (14.0/5.0)
Si colocación > 7 y reten del balón > 6 y aceleración <= 75 y visión > 45 y penales <= 64.	Entonces= MEDIO (21.0/10.0)

Tabla 7. Resultados del algoritmo PART con el conjunto de prueba de datos numéricos.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	10	1	1	6	18	55.6	44.4
PORTERO	0	8	0	0	8	100	0
DEFENSA	1	1	21	3	26	80.8	19.2
MEDIO	3	1	2	15	21	71.4	28.6
	Matriz de confusión 54 en la diágnonal				73	73.97	26.03

Tabla 8. Resultados del algoritmo C4.5 con el conjunto de entrenamiento de datos nominales.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	70	1	9	21	101	69.3	30.7
PORTERO	4	42	4	2	52	81	19
DEFENSA	14	6	95	15	130	73.1	26.9
MEDIO	9	11	23	114	157	72.6	27.4
	Matriz de confusión 321 en la diágnonal				440	72.95	27.05

Se creó una base de datos con los valores nominales y se utilizó el algoritmo C4.5, sobre el conjunto de entrenamiento, dando como resultado un 72.95%. Hay un porcentaje parecido de instancias clasificadas correctamente que las obtenidas por los valores numéricos. La matriz de confusión del entrenamiento de datos nominales con el algoritmo C4.5 se puede observar en la tabla 8.

El árbol cambia un poco con relación al creado con los valores numéricos, es mucho más amplio y se evalúan diversos atributos para la clasificación. En este caso toma como nodo raíz la habilidad Estirada.

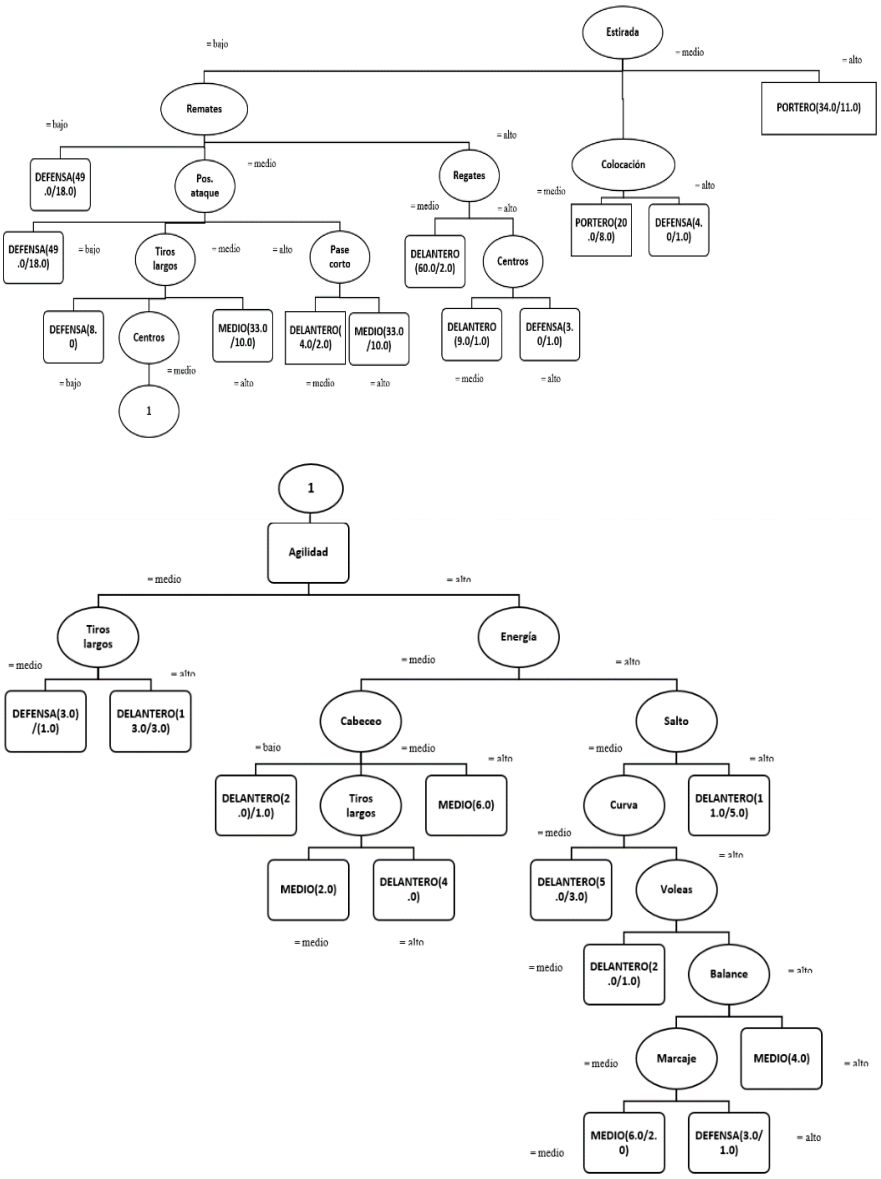


Fig. 4. Árbol de decisión con datos nominales.

Al realizar el proceso de prueba con los datos de los jugadores del torneo pasado se encontraron los siguientes resultados: 50 jugadores que pertenecen al 68.49%, fueron clasificados de forma correcta, los cuales pueden observarse en la matriz de confusión en la tabla 9.

En la cual si su valor es bajo se compara los remates, si es medio, compara la colocación, y si esta es alta se determina automáticamente que es un portero.

Tabla 9. Resultados del algoritmo C4.5 con el conjunto de prueba utilizando datos nominales.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	13	0	2	3	18	72.2	27.8
PORTERO	1	6	1	0	8	75	25
DEFENSA	0	5	19	2	26	73.1	26.9
MEDIO	2	1	6	12	21	57.1	42.9
	Matriz de confusión 50 en la diágnonal				73	68.49	31.51

Tabla 10. Resultados del algoritmo PART con el conjunto de entrenamiento datos nominales.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	63	1	13	24	101	62.4	37.6
PORTERO	4	41	5	2	52	79	21
DEFENSA	6	6	105	13	130	80.8	19.2
MEDIO	10	11	21	115	157	73.2	26.8
	Matriz de confusión 324 en la diágnonal				440	73.64	26.36

Tabla 11. Conjunto de reglas con valores nominales utilizando el algoritmo PART.

Si retén del balón = alto y pase corto = bajo y estirada = alto y fuerza = alto.	Entonces= PORTERO (14.0/5.0)
Si colocación = alto y pase corto = bajo y fuerza = medio y salto = medio y reacciones = alto.	Entonces= PORTERO (8.0/2.0)
Si regates = medio y entrada = alto y visión = bajo y vel. sprint = medio y agresividad = alto.	Entonces= DEFENSA (13.0)
Si barrida = alto y pase corto = medio.	Entonces= DEFENSA (27.0/10.0)
Si intercepción = medio y barrida = alto.	Entonces= MEDIO (15.0/4.0)
Si control = medio y visión = medio y compostura = medio.	Entonces= DELANTERO (22.0/10.0)
Si barrida = alto y pase largo = medio.	Entonces= DEFENSA (18.0/6.0)
Si barrida = alto y marcaje = alto y compostura = medio y aceleración = medio.	Entonces= MEDIO (5.0)
Si barrida = alto y compostura = alto y barrida = alto y compostura = alto y balance = alto y penales = medio.	Entonces= MEDIO (14.0/5.0)
Si barrida = alto y compostura = alto y centros = alto.	Entonces= DEFENSA (14.0/5.0)
Si agresividad = medio y barrida = bajo y compostura = alto y marcaje = bajo.	Entonces= DELANTERO (12.0/2.0)
Si agresividad = medio y agilidad = alto y barrida = medio.	Entonces= MEDIO (21.0/8.0)

Tabla 12. Resultados del algoritmo PART con el conjunto de prueba de datos nominales.

CLASIFICACIÓN	DELANTERO	PORTERO	DEFENSA	MEDIO	TOTAL	ACIERTO %	ERROR %
DELANTERO	13	0	2	3	18	72.2	27.8
PORTERO	1	6	0	1	8	75	25
DEFENSA	1	3	21	1	26	80.8	19.2
MEDIO	2	0	7	12	21	57.1	42.9
	Matriz de confusión 52 en la diágnonal				73	71.23	28.77

El atributo colocación es muy parecido al árbol generado con datos numéricos, ya que los resultados son los mismos, se puede determinar que es defensa o es portero (ver figura 2). Esta es sólo una de las hojas donde se puede analizar las habilidades para determinar la posición de juego. Este árbol se puede observar en la figura 4.

Al igual que con los datos numéricos, se crearon reglas de clasificación utilizando el algoritmo PART, al realizar el entrenamiento mostró un 73.64% de acierto, dando un total de 50 reglas. La matriz de confusión se muestra en la tabla 10 y las reglas que más destacan se encuentran en la tabla 11.

Al realizar la fase de pruebas con los datos de los jugadores del torneo pasado se obtuvo un 71.23% de instancias clasificadas correctamente. Este resultado junto con la matriz de confusión pueden observarse en la tabla 12.

6. Conclusiones y trabajo futuro

El futbol es un deporte en donde se ven inmersas habilidades tanto deportivas, mentales como físicas, las cuales cada jugador tiene que desarrollar y mejorar día con día. En este trabajo se presentaron dos modelos, uno utiliza árboles de decisión y el otro reglas de clasificación para determinar la posición en la que un jugador de futbol soccer tendrá posiblemente, un mejor desempeño con base en las habilidades que tiene o demuestra.

Este modelo puede ser de gran ayuda si es utilizada en pruebas o visorias, así como en los entrenamientos en los equipos tanto profesionales como amateur. Se presentaron los modelos con datos numéricos y nominales, que pueden usarse dependiendo de la precisión de la información que se tenga al momento de clasificar al jugador.

En el caso de los datos numéricos se pudo observar que se tiene una mayor certeza con el algoritmo C4.5 teniendo un 71.59% de instancias clasificadas correctamente al realizar el entrenamiento, y un 79.45% en la fase de pruebas, además el árbol de decisión es relativamente corto y tiene mejor definidas las clases, mientras que el algoritmo PART con un 73.64% de certeza en la fase de entrenamiento y un 71.23%, trabaja mejor con datos nominales donde sus reglas son cortas y definen que tan bueno debe ser un jugador en ciertas habilidades para determinar su posición. Ambos algoritmos muestran que un portero sobresale por tener una mejor colocación y reflejos, mientras que la defensa se caracteriza por una mejor compostura y tener mejores tiros largos. La posición medio tiene una mejor visión del campo y retención del balón. Finalmente los delanteros puede observarse que su fortaleza es la aceleración, el cabeceo y el salto.

Se considera que se cumplieron los objetivos planteados, al poder generar un clasificador que permita a un entrenador o director técnico apoyarse para definir la posición de juego de un jugador dándole certeza de las funciones que debe realizar, evitando con ello el tener que probarlo en todas las posiciones tratando de encontrar la mejor para éste.

Se plantea como trabajo futuro mejorar el porcentaje de precisión de la clasificación de jugadores, ya sea tomando en cuenta otros factores tales como la complexión física de estos, el rendimiento que pueda llegar a tener en los entrenamientos, así como el experimentar con otros algoritmos de minería de datos, o en su defecto con otras técnicas de inteligencia artificial.

Referencias

1. Jover, F.: Hábitos de entrenamiento y lesiones deportivas en la selección murciana de baloncesto 2007. Revista Internacional de Medicina y Ciencias de la Actividad Física y el Deporte, 8(1), pp. 146–160 (2008)

2. Huang, K., Chang, W.: A neural network method for prediction of 2006 world cup football game. In: 2010 International Joint Conference on Neural Networks (IJCNN), IEEE World Congress on Computational Intelligence, 1, pp. 1–8 (2010)
3. WEKA: http://www.cs.waikato.ac.nz/ml/weka/ (2018)
4. Fayyad, U., Piatetsky-Shapiro, G., Smith, P.: From data mining to Knowledge Discovery and Data Mining. A.I Magazine, 17(1), pp. 37–54 (1996)
5. FIFA: http://es.fifa.com/ (2018)
6. FMF: http://www.femexfut.org.mx (2018)
7. Ruiz, R.: Minería de datos como soporte a la toma de decisiones empresariales en una arquitectura SOA, Barranquilla. Ed. Coruniamericana, 1(1), pp. 25–34 (2013)
8. Medina, J., Sandi-Pinheiro, M., Andux, C.: Evaluación del rendimiento de los voleibolistas mediante minería de datos. Revista Ingeniería industrial, CUJAE, 26(2), pp. 47–52 (2005)
9. Sanhueza, R.: Utilización de Naive Bayes para predicción de victorias en jugadores de la Asociación de Tenistas Profesionales. Memorias de pregrado Ingeniería civil en computación, Universidad de Talca, Chile (2014)
10. Pérez, F.: Sistema de predicción de apuestas deportivas: una aproximación a la Quiniela. Tesis de grado, Universidad Carlos III, De Madrid, España (2014)
11. Quinlan, J., Kumar, V., Wu, X.: The Top 10 algorithms in data mining. 14(1), pp. 1–37 (2008)
12. Berry, M., Gordon, L.: Data mining Techniques. Canada: Wiley Computer Publishing (2004)
13. Ramos, M.: Estudios en finanzas y contabilidad: España y América Latina. Estado del arte y las nuevas metodologías aplicadas, 15, pp. 328–351, Madrid, España (2014)
14. Robles, Y., Sotolongo, A.: Integración de los algoritmos de minería de datos 1R, PRISM e ID3 a PostgreSQL. Journal of Information Systems and Technology Management, 10(1) (2013)

ISSN 1870-4069

Caso de estudio de análisis de sentimientos en Twitter: Tratado de libre comercio de América del Norte

Diego Aguilar, Grigori Sidorov, Ildar Batyrshin

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

diego.aguilar.m15@gmail.com, sidorov@cic.ipn.mx, batyr1@gmail.com

Resumen. En este artículo se implementan técnicas de minería de texto, procesamiento del lenguaje natural y aprendizaje automático para realizar una análisis de sentimientos en Twitter; utilizando para ello el tema central del Tratado de libre comercio de América del Norte. Se muestran los resultados de aplicar diferentes algoritmos de aprendizaje automático; donde se concluye que al usar en conjunto los algoritmos se obtiene el mejor rendimiento con el 87.87 % de exactitud. Se implementó una aplicación en tiempo real para el análisis de sentimientos en Twitter.

Palabras clave: Análisis de sentimientos, minería de texto, aprendizaje automático, Twitter, Tlcan.

Case Study of Sentiment Analysis in Twitter: North American Free Trade Agreement

Abstract. This paper text mining techniques, natural language processing and automatic learning are implemented in order to develop a sentiment analysis on tweets with the topic of North American Free Trade Agreement (NAFTA). The results of applying different machine learning algorithms are shown; where it is concluded that the conjuction of all algorithms obtains the best performance with the 87.87 % of precision. A real time application was implemented for the analysis of feelings on Twitter.

Keywords: Sentiment analysis, text mining, machine learning, Twitter, Nafta.

1. Introducción

Recientemente se ha incrementado el interés por la extracción de conocimiento de manera automática en grandes bancos de datos en los campos de economía, ciencias, salud, etc.

Además actualmente se cuenta con un gran flujo de información en redes sociales y esta es una gran oportunidad para desarrollar algoritmos que en tiempo real generen información útil del mundo que nos rodea.

1.1. Tratado de Libre Comercio de América del Norte

El tratado de libre comercio de América del Norte (TLCAN) es un acuerdo entre Canadá, Estados Unidos y México para crear una zona en donde se reduzcan los costos de comercio e intercambio de bienes entre los países involucrados.

Originalmente el pacto lo comprendían Canadá y Estados Unidos; sin embargo a principios de los noventa se anexó a México al TLCAN iniciando así una nueva era de acurdos diplomáticos entre los tres países. El TLCAN tiene como principales objetivos:

- Facilitar el comercio de bienes y servicios,
- Promover competencia económica,
- Impulsar la inversión entre los tres países,
- Proteger los derechos de propiedad intelectual,
- Establecer una cooperación trilateral para mejorar los beneficios del tratado.

El TLCAN ha sido motivo de discusiones en torno al beneficio que ofrece a sus partes, en donde hay opiniones encontradas las cuales lo apoyan [3] o bien indican que ha traído dificultades políticas y económicas a los países involucrados [2].

TLCAN en la actualidad Durante la campaña presidencial de Donald Trump se anunció que se buscaría una renegociación del TLCAN y en caso de no lograr un acuerdo satisfactorio su gobierno abandonaría el tratado. Esto provocó discusiones y desacuerdos entre los países cuando Trump se convirtió en presidente de Estados Unidos.

Lo anterior hace que las negociaciones del TLCAN y su estudio en redes sociales generan información valiosa respecto al sentimiento general que se tiene sobre el tema.

El resto del artículo esta comprendido como sigue: La Sección 2 muestra el estado del arte de los trabajos relacionados al análisis de sentimientos en Twitter. La descripción de los métodos de minería de texto, procesamiento de lenguaje natural y algoritmos de aprendizaje automático son descritos en la Sección3. Los resultados obtenidos se muestran en la Sección 4 respectivamente. Finalmente, en la Sección 5 son presentadas las conclusiones y el trabajo a futuro.

2. Trabajos relacionados

Como se menciona en la Sección 1 el interés por parte de la comunidad científica para el análisis de textos ha aumentado. No obstante la mayor parte de las investigaciones realizadas se enfocan en el idioma inglés, para los cuales se han alcanzado altos porcentajes de clasificación correcta y se han generado corpus de calidad [1].

2.1. Análisis de sentimientos en Twitter

Específicamente la plataforma de Twitter se ha utilizado ampliamente para tareas de agrupamiento, clasificación de textos, detección de tópico, etc.

En el taller TASS [7] del año 2012 se propuso un corpus de Tweets en español para realizar las tareas de análisis de sentimientos y clasificación de tópicos. El corpus contiene un total de 70000 tweets distribuidos en 10 temas y escritos por 200 conocidas personalidades de habla hispana. Durante el taller el porcentaje más alto para análisis de sentimientos obtuvo un 71.12 % de precisión utilizando tres clases (*pos, neg, neu*).

Antonio Fernández et al. [1] en su investigación realizaron una comparación de diversa técnicas de procesamiento de lenguaje natural (PLN) y aprendizaje automático (ML por sus siglas en inglés *machine learning*) para un corpus de Tweets en español con la finalidad de realizar tareas de identificación de tópicos y análisis de sentimientos.

Concluyeron que aunque la mayoría de los trabajos se centra en el idioma inglés, recientemente ha crecido la investigación en otros idiomas. Para la selección e tópicos la precisión más alta alcanzada fue 58 %, mientras un 42 % para clasificación de sentimientos. En su estudio utilizaron los algoritmos Ibk, Complement Naive Bayes, Naive Bayes Multinomial, Random Committe y SMO.

En el trabajo de Sidorov et al. [6] se realizó un estudio exhaustivo para análisis de sentimientos en Twitter, utilizando la herramienta WEKA [1] se comprobaron los resultados de clasificación de tweets en español para los algoritmos Naive Bayes, J48 y SVM. Como parte del preprocesamiento de datos utilizaron módulos para corrección de errores, POS-tagging y negación de sentimientos. Los autores concluyen que los parámetros ideales de configuración son: (1) utilizar unigramas como características, (2) usar solamente las clases *positivo* y *negativo*, (3) un corpus de al menos 3, 000 tweets para entrenamiento, (4) un corpus balanceado arrojó resultados peores que uno des-balanceado y (5) las SVM presentan la mejor clasificación.

3. Materiales y métodos

El panorama general para extraer información de sentimientos en tweets puede agruparse en cuatro pasos. Primero se debe reunir un corpus dedicado al tema a tratar (en este caso tweets relacionados al TLCAN); después se realizan métodos de PLN para limpiar obtener las características de los tweets. Finalmente, se entrenan los algoritmos de ML y se evalúan con criterios de medición estándar.

3.1. Corpus de tweets

Twitter es una plataforma de red social en donde los usuarios pueden expresar pensamiento o noticias en un párrafo breve (actualmente el límite es de 280

[1] Disponible en linea: https://www.cs.waikato.ac.nz/ml/weka/

caracteres). En el año 2018 Twitter cuenta con 330 millones de usuarios activos al mes con soporte para 40 idiomas [4].

Con el API de Twitter para desarrolladores obtuvimos un corpus con 5054 tweets relacionados al tema TLCAN, es decir que en su texto incluyeran "#TLCAN" para el periodo de Abril a Mayo del 2018. Los tweets fueron clasificados de forma manual (*Gold standard*) en una división de tres clases:

POS Ejemplo: "EU optimista ante TLCAN."
NEG Ejemplo: "No ha sido fácil negociar el TLCAN con México y Canadá: Trump."
NEU Los tweets neutrales son de carácter informativo o bien contienen una combinación de sentimientos positivos y negativos. Ejemplo: "Pese avances en TLCAN, no hubo luz en puntos críticos."

La Tabla 1 muestra la distribución de clase en el corpus; como se observa, las clases POS y NEG se encuentras más balanceadas en comparación con la clase NEU que predomina en el corpus.

Tabla 1. Distribución de clases en el corpus.

Clase	No. tweets	Porcentaje
POS	1,342	24.1 %
NEG	1,618	28.9 %
NEU	2,625	47 %
Total	5,585	100 %

3.2. Preprocesamiento

Cabe mencionar que la tarea de clasificación de sentimientos es difícil incluso para los humanos, en donde a veces no posible llegar a un acuerdo respecto a la polaridad del tweet [7].

Además los textos en tweets están sujetos a errores de ortografía, abreviaciones, modismos y humor o sarcasmo [6].

De forma similar al proceso seguido en [6]; utilizamos los siguientes métodos para preprocesar los tweets:

Corrección de errores: Para corregir algunos de los errores en el corpus se utilizó un diccionario en donde manualmente se introducen las correcciones ortográficas a los errores más comunes. Por ejemplo, los acentos en las preguntas "Qué, Cómo, Cuál, Dónde".

Además se eliminaron los símbolos extras como "(", "), ", "!", ".", ":", ";", etc.

Etiquetas especiales: Una característica de los tweets es la presencia de *hashtags (#)* los cuales generalmente son utilizados para etiquetas tweets en un tema en particular. Además, frecuentemente los tweets contienen URLs con direcciones a páginas de noticias o bien utilizan *emojis*.

Para lidiar con lo anterior se realiza una limpieza del texto. en un primer acercamiento se cambiaron los URL y emojis por etiquetas especiales (URL y $SMILEY$); no obstante, en un segundo enfoque se eliminaron por completo estas etiquetas de los tweets, ya que así se mejoraron los resultados de clasificación.

Por otra parte, los *hashtags* generalmente son escritos utilizando un modelo de *CamelCase* si son varias palabras juntas, e.g "#FelizJueves". Por lo cual se utilizó un método para eliminar el símbolo (#) y separarlos en palabras, de tal forma que el ejemplo anterior sería: "Feliz Jueves".

POS-tagging: Después de realizar una limpieza a los textos utilizamos el paquete *Natural Language Toolkit* (NLTK) de Python para el etiquetado de *Part of Speech*; una vez realizado este procedimiento obtenemos la clasificación de las palabras en: verbos, adjetivos, sustantivos, etc. De este conjunto de etiquetas realizamos un filtro para conservar solamente los adjetivos, adverbios y verbos.

Lematización: En un primer enfoque se utilizó la lematización de las palabras para reducir el número de características en el vector de entrada; no obstante, se obtuvieron mejores resultados al no utilizar este método por lo que se opto por no utilizarlo en la versión de tiempo real.

Negación de polaridad: Para el análisis de sentimientos es muy importante considerar la negación del sentido de la oración, es decir que el uso de ciertas palabras pueden cambiar el significado del texto. Particularmente, en este trabajo cuando se encuentra la palabra "no", por ejemplo en "no estoy feliz"; se utiliza el método descrito en [6] y se anexa a la palabra inmediata siguiente. Quedando así la oración anterior como: "no_estoy feliz".

Es importante destacar que existen otros métodos para combatir la negación, en [1] al encontrar la palabra "no" cambian la polaridad de las siguientes tres palabras. Además, existen adjetivos como "muy" o "poco" que acentúan o disminuyen el valor del sentimiento [5]. En este trabajo son omitidas este tipo de palabras.

3.3. Clasificadores

Usamos clasificadores implementados en las paqueteras NLTK y Sklearn de Python, dichos algoritmos son: Naive Bayes, gradiente descendente estocástico (SGDC), máquina de vectores de soporte (LinearSVC).

Los algoritmos utilizan de entrada un vector de características formadas por uni-gramas y filtradas usando POS-tagging.

Del corpus de tweets utilizamos el 80 % para entrenamiento y el 20 % para pruebas. Para estabilizar los resultados obtenidos por los clasificadores utilizamos el método *k-cross validation*.

3.4. Medidas de evaluación

Calculamos la exactitud de cada uno de los clasificadores en el set de pruebas con la Ecuación 1:

$$exactitud = \frac{\text{respuestas correctas}}{\text{total de respuestas}}. \quad (1)$$

Utilizamos la Ecuación 2 para calcular la exhaustividad de los clasificadores:

$$exhaustividad = \frac{\text{respuestas correctas}}{\text{total de posibles respuestas correctas}}. \quad (2)$$

Finalmente, la medida F1 utiliza los valores anteriores para mostrar un rendimiento general (Ecuación 3):

$$F1 = \frac{2 \times exactitud \times exhaustividad}{exactitud \times exhaustividad}. \quad (3)$$

3.5. Votación de clase

Para la clasificación de tweets en tiempo real utilizamos un votador, el cual clasifica el texto con las mejores versiones de los algoritmos de ML y calcula la moda de la clase pronosticada. Si existe un acuerdo entre los votadores con un umbral mayor al 70 % entonces se clasifica al tweet con el sentimiento acordado, en caso contrario, no existe un acuerdo entre los votantes y el tweet se deshecha.

4. Resultados

En esta sección se describen los resultados obtenidos del entrenamiento. La Tabla 2 muestra los mejores porcentajes de exactitud por algoritmo; también se muestra el resultado de combinar todos los clasificadores en un votador, cuya exactitud alcanzó un 87.8 % en el conjunto de pruebas.

Tabla 2. Resultados para algoritmos de entrenamiento.

Algoritmo	Exactitud	Exhaustividad	F1
Naive Bayes	65.86 %	65.92 %	65.76 %
Multinomial NB	66.40 %	64.87 %	65.44 %
C4.5	65.02 %	64.44 %	64.66 %
Bernoulli NB	66.81 %	64.41 %	62.90 %
Regresión logística	70.79 %	68.00 %	69.03 %
SGDC	68.22 %	67.12 %	67.22 %
LinearSVC	69.84 %	69.19 %	69.42 %
Votador	88.60 %	87.35 %	88.44 %

Utilizando el API de Twitter para realizar un streaming de datos es posible obtener en tiempo real todos los tweets relacionados al tema TLCAN. Después, son clasificados con el votador y si supera el umbral mencionado en la Sección 3.5 se guarda el resultado para finalmente visualizarlo en dos gráficas.

Análisis de sentimientos en el tiempo: La Figura 1 muestra la evolución del sentimiento general sobre el tema TLCAN. En el eje horizontal se muestra el tiempo en donde un tweet se publicó; mientras que en el eje vertical se muestra el valor del sentimiento, cuyo rango se encuentra en $[-1, 1]$

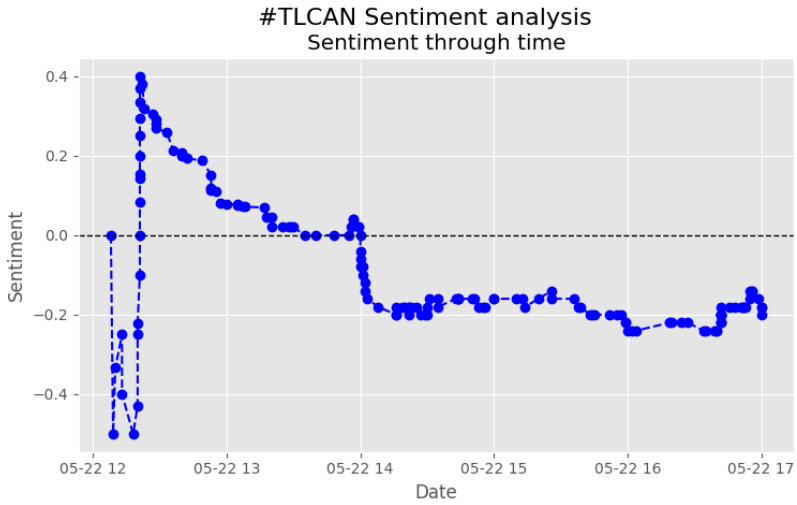


Fig. 1. Análisis de sentimientos a través del tiempo.

El valor de asociación es calculado a través de un promedio móvil simple. Si tenemos un vector $X = [x_1, x_2, \ldots, x_n]$ de tamaño n, donde cada x_i representa el valor de una clase de sentimiento, entonces su promedio móvil es calculado como se muestra en la Ecuación 4. Para la Figura 1 se utiliza $k = 50$:

$$sma_i = \frac{\sum_j^{i-1} x_j}{k}, \text{donde:} i > j. \tag{4}$$

Distribución del sentimiento: La Figura 2 muestra la distribución porcentual del sentimiento en tweets respecto al tema TLCAN. Como se observa, para este periodo de tiempo la mayoría de los tweet tenían cargado un sentimiento negativo; esto posiblemente se deba a que ocurrió un evento externo que afecto a la negociación del tratado (por ejemplo algún anuncio político del presidente de USA).

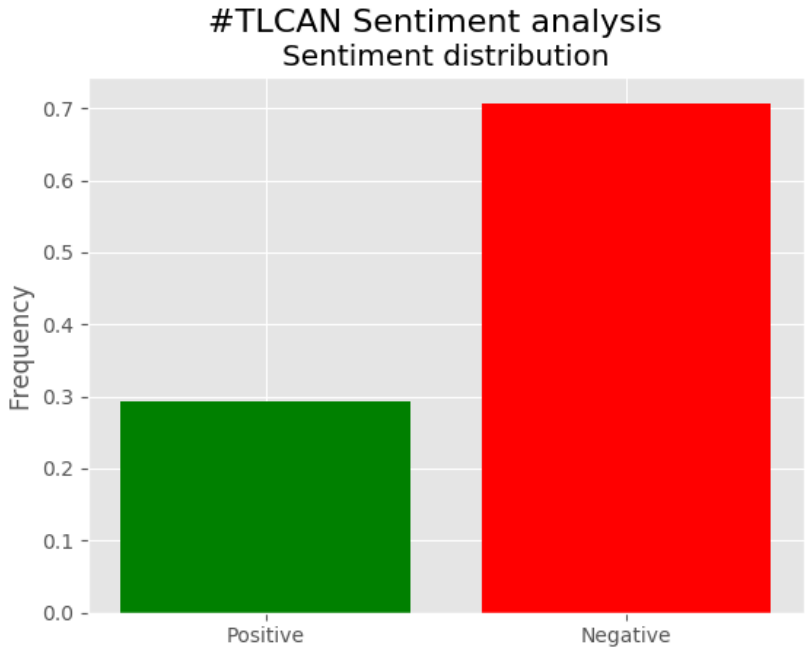


Fig. 2. Distribución porcentual de sentimientos, se oculta la etiqueta 'NEU' debido a que no aporta información para el sentimiento general sobre el tema.

5. Conclusiones y trabajo a futuro

En análisis de sentimientos en Twitter es un campo que promete mucho aún y es posible explotar las capacidades de lo algoritmos de ML para obtener información valiosa del mundo que nos rodea; a través del análisis de grandes cantidades de datos generados por los usuarios diariamente.

En el presente trabajo se presentó una solución para el análisis de sentimientos de tweets en español con el tema TLCAN. Utilizando diversos algoritmos de ML encontramos que se obtiene un mejor resultado combinando los clasificadores en una capa de votación, con una exactitud del 88.6 %. También creemos que la precisión de cada uno de los algoritmos puede mejorar si se utilizan características más sofisticadas y algunos métodos de preprocesamiento encontrados en el estado del arte.

También se presentó una aplicación para la clasificación y visualización de sentimientos en tiempo real, a través del streaming de tweets, usando el API proporcionado por Twitter. Estas aplicaciones pueden ser aprovechadas de diversas formas; por ejemplo, para analizar el efecto que tienen los eventos políticos o sociales sobre algún tema en específico.

Como trabajo a futuro proponemos utilizar otros temas políticos con su propio corpus de tweets, e.g opiniones de candidatos presidenciales en Twitter. Además, se puede diseñar un método que identifique eventos importante que afecten al sentimiento del tema tratado y a su vez poder visualizarlo en la gráfica. Finalmente, es posible utilizar diferentes métodos de visualización, así como calcular el valor general del sentimiento respecto al tema utilizando otras medidas.

Agradecimientos. Quisiera agradecer a mis compañeros Carolina Martín de Campo y Christian Maldonado por las sugerencias e ideas que aportaron al proyecto. Agradecemos el apoyo del proyecto CONACYT 240844 y el proyecto SIP 20181849.

Referencias

1. Anta, A.F., Chiroque, L.N., Morere, P., Santos, A.: Sentiment analysis and topic detection of Spanish tweets: A comparative study of of NLP techniques. Procesamiento del lenguaje natural 50, 45–52 (2013)
2. Buendia Rice, E.A.: Las promesas incumplidas del Tratado de Libre Comercio de América del Norte (TLCAN). Análisis Económico 29(72) (2014)
3. Moreno Brid, J.C., Rivas Valdivia, J.C., Ruiz Nápoles, P.: La economía mexicana después del TLCAN. Revista Galega de Economía 14(1-2) (2005)
4. Newberry, C.: 28 twitter statistics all marketers need to know in 2018. https://blog.hootsuite.com/twitter-statistics/ (2018)
5. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: Computing attitude and affect in text: Theory and applications, pp. 1–10. Springer (2006)
6. Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., Gordon, J.: Empirical study of machine learning based approach for opinion mining in tweets. In: Mexican international conference on Artificial intelligence. pp. 1–14. Springer (2012)
7. Villena Román, J., Lana Serrano, S., Martínez Cámara, E., González Cristóbal, J.C.: TASS workshop on sentiment analysis at SEPLN (2013)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2018
Printing 500 / Edición 500 ejemplares