

Mejorando el clasificador asociativo de patrones en bases de datos multi-clase

Roberto O. Cruz L., Raúl Santiago M. y Héctor J. Puga S.

Instituto Tecnológico de León, Av. Tecnológico, Industrial Julián de Obregón,
León, Guanajuato, 37000, México
{robanac,rsantiago66}@gmail.com,pugahector@yahoo.com

Resumen En el siguiente artículo se muestra la modificación del clasificador supervisado Clasificador Asociativo de Patrones (CAP) donde se implementa una modificación al proceso de traslación de ejes obteniendo una clasificación en cascada. Se pone a prueba la modificación y se clasifican distintas bases de datos multi-clase observando su comportamiento en distintos escenarios. En la interpretación de resultados se utiliza el método de validación K-fold y la prueba estadística no paramétrica de Wilcoxon, reforzando los resultados obtenidos. Con la modificación al algoritmo CAP se obtiene una mejora sustancial así como mejores rendimientos de clasificación.

Palabras clave. Memorias asociativas, rendimiento de clasificación, CAP, cascada.

1. Introducción

El área de Reconocimiento de Patrones es un campo donde existe una amplia variedad de técnicas que compiten entre sí por ofrecer los mejores resultados. Entre ellas las Redes Neuronales, Maquinas de Vector Soporte, Knn, Bayesiano y Mínima Distancia, por citar algunas. En este campo es donde nuestra propuesta de investigación se desenvuelve y utiliza el modelo de Memorias Asociativas en procesos de clasificación.

En [1] mencionan lo siguiente sobre las Memorias Asociativas aplicadas a la clasificación de patrones: *“In acts of associative memory, the system takes in a pattern and emits another patter which is representative of a general group of patterns. It thus reduces the information somewhat, but rarely to the extent that pattern classification does”*. Es entonces que las Memorias Asociativas son solo un sistema que toma un patrón y emite otro, el cual es generalmente representativo de un grupo de patrones ocasionando una reducción de la información, pero raramente en la medida de clasificar patrones.

En el año del 2002 [2] propone el primer clasificador supervisado basado en Memorias Asociativas llamado Clasificador Híbrido Asociativo con Traslación (CHAT) y mostró resultados promisorios de eficacia, compitiendo con otros clasificadores en el ámbito de clasificación de patrones más longevos. En el año 2003 los creadores de CHAT lo renombraron por Clasificador Asociativo de Patrones (CAP) [3].

En las pruebas desarrolladas en [4] se muestran rendimientos de clasificación mayores al 90% usando bases de datos (BD) bi-clase y se plasma un procedimiento multi-clase elitista de selección de patrones. Este procedimiento elitista no es común en procesos reales de clasificación y bajo escenarios no controlados de selección de los patrones.

En este trabajo se propone una extensión al CAP para aplicarlo a problemas de clasificación multi-clase. Esta extensión se ha denominado como *Técnica en Cascada*. La propuesta mejora el rendimiento de clasificación en BD multi-clase con respecto al modelo original del CAP y evita cualquier pre-procesamiento de los datos. La propuesta muestra un mejor rendimiento de clasificación sobre BD ampliamente utilizadas en pruebas de clasificación como la BD Iris Plant. Además se establece un marco comparativo con la versión original del CAP aplicando técnicas estadísticas.

Este artículo está organizado de la manera siguiente: En la sección 2 se muestra el funcionamiento general del CAP; En la sección 3 se muestra el diseño de la técnica Cascada y los elementos estructurales que la describen; En la sección 4 se pone a prueba la técnica en cascada usando distintas BD, se implementan técnicas de validación y selección de datos como *K-Fold Validation* y la prueba estadística no paramétrica de *Wilcoxon* para determinar qué algoritmo es mejor; y para finalizar, en la sección 5 se muestran algunas conclusiones.

2. CAP

Para comprender la mejora del algoritmo CAP para BD multi-clase con la técnica cascada, primero se ilustrará el funcionamiento del CAP y las ventajas que presenta este algoritmo. En [4] se consideran como principales características del clasificador los siguientes puntos:

- El CAP combina los principios de álgebra de matrices que utiliza el Linear Associator [5] y el criterio de clasificación de la Lernmatrix [6];
- Siendo una Memoria Asociativa que funciona como clasificador cuenta con las fases de aprendizaje y de recuperación;
- Para la fase de aprendizaje y posteriormente para la fase de recuperación se realiza una traslación de todos los patrones del conjunto fundamental empleando un vector medio, generado a partir de un conjunto fundamental;
- La posición del vector medio da origen a un nuevo conjunto de ejes coordenados;
- En la fase de recuperación, si se desea clasificar un patrón desconocido que no exista dentro del conjunto fundamental, es necesario trasladar al nuevo espacio de clasificación definido a través del vector medio.

En forma general, el algoritmo del CAP es el siguiente:

1. Sea un conjunto fundamental de patrones de entrada de dimensión n con valores reales en sus componentes (a la manera del Linear Associator), que se aglutinan en m clases diferentes.
2. A cada un de los patrones de entrada que pertenece a la clase k se le asigna el vector formado por ceros, excepto en la coordenada k -ésima, donde el valor es uno (a la manera de la Lernmatrix).

3. Se calcula el vector medio del conjunto de patrones muestra o prototipos con que se cuenta.
4. Se toman las coordenadas del vector medio a manera de centro de un nuevo conjunto de ejes coordenados.
5. Se realiza la traslación de todos los patrones del conjunto fundamental.
6. Aplicamos la fase de aprendizaje, que es similar a la del Linear Associator.
7. Aplicamos la fase de recuperación, que es similar a la que usa la Lernmatrix.
8. Trasladamos todo patrón a clasificar en los nuevos ejes.
9. Procedemos a clasificar los patrones desconocidos.

En la Figura 1 se muestra en a) una representación de dos clases y el vector medio que se genera bajo la propuesta del CAP. Este vector dará origen al nuevo espacio de clasificación, además se ilustra en b) el nuevo espacio de clasificación y los patrones trasladados con respecto al vector medio. Se observa como la traslación garantiza una separación entre las clases, intuitivamente es un efecto exclusivo para clasificación bi-clase.

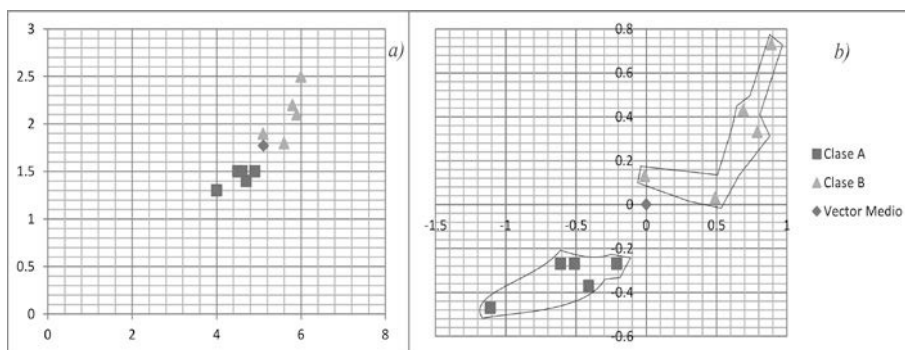


Fig. 1. Patrones simulados en dos clases y la traslación de los mismos (Fuente: Elaboración propia)

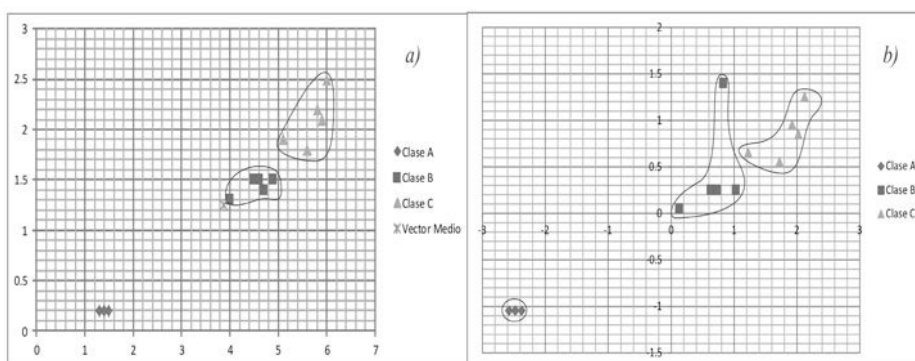


Fig. 2. Simulación de patrones en tres clases y la traslación de los mismos (Fuente: Elaboración propia)

En la Figura 2 se muestra en a) una representación de tres clases y el vector medio que se genera bajo la propuesta del CAP. Este vector dará origen al nuevo espacio de clasificación, además se ilustra en b) el nuevo espacio de clasificación y los patrones trasladados con respecto al nuevo origen.

Como puede observarse en el ejemplo de la Figura 2 para un problema multi-clase el CAP agrupa dos clases como si fuese una sola donde el vector medio no puede garantizar una separabilidad entre las clases. De forma general el CAP tiende a obtener bajos rendimientos en procesos de clasificación multi-clase.

3. Técnica cascada

Nuestra propuesta consiste en transformar un proceso multi-clase en múltiples procesos bi-clase. Los procesos de clasificación bi-clase dependen del total de clases con las que se cuente en la BD. La modificación al CAP tiene las siguientes consideraciones:

1. El número de niveles define la cantidad de bloques en los que se desarrolla el proceso de clasificación.
2. En cada bloque se obtiene un vector medio, que dará origen a un nuevo espacio de clasificación.
3. Como los vectores medios van cambiando, la matriz de entrenamiento tiene que ser calculada en cada bloque o nivel.
4. En cada bloque se obtiene un resultado de clasificación para una clase de la base de datos.

Por ejemplo, si una BD cuenta con cuatro clases el proceso de clasificación se desarrolla en cuatro etapas, cada etapa se encarga de clasificar todos los elementos que describen a una clase en particular, después de clasificar los primeros patrones se continúa con los restantes y así sucesivamente hasta terminar la clasificación de todas las clases involucradas en el planteamiento inicial. En la Figura 3 se muestra un esquema de nuestro planteamiento.

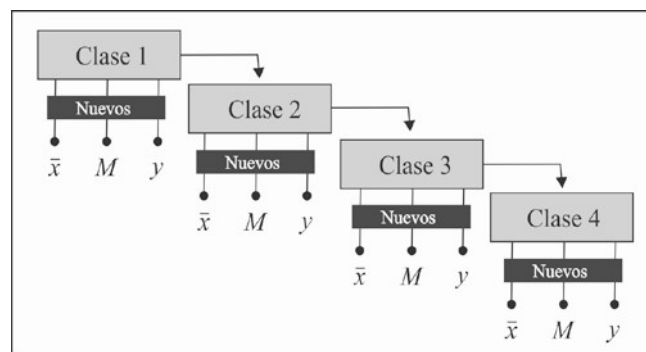


Fig. 3. Representación de Niveles de la Cascada donde \bar{x} representa cada vector medio, M representa la nueva matriz de entrenamiento y y el resultado de clasificación (Fuente: Elaboración propia)

Ya se menciona anteriormente que al obtener un vector medio en cada nivel o bloque se genera un nuevo proceso de clasificación bi-clase. En la siguiente sección se describe cómo se generan los vectores medios de nuestra propuesta.

3.1. Obtención de los vectores medios (VM)

El vector promedio de una clase se puede visualizar como un elemento representativo de dicho conjunto de vectores. Si tenemos m vectores que representan a una clase A el vector promedio \bar{x} se define de la siguiente manera

$$\bar{x} = \frac{\sum_{i=1}^m v_i}{m} \quad (1)$$

Siendo v_i un patrón en la clase A .

Bajo la propuesta del CAP, el vector medio obtenido genera un nuevo espacio de clasificación. Ahora:

Suponiendo que se tienen tres clases distintas A, B y C , el vector medio de estas no puede garantizar una separabilidad entre las clases. La propuesta de cascada genera artificialmente dos clases, una de ellas es la clase que se desea separar y la otra la aglutinación de las clases restantes. Si se desean clasificar los patrones correspondientes a la clase A , es necesario conocer un origen que de lugar a un espacio de clasificación. El nuevo origen del espacio de clasificación se realiza a través de la unión de las clases B y C (clases restantes) como una sola. A esta unión se le ha denominado *clase nube* y la clase que se está clasificando se le denomina *clase principal*. Después se calcula el vector promedio de la *clase nube*, ya no es necesario calcular el promedio con todos los vectores ya que es suficiente con obtener el promedio de los vectores medios \bar{x}_2 , de la clase B y \bar{x}_3 , de la clase C . Esto da origen a un vector medio del bloque que hemos definido como \bar{n} .

En la Figura 4 se muestra la *clase principal* A junto a su vector promedio \bar{x}_1 , además se muestra una representación de la *clase nube* BC y el vector promedio de la nube.

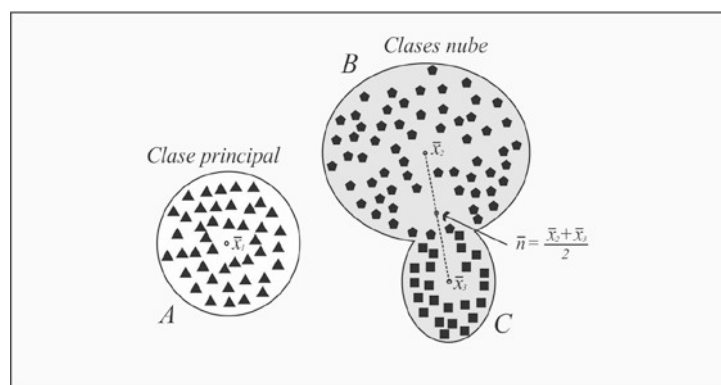


Fig. 4. Ejemplo de Clase principal y Clases nube, vectores promedio $\bar{x}_1, \bar{x}_2, \bar{x}_3$ y \bar{n} (Fuente: Elaboración propia)

Una vez obtenidos el vector promedio de la *clase principal* que en este caso es \bar{x}_1 y el vector promedio de las *clases nube* \bar{n} , el nuevo origen del conjunto de ejes coordenados o *vector medio* queda definido como

$$VM = \frac{\bar{x}_1 + \bar{n}}{2} \quad (2)$$

Como se puede ver en la ecuación 2 solo es necesario calcular el vector medio VM entre el vector promedio de la *clase principal* y el vector promedio de las *clases nube*.

En la Figura 5 se muestra el VM cuando la clasificación se realiza con la clase A .

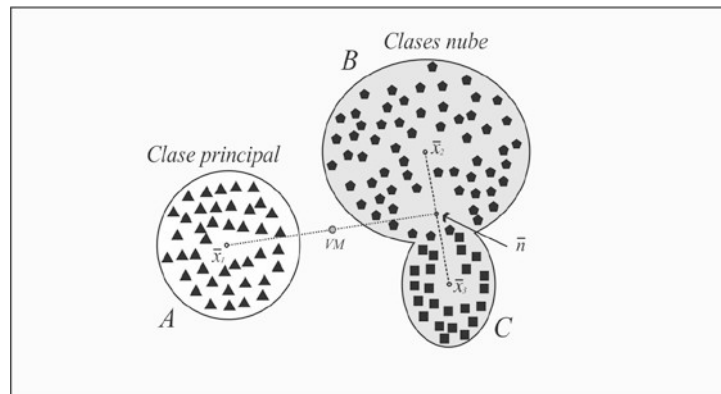


Fig. 5. Vector Medio en clasificación de clase A (Fuente: Elaboración propia)

Es fundamental señalar que todo el procedimiento de obtención de VM se desarrolla por clase, es decir, en su momento todas ellas llegan a ser *clase principal* y las restantes *clases nube*, obteniendo en cada bloque un nuevo VM .

3.2. Algoritmo CAP cascada

Algoritmo General de CAP Cascada

Se detalla el algoritmo general del CAP Cascada:

- Sea un conjunto de n clases distintas con m vectores cada una;
- Sea un conjunto fundamental de patrones de entrada de dimensión j con valores reales en sus componentes (a la manera del Linear Associator), que se aglutinan en n clases distintas;
- A cada uno de los patrones de entrada que pertenece a la clase k se le asigna el vector formado por ceros, excepto en la coordenada k -ésima, donde el valor es uno (a la manera de la Lernmatrix);
- Se calculan los vectores promedio de cada clase;
- Se implementa n veces el *algoritmo de clasificación por clase*;

Se unen los resultados de clasificación.

Algoritmo de Clasificación por Clase

Este procedimiento se realiza por cada clase:

- Se calcula el *vector medio* con respecto al vector promedio de la *clase primaria* y al vector promedio de las *clases nube*;
- Se toman las coordenadas del *vector medio* a manera de centro de un nuevo conjunto de ejes coordenados;
- Se realiza la traslación de todos los patrones del conjunto fundamental;
- Aplicamos la fase de aprendizaje, que es similar a la del Linear Associator;
- Aplicamos la fase de recuperación, que es similar a la que usa la Lernmatrix;
- Trasladamos todo patrón desconocido de la *clase primaria* a clasificar en los nuevos ejes;
- Procedemos a clasificar los patrones desconocidos;
- Se almacenan los resultados de clasificación.

4. Resultados

En esta sección es necesario mencionar cuáles son las bases de datos con las que se está poniendo a prueba el CAP y la propuesta CAP Cascada. Todas ellas se obtuvieron del UCI Machine Learning Repository de la universidad de California [7], este repositorio cuenta con alrededor de 1000 citas lo cual otorga a la fuente alta confiabilidad. Las bases de datos utilizadas son las siguientes:

Table 1. Características de las Bases de Datos.

BD	Instancias	Número Atributos	Clases	Año
Iris	150	4	3	1988
Wine	178	13	3	1991
Glass	214	9	6	1987
Statlog Project S. Vehículos	846	18	4	1986
Segmentation	210	19	7	1990

Las pruebas se desarrollaron en una PC genérica con procesador Intel Core 2 Quad a 2.40Ghz, 4 Gb de Ram, disco duro de 145 Gb, sistema operativo Windows 7 de Microsoft y la programación se realizó en lenguaje Java.

4.1. Comparativo entre CAP y CAP cascada

En esta sección se muestran los rendimientos de clasificación que se obtienen del CAP y CAP Cascada en cada una de las bases de datos descritas anteriormente, ambos clasificadores son sometidos en un mismo escenario de clasificación donde se implementa una validación cruzada de K-Fold [8] usando $K = 10$, este proceso se realiza 100 veces obteniendo un rendimiento promedio de cada clasificador en cada una de las bases de datos. Para determinar que clasificador obtiene mejor rendimiento

se implementa la prueba estadística no paramétrica de Wilcoxon con todos los promedios de rendimiento obtenidos.

En la Tabla 2 se muestran los porcentajes de rendimientos promedio de clasificación del CAP y CAP Cascada en cada una de las bases de datos

Table 2. Porcentajes de Rendimiento promedio de clasificación, usando K-Fold con $K = 10$ en 100 épocas de clasificación.

BD	CAP	CAP Cascada	Aumento de Porcentaje
Iris Plant	69.04	79.40	10.35
Wine	68.18	76.27	8.09
Glass	50.04	81.04	31.00
Statlog Project S. Vehiculos	38.52	62.08	23.55
Segmentation	62.21	73.97	11.75

En la Figura 6 se muestran gráficas de dos de los comportamientos de clasificación de la Tabla 2, donde las líneas rojas corresponden al CAP y las líneas azules al CAP Cascada

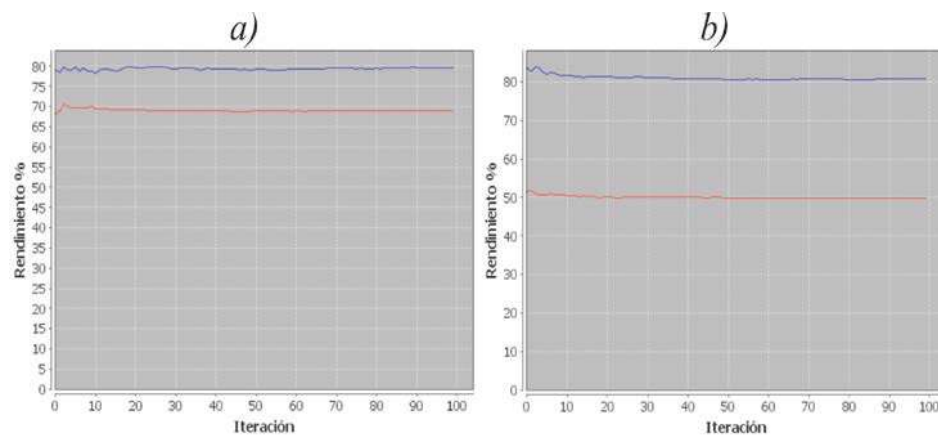


Fig. 6. a) Gráfica de rendimientos de clasificación en BD Iris Plant, b) Gráfica de rendimientos en BD Glass (Fuente: Elaboración propia)

4.2. Prueba Wilcoxon

La prueba estadística de Wilcoxon [9] es una herramienta matemática la cual determina si dos distribuciones son iguales, es considerada como una prueba no paramétrica ya que analiza experimentos que producen mediciones de respuesta que son difíciles de interpretar o cuantificar. Este método es bueno para resolver suposiciones generales acerca de las características de cualquier distribución de probabilidad o parámetros involucrados en un problema inferencial.

Esta prueba ayuda a determinar qué clasificador es *mejor*, para implementar la prueba es necesario definir una Hipótesis nula H_0 , la cual en todos los casos refiere a que las dos o más distribuciones comparadas son iguales. Para este ejemplo se espera que cuando la distribución de CAP y la distribución de CAP Cascada sean comparadas exista una diferencia notable entre ellas. De ser así, la hipótesis H_0 estaría siendo rechazada.

Para determinar qué clasificador es mejor se toman en cuenta los siguientes puntos:

- Es necesario tener como mínimo 5 instancias de prueba para la prueba básica;
- Iniciamos los parámetros $\alpha = 0.1$ y $T_{critico} = 1$ (parámetros definidos por Wilcoxon para un prueba de cinco instancias)[9];
- Los escenarios de clasificación deben ser los mismos, siendo los patrones de entrenamiento y de clasificación los mismos para CAP como para CAP Cascada, es por eso que utilizamos los datos obtenidos en la validación K-Fold mostrados en la Tabla 2;
- Definimos dos distribuciones $distX$ y $distY$ las cuales pertenecen a CAP y CAP Cascada respectivamente.

Los resultados de la prueba Wilcoxon son los siguientes:

Table 3. Cálculos de Wilcoxon con rendimientos de clasificación de CAP y CAP Cascada

BD	$distX$	$distY$	D	$D_{Absoluta}$	Rango
Iris Plant	69.04	79.40	-10.35	10.35	2
Wine	68.18	76.27	-8.09	8.09	1
Glass	50.04	81.04	-31.00	31.00	5
Statlog Project S. Vehículos	38.52	62.08	-23.55	23.55	4
Segmentation	62.21	73.97	-11.75	11.75	3

Después de los resultados mostrados en la Tabla 3 $T- = 15$ y $T+ = 0$, utilizando la literatura [9] se concluye que $T+ < T_{critico}$, por lo tanto la hipótesis nula H_0 se rechaza, y sabemos que la distribución $distY$ (CAP Cascada) se encuentra más hacia la derecha teniendo los valores más altos de ambas distribuciones y por lo tanto es mejor.

5. Conclusiones

Los resultados experimentales obtenidos en este trabajo muestran que el CAP tiene un menor rendimiento en clasificación multi-clase en comparación con procesos bi-clase, esto limita al algoritmo de clasificación para ser aplicado de manera general.

Nuestra propuesta de clasificación en cascada mejora sustancialmente al modelo original del CAP sin tener que utilizar un pre-procesamiento de los datos y siguiendo el mismo modelo. Finalmente la prueba de Wilcoxon valida que en el escenario de clasificación que se usa en este artículo, la propuesta cascada es superior al algoritmo CAP original.

Referencias

1. Duda,R.O.,Hart, P.E. & Stork,D.G.: Pattern Classification.Wiley,USA (2001)
2. Santiago Montero, R., Díaz-de-León Santiago, J.L. & Yáñez Márquez, C.: Clasificador híbrido de patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen. In Díaz-de-León Santiago, J.L. & Yáñez Márquez, C. (Eds.) "Reconocimiento de Patrones. Avances y Perspectivas", Special Edition,Vol. 1, ISBN 970-36-0194-4, CIC-IPN, México, pp. 449-460.Research on Computing Science Series (2002)
3. Santiago Montero, R., Díaz-de-León Santiago, J.L. & Yáñez Márquez, C.: Clasificador asociativo de patrones: avances teóricos. In Díaz de León-Santiago, J.L., González-Santos, G. & Figueroa-Nazuno, J. (Eds.) "Avances en: Ciencias de la Computación", Special Edition, Vol. 3, ISBN 970-36-0194-4, CIC-IPN, México, pp. 257-267.Research on Computing Science Series (2003)
4. Santiago Montero, R.: Clasificador Híbrido de Patrones basado en la Lernmatrix de Steinbuch y el linear Associator de Anderson-Kohonen. Tesis de Maestría, Instituto Politécnico Nacional-Centro de Investigación en Computación (2003)
5. Yáñez, C., Díaz de León, J.L.: Linear Associator de Anderson-Kohonen. Colección Memorias Asociativas, Instituto Politécnico Nacional-Centro de Investigación en Computación (2001)
6. Yáñez, C., Díaz de León, J.L.: Lernmatrix de Steinbuch. Colección Memorias Asociativas, Instituto Politécnico Nacional-Centro de Investigación en Computación (2001)
7. Frank, A. & Asuncion, A.: UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science (2010)
8. Kohavi, Ron.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Article, Computer Science Department-Stanford University (1995)
9. Mendenhall, W.: Estadística Matemática con Aplicaciones. Iberoamericana (1986)