

Modelo de clasificación supervisada usando la función de Heaviside

Andrés García Floriano

Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Unidad de Informática,
México

agarciafl@ipn.mx

Resumen. En este artículo presentamos los fundamentos teóricos de un nuevo modelo de clasificación que se basa en el enfoque asociativo de reconocimiento de patrones: clasificador de Heaviside. Como su nombre indica, sus dos fases, aprendizaje y clasificación, se basan en la función de Heaviside. La efectividad del modelo propuesto puede verificarse mediante los resultados de un estudio comparativo en el que el clasificador se probó con otros siete modelos de reconocimiento de patrones en 20 conjuntos de datos. Los resultados experimentales indican que el modelo es competitivo en el estado de la técnica. Es de destacar que con un conjunto de datos, nuestro clasificador logró el 100% del rendimiento, validado con *10-fold cross validation*, mientras que en su peor rendimiento logró un poco más del 50%. Los resultados obtenidos fueron validados por la prueba no paramétrica de Wilcoxon, que proporciona certeza estadística a los resultados de la comparación de rendimiento entre modelos.

Palabras clave. Clasificación de patrones, aprendizaje supervisado, función de Heaviside, prueba estadística no paramétrica.

Supervised Classification Model Using the Heaviside Function

Abstract. In this paper, we discuss the theoretical foundations of a new classification model which is based on the associative approach of Pattern Recognition: Heaviside's Classifier. As its name suggests its both phases, learning and classification, are based on the Heaviside's function. The effectivity of the proposed model can be verified by the results of a comparative study where the classifier was tested against other seven pattern recognition models on 20 datasets. Experimental results indicate that the model is competitive in the state of the art. It is noteworthy that with one dataset, our classifier achieved the 100% of

performance, validated with 10-fold cross-validation, while in its worst performance it achieved a little above of 50%. The obtained results were validated by the Wilcoxon non parametric test, which provides statistical certainty to the results of the performance comparison between models.

Keywords. Pattern classification, supervised learning, Heaviside function, non parametric statistical test.

1. Introducción

Durante las últimas décadas del siglo XX surgió el interés por comprender cómo es que los seres humanos somos capaces de interactuar con nuestro entorno: somos capaces de reconocer nuestra habitación, podemos realizar secuencias de actividades, también podemos reconocer objetos y con base en lo que reconocemos podemos tomar decisiones y de esta forma podemos realizar acciones como seleccionar nuestra vestimenta o ir a algún lugar [1].

El reconocimiento de patrones (RP) es una ciencia que contempla cuatro tareas básicas: la recuperación, la clasificación de patrones, la regresión y el agrupamiento o *clustering*. La tarea de clasificación, en la cual se centra este artículo, está relacionada con la asignación de etiquetas a objetos que pertenecen a clases que son diferentes entre sí; en el caso más simple los patrones u objetos son mutuamente excluyentes en relación con las clases [2].

Es importante resaltar que el Reconocimiento de Patrones es una ciencia interdisciplinaria que se basa en conceptos de ciencias formales, como las matemáticas, para solucionar problemas

relacionados con ciencias como la medicina o las ciencias sociales. Por ello, un científico experto en el área del Reconocimiento de Patrones debe tener interacción con expertos de otras áreas [3].

En la actualidad, muchos de los avances relacionados con el reconocimiento de patrones difieren con el enfoque clásico propuesto por la Inteligencia Artificial, y en realidad se basan en un enfoque más reciente denominado Inteligencia Computacional o *Computational Intelligence*. A partir de las características definidas por su creador, James Bezdek, la Inteligencia Computacional es el conjunto de modelos computacionales y herramientas inteligentes capaces de aceptar datos numéricos, provenientes de sensores, para procesarlos de forma eficiente y por lo tanto generar respuestas confiables y rápidas y con alta tolerancia a los fallos [4, 5].

Durante los últimos 70 años ha surgido una gran cantidad de modelos destinados a resolver las cuatro tareas base del RP; estos modelos se agrupan en un conjunto de enfoques o superestructuras conceptuales, que a grandes rasgos están basados en métricas, en probabilidad, en modelos matemáticos de las neuronas humanas, en árboles de decisión e incluso en teorías de aprendizaje estadístico que fueron creadas exprofeso, entre otras [6].

Es importante mencionar que derivado de los trabajos mencionados anteriormente, surgieron un conjunto de modelos matemáticos orientados a facilitar la tarea de la recuperación de patrones. Estos modelos conocidos como Memorias Asociativas [7], han sido la base de lo que es prácticamente una nueva superestructura, de la cual han sido creados casi dos decenas de modelos originales, los cuales brindan comportamientos competitivos respecto a los modelos del estado del arte [8].

El modelo de clasificación que se resume en este documento, está basado en esta superestructura de las Memorias Asociativas; sólo que, a diferencia de otros modelos basados en esta superestructura, en este artículo se permite transformar los datos a cualquier sistema numérico posicional base b , donde b es un número entero mayor que 1 [9], así como el uso de la función de Heaviside [10]; como restricciones del modelo, se estableció que los patrones a procesar

únicamente deben estar formados por números mayores o iguales que 0.

El resto de este resumen se estructura de la siguiente forma: en la siguiente sección se detallarán los modelos pertenecientes a la superestructura asociativa del reconocimiento de patrones; posteriormente, en la sección 3 se mencionarán brevemente las herramientas matemáticas empleadas, para finalmente presentar algunos de los resultados experimentales y las conclusiones a las que se llegó en este artículo.

2. Superestructura asociativa del reconocimiento de patrones

Los modelos de clasificación de patrones parten de diversas superestructuras o bases teóricas, las cuales han propiciado el surgimiento de los llamados enfoques del Reconocimiento de Patrones. Entre los principales enfoques destacan los siguientes:

- **Enfoque estadístico-probabilístico.** Basado principalmente en la teoría de la probabilidad condicional. De este enfoque han surgido modelos como el clasificador Bayesiano, el clasificador Naïve Bayes [2,11] y las redes Bayesianas.
- **Clasificadores basados en métricas.** En este enfoque se aprovechan las propiedades de las métricas para realizar la clasificación de patrones; entre los que se destacan los clasificadores de distancia mínima (denominado por algunos clasificador Euclidiano) [2] y los clasificadores de los vecinos más cercanos (k-NN) [12] o variantes como el clasificador IBK [13].
- **Clasificadores basados en árboles de decisión:** los cuales pueden definirse como modelos de clasificación que mapean los rasgos de un patrón (los cuales se modelan en las ramas del árbol) a conclusiones acerca del valor de la clase del patrón (las hojas del árbol); entre los modelos de clasificación destacados se encuentran en C4.5 [14] y el *Random Forest* [15].

- **Enfoque neuronal.** Este enfoque propone modelos que realizan la tarea de clasificar o recuperar patrones a partir de la representación matemática de las neuronas y sus interconexiones [16,17]. De este enfoque en particular, han surgido modelos emergentes como el que se denomina *Deep Learning* [18].
- **Enfoque asociativo.** Este enfoque surgió en el año 2002 en el Centro de Investigación en Computación, ha aprovechado las características de algunos modelos de memorias asociativas, en especial las Alfa-Beta [8], para generar más de una decena de modelos originales que han logrado resultados competitivos con respecto a modelos importantes en el estado del arte del Reconocimiento de Patrones.

A partir del surgimiento de las denominadas Memorias Asociativas Alfa-Beta, se han producido diversos trabajos de investigación, los cuáles están orientados a proponer soluciones a tres de las cuatro tareas que realiza el Reconocimiento de patrones, a saber: clasificación, regresión y recuperación.

El primero de estos trabajos se denominó CHAT (Clasificador Híbrido Asociativo con Traslación) [19] y se basa en los mejores principios de la Lernmatrix de Steinbuch [20] y del *Linear Associator* de Anderson-Kohonen [21] para realizar tareas de clasificación de patrones. Este modelo se caracteriza por evitar la saturación de la Lernmatrix y por otra parte solventa el problema del *Linear Associator* de requerir patrones ortonormales. A pesar de los buenos resultados obtenidos, este modelo aún tenía diversas áreas de oportunidad, como la descrita en la referencia [22], donde se explotó el concepto de *One-Hot Vector* y el procesamiento de vectores en base 2.

Además de los trabajos citados anteriormente, dentro de la superestructura asociativa del reconocimiento de patrones ha destacado el uso del Clasificador Gamma, el cual está basado en los operadores originales alfa y beta y ha probado ser exitoso al momento de enfrentar tareas de clasificación o de predicción de series de tiempo, como la contaminación del aire de la ciudad de México [23] o la producción de petróleo [24].

Cabe mencionar que, aunque el modelo original de memorias Alfa-Beta ha servido de base para otros modelos, también han sido aplicadas exitosamente en diversos escenarios, como en el propuesto en [25], donde se usan dichas memorias para clasificar patrones en bancos de datos de bioinformática. También se ha modificado el modelo original de las memorias Alfa-Beta, de forma tal que se han reducido sus limitantes iniciales y se ha aplicado exitosamente como modelo de clasificación de patrones, validando los resultados con una prueba de significancia no paramétrica [26].

Una de las áreas en las que los modelos asociativos también han destacado por ofrecer efectividad y eficiencia es en el área de finanzas; en este rubro destaca el *Naïve Associative Classifier* (NAC) el cual, a partir de un operador asociativo y de una regla para tratar con datos perdidos, es posible tratar problemas del área de finanzas con resultados competitivos respecto a otros modelos del estado del arte [27]. Posteriormente, en otro trabajo se comprobó que era posible mejorar los resultados de este modelo generando pesos a los rasgos a evaluar, a través de la Evolución Diferencial [28].

Finalmente, no todos los trabajos desarrollados en esta área han estado relacionados con modelos de reconocimiento de patrones. En su momento, se ha incursionado en el debate sobre la inclusión de las Tecnologías de la Información y las comunicaciones en el ámbito educativo [29]; posteriormente este análisis se profundizó para cursos de posgrado [30]; de igual forma, se ha trabajado en el modelado de la evolución de las redes sociales en el ámbito académico [31].

3. Herramientas matemáticas

El modelo que se propuso en este artículo se basó en los conceptos matemáticos que se enlistan a continuación.

3.1. La función de Heaviside

Definición 1.

Sea x un número real. La función de Heaviside de x se define mediante la siguiente expresión [10]:

$$H(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

3.2. Lattices y estructuras algebraicas ordenadas

Definición 2 (máximo y mínimo).

Sea $(E; \leq)$ un conjunto parcialmente ordenado, entonces:

Un elemento maximal, es un elemento $m \in E$ que no está contenido en ningún otro elemento de E , esto es:

$$p \in E, m \leq p \Rightarrow m = p.$$

Un máximo elemento $m \in E$ es un elemento que contiene a cada elemento de E , esto es:

$$\forall p \in E \Rightarrow p \leq m.$$

Un elemento minimal es un elemento $n \in E$ que no contiene a ningún otro elemento de E , esto es:

$$p \in E, p \leq n \Rightarrow p = n.$$

El mínimo elemento $n \in E$ es un elemento contenido por todos los demás elementos de E , esto es:

$$\forall p \in E \Rightarrow n \leq p.$$

Un conjunto parcialmente ordenado está limitado o acotado si tiene ambos elementos: un máximo y un mínimo.

Definición 3 (límite inferior y superior).

Si E es un conjunto ordenado y F es un subconjunto de E , entonces se dice que $x \in E$ es un límite inferior de F si $\forall y \in F x \leq y$ y un límite superior de F si $\forall y \in F y \leq x$.

Definición 4 (ínfimo y supremo).

Si E es un conjunto ordenado y F es un subconjunto de E , entonces el ínfimo se define como el mayor de los límites inferiores. Por dualidad se dice que el supremo se define como el menor de los límites superiores.

Definición 5 (lattice).

Una *lattice* es un conjunto ordenado, en el cual cada par de elementos (y por lo tanto cada subconjunto finito) tiene un ínfimo y un supremo. Notación: $(E; \wedge, \vee, \leq)$.

3.3. Representación de números enteros

Típicamente, en las aplicaciones cotidianas los números enteros se representan mediante notación decimal o de base 10. Por ejemplo, el número 965 denota la suma de productos $9 \cdot 10^2 + 6 \cdot 10 + 5$, aunque en algunos casos conviene utilizar otras bases diferentes a 10. Por ejemplo, las computadoras utilizan notación binaria (con base 2) para realizar cálculos aritméticos y también usan la notación octal (base 8) o hexadecimal (base 16) para representar caracteres como letras o dígitos. De hecho, es posible utilizar cualquier entero positivo mayor que 1 como base para expresar los enteros. Para fines de procesar los bancos de datos, en este artículo se utilizaron las definiciones descritas a continuación, y que fueron tomadas de [9, 32].

Definición 6.

Sea b un entero positivo mayor que 1. Entonces, si n es un entero positivo en base 10, n se expresa en términos de una base b de la siguiente forma:

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots + a_1 b + a_0,$$

donde b es un entero no negativo, a_0, a_1, \dots, a_k son enteros no negativos menores que b y $a_k \neq 0$. Esta representación se denomina expresión de n en términos de una base b , y se denota por $(a_k a_{k-1} \dots a_1 a_0)_b$. Por ejemplo $(245)_8$ representa la expresión $2 \cdot 8^2 + 4 \cdot 8 + 5 = 165$. Para expresar un número decimal en términos de una base b , se debe seguir este algoritmo:

Algoritmo 1 (conversión de base)

1. Dividir n entre b para obtener el cociente y el residuo: $n = bq_0 + a_0$, $0 \leq a_0 < b$
2. El resto a_0 , es el dígito situado más a la derecha de la expresión en base b de n .
3. Dividir q_0 entre b : $q_0 = bq_1 + a_1$, $0 \leq a_1 < b$
4. Vemos que a_1 es el segundo dígito por la derecha de la expresión de n en base b . El proceso continúa dividiendo sucesivamente el cociente por b , obteniendo como residuos los dígitos de la representación en base b .
5. El proceso concluye si el cociente es cero.

4. Modelo de clasificación propuesto

El modelo de clasificación del artículo basa su funcionamiento en las siguientes hipótesis.

Hipótesis 1. Dado que el clasificador de Heaviside es supervisado, se asume que se tiene acceso a un banco de patrones para aprendizaje supervisado, al estilo de los que se incluyen en el repositorio de la UCI [33].

Hipótesis 2. Los patrones pueden ser representados como vectores de dimensión finita, cuyas componentes son números reales. Si hay números negativos, éstos se transformarán en componentes reales no negativas. Además, es posible truncar o redondear estos números reales, para obtener vectores de dimensión finita con componentes racionales no negativas representadas por un número no negativo y un número finito de decimales.

Hipótesis 3. Es posible encontrar un escalamiento que permita transformar esos racionales en números enteros no negativos.

Las tres hipótesis previas nos permiten transformar el banco de datos original en un banco de vectores de cardinalidad finita y de dimensión finita, cuyas componentes son números enteros no negativos. De aquí se desprende la siguiente hipótesis.

Hipótesis 4.- Es posible representar cada una de las componentes de los vectores del banco transformado, en términos de un sistema numérico posicional de base b , donde b es un número entero mayor que 1. Para lograrlo, simplemente se calcula la expansión de esa componente, que corresponda a la base b del sistema numérico.

Tras un análisis extenso de algunos conceptos de las memorias asociativas, de las *lattices* y en las estructuras algebraicas ordenadas, llegamos a la conclusión de que el clasificador de Heaviside operará en dos modos diferentes: el modo HI que se corresponde con el ínfimo y ayudará en la clasificación de patrones contaminados con ruido sustractivo, y el modo HS que se corresponde con el supremo y será de utilidad en la clasificación de patrones alterados con ruido aditivo [34].

Cabe destacar que mientras que la operación de aprendizaje L es igual para ambos modos del

clasificador, no sucede lo mismo con la operación de clasificación C , la cual varía con cada modo.

4.1. La operación L para ambos modos del clasificador de Heaviside

Definición 7 Sean A_i^k y A_j^k la i -ésima y la j -ésima componentes del k -ésimo patrón del conjunto A de aprendizaje, en términos de un sistema numérico posicional base b . Entonces, para ambos modos del clasificador de Heaviside se define la operación de aprendizaje L :

$$L(A_i^k, A_j^k) = [H(A_i^k + A_j^k + b)] \cdot [(A_i^k + b) - (A_j^k + 1)],$$

la cual está sujeta a las siguientes propiedades:

Propiedad L1:

Sea $b \in \mathbb{Z}$ tal que $b > 1$. Si $A_i^k = A_j^k = 0$. Entonces $L(A_i^k, A_j^k) = b - 1$.

Propiedad L2:

Sea $b \in \mathbb{Z}$ tal que $b > 1$. Entonces $L(b - 1, b - 1) = L(0, 0) = b - 1$.

Propiedad L3:

Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y \in \mathbb{Z}$ tales que $x \leq y$, con $0 \leq x \leq b - 1$ y $0 \leq y \leq b - 1$. Entonces se cumple: $L(x, y) \leq L(y, x)$.

Propiedad L4:

Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y, z \in \mathbb{Z}$ tales que $x \leq y$, con $0 \leq x \leq b - 1, 0 \leq y \leq b - 1$ y $0 \leq z \leq b - 1$. Entonces: $L(x, z) \leq L(y, z)$

Propiedad L5:

Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y, z \in \mathbb{Z}$ tales que $x \leq y$, con $0 \leq x \leq b - 1, 0 \leq y \leq b - 1$ y $0 \leq z \leq b - 1$. Entonces: $L(z, x) \geq L(z, y)$

Propiedad L6:

Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y, z \in \mathbb{Z}$ tales que $0 \leq x \leq b - 1, 0 \leq y \leq b - 1$ y $0 \leq z \leq b - 1$. Entonces: $L[(x \vee y), z] = L(x, z) \vee L(y, z)$.

Propiedad L7:

Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y, z \in \mathbb{Z}$ tales que $0 \leq x \leq b - 1, 0 \leq y \leq b - 1$ y $0 \leq z \leq b - 1$. Entonces: $L[(x \wedge y), z] = L(x, z) \wedge L(y, z)$.

4.2. La operación C para el modo HI del clasificador de Heaviside

A diferencia de la operación L, la operación C se maneja de forma diferente para cada modo del clasificador de Heaviside. Se denomina C^l para el modo HI y C^s para el modo HS.

Definición 8. Sean M_{ij} y P_j^k la ij -ésima componente del clasificador de Heaviside M y la j -ésima componente del k -ésimo patrón del conjunto P de prueba, respectivamente. Entonces, para el modo operativo HI del clasificador de Heaviside se define C^l :

$$C^l(M_{ij}, P_j^k) = H[(M_{ij} + 2) - (b - P_j^k)] \\ \cdot H[(2b - 1) - (M_{ij} + P_j^k)] \\ \cdot [(M_{ij} + 1) - (b - P_j^k)]$$

Definición 9. Sean M_{ij} y P_j^k la ij -ésima componente del clasificador de Heaviside M y la j -ésima componente del k -ésimo patrón del conjunto P de prueba, respectivamente. Entonces, para el modo operativo HI del clasificador de Heaviside se define C^s :

$$C^s(M_{ij}, P_j^k) = H[(M_{ij} + 2) - (b - P_j^k)] \\ \cdot H[(2b - 1) - (M_{ij} + P_j^k)] \\ \cdot [(M_{ij} + 1) - (b - P_j^k)] \\ + (H[(b - 1) - (M_{ij} + P_j^k)] \\ + H[(M_{ij} + 2) - (2b - P_j^k)]) \\ \cdot (b - 1),$$

Propiedades de C, en ambos modos:

Propiedad C1:

Sea $b \in \mathbb{Z}$ tal que $b > 1$. Entonces $C(b, 0) = 1$.

Propiedad C2:

Sea $b \in \mathbb{Z}$ tal que $b > 1$. Entonces $C(b - 1, 0) = 0$.

Propiedad C3:

Sea $b \in \mathbb{Z}$ tal que $b > 1$. Entonces $C(0, b - 1) = 0$.

Propiedad C4:

$C(0, 0) = 0$ para el modo HI y $C(0, 0) = b - 1$ para el modo HS.

Propiedad C5:

Sean $x, y, z \in \mathbb{Z}$ tales que $0 \leq x \leq 2(b - 1)$, $0 \leq y \leq 2(b - 1)$, $0 \leq z \leq b - 1$. Si $x \leq y$, entonces $C(x, z) \leq C(y, z)$.

4.3. Algoritmo del clasificador de Heaviside

A partir del planteamiento de las fases de aprendizaje y clasificación del clasificador de Heaviside, generamos un algoritmo que posteriormente fue programado y probado en diversos bancos de datos (Capítulo 5). A continuación, presentamos el algoritmo del Clasificador de Heaviside, el cual contempla las siguientes etapas: preprocesamiento, transformación de los patrones, entrenamiento (aprendizaje) y clasificación.

Algoritmo 2. Clasificador de Heaviside en modo operativo HI

Eta de preprocesamiento

1. Para cada patrón del banco de datos.
2. Buscar si hay componentes negativas
3. Si hay componentes negativas, buscar aquella que tenga el menor valor y restarla de las demás componentes del patrón.
4. Si no hay componentes negativas el patrón no se modifica.
5. Buscar si existen componentes con punto decimal en el banco de datos.
6. Si existen componentes decimales en el banco de datos, buscar de entre todas éstas el número máximo de dígitos decimales y almacenarlo en la variable d . Si no hay componentes decimales, $d=0$.
7. Para cada patrón del banco de datos, multiplicar todas las componentes por 10^d .

Eta de transformación

8. Elegir como base un número entero positivo b mayor que 1.
9. Por cada patrón del banco de datos
10. Transformar cada componente en términos de la base b .
11. Crear el patrón transformado a partir de la concatenación de las componentes transformadas, en términos de la base b .

Eta de entrenamiento

12. Para cada patrón de dimensión n $A^k \in A$ $k = 1, 2, \dots, |A|$ construir la matriz: $M_k = [A^k \boxtimes_L (A^k)^t]_{n \times n}$
13. Aplicar el ínfimo a las matrices obtenidas en el paso 1 para obtener la memoria asociativa M :

$$M = \bigwedge_{k=1}^{|A|} [A^k \boxtimes_L (A^k)].$$

La componente ij -ésima de la memoria asociativa M está dada así:

$$m_{ij} = \bigwedge_{k=1}^{|A|} L(A_i^k, A_j^k) \quad \forall i \in \{1, 2, \dots, n\} \text{ y } \forall j \in \{1, 2, \dots, n\}.$$

Etapa de clasificación

Para cada patrón $P^d \in P$ hacer lo siguiente:

- Presentar el patrón $P^d \in P$ $d \in \{1, 2, \dots, |P|\}$, de dimensión n , a la memoria asociativa M y realizar la operación:

$$R = M \cup_{C^I} P^d.$$

- Se obtendrá el patrón R , de dimensión n , cuya i -ésima componente es:

$$\begin{aligned} R_i &= (M \cup_{C^I} P^d)_i = \bigvee_{j=1}^n C^I(m_{ij}, P_j^d) \\ &= \bigvee_{j=1}^n C^I \left\{ \left[\bigwedge_{k=1}^{|A|} L(A_i^k, A_j^k) \right], P_j^d \right\}. \end{aligned}$$

- Buscar en A el patrón recuperado R o en su defecto el patrón más parecido¹ a R .
- Asignar a P^d la clase del patrón encontrado en A para R .

En esta sección se presentó el algoritmo del clasificador de Heaviside en el modo HI. El algoritmo correspondiente modo HS puede obtenerse aplicando dualidad en las etapas de entrenamiento y de clasificación, y se presenta a continuación.

Algoritmo 3. Clasificador de Heaviside en modo operativo HS

Etapa de preprocesamiento

- Para cada patrón del banco de datos.
- Buscar si hay componentes negativas
- Si hay componentes negativas, buscar aquella que tenga el menor valor y restarla de las demás componentes del patrón.

¹ En este trabajo de tesis se utilizó la idea de asignar la clase del vecino más cercano.

- Si no hay componentes negativas el patrón no se modifica.
- Buscar si existen componentes con punto decimal en el banco de datos.
- Si existen componentes decimales en el banco de datos, buscar de entre todas éstas el número máximo de dígitos decimales y almacenarlo en la variable d . Si no hay componentes decimales $d=0$.
- Por cada patrón del banco de datos
- Multiplicar todas las componentes por 10^d .

Etapa de transformación

- Elegir como base un número entero positivo mayor que 1.
- Por cada patrón del banco de datos
- Transformar cada componente en términos de una base b elegida.
- Crear el patrón transformado a partir de la concatenación de las componentes transformadas en términos de la base b .

Etapa de entrenamiento

- Por cada patrón, de dimensión n , $A^k \in A$ $k = 1, 2, \dots, |A|$ construir la matriz:

$$M_k = [A^k \boxtimes_L (A^k)^t]_{n \times n}.$$

- Aplicar el ínfimo a las matrices obtenidas en el paso 1 para obtener la memoria asociativa M :

$$M = \bigwedge_{k=1}^{|A|} [A^k \boxtimes_L (A^k)]$$

La componente ij -ésima de la memoria asociativa M está dada de la siguiente manera:

$$m_{ij} = \bigwedge_{k=1}^{|A|} L(A_i^k, A_j^k) \quad \forall i \in \{1, 2, \dots, n\} \text{ y } \forall j \in \{1, 2, \dots, n\}.$$

Etapa de clasificación

- Por cada patrón $P^d \in P$ hacer lo siguiente:

16. Presentar el patrón $P^d \in P$ $d \in \{1, 2, \dots, |P|\}$, de dimensión n , a la memoria asociativa M y realizar la operación:

$$R = M \mathbin{\frown}_{C^S} P^d$$

17. Se obtendrá el patrón R , de dimensión n , cuya i -ésima componente se expresa como:

$$\begin{aligned} R_i &= (M \mathbin{\frown}_{C^S} P^d)_i \\ &= \bigwedge_{j=1}^n C^S(m_{ij}, P_j^d) \\ &= \bigwedge_{j=1}^n C^S \left\{ \left[\bigvee_{k=1}^{|A|} L(A_i^k, A_j^k) \right], P_j^d \right\} \end{aligned}$$

18. Buscar en A el patrón recuperado R o en su defecto el patrón más parecido² a R .
19. Asignar a P^d la clase del patrón encontrado en A para R .

5. Pruebas de rendimiento

En esta sección se reportan los resultados de una de las dos rondas de pruebas a las que fue sometido en clasificador de Heaviside. En estas pruebas se usaron bancos de datos de los repositorios KEEL [35] y UCI [33].

5.1. Clasificadores comparados

Para esta ronda de pruebas se decidió comparar el desempeño del clasificador de Heaviside con los siguientes modelos:

- AdaBoost M1 [36].
- Metaclasificador utilizando Bagging [37].
- Clasificador métrico IB1 [13].
- Clasificador Naïve Bayes [11].
- Perceptrón Multicapa [17].
- Metaclasificador RandomForest [15].
- Máquina de Soporte Vectorial con Kernel Gaussiano [38,39].

² En este trabajo de tesis se utilizó la idea de asignar la clase del vecino más cercano.

5.2. Bancos de datos utilizados en los experimentos

Para los experimentos reportados en el presente trabajo, se utilizaron los bancos de datos enlistados a continuación, tomados de los repositorios Keel [35] y UCI [33]:

- **Balance:** banco de datos creado para modelar resultados de experimentos relacionados con la psicología y las ciencias del conocimiento.
- **Banknote authentication:** banco de datos creado a partir de imágenes obtenidas para la autenticación de billetes.
- **Breast Tissue:** en este banco de datos se registran mediciones de impedancia que fueron hechas al tejido mamario de un conjunto de pacientes.
- **Cardiocotography:** este banco de datos contiene mediciones del ritmo cardiaco fetal y rasgos de contracción uterina que han sido analizados y clasificados por expertos.
- **Climate Model Simulation Crashes:** a partir de muestras de 18 parámetros de modelos de simulación del clima, este banco de datos busca detectar qué rasgos provocan fallos en las simulaciones de modelos climáticos.
- **Column vertebral:** ya mencionado en la lista anterior.
- **Connectionist Bench:** este banco de datos fue creado originalmente para entrenar una red neuronal que sea capaz de discriminar entre señales de sonar que hayan rebotado en materiales como el metal y aquellas que hayan rebotado en roca.
- **Default of credit card clients:** banco de datos creado para predecir la capacidad de

Tabla 1. Resultados experimentales (parte 1)

Clasificador	CH-HI	CH-HS	ABM1	BGG
Banco de datos				
Balance	86.70%	86.43%	81.92%	82.72%
Banknote authentication	97.79%	97.93%	99.64%	98.69%
Breast Tissue	66.95%	64.93%	71.70%	73.58%
Cardiocotography	90.39%	87.82%	93.24%	94.45%
Climate Model Simulation Crashes	100%	100%	94.26%	92.22%
Connectionist Bench	81.34%	82.41%	82.69%	76.92%
Default of credit card clients	76.76%	76.69%	78.38%	81.70%
Fertility	88%	88%	88%	88%
Ionosphere	92.22%	91.16%	92.31%	91.17%
Led7Digit	60.74%	61.26%	71%	72.80%
Movement_LIBRAS	85.24%	86.33%	74.44%	69.72%
QSAR biodegradation	81.84%	81.65%	86.16%	84.83%
Ring	90.43%	90.154%%	94.36%	93.76%
Seeds	90.67%	90.43%	93.33%	92.86%
Spambase	82.41%	81.29%	92.82%	94.06%
Texture	98.14%	93.68%	97.58%	95.16%
Twonorm	50.93%	51.10%	94.18%	93.55%
Vertebral column	83.26%	83.13%	81.29%	82.58%
Wine quality	60.29%	60.07%	64.23%	62.98%
Yeast	58.18%	57.81%	59.2992	60.04%

- un grupo de clientes Taiwanesees para hacer el pago de sus tarjetas de crédito.
- **Fertility:** datos de fertilidad de 100 pacientes masculinos. Se busca establecer una relación entre la calidad del esperma y factores como entorno socio-demográfico, aspectos del medio ambiente, estado de salud y estilo de vida.
- **Ionosphere:** Banco de datos creado a partir de muestras, obtenidas mediante radar, de la ionósfera.
- **Led7Digit:** Este banco de datos representa todos los dígitos que se pueden representar con un display de 7 segmentos.
- **Movement_LIBRAS:** Banco de datos compuesto por 24 clases; cada clase representa una letra en la LIBRAS (lengua brasileña de señas).
- **QSAR biodegradation:** banco de datos creado para clasificar poco más de 1000 sustancias químicas en lista para biodegradación o no lista para biodegradación.
- **Ring:** ya descrito en la sección anterior.

Tabla 2. Resultados experimentales, parte 2

Clasificador	CH-HI	CH-HS	IB1	NB
Banco de datos				
Balance	86.70%	86.43%	84.80%	90.56%
Banknote authentication	97.79%	97.93%	99.85%	84.26%
Breast Tissue	66.95%	64.93%	71.70%	66.98%
Cardiocotography	90.39%	87.82%	91.16%	82.36%
Climate Model Simulation Crashes	100%	100%	89.44%	95%
Connectionist Bench	81.34%	82.41%	86.54%	67.79%
Default of credit card clients	76.76%	76.69%	73.02%	69.62%
Fertility	88%	88%	83%	88%
Ionosphere	92.22%	91.16%	86.32%	82.62%
Led7Digit	60.74%	61.26%	69.80%	70.20%
Movement_LIBRAS	85.24%	86.33%	85.83%	62.78%
QSAR biodegradation	81.84%	81.65%	84.46%	75.92%
Ring	90.43%	90.154%%	75.05%	97.97%
Seeds	90.67%	90.43%	94.29%	91.43%
Spambase	82.41%	81.29%	90.89%	79.60%
Texture	98.14%	93.68%	99.13%	77.51%
Twonorm	50.93%	51.10%	94.76%	97.86%
Vertebral column	83.26%	83.13%	78.39%	83.23%
Wine quality	60.29%	60.07%	63.98%	54.72%
Yeast	58.18%	57.81%	52.16%	58.56%

- **Seeds:** banco de datos creado a partir de mediciones hechas en tres diferentes especies de trigo.
- **Spambase:** Banco de datos creado a partir de la identificación de mensajes de correo no deseado (spam).
- **Texture:** Banco de datos creado a partir de 11 diferentes tipos de textura presentes en imágenes digitales.
- **Twonorm:** Banco de datos sintético cuyas clases se generan a partir de una distribución multinomial multivariable.
- **Wine quality:** banco de datos en el que se registra la calidad del vino tinto, del norte de Portugal, a partir de pruebas físico-químicas.
- **Yeast:** banco de datos creado para predecir los sitios de localización de proteínas en células.

5.3. Resultados experimentales

Debido a que se está probando el rendimiento del clasificador en problemas relacionados con una o más clases, se decidió emplear el *Accuracy*

Tabla 3. Resultados experimentales, parte 3

Clasificador	CH-HI	CH-HS	MLP	RF	SVMG
Banco de datos					
Balance	86.70%	86.43%	91.84%	81.28%	89.44%
Banknote authentication	97.79%	97.93%	99.93%	99.27%	100%
Breast Tissue	66.95%	64.93%	63.21%	71.70%	20.75%
Cardiocotography	90.39%	87.82%	91.96%	94.92%	80.01%
Climate Model Simulation Crashes	100%	100%	92.96%	93.33%	91.48%
Connectionist Bench	81.34%	82.41%	82.21%	83.17%	65.87%
Default of credit card clients	76.76%	76.69%	81.78%	81.71%	77.94%
Fertility	88%	88%	90%	86%	88%
Ionosphere	92.22%	91.16%	91.17%	92.88%	93.45%
Led7Digit	60.74%	61.26%	70.60%	70.40%	71.80%
Movement_LIBRAS	85.24%	86.33%	79.44%	83.06%	36.39%
QSAR biodegradation	81.84%	81.65%	86.73%	86.82%	84.93%
Ring	90.43%	90.154% ^o	91.06%	95.30%	50.46%
Seeds	90.67%	90.43%	95.24%	93.81%	90.48%
Spambase	82.41%	81.29%	91.49%	95.48%	84.49%
Texture	98.14%	93.68%	99.81%	97.87%	94.47%
Twonorm	50.93%	51.10%	96.84%	96.62%	97.73%
Vertebral column	83.26%	83.13%	85.48%	83.55%	48.39%
Wine quality	60.29%	60.07%	61.48%	70.04%	58.22%
Yeast	58.18%	57.81%	58.29%	61.39%	42.92%

como medida principal de desempeño; el método de validación elegido fue el *10-Fold Cross-Validation*, a fin de otorgar certeza estadística al resultado [40].

En las tablas 1, 2 y 3 se presentan los resultados obtenidos tras comparar el desempeño del clasificador de Heaviside con otros modelos del estado del arte.

Las siglas presentadas en las tablas anteriores, se enlistan a continuación.

- CH-HI: clasificador de Heaviside modo HI.
- CH-HS: clasificador de Heaviside modo HS.
- ABM1: metaclasificador AdaBoost M1
- BGG: metaclasificador Bagging.

- IB1: clasificador Instance Based con K=1.
- NB: clasificador Naive Bayes.
- MLP: perceptrón multicapa.
- RF: clasificador Random Forest.
- SVMG: Máquina de Soporte Vectorial con kernel Gaussiano.

Como es posible verificar en las tablas previas, el desempeño del Clasificador de Heaviside es equiparable al de los otros modelos de clasificación utilizados en esta prueba. En la tabla 5 se presentan los resultados de aplicar la prueba de Wilcoxon [41] a los resultados obtenidos entre el clasificador de Heaviside, en su modo HI, con los demás modelos involucrados.

Tabla 4. Resultados de la prueba de Wilcoxon.

Clasificadores	p
Heaviside HS - Heaviside HI	0.078
AdaBoost M1 - Heaviside HI	0.117
Bagging - Heaviside HI	0.243
IB1 - Heaviside HI	0.911
Naïve Bayes - Heaviside HI	0.136
Multilayer Perceptron - Heaviside HI	0.033
Random Forest - Heaviside HI	0.025
SVM Gaussian K - Heaviside HI	0.171

Cabe destacar que en la prueba de Wilcoxon se asume como Hipótesis nula que el comportamiento de los clasificadores es similar, y como hipótesis alterna que sí hay diferencias significativas entre los clasificadores.

Para que se descarte la hipótesis nula, es necesario que la prueba obtenga un valor-p menor que 0.05; de forma que los resultados de esta prueba tendrán un 95% de confianza. En la tabla 4, se presentan los resultados de la prueba.

Como es posible observar, los resultados obtenidos en esta prueba revelan que el desempeño del Clasificador de Heaviside es similar al de los demás clasificadores evaluados.

5.4. Aplicaciones

Una vez concluida la etapa de formulación y las primeras pruebas de funcionamiento del Clasificador propuesto en este artículo, propusimos algunas situaciones en las que su uso resultó conveniente.

En esta sección mencionaremos dos aplicaciones de nuestro modelo: una para la clasificación inteligente de textos escolares [42], mientras que la segunda aplicación está relacionada con la detección rápida de la Degeneración Macular Dependiente de la edad [43], en la cual la mejor clasificación fue realizada por las Support Vector Machines, aunque en una modificación posterior del algoritmo de procesamiento de imágenes, el clasificador de Heaviside obtuvo los mejores resultados [44].

6. Conclusiones

En este artículo se propuso un nuevo modelo para la clasificación supervisada de patrones, el cual está basado en la función de Heaviside. Cabe destacar que, tras un extenso análisis del estado del arte del reconocimiento de patrones, no se encontraron modelos de cómputo inteligente que estén basados en la función de Heaviside, con la excepción de los modelos primitivos de las Redes Neuronales o de función de decisión de las SVM

El clasificador propuesto trabaja con patrones cuyas componentes están en términos de números enteros no negativos, los cuales pueden ser obtenidos a partir patrones con componentes numéricas reales, sólo basta transformarlas a valores con base entera mayor que 1.

El modelo propuesto en este trabajo se basa en la función de Heaviside y en operaciones propias de las *lattices* y las estructuras algebraicas ordenadas. La función de aprendizaje se denomina L; mientras que la operación que permite realizar la clasificación de patrones se denomina C.

Las operaciones de aprendizaje y clasificación de patrones propuestas en este artículo están sustentadas en un conjunto de definiciones, lemas y teoremas que demuestran formalmente el rendimiento y las capacidades del clasificador de Heaviside. Después de probar el clasificador de Heaviside en varios bancos de datos relacionados con Reconocimiento de Patrones, y de comparar sus resultados con los obtenidos por algunos de los modelos más importantes de clasificación de patrones, destaca el hecho de que el clasificador propuesto tiene un desempeño satisfactorio, ya que ofrece un desempeño bastante competitivo. También es importante resaltar que la validez de los experimentos realizados fue sustentada mediante en al menos una prueba de significancia estadística; las diferencias no fueron significativas.

Además de las pruebas realizadas con bancos de datos de repositorios importantes, se propusieron al menos dos aplicaciones para el Clasificador de Heaviside. En dichas aplicaciones fue necesario diseñar un método para extraer rasgos útiles a partir de datos en bruto, y a partir de los rasgos seleccionados fue posible generar los bancos de datos adecuados para la clasificación de los patrones; además, se hicieron

experimentos en donde se comprobó que nuestro clasificador obtiene, al menos, resultados competitivos contra otros modelos destacados en el estado del arte.

A pesar de que el clasificador de Heaviside ofreció resultados competitivos en la mayoría de los bancos de datos donde fue probado, debemos mencionar que este modelo está, en cierta forma, limitado por su complejidad temporal; ya que en bancos de datos de alta dimensionalidad las etapas de entrenamiento y clasificación consumirán mucho tiempo de cómputo. Sin embargo, es posible reducir el consumo temporal fijando valores de base relativamente altos, respecto al banco de datos en cuestión.

A continuación, presentamos los trabajos a futuro que consideramos se deben realizar para mejorar los resultados de este artículo:

- Revisar a fondo la formulación de las operaciones L y C en busca de formulaciones alternas que optimicen su funcionamiento.
- Profundizar en el estudio de las propiedades de ambas operaciones, para potencializar sus alcances y reducir el efecto de sus limitaciones.
- Proponer una regla para el manejo de valores perdidos, de manera que no sea necesario realizar imputación.
- Realizar pruebas en más bancos de datos, pertenecientes a repositorios públicos y privados, y que tengan alta complejidad de datos.
- Continuar con la generación de escenarios de aplicación para el modelo propuesto en este trabajo de tesis.
- Incursionar en el campo del cómputo paralelo implementando el algoritmo del clasificador de Heaviside en arquitecturas paralelas (CUDA, Open MPI, OpenCL, entre otras); y aplicarlo en bancos de datos pertenecientes a *big data*.
- Investigar el mejor valor de base para un determinado banco de datos.

Agradecimientos

El autor de este artículo desea agradecer a los Drs. Cornelio Yáñez Márquez y Oscar Camacho Nieto, por su asesoría y apoyo en este proceso. De igual forma, se agradece al IPN por la formación recibida tanto a nivel Licenciatura, en la Maestría y en el Doctorado y por la Beca de Estímulo Institucional para la Formación de Investigadores. Finalmente se agradece al Consejo Nacional de Ciencia y Tecnología de México (CONACYT) por las becas recibidas para los estudios de Maestría y Doctorado.

Referencias

1. **Marques de Sá, J.P. (2001).** *Pattern Recognition Concepts, Methods and Applications*. Springer Verlag Berlin Heidelberg. DOI: 10.1007/978-3-642-56651-6.
2. **Duda, R.O., Hart, P.E., & Stork, D.E. (2001).** *Pattern Classification*.
3. **Díaz de León, S.J.L., Sánchez Garfias, F.A., & Yáñez Márquez, C. (2003).** *Reconocimiento de Patrones: Enfoque Probabilístico-Estadístico*. Centro de Investigación en Computación del IPN.
4. **Bezdek, J.C. (1994).** What is Computational Intelligence? *Computing Intelligent Imitating Life IEEE Press*, pp. 1–12.
5. **Konar, A. (2005).** *Computational Intelligence: Principles, Techniques and Applications*. Springer Berlin Heidelberg.
6. **García Floriano, A., Camacho Nieto, O., & Yáñez Márquez, C.** Clasificador de Heaviside. *Nova Science*, Vol. 7, No. 14, pp. 365–397.
7. **Fernandes, B.J.T. & Cavalcanti, G.D.C. (2014).** Ren TI. Constructive Autoassociative Neural Network for Facial Recognition. *PLoS ONE*, pp. 1–23. DOI:10.1371/journal.pone.0115967.
8. **Yáñez Márquez, C., López Yáñez, I., Aldape Pérez, M., Camacho Nieto, O., Argüelles Cruz A.J., & Villuendas Rey, Y. (2018).** Theoretical Foundations for the Alpha-Beta Associative Memories: 10 Years of Derived Extensions, Models, and Applications. *Neural Process Lett*, Vol. 48, No. 2, pp. 811–847. DOI: 10.1007/s11063-017-9768-2.
9. **Rosen, K.H. (1999).** *Discrete Mathematics and Its Applications*. McGraw-Hill Inc.
10. **Abramowitz, M. & Stegun, I.A. (1964).** *Handbook of Mathematical Functions with Formulas, Graphs*

- and Mathematical Tables. *National Bureau of Standards*.
11. **Tang, B., Kay, S., & He, H. (2016).** Toward Optimal Feature Selection in Naive Bayes for Text Categorization. *IEEE Trans Knowl Data Eng*, Vol. 28, No. 9, pp. 2508–2521. DOI: 10.1109/TKDE.2016.2563436.
 12. **Hart, P. (1968).** The condensed nearest neighbor rule. *IEEE Trans Inf Theory*, Vol. 14, No. 3, pp. 515–516. DOI: 10.1109/TIT.1968.1054155.
 13. **Aha, D.W., Kibler, D., & Albert, M.K. (1991).** *Instance-based learning algorithms*. Mach Learn, Vol. 6, No. 1, pp. 37–66. DOI: 10.1007/BF00153759.
 14. **Quinlan, J.R. (1996).** Improved use of continuous attributes in C4. 5. *Journal Artif Intell Res*, Vol. 4, pp. 77–90. DOI: 10.1613/jair.279.
 15. **Breiman, L. (2001).** *Random Forests*. Mach Learn; Vol. 45, No. 1, pp. 5–32. DOI: 10.1023/A:1010933404324.
 16. **Rosenblatt, F. (1957).** *The Perceptron — A Perceiving and Recognizing Automaton*. Cornell Aeronautical Laboratory, Inc., Vol. 65, No. 6, pp. 386–408, DOI: 10.1037/h0042519.
 17. **Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986).** Learning representations by back-propagating errors. *Cognitive Modeling*, pp. 213–220. DOI: 10.1038/323533a0.
 18. **LeCun, Y., Bengio, Y., & Hinton, G. (2015).** Deep learning. *Nature Research Journal*, pp. 436–444. DOI: 10.1038/nature14539.
 19. **Santiago Montero, R. (2003).** *Clasificador híbrido de patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen*. Centro de Investigación en Computación. pp. 1–96.
 20. **Steinbuch, K. & Die, L. (1961).** *Biological Cybernetics*. DOI: 10.1007/BF00293853.
 21. **Kohonen, T. (1972).** *Correlation Matrix Memories*. *IEEE Trans Computers*, Vol. C-21, No. 4, pp. 353–359. DOI: 10.1109/TC.1972.5008975.
 22. **Uriarte Arcia, A.V., López-Yáñez, I., Yáñez-Márquez, C. (2014).** One-Hot Vector Hybrid Associative Classifier for Medical Data Classification. *PLoS ONE*. DOI: 10.1371/journal.pone.0095715.
 23. **Yáñez-Márquez, C., López-Yáñez, I., & de la Luz Sáenz-Morales, G. (2008).** Analysis and Prediction of Air Quality Data with the Gamma Classifier. **Ruiz-Shulcloper J, Kropatsch WG, (eds).** *Progress Pattern Recognit. Image Anal. Appl.*, Springer Berlin Heidelberg. pp 651–658. DOI: 10.1007/978-3-540-85920-8_79.
 24. **López-Yáñez, I., Sheremetov, L., & Yáñez-Márquez, C. (2014).** A novel associative model for time series data mining. *Pattern Recognition Lett*, Vol. 41, pp. 23–33. DOI: 10.1016/j.patrec.2013.11.008.
 25. **Godínez, I.R., López-Yáñez, I., & Yáñez-Márquez, C. (2009).** Classifying Patterns in Bioinformatics Databases by Using Alpha-Beta Associative Memories. **Sidhu AS, Dillon TS, (eds).** *Biomed. Data Appl.*, Berlin, Heidelberg: Springer, pp. 187–210. DOI: 10.1007/978-3-642-02193-0_8.
 26. **Ramírez-Rubio, R., Aldape-Pérez, M., Yáñez-Márquez, C., López-Yáñez, I., & Camacho-Nieto, O. (2017).** Pattern classification using smallest normalized difference associative memory. *Pattern Recognition Lett*, Vol. 93, pp. 104–112. DOI: 10.1016/j.patrec.2017.02.013.
 27. **Villuendas-Rey, Y., Rey-Benguría, C.F., Ferreira-Santiago, Á., Camacho-Nieto, O., & Yáñez-Márquez, C. (2017).** The Naïve Associative Classifier (NAC): A novel, simple, transparent, and accurate classification model evaluated on financial data. *Neurocomputing*, Vol. 265, pp. 105–15. DOI: 10.1016/j.neucom.2017.03.085.
 28. **Serrano-Silva, Y.O., Villuendas-Rey, Y., & Yáñez-Márquez, C. (2018).** Automatic feature weighting for improving financial Decision Support Systems. *Decis Support Syst*, Vol. 107, pp. 78–87. DOI:10.1016/j.dss.2018.01.005.
 29. **Moreno-Moreno, P., Yanez-Marquez, C., & Moreno-Franco. O.A. (2009).** The new informatics technologies in education debate. *Int J Technol Enhanc Learn*, Vol. 1, No. 4, pp. 327–341. DOI: 10.1504/IJTEL.2009.030782.
 30. **López-Yáñez, I., Yáñez-Márquez, C., Camacho-Nieto, O., Aldape-Pérez, M., & Argüelles-Cruz, A.J. (2015).** Collaborative learning in postgraduate level courses. *Computers in Human Behavior*, Vol. 51, Part. B, pp. 938–944. DOI:10.1016/j.chb.2014.11.055.
 31. **Lytras, M.D., Mathkour, H.I., Abdalla, H., Yanez-Marquez, C., & de Pablos, P.O. (2014).** The Social Media in Academia and Education Research Revolutions and a Paradox: Advanced Next Generation Social Learning Innovation. *Journal of Universal Computer Science*, Vol. 20, No. 15, pp. 1987–1994.
 32. **Andreescu, T. & Andrica, D. (2009).** *Number Theory: Structures, Examples, and Problems*. Edición Boston Birkhäuser. DOI: 10.1007/b11856.
 33. **Dua, D. & Graff, C. (2017).** *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences.

34. **Yáñez-Márquez, C. & Díaz de León, S.J.L. (2003).** Memorias Asociativas Basadas en Relaciones de Orden y Operaciones Binarias. *Computación y Sistemas*, Vol. 6, No. 4. pp. 300–311.
35. **Alcalá-Fernández, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herre, F. (2011).** KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *J Mult-Valued Log Soft Computing*, Vol. 7, pp. 255–287.
36. **Freund, Y. & Schapire, R.E. (1996).** Experiments with a new boosting algorithm. *Proceedings Thirteenth International Conference, Machine Learning*, pp. 1–9.
37. **Breiman, L. (1996).** Bagging Predictors. *Mach Learning*, Vol. 24, No. 2, pp. 123–140. DOI: 10.1023/A:1018054314350.
38. **Cortes, C. & Vapnik, V. (1995).** Support-vector networks. *Mach Learning*, Vol. 20, No. 3, pp. 273–297. DOI: 10.1007/BF00994018.
39. **Vapnik, V.N. (1998).** *Statistical Learning Theory*. 1st ed. Wiley-Interscience.
40. **Wong, T.T. (2015).** Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, Vol. 48, No. 9, pp. 2839–2846. DOI: 10.1016/j.patcog.2015.03.009.
41. **Wilcoxon, F. (1945).** Individual comparisons by ranking methods. *Biometrics*, pp. 192–202. DOI: 10.1007/978-1-4612-4380-9_16.
42. **García-Floriano, A., Ferreira-Santiago, Á., Yáñez-Márquez, C., Camacho-Nieto, O., Aldape-Pérez, M., & Villuendas-Rey, Y. (2017).** Social Web Content Enhancement in a Distance Learning Environment: Intelligent Metadata Generation for Resources. *International Review of Research in Open and Distributed Learning*, Vol. 18, No. 1, pp. 161–176. DOI: 10.19173/irrodl.v18i1.2646.
43. **García-Floriano, A., Ferreira-Santiago, Á., Camacho-Nieto, O., & Yáñez-Márquez, C.A. (2017).** Machine learning approach to medical image classification: Detecting age-related macular degeneration in fundus images. *Computer & Electrical Engineering*, Vol. 75, DOI: 10.1016/j.compeleceng.2017.11.008.
44. **García-Floriano, A., Yáñez-Marquez, C., & Camacho-Nieto, O. (2018).** Detection of Age-Related Macular Degeneration in Fundus Images by an Associative Classifier. *IEEE Latin American Transactions*, Vol. 16, No. 3, pp. 933–939. DOI: 10.1109/TLA.2018.8358676.

Article received on 01/09/2019; accepted on 07/10/2019.
Corresponding author is Andrés García Floriano.