ISSN 1870-4069

Reducción de dimensionalidad en microarreglos: GN y GA

Flor Alejandra Romero-Montiel, Katya Rodríguez-Vázquez

UNAM, Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, México
f.alejandra.r.m@gmail.com, katya.rodriguez@iimas.unam.mx

Resumen. Los microarreglos de DNA surgen de la necesidad de analizar la cantidad de información procedente de los grandes proyectos de secuenciación de genomas [1]. Permiten investigar el nivel de expresión de genes en una muestra. Usando esta información es posible diagnosticar y predecir enfermedades, por lo tanto, la tarea de clasificación de microarreglos de DNA es importante en bioinformática [2]. En este trabajo se mostrará como un neurón generalizado puede usarse en la tarea de clasificación de microarreglos. En la metodología propuesta se inicia por seleccionar un conjunto de genes usando un algoritmo genético, después el neurón generalizado es entrenado con un algoritmo genético. Finalmente la precisión de la metodología es probada clasificando tres bases de datos de microarreglos de DNA: *Leukemia benchmarck* $ALL - AML$, *Colon Tumor* y *Prostate cancer*.

Palabras clave: microarreglos, algoritmos géneticos, clasificación, redes neuronales, reconocimiento de patrones.

Reduction of Dimensionality in Microarrays: GN and GA

Abstract. DNA microarrays arise from the need to analyze the amount of information coming from large genome sequencing projects [1]. They allow to investigate the level of gene expression in a sample. Using this information it is possible to diagnose and predict diseases, therefore, the task of classifying DNA microarrays is important in bioinformatics [2]. This paper we will show how a generalized neuron can be used in the task of microarray classification. In the proposed methodology begins by selecting a set of genes using a genetic algorithm, then the generalized neuron is trained with a genetic algorithm. Finally, the accuracy of the methodology is tested by classifying three databases of DNA microarrays: *Leukemia benchmarck* $ALL - AML$, *Colon Tumor* and *Prostate cancer*

Keywords: microarrays, genetic algorithms, classification, neural networks, pattern recognition.

1. Introducción

Los microarreglos de DNA son utilizados para la cuantificación masiva de la expresión de genes. Este análisis permite diagnosticar enfermedades, identificar diferentes tumores, seleccionar para un paciente específico el mejor tratamiento para resistir una enfermedad [3]. Para obtener esta información se han aplicado algunas líneas de investigación de la Inteligencia Arficial (IA). Una rama de IA, llamada clasificación de patrones, consiste en la identificación de diferentes clases o grupos asociados con una enfermedad particular, por ejemplo tipos de cáncer o detección en un paciente. El microarreglo de DNA tiene una enorme cantidad de genes para analizar (del orden de los miles) y un pequeño número de muestras, esto es una desventaja cuando se utiliza una tarea de aprendizaje y reconocimiento, por lo que es un reto para los métodos de clasificación actuales [4].

Las redes neuronales artificiales (ANN por sus siglas en inglés) son famosos modelos computacionales usados para clasificación y otras tareas [5]. En los trabajos encontrados en la literatura se puede notar que comparten una característica: los autores buscan reducir la dimensionalidad de los genes, por que muchos de ellos son irrelevantes y la consecuencia de conservarlos es costosa, aumenta el tiempo de cómputo, la alta complejidad y el bajo rendimiento en la clasificación o predicción de una enfermedad [6].

La red neuronal generalizada o neurón generalizado (GN) es otro tipo de ANN que fue desarrollado con el objetivo de reducir el diseño de una red neuronal, pues no tiene muchas conexiones, con un buen rendimiento en comparación con el ANN clásico, además de ser fácil de implementar.

Algunos trabajos aplicaron GN para resolver funciones de aproximación, para calcular estimaciones de densidad , predicción y problemas de clasificación [7,8]. En este trabajo se analiza un problema de clasificación de microarreglos de DNA. La metodología consiste en reducir primero la dimensionalidad de los genes usando un algoritmo genético, luego clasificar los datos con un GN donde algunos parámetros son determinados mediante un algoritmo genético.

Este trabajo se divide en seis secciones: en la segunda se describe el concepto de neurón generalizado. A continuación se describe el algoritmo genético. Además la metodología propuesta se describe en la sección cuatro, seguido por los resultados experimentales en la sección cinco. Finalmente las conclusiones de este trabajo se encuentran en la sección seis.

2. Neurón generalizado

El neurón generalizado (GN) fue propuesto en [9], ha sido aplicado en problemas de clasificación y aproximación de funciones.

La estructura general del modelo son dos funciones de agregación (suma y producto) y dos funciones de transferencia (sigmoidal y gaussiana). La función sigmoidal característica (f_1) se usa con la función de suma $\sum_1$, mientras que la función gaussiana (f_2) se usa con la función producto $\prod$. La salida de la parte

$\sum_1$ con la función sigmoidal de activación es :

$$O_{\sum} = f_1(S_{net}) = \frac{1}{1 + exp(-\lambda_s * S_{net})}, \tag{1}$$

donde:

$$S_{net} = \sum W_{\sum i} X_i + X_{0\sum},$$

y λ_s, $X_{O\sum}$ son la ganancia y el sesgo de la parte $\sum$ respectivamente. La salida de la parte $\prod$ con la función de activación gaussiana para f_2 es:

$$O_{\prod} = f_2(P_{net}) = exp(-\lambda_p * P_{net}^2), \tag{2}$$

donde

$$P_{net} = \prod W_{\prod i} X_i * X_{O\prod},$$

y λ_p, $X_{O\prod}$ son la ganancia y el sesgo de la parte $\prod$, respectivamente. La salida final O_{pk} del neurón es una función de las dos salidas $O_{\sum}$ y $O_{\prod}$ con los pesos W y $1 - W$ respectivamente y se puede escribir como:

$$O_{GN} = W * O_{\sum} + (1 - W) * O_{\prod}. \tag{3}$$

Para problemas de múltiples salidas, diversos modelos de neurón generalizado en paralelo son requeridos. El número de pesos en el caso del neurón generalizado es el doble del número de entradas más dos pesos del sesgo, más dos pesos de la ganancia más un peso que corresponde a W (porcentaje de contribución de cada una de las partes de la estructura del neurón: suma y producto). Esto es mucho menos comparado con el número de pesos en una red multicapa [10]. Al reducir el número de pesos desconocidos el tiempo de entrenamiento se reduce.

3. Algoritmo genético

Los algoritmos genéticos (GA) son estrategias de búsqueda estocástica basados en el mecanismo de selección natural, imitando a la evolución biológica como estrategia para resolver problemas [11]. Los GA trabajan sobre un conjunto de potenciales soluciones, llamado *población*. Esta población está compuesta de una serie de soluciones llamadas *individuos* y un individuo está conformado por una serie de posiciones que representan cada una de las variables involucradas en los procesos de optimización y que son llamados *cromosomas*. Estos cromosomas están compuestos por una cadena de símbolos que en muchos casos está presentada en números binarios, aunque también es posible usar codificación hexadecimal, octal, real, etc.

En un GA cada individuo está definido como una estructura de datos que representa una posible solución del espacio de búsqueda del problema. Las estrategias de evolución trabajan sobre los individuos, que representan las soluciones del problema, por lo que estos *evolucionan* a través de *generaciones*. Dentro de

la población cada individuo es diferenciado de acuerdo con su valor de *aptitud* o $fitness$, que es obtenido usando algunas medidas de acuerdo con el problema a resolver. Para la obtención de las próximas generaciones se crean nuevos individuos, llamados $hijos$, utilizando dos estrategias de evolución básicas como son el operador de cruce y el de mutación.

En Algoritmo 1.1 se muestra el pseudocódigo del algoritmo genético simple.

Algorithm 1.1 Algoritmo Genético Simple.

```
S ← Generar una población inicial S
while no se alcance la condición de parada do
   evaluar cada individuo de la población S
  for i = 1 hasta tamaño (S)/2 do
    Seleccionar dos individuos de la generación anterior.
    Cruzar con cierta probabilidad los dos individuos obteniendo dos
    descendientes.
    Mutación de individuos.
    Insertar los dos descendientes mutados en la nueva generación.
  end for
end while
return la mejor solución encontrada.
```

4. Metodología para clasificar microarreglos de DNA

La metodología usada en este trabajo para desempeñar una tarea de clasificación binaria de microarreglos de DNA se divide en dos etapas:

- La primera dedicada a seleccionar el conjunto de genes que mejor describen el microarreglo de DNA.
- La segunda enfocada a entrenar un neurón generalizado para mejorar la precisión de la tarea de clasificación.

y la representación esquemática se puede observar en la figura 1. El primer paso propone una reducción de dimensionalidad del microarreglo de DNA, aminorando el número de características, con las cuales será entrenado el neurón generalizado. El segundo paso consiste en utilizar esta información para entrenar un neurón generalizado y realizar la clasificación. Estos dos pasos son repetidos hasta que el número máximo de iteraciones es alcanzado. Una descripción más detallada de las etapas se describen a continuación:

4.1. Reducción de dimensionalidad

De acuerdo con [12], la selección del conjunto con los mejores genes puede ser definido en términos de un problema de optimización. En este trabajo se usó un algoritmo genético, (*algoritmo genético* 2) para explorar el espacio de soluciones, los individuos son subconjuntos de genes del microarreglo de DNA de longitud

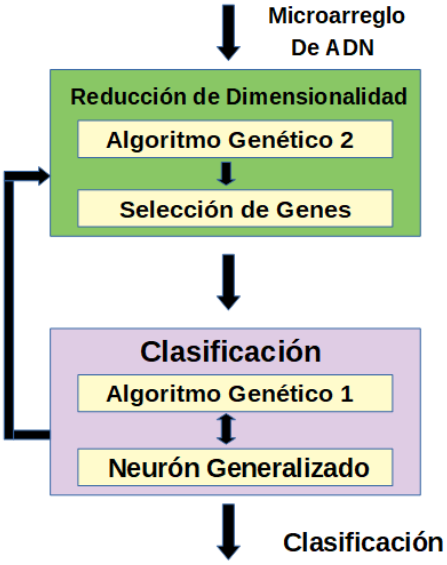

Fig. 1. Esquema de la metodología propuesta.

fija. La función fitness usada, es el mínimo número de elemento mal etiquetados al usar esos genes para entrenar un GN, es decir, con la base propuesta se entrenan distintos modelos de GN, cada uno de ellos con un resultado de clasificación distinto, tomando como valor de fitness la clasificación con menos errores. Un ejemplo de codificación es:

$$[19, 58, 7, 325, 205]. \tag{4}$$

Las entradas se encuentran entre 0 y el número de características que tiene la base original menos uno, indican los índices de las columnas que serán tomadas para formar una nueva base de datos. La cruza usada es, dados los individuos $Ind_1 = [I_{11}, I_{12}, ..., I_{1n}]$ y $Ind_2 = [I_{21}, I_{22}, ..., I_{2n}]$, se generan dos descendientes, $H_k = [h_{k1}, h_{k2}, ..., h_{kn}]$, $k = 1, 2$ donde:

$$H_1 = Ind_1 + \alpha * (Ind_2 - Ind_1), \tag{5}$$
$$H_2 = Ind_2 + \alpha * (Ind_1 - Ind_2), \tag{6}$$

con $\alpha \in [-0.25, 1.25]$. La mutación es un operador unario, dado un individuo $Ind_1 = [I_{11}, I_{12}, ..., I_{1n}]$ este tiene una probabilidad de mutar del 10 % (en la mayoría de los casos es baja), si el individuo muta, cada entrada I_{1k} tendrá la probabilidad $\frac{1}{n}$ de mutar, es decir:

$$Im_{1k} = I_{1k} + (\beta - 0.5) * (p_2 - p_1) * 0.1, \tag{7}$$

donde $\beta \in (0, 1)$, $p_2 = 4$ *y* $p_1 = -4$ después de aplicar estos operadores, se aplica la función piso a todos los individuos, comprobando en cada individuo que las

entradas son todas distintas entre sí, de no ser el caso se reemplazará con un aleatorio.

4.2. Clasificación usando un GN

Una vez propuesto el conjunto de características, estos genes formarán una nueva base de datos, que es particionada en dos conjuntos: entrenamiento y prueba. El conjunto de entrenamiento posee el 70 % y el conjunto de prueba el 30 % restante. Esta partición se realiza de forma aleatoria para asegurar que los conjuntos contengan elementos con ambas etiquetas. Después el GN es entrenado con un algoritmo genético, para términos prácticos llamado *algoritmo genético* 1, con codificación real. Las soluciones (individuos), generados con el algoritmo genético 1, codifican la estructura del GN en términos de los pesos sinápticos $(W_\Sigma, W_\prod, W)$, sesgo y parámetros de la función de activación (λ) para cada tipo de neurona $(\Sigma\ y\ \prod)$. Un ejemplo de codificación es :

$$[0.98, 2.12, -3.45, -0.34, 1.52, \\ 0.872, -3.369, 2.548, -0.125, 1.417, \\ -2.834, 0.723, -1.396, 2.196, -3.592] \quad (8)$$

Las entradas se encuentran en el intervalo $(-4, 4)$, si el número de características en la base es 5, la longitud de los individuos es de $(2 * 5) + 5$. La cruza y mutación son las mencionadas anteriormente.

La función escalón es aplicada a la salida del GN para determinar la clase a la que pertenece. En la figura 2 se muestra la gráfica de dicha función. La función fitness está basada en el número de elementos mal etiquetados, se obtiene la matriz de confusión comparando las etiquetas reales con las etiquetas propuestas, el número de falsos negativos y falsos positivos es el valor de la función fitness que se busca minimizar. Una vez entrenado el GN, se procede a evaluar la capacidad de generalización usando el conjunto de prueba, utilizando las métricas de precisión, recall y f1-score.

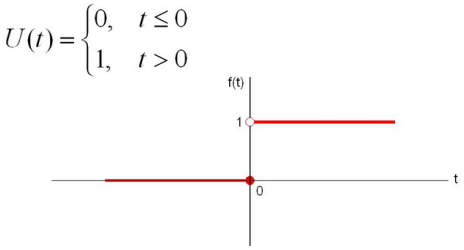

Fig. 2. Gráfica de la función escalón de Heaviside.

5. Resultados experimentales

En esta sección, se analizarán los resultados obtenidos con la metodología propuesta para determinar la precisión, usando tres bases de microarreglos distintas. En todos los casos, para seleccionar el mejor conjunto de genes usando el algoritmo genético 2, con codificación entera, se definieron los siguientes parámetros: tamaño de la población = 30 y máximo número de ciclos= 30, selección por torneo, probabilidad de cruza= 100 %, probabilidad de mutación del individuo = 10 %, probabilidad de mutación de cada gen= $1/(número\ de\ genes)$ y elitismo. En los casos donde la base de datos se encuentra dividida en datos de entrenamiento y prueba, se reagrupa y se realiza una partición aleatoria, 70 % para entrenamiento y 30 % para prueba.

- La metodología fue aplicada para clasificar dos tipos de cáncer: la leucemia linfocítica aguda y la leucemia mieloide aguda, en la base de datos *Leukemia* $ALL - AML$.
  Leukemia benchmark $ALL - AML$ contiene las medidas correspondientes a muestras de ALL y AML de médula ósea y sangre periférica. Originalmente se compone de 38 muestras para entrenamiento (27 ALL *y* 11 AML) y 34 muestras para prueba (20 ALL *y* 14 AML) donde cada muestra contiene información de 7129 expresiones génicas, al realizar la partición se tienen 50 muestras en el conjunto de entrenamiento y 22 en el conjunto de prueba. En la tabla 1 se muestran algunos de los resultados obtenidos.

Tabla 1. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
3	0.76	0.96	0.62	0.93	0.62	0.94
15	0.91	0.96	0.86	0.91	0.88	0.94
20	0.87	0.92	0.84	0.89	0.85	0.90
25	0.90	0.94	0.92	0.95	0.91	0.95
30	0.80	0.93	0.71	0.95	0.77	0.95

  Después de concluir los experimentos, se puede notar que el mejor resultado obtenido fue al usar 15 genes, obteniendo un precisión del 96 % en el conjunto de prueba.
- La base de datos *colon Tumor* contiene 62 muestras, de las cuales 22 son positivas para Tumor en colon y 40 negativas. Al realizar la partición se tienen 43 muestras para el entrenamiento y 19 para prueba. Cada muestra contiene 2000 genes. En la tabla 2 se muestran algunos de los resultados obtenidos.

Tabla 2. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
10	0.82	0.95	0.81	0.88	0.81	0.91
15	0.82	0.95	0.87	0.85	0.81	0.91
20	0.83	1.0	0.87	1.0	0.85	1.0
30	0.94	0.93	0.96	0.86	0.95	0.87

Después de concluir los experimentos, se observa que los mejores resultados se obtuvieron al usar 20 genes, obteniendo una precisión de 83 % en el conjunto de entrenamiento y 100 % en el conjunto de prueba.

- La base *Prostate cancer* se compone de 102 muestras para entrenamiento (52 con tumor de próstata y 50 con no-tumor de próstata "normal") El conjunto de prueba contiene 25 muestras con tumor de próstata y 9 normal, cada muestra contiene 12,600 genes. Al realizar la partición se tienen 95 muestras para el entrenamiento y 41 para prueba.En la tabla 3 se muestran algunos de los resultados obtenidos.

Tabla 3. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
10	0.74	0.78	0.75	0.78	0.74	0.78
20	0.80	0.86	0.79	0.85	0.79	0.85
30	0.86	0.87	0.87	0.89	0.86	0.87
40	0.94	0.90	0.95	0.90	0.94	0.90
50	0.84	0.89	0.83	0.89	0.83	0.89

Después de concluir los experimentos, se observa que los mejores resultados se obtuvieron al usar 40 genes, obteniendo una precisión de 94 % en el conjunto de entrenamiento y 90 % en el conjunto de prueba.

6. Conclusiones

Los diversos experimentos permiten determinar el comportamiento de la metodología propuesta en la clasificación de microarreglos de DNA. Durante la primera etapa, se aplicó con éxito una reducción de dimensionalidad sobre los conjuntos de datos *Leukemia benchmarck* $ALL-AML$, $Colon\ Tumor$ y *Prostate cancer* para seleccionar el conjunto de genes que mejor describe una enfermedad en particular, utilizando un algoritmo genético. El problema de reducción de dimensionalidad puede tratarse como un problema de optimización, debido a que la disminución dimensional de un microarreglo de DNA puede verse como un

problema combinatorio que trata de encontrar entre millones de genes los más relevantes.

Los resultados obtenidos utilizaron en algunos casos menos del uno por ciento de los genes para realizar una tarea de detección o clasificación.
En la segunda etapa, se evaluó el desempeño del GN, los resultados obtenidos mostraron que todo el conjunto de datos se resolvió con una buena precisión, por lo que el algoritmo genético es una buena técnica para entrenar un GN. Estos GN fueron entrenados usando el conjunto de genes propuestos en la primera etapa.

Finalmente, se puede concluir que el GN entrenado con la metodología propuesta es capaz de detectar, predecir y clasificar una enfermedad con una precisión aceptable.

Trabajos futuros incluirán la comparación del desempeño del neurón generalizado con otras arquitecturas de ANN.

Agradecimientos. Se agradece al posgrado de Ciencia e Ingeniería de la Computación de la UNAM, así como al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado.

Referencias

1. López, M., Mayorquin, P., Vega, M.: Aplicación de los microarrays y biochips en salud humana. Madrid: Fundación Española para el Desarrollo de la Investigación en Genómica y Proteomica (2005)
2. Garro, B. A., Rodríguez, K., Vazquez, R.: Generalized neurons and its application in DNA microarray classification. In: Evolutionary Computation (CEC), 2016 IEEE Congress on, pp. 3110–3115, IEEE (2016)
3. Illana-Rico, E.: Análisis bioinformático de datos de expresión genética obtenidos mediante tecnología de microarrays. Jaén: Universidad de Jaén (2018)
4. Cano Gutiérrez, C.: Extracción de conocimiento de microarrays y literatura biomédica para el estudio de la regulación genética. Tesis doctoral], Granada: Editorial de la Universidad de Granada (2010)
5. Huynh, Hieu Trung, Kim, Jung-Ja, Won, Yonggwan: Classification study on DNA micro array with feed forward neural network trained by singular value decomposition. International Journal of Bio-Science and Bio-Technology, Vol. 1, pp. 17–24 (2009)
6. Plous, C.: Microarreglos de ADN y sus aplicaciones en investigaciones biomédicas. Revista CENIC. Ciencias Biológicas, Núm. 2, pp. 132–135 (2007)
7. Rizwan, M., Jamil, M., Kothari, D.P.: Generalized neural network approach for global solar energy estimation in India. IEEE Transactions on Sustainable Energy, Vol. 3, pp. 576–584, IEEE (2012)
8. Kiran, R., Jetti, S. R., Venayagamoorthy, G. K.: Online training of a generalized neuron with particle swarm optimization. In: Neural Networks (IJCNN'06), International Joint Conference on, pp. 5088–5095, IEEE (2006)
9. Kulkarni, R., Venayagamoorthy, G. K.: Generalized neuron: Feedforward and recurrent architectures. Neural Networks, Vol. 22, pp. 1011–1017 (2009)

10. Peterson, L. E., Ozen, M., Erdem, H., Amini, A., Gomez, L., Nelson, C., Ittmann, M.: Artificial neural network analysis of DNA microarray-based prostate cancer recurrence. In: Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB'05), Proceedings of the 2005 IEEE Symposium on, pp. 1–8, IEEE (2005)
11. Arranz de la Peña, J., Parra Truyol, A.: Algoritmos genéticos. Vol. 20, pp. 06–07 (2007)
12. Garro, B., Rodríguez, K., Vázquez, R.: Classification of DNA microarrays using artificial neural networks and ABC algorithm. Applied Soft Computing, Vol. 38, pp. 548–560 (2016)