

Revista Electrónica Nova Scientia

**Clasificador de Heaviside
Heaviside's Classifier**

**Andrés García Floriano¹, Oscar Camacho Nieto² y Cornelio
Yáñez Márquez³**

¹Doctorado, Centro de Investigación en Computación del IPN, Ciudad de México

²Centro de Innovación y Desarrollo Tecnológico en Cómputo del IPN

³Laboratorio, Centro de Investigación en Computación del IPN

México

Cornelio Yáñez Márquez. E mail: cyanez@cic.ipn.mx

Resumen

En este trabajo se presenta un nuevo modelo de clasificación de patrones, en modo supervisado, cuyo diseño y operación se basa en la función de Heaviside; este Clasificador de Heaviside es del tipo *one shot*, lo cual garantiza que el nuevo modelo carecerá de problemas de convergencia.

Para lograr lo anterior, se proponen dos operaciones originales, llamadas L y C, en cuyo diseño interviene de manera relevante la función de Heaviside. La fase de aprendizaje de patrones del nuevo modelo está sustentada en la operación original L, mientras que la fase de clasificación de patrones descansa en la efectividad de la nueva operación C.

Con objeto de fundamentar teóricamente el Clasificador de Heaviside, se enuncian y demuestran algunos lemas, teoremas y corolarios que exhiben propiedades relevantes de las nuevas operaciones, las cuales inciden directamente en el desempeño del nuevo modelo.

En pruebas experimentales preliminares, cuyos resultados se incluyen en el presente artículo, se ha aplicado el Clasificador de Heaviside en algunos bancos de datos conocidos y utilizados por la comunidad académica internacional. El análisis de los datos generados en las pruebas experimentales, evidencian que el desempeño del nuevo modelo es competitivo, y en algunos casos superior, con respecto a modelos destacados en el estado del arte de temas afines a la Inteligencia Computacional, la Minería de Datos, el Reconocimiento y la Clasificación de Patrones (en modo de aprendizaje supervisado).

Palabras Clave: Inteligencia Computacional, Aprendizaje Supervisado, Clasificación de Patrones, Función de Heaviside

Recepción: 20-12-2014

Aceptación: 23-04-2015

Abstract

In this work, a new pattern classification model, working in supervised mode, is presented. The design and operation of this model is based on the Heaviside function. Also, this Heaviside Classifier is of the one-shot kind, which guarantees that the new model will not have any convergence problem.

In order to achieve the former, two original operations, called L and C, are proposed; the Heaviside function strongly intervenes in the design of these operations. The pattern learning phase of the new model is based on the original operation L, while the pattern classification phase relies on the effectiveness of the new operation C.

With the goal of theoretically substantiating the Heaviside Classifier, some lemmas, theorems, and corollaries are stated and proved. These theorems exhibit relevant properties of the new operations, which in turn affect directly the performance of the new model.

In preliminary experimental tests, included in this paper, the Heaviside Classifier has been applied to some data sets known and used by the international academic community. The data obtained from the experimental tests show that the new model performance is competitive, and in some cases superior, with respect to outstanding models in the state of the art on topics related to Computer Intelligence, Data Mining, Pattern Recognition, and Pattern Classification (in the supervised learning mode).

Keywords: Computational Intelligence, Supervised Learning, Pattern Classification, Step Unit Function.

Introducción

La creación y desarrollo de algoritmos inteligentes tiene relevancia en áreas tales como el Reconocimiento y Clasificación de Patrones, la Minería de Datos y la Inteligencia Computacional (Theodoridis y Koutrombas 2009, 7).

Son cuatro las tareas principales asociadas con los algoritmos inteligentes: recuperación, regresión, clasificación y agrupamiento; las tres primeras tareas se ubican en el paradigma de aprendizaje supervisado, mientras que la última se asocia con el aprendizaje no supervisado (Duda, Hart y Stork 2000, 12).

Dado un algoritmo específico, la base matemática que se incluye en su diseño y operación determina el enfoque al que pertenece. Por ejemplo, hay algoritmos que se basan en el concepto de métrica y en las propiedades de los espacios métricos para hacer la clasificación de los patrones (Cover 1967, 23); de igual manera, han surgido clasificadores basados en el enfoque estadístico-probabilístico, los cuales están basados en el teorema de Bayes (Marques de Sá 2001, 90); y de forma similar, existen clasificadores creados a partir del enfoque neuronal, cuya base conceptual consiste en modelos matemáticos de la neurona del cerebro humano (Du y Swami 2014, 10). Estos son sólo algunos ejemplos de la gran variedad de enfoques que existe para reconocer y clasificar patrones; hay muchos más.

Cada enfoque tiene sus prototipos. Por ejemplo, el Clasificador Bayesiano es el algoritmo más conocido en el enfoque estadístico-probabilístico, mientras que el Perceptrón Multicapa es el prototipo del enfoque neuronal. Al enfoque basado en métricas pertenece uno de los clasificadores más eficaces, no obstante su simplicidad: el clasificador basado en los vecinos más cercanos (*k*-NN, por *k*-Nearest Neighbor). Sin embargo, la mayoría de los algoritmos comparten características de más de un enfoque y de otras áreas de la ciencia. Lo anterior es notorio en las máquinas de soporte vectorial (SVM, siglas de *Support Vector Machines*), en cuyo algoritmo interviene de manera importante el concepto de optimización: las SVM maximizan el margen entre los patrones en el límite de las clases (estos patrones son llamados vectores de soporte) (Cortes y Vapnik 1995, 275).

La propuesta del presente artículo se enfoca en la tarea de clasificación supervisada de patrones; por ello, se asume que se cuenta con un banco de patrones divididos previamente en clases o etiquetas. El modelo aquí propuesto se ubica en el enfoque asociativo de clasificación de patrones (Yáñez Márquez 2002, 30). Al igual como sucede en los demás enfoques, éste también posee una base conceptual que rige el diseño y operación de los algoritmos: aquí, la base conceptual consiste en el establecimiento de asociaciones entre patrones, a través de la aplicación de entes matemáticos específicos para cada algoritmo.

El ente matemático que se ha elegido para el diseño y operación del modelo aquí propuesto, es la función de Heaviside (Abramowitz y Stegun 1972, 1020). Así, en este artículo se propone un nuevo modelo de clasificación de patrones, en modo supervisado, cuyo diseño y operación se basa en la función de Heaviside; este Clasificador de Heaviside es del tipo one shot, lo cual garantiza que el nuevo modelo carecerá de problemas de convergencia.

El resto del artículo está organizado como sigue: en la segunda sección se presenta un resumen conciso del surgimiento y desarrollo del enfoque asociativo de clasificación de patrones, a fin de contextualizar el modelo aquí propuesto. Tomando como base este contenido, en la sección tercera se describe y fundamenta, a través de lemas, teoremas y corolarios, el Clasificador de Heaviside. La cuarta sección, Resultados experimentales, es importante porque da soporte experimental al nuevo modelo, justificando así su inclusión en el estado del arte. Inmediantamente después, se incluye la relevante sección de Discusión y conclusiones, para finalmente cerrar con las Referencias.

Enfoque asociativo del reconocimiento y clasificación de patrones

¿Cuántas veces nos sucede que no podemos recordar la melodía o la letra de una canción de nuestro agrado? Sin embargo, en el mismo instante en que escuchamos un pequeño trozo de la melodía o un fragmento de la letra, somos capaces de recordar la melodía o la letra completas de la canción en cuestión, sin esfuerzo adicional. Similarmente, los seres humanos somos capaces de reconocer el rostro de una persona, no obstante que sólo veamos una parte de la cara, o que la persona se haya puesto peluca o lentes, o se haya quitado la barba o el bigote, por ejemplo.

En relación con este tipo de habilidades, se dice que la mente humana es asociativa (Anderson y Rosenfeld 1990, 271); esto significa que los seres humanos poseemos memoria asociativa, dado que podemos recordar, mediante asociaciones, lo que hemos aprendido. Esto ocurre con objetos, seres vivos, conceptos e ideas abstractas, incluso cuando hay contaminación o alteración (como en el caso de los rostros con lentes o sin barba, o las melodías incompletas).

Algunos grupos de investigación se interesan en crear modelos matemáticos que se comporten como memorias asociativas. Con base en esos modelos, se crean, diseñan y operan sistemas (software o hardware) que sean capaces de aprender y recordar objetos, seres vivos, conceptos e ideas abstractas. Para lograr lo anterior, es preciso representar esos objetos o ideas como patrones, lo cual se realiza usualmente a través de vectores columna de dimensión finita con valores reales, o racionales, o enteros, o booleanos, o mezclas de algunos de ellos (Uriarte-Arcia, López-Yáñez y Yáñez-Márquez 2014, 4).

Típicamente, un modelo asociativo clasificador de patrones consta de dos fases: aprendizaje y clasificación. Si A^k es un patrón perteneciente al conjunto de entrenamiento que pertenece a la clase k, la fase de aprendizaje se esquematiza de la siguiente manera (Figura 1):

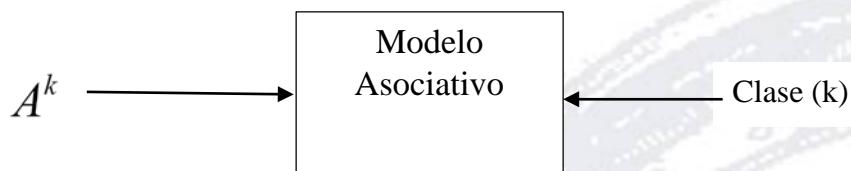


Figura 1. Fase de aprendizaje de un modelo asociativo.

Cuando el modelo asociativo ha concluido la fase de aprendizaje, ya está listo para clasificar patrones de clase desconocida. Si A' es un patrón cuya clase se desconoce, la fase de clasificación del modelo asociativo se esquematiza así (Figura 2):

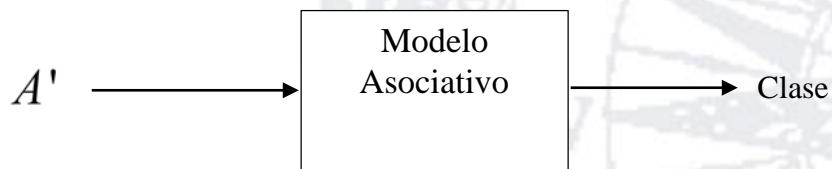


Figura 2. Fase de clasificación de un modelo asociativo.

El primer modelo matemático de memoria asociativa de que se tiene noticia es la Lernmatrix de Steinbuch, desarrollada en 1961 por el científico alemán Karl Steinbuch (Steinbuch 1961, 39); no obstante la importancia de su modelo y las potenciales aplicaciones, el trabajo pasó casi inadvertido. El siguiente modelo de memoria asociativa surgió 8 años después: en 1969, tres científicos escoceses crearon el Correlograph, el cual es un dispositivo óptico elemental capaz de funcionar como una memoria asociativa. En palabras de los autores "el sistema es tan simple, que podría ser construido en cualquier laboratorio escolar de física elemental" (Willshaw, Buneman y Longuet-Higgins 1969, 961).

El año de 1972 fue testigo de la generación de cuatro nuevos modelos asociativos relevantes. Al inicio del año, James A. Anderson de la UCLA publicó su *Interactive Memory* (Anderson 1972, 202); en abril, Teuvo Kohonen, profesor de la Helsinki University of Technology, presentó ante el mundo sus *Correlation Matrix Memories* (Kohonen 1972, 355); tres meses después, Kaoru Nakano de la University of Tokyo, dio a conocer su *Associatron* (Nakano 1972, 382); y en el ocaso del año, Shun-Ichi Amari, profesor de la University of Tokyo, publicó un trabajo teórico donde continuaba con sus investigaciones sobre las *Self-Organizing Nets of Threshold Elements* (Amari 1972, 1200). Los trabajos de Anderson y Kohonen, en conjunto, dieron lugar al *Linear Associator*.

Si 1972 fue el año de los pioneros en el área de las memorias asociativas, 1982 fue el año del científico estadounidense John J. Hopfield. Su artículo de ese año publicado por la National Academy of Sciences, impactó positivamente y trajo a la palestra internacional su modelo asociativo (Hopfield 1982, 2556). La razón es que el modelo Hopfield, que funciona como una red neuronal y también como una memoria asociativa, impulsó de nuevo al área de las redes neuronales, después de sufrir más de una década de estancamiento.

Los trabajos de Hopfield causaron excitación en el mundo de las memorias asociativas y las redes neuronales, y las actividades de investigación en el área se incrementaron. No obstante la cantidad de nuevos modelos propuestos después del éxito de la memoria Hopfield de 1982, no sucedió nada realmente trascendente hasta 1998, cuando aparecieron las memorias asociativas morfológicas (Ritter, Sussner y Diaz-de-Leon 1998, 286). La diferencia fundamental entre estos

modelos y las memorias asociativas clásicas como el Linear Associator y la Memoria Hopfield, es que mientras éstas basan su operación en la suma y multiplicación usuales, las memorias morfológicas se basan en las operaciones morfológicas de dilatación y erosión. Los modelos morfológicos rompieron el esquema utilizado a través de los años en los modelos clásicos, que utilizan operaciones convencionales entre vectores y matrices para la fase de aprendizaje, y suma de productos para recuperar patrones.

Las memorias asociativas morfológicas usan máximos o mínimos de sumas para la fase de aprendizaje, y máximos o mínimos de sumas para la fase de recuperación. Con este nuevo esquema, superaron claramente a las memorias asociativas clásicas.

La aparición, desarrollo, aplicaciones y consolidación de las memorias asociativas morfológicas en 1998 marcó un hito en el campo de las memorias asociativas, porque superaron en prácticamente todos los aspectos de interés, de súbito, a los modelos conocidos. Este hecho se convirtió en una fuerte motivación para la creación y desarrollo de las memorias asociativas Alfa-Beta, en México, durante el año 2002 (Yáñez Márquez 2002, 301).

Las memorias asociativas Alfa-Beta constituyen un modelo alternativo a las memorias asociativas morfológicas. Los entes matemáticos que sustentan a los modelos Alfa-Beta incluyen dos operaciones binarias inventadas ex profeso, cuyos operadores fueron bautizados arbitrariamente con las dos primeras grafías del alfabeto griego: Alfa y Beta. Alfa se usa en la fase de aprendizaje, y Beta en la fase de clasificación.

Sean los conjuntos $W = \{0,1\}$ y $Y = \{0,1,2\}$. Los operadores Alfa(α) y Beta(β) se definen en forma tabular, como se ilustra en las siguientes tablas:

$$\alpha : W \times W \rightarrow Y$$

Tabla 1. Operación alfa.

$w \in W$	$y \in W$	$\alpha(w, y)$
0	0	1
0	1	0
1	0	2
1	1	1

$$\beta : Y \times W \rightarrow Y$$

Tabla 2. Operación beta.

$y \in Y$	$w \in W$	$\beta(y, w)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

A más de una década de su creación, los modelos Alfa-Beta se han consolidado, y las actividades del Grupo de Investigación Alfa-Beta giran en torno a estos modelos, sus extensiones y sus aplicaciones. El enclave académico del Grupo es el Centro de Investigación en Computación del Instituto Politécnico Nacional, y los nuevos modelos basados en los operadores originales se desarrollan como tesis de doctorado en ciencias de la computación, además de que se generan publicaciones nacionales e internacionales en revistas arbitradas.

Así se han creado, entre otros modelos, las redes neuronales Alfa-Beta sin pesos (Argüelles Cruz 2007, 37), los modelos Alfa-Beta difusos (Sánchez Garfias 2009, 33) y las memorias asociativas Alfa-Beta bidireccionales (Acevedo-Mosqueda, Yáñez-Márquez y López-Yáñez 2007, 25). Tomando como base los operadores Alfa y Beta originales, se creó el Clasificador Gamma, en cuyo algoritmo interviene un código especial diseñado también en el seno del Grupo: el código Johnson-Möbius (López-Yáñez, Argüelles-Cruz, Camacho-Nieto y Yáñez-Márquez 2011, 696).

El advenimiento del algoritmo del Clasificador Gamma abrió una veta muy interesante de nuevas aplicaciones de éste y de algunos algoritmos híbridos Alfa-Beta. Se exploran áreas de aplicación tan disímbolas como la predicción de contaminantes atmosféricos y los componentes de polución arrojados por el escape de un automóvil, procesos de predicción en ingeniería de software, detección de fracturas craneales en radiografías, reconocimiento de conceptos en lattices y reconocimiento de voz emotiva, entre otras.

Recientemente, los modelos asociativos Alfa-Beta han arrojado resultados muy prometedores al realizar predicciones en series de tiempo correspondientes a la producción de pozos petroleros (López Yáñez, Sheremetov y Yáñez Márquez 2014, 25); lo mismo que en el diseño e implementación de sistemas de soporte para la toma de decisiones en el ámbito médico (Aldape-Pérez, Yáñez-Márquez, Camacho-Nieto y Argüelles-Cruz 2012, 289).

El Grupo Alfa-Beta está innovando constantemente. En este contexto, se ha iniciado un conjunto de proyectos y actividades de investigación que involucran la fusión o extensión de los operadores Alfa y Beta con algunas funciones especiales. Es el caso de la función de Heaviside, cuando el primer autor de este artículo se percató de que con esta función era posible representar extensiones de los operadores Alfa y Beta, más allá del sistema binario.

Mediante un proceso inductivo, se procedió a realizar experimentos de aprendizaje y clasificación de patrones en el sistema numérico de base 3, luego en el de base 4 y después en el de base 5. Los resultados fueron buenos, porque en algunos casos el desempeño de los nuevos clasificadores era superior al modelo original en binario.

Acto seguido, el equipo de investigación procedió a generar las expresiones para las fases de aprendizaje y clasificación de patrones, extendido a un sistema numérico de base b , donde b es un entero mayor que 1. A partir de los buenos resultados, se procedió a crear un nuevo modelo clasificador de patrones basado en la función de Heaviside: el Clasificador de Heaviside, cuya descripción y fundamentación matemática, a través de lemas, teoremas y corolarios, se presenta a continuación.

Clasificador de Heaviside

Algunas hipótesis acerca del modelo propuesto

El modelo que se está proponiendo es del tipo *one shot*, y por ello la clasificación se realiza en un solo paso. Una premisa que subyace al presente trabajo, es que esta característica permitirá superar las limitaciones que exhiben los modelos del estado del arte, los cuales en su mayoría son iterativos.

Hipótesis 1.- Dado que el Clasificador de Heaviside es supervisado, se asume que se tiene acceso a un banco de patrones clasificado previamente en un número finito de clases, al estilo de los que se incluyen en el prestigioso repositorio de la UCI (Lichman,2013).

Hipótesis 2.- Los patrones pueden ser representados como vectores de dimensión finita, cuyas componentes son números reales. Si hay números negativos, es posible aplicar una transformación para obtener sólo componentes reales no negativas. Además, es posible truncar o redondear estos números reales no negativos, a fin de obtener vectores de dimensión finita con componentes racionales no negativas; estos valores se representan mediante números racionales que contienen un número no negativo y un número finito de decimales.

Hipótesis 3.- Es posible encontrar un escalamiento que permita transformar esas componentes racionales con un número finito de decimales, en números enteros no negativos.

Las tres hipótesis previas 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, y última, 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.

Después de aplicar al banco de datos original los procesos operativos involucrados en cada una de las hipótesis previas, se tendrá disponible un banco de vectores con componentes enteras no

negativas. La representación simbólica de estas componentes enteras no negativas, se realizará respecto de una base b de algún sistema numérico posicional.

Para ilustrar un poco lo mencionado en las hipótesis anteriores, considérese el siguiente conjunto de patrones, en donde cada fila representa un patrón diferente:

0.222222	0.074074	0.962963	0	2.88889	0	-2.88889	5.77778	-2.88889	2.88889	1	-2.0944
2	2.44444	23.1111	21.6667	30.1111	17.5556	-4.33333	21	-16.6667	30.1111	0.415524	-1.74074
5.66667	5.79271	58.3333	51.8889	72.3333	50.7778	-19.3333	42	-22.6667	72.3333	0.299654	-2.03587
1.61111	2.41852	14.9259	11.8889	13.7778	19.1111	-9.11111	-3.44444	12.5556	19.1111	0.386456	2.36215

El valor mínimo del patrón 1 es -2.88889, al restar este valor al patrón obtenemos el siguiente patrón transformado:

3.111112	2.962964	3.851853	2.88889	5.77778	2.88889	0	8.66667	0	5.77778	3.88889	0.79449
----------	----------	----------	---------	---------	---------	---	---------	---	---------	---------	---------

En el patrón 2 el valor mínimo es -16.6667; al restar este valor de las componentes del patrón se obtiene lo siguiente:

18.6667	19.11114	39.7778	38.3334	46.7778	34.2223	12.33337	37.6667	0	46.7778	17.082224	14.92596
---------	----------	---------	---------	---------	---------	----------	---------	---	---------	-----------	----------

En el patrón 3 el valor mínimo es -22.6667; al restar este valor de las componentes del patrón se obtiene el siguiente patrón transformado:

28.33337	28.45941	81	74.5556	95	73.4445	3.3334	64.6667	0	95	22.966354	20.63083
----------	----------	----	---------	----	---------	--------	---------	---	----	-----------	----------

Finalmente, en el patrón 4 el valor mínimo es -9.11111; al restar este valor del patrón se obtiene el siguiente patrón transformado:

10.72222	11.52963	24.03701	21.00001	22.88891	28.22221	0	5.66667	21.66671	28.22221	9.497566	11.47326
----------	----------	----------	----------	----------	----------	---	---------	----------	----------	----------	----------

Por lo tanto, el banco de datos transformado queda de la siguiente forma:

3.111112	2.962964	3.851853	2.88889	5.77778	2.88889	0	8.66667	0	2.88889	3.88889	0.79449
18.6667	19.11114	39.7778	38.3334	46.7778	34.2223	12.33337	37.6667	0	46.7778	17.082224	14.92596

28.33337	28.45941	81	74.5556	95	73.4445	3.3334	64.6667	0	95	22.966354	20.63083
10.72222	11.52963	24.03701	21.00001	22.88891	28.22221	0	5.66667	21.66671	28.22221	9.497566	11.47326

Al revisar el banco de datos transformado se obtiene que el máximo número de dígitos decimales que tienen las componentes es de 6, por lo tanto todas las componentes de todos los patrones del banco de datos serán multiplicadas por 10^6 . El banco de datos resultante queda de la siguiente forma:

3111112	2962964	3851853	2888890	5777780	2888890	0	8666670	0	5777780	3888890	794490
18666700	19111140	39777800	38333400	46777800	34222300	12333370	37666700	0	46777800	17082224	14925960
28333370	28459410	81000000	74555600	95000000	73444500	3333400	64666700	0	95000000	22966354	20630830
10722220	11529630	24037010	21000010	22888910	28222210	0	5666670	21666710	28222210	9497566	11473260

El banco de datos ha quedado en términos de números enteros no negativos, lo cual satisface las primeras tres hipótesis. A partir de esta representación es posible transformar las componentes, y por lo tanto el banco de datos, en términos de un sistema numérico base b en donde $b > 1$, lo cual satisface la hipótesis 4.

La función de Heaviside y los métodos de validación

Aquí es donde entra de lleno la Función de Heaviside. No obstante que en la literatura de las funciones especiales la Función de Heaviside aparece definida de diversas maneras, en este trabajo se adoptará la definición incluida en una de las referencias más relevantes y respetadas en el área: el Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables de Abramowitz y Stegun (Abramowitz y Stegun 1972, 1020), la cual es avalada por importantes obras como (Bracewell 2000, 61).

Definición 1: Sea x un número real. La Función de Heaviside de x se define mediante la siguiente expresión:

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

Asumiendo que ya se cuenta con un banco de vectores con componentes enteras no negativas (al que llamaremos B), representadas éstas respecto de una base b de algún sistema numérico

posicional, se procede a aplicar un método de validación estándar, tal como hold-out, leave-one-out o k-fold-cross-validation (el cual es implementado normalmente con el parámetro k=10) (Witten y Frank 2011, 152).

De la aplicación de uno de estos métodos de validación de modelos en el banco de vectores B, se obtienen dos conjuntos de patrones, mutuamente exclusivos: el conjunto de aprendizaje (al que se denominará A) y el conjunto de prueba (que se nombrará como P) cuya unión es, precisamente, el conjunto B.

Operaciones originales L y C

Para su funcionamiento, el Clasificador de Heaviside consta de dos operaciones originales: una para la fase de aprendizaje (a la que llamamos L) y otra para la fase de clasificación de patrones (a la que llamamos C).

Considerando que el modelo de esta propuesta trabaja con operaciones propias de las lattices y estructuras algebraicas ordenadas (Blyth 2005, 20), existen dos modos operativos del Clasificador de Heaviside: el modo que corresponde al supremo (HS) y el modo que corresponde al ínfimo (HI).

Mientras que la operación de aprendizaje L es igual para ambos modos, no sucede lo mismo con la operación de clasificación C, la cual varía con cada modo; específicamente, para el modo HI se ocupa sólo uno de los términos de la operación genérica C.

Nota 1: Dado que el modo HS es el dual del modo HI, en este artículo sólo se describe lo concerniente al modo HI; el modo HS se puede obtener por dualidad.

Definición 2: Si A_i^k y A_j^k son la i -ésima y la j -ésima componentes del k -ésimo patrón del conjunto A de aprendizaje, respectivamente, y si M_{ij} y P_j^k son la ij -ésima componente del modelo del Clasificador de Heaviside M y la j -ésima componente del k -ésimo patrón del conjunto

P de prueba, respectivamente, el modo operativo HI del Clasificador de Heaviside utiliza las siguientes operaciones de aprendizaje y clasificación de patrones:

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

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

Definición 3: Sean A^k y A^d dos patrones que pertenecen al conjunto de aprendizaje A en un Clasificador de Heaviside en modo operativo HI. Se define el producto externo bajo L de ambos patrones, y se denota como $A^k A^d$, a una matriz cuya ij -ésima componente se calcula así:

$$L(A_i^k, A_j^d)$$

Siendo A_i^k y A_j^d la i -ésima y la j -ésima componentes del k -ésimo y d -ésimo patrón del conjunto A de aprendizaje, respectivamente.

Nota 2: Si la dimensión de los patrones de A es n, entonces el producto externo bajo L es una matriz de dimensiones $n \times n$.

Nota 3: Es posible que $k=d$.

Fases de aprendizaje y clasificación de patrones

Definición 4: Fase de Aprendizaje. La ij -ésima componente del modelo del Clasificador de Heaviside M, en modo operativo HI dado un conjunto de aprendizaje A, se define por la siguiente expresión:

$$M_{ij} = \min L(A_i^k, A_j^k)$$

Nota 4: El valor mínimo se toma con respecto al índice k, que va de 1 a $\text{card}(A)$.

Definición 5: Fase de Clasificación. Sea M un Clasificador de Heaviside en modo operativo HI, y sea P^d un patrón del conjunto de prueba P de dimensión n; la clase que se le asigna a este patrón es la clase del patrón que se recupera al realizar la siguiente operación con la que se obtiene la i -ésima componente del patrón recuperado R:

$$R_i = \max[C(M_{ij}, P_j^d)]$$

Nota 5: El valor máximo se toma con respecto al índice j, que va de 1 a n.

Nota 6: Por definiciones 4 y 5, se tiene: $R_i = \max[C(\min L(A_i^k, A_j^k), P_j^d)]$

Fundamentación teórica del Clasificador de Heaviside

Con base en las definiciones previas, se diseñaron y se demostraron los lemas, teoremas y corolarios que describen teóricamente el funcionamiento del Clasificador de Heaviside, sus alcances y sus limitaciones.

Teorema 1.- Sea $b \in Z$ tal que $b > 1$, entonces $\forall x \in Z$ tal que $0 \leq x \leq b - 1$ se cumple $L(x, x) = b - 1$.

Demostración:

Por hipótesis y por la definición 1 se tiene:

$$H[x + x + b] = H(2x + b) = 1, \text{ dado que } 2x + b \geq b > 0$$

$$\text{Y además se cumple que: } [x + b] - [x + 1] = b - 1$$

Al realizar las operaciones, resulta lo siguiente:

$$L(x, x) = [H(x + x + b)] \cdot [(x + b) - (x + 1)] = (1) \cdot (b - 1) = b - 1$$

q.e.d.

Corolario 1.1:

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

Demostración:

Dado que se cumple la hipótesis del Teorema 1, éste se puede aplicar directamente, haciendo $x=0$:

$$L(0, 0) = L(x, x) = b - 1$$

q.e.d.

Corolario 1.2:

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

Demostración:

Dado que se cumple la hipótesis del Teorema 1, éste se puede aplicar directamente, haciendo $x=b-1$:

$$L(b-1, b-1) = L(x, x) = b-1 = L(0,0)$$

q.e.d.

Lema 1.- Un Clasificador de Heaviside M en modo operativo HI contiene sólo valores $b-1$ en su diagonal principal.

Demostración:

De acuerdo con la definición 4, las entradas de la diagonal principal de M se calculan con la siguiente expresión:

$$M_{ii} = \min L(A_i^k, A_i^k)$$

Pero por el Teorema 1, el valor de $L(A_i^k, A_i^k) = b-1$ es constante y no depende del índice k ; por ello, se tiene $M_{ii} = \min L(A_i^k, A_i^k) = L(A_i^k, A_i^k) = b-1$ para todo valor de i .

q.e.d.

Lema 2.- Sea $b \in \mathbb{Z}$ tal que $b > 1$, y sean $x, y \in \mathbb{Z}$ tales que $0 \leq x \leq b-1$ y $0 \leq y \leq b-1$. Entonces se cumple la siguiente expresión: $C[L(x, y), y] = x$.

Demostración:

Por definición 2:

$$L(x, y) = [H(x+y+b)] \cdot [(x+b)-(y+1)] = (1) \cdot [(x+b)-(y+1)] = x+b-y-1$$

$$C[L(x, y), y] = H[(L(x, y)+2)-(b-y)] \cdot H[(2b-1)-(L(x, y)+y)] \cdot [(L(x, y)+1)-(b-y)]$$

Por otro lado:

$$H[(L(x, y)+2)-(b-y)] = H[(x+b-y-1+2)-(b-y)] = H[x+b-y-1+2-b+y] = H[x+1] = 1$$

$$H[(2b-1)-(L(x, y)+y)] = H[2b-1-x-b+y+1-y] = H[b-x] = 1$$

$$(L(x, y)+1)-(b-y) = x+b-y-1+1-b+y = x$$

Sustituyendo en la expresión de C:

$$C[L(x, y), y] = (1) \cdot (1) \cdot (x) = x$$

q.e.d.

Lema 3.- Sea M un Clasificador de Heaviside en modo operativo HI, y sea A^d un patrón del conjunto de aprendizaje A, entonces:

$$R_i = \max[C(M_{ij}, A_j^d)] \leq A_i^d$$

Demostración:

Por definición 4, se tiene que:

$$M_{ij} = \min L(A_i^k, A_j^k)$$

Pero es claro que para cualquier valor fijo d del índice k se cumple la siguiente desigualdad:

$$\min L(A_i^k, A_j^k) \leq L(A_i^d, A_j^d)$$

Aplicando la operación C con A_j^d en ambos lados de la desigualdad y tomando el máximo con respecto al índice j:

$$\max C[\min L(A_i^k, A_j^k), A_j^d] \leq \max C[L(A_i^d, A_j^d), A_j^d]$$

Por nota 6, se tiene que si $P^d = A^d$:

$$R_i = \max[C(\min L(A_i^k, A_j^k), A_j^d)]$$

Y por transitividad con la expresión previa:

$$R_i \leq \max C[L(A_i^d, A_j^d), A_j^d]$$

Por Lema 2, se tiene el resultado, porque max es con respecto al índice j, no al i:

$$R_i \leq A_i^d$$

q.e.d.

Lema 4.- Sea M un Clasificador de Heaviside en modo operativo HI, y sea A^d un patrón del conjunto de aprendizaje A, entonces:

$$R_i \geq A_i^d$$

Demostración:

Al hacer $j=i$ en la expresión $R_i = \max[C(M_{ij}, A_j^d)]$ se tiene:

$$R_i = \max[C(M_{ij}, A_j^d)] \geq C(M_{ii}, A_i^d)$$

Pero por Lema 1, para cualquier valor de k (incluyendo d) se tiene:

$$M_{ii} = \min L(A_i^k, A_i^k) = L(A_i^k, A_i^k) = b - 1 = L(A_i^d, A_i^d)$$

Por lo que:

$$R_i \geq C(L(A_i^d, A_i^d), A_i^d)$$

Y finalmente, por Lema 2:

$$R_i \geq A_i^d$$

q.e.d.

Teorema 2.- Un Clasificador de Heaviside M en modo operativo HI, clasifica correctamente cualquier patrón A^d del conjunto de aprendizaje A .

Demostración:

Por Lemas 3 y 4, para un valor de índice de patrón fijo d , se cumplen simultáneamente estas dos desigualdades: $R_i \leq A_i^d$ y $R_i \geq A_i^d$ para todos los valores del índice i , el cual representa la dimensión de los patrones de aprendizaje.

Las dos desigualdades simultáneas equivalen a la igualdad $R^d = A^d$, por lo que el patrón recuperado es precisamente A^d , lo cual significa que se le asigna su propia clase; es decir, es clasificado correctamente.

q.e.d.

En el teorema 2 el índice d se fijó arbitrariamente, por lo que la conclusión es válida para cualquier patrón del conjunto de aprendizaje A .

De aquí surge una conclusión MUY RELEVANTE, en relación con el funcionamiento del Clasificador de Heaviside en modo operativo HI: **EL CLASIFICADOR DE HEAVISIDE EN MODO OPERATIVO HI CLASIFICA CORRECTAMENTE TODOS LOS PATRONES DEL CONJUNTO DE APRENDIZAJE.**

¿Por qué es tan relevante esta conclusión? Por la sencilla razón de que, entre la mirada de algoritmos inteligentes para clasificación de patrones que existen en el estado del arte, sólo una mínima cantidad de ellos clasifican correctamente todo el conjunto de aprendizaje.

Para el modo operativo HS del Clasificador de Heaviside, es posible demostrar un teorema similar, lo que potencia fuertemente las bondades y alcances del nuevo modelo.

Pero este Clasificador también posee limitaciones, como cualquier obra del quehacer humano, en los ámbitos cotidiano, artístico, educativo, industrial o científico: por ejemplo, ¿qué ocurre si el conjunto de aprendizaje A contiene patrones con diferentes dimensiones? Pues ocurre que al tomar los máximos sobre el índice j de la dimensión, las operaciones fallarían rotundamente y los teoremas 1 y 2 no se cumplirían.

Una manera adecuada, además del método basado en la teoría matemática, de visualizar los alcances y limitaciones de este modelo, es la realización de experimentos con patrones extraídos de la realidad cotidiana del quehacer humano, en sus diferentes ámbitos.

Resultados experimentales

Se hizo un estudio comparativo entre el modelo de clasificación propuesto contra nueve clasificadores reconocidos en el estado del arte del Reconocimiento de Patrones, los modelos comparados fueron los siguientes:

1. Memorias Morfológicas Min (Ritter, Sussner, y Diaz-de-Leon, 1998, p. 285) (este modelo de memorias asociativas se basa en las dos operaciones básicas de la morfología matemática),

2. Clasificador 1NN (Cover y Hart, 1967, p. 21) (este clasificador asigna a los patrones de prueba la clase del vecino más cercano a partir de una función de distancia),
3. K-NN con K=3 (Duda, Hart, y Stork, 2001, p. 177) (este modelo asigna a los patrones de prueba la clase mayoritaria entre sus tres vecinos más cercanos, en caso de tener un empate la clase será asignada de forma aleatoria),
4. Naïve Bayes (Duda, Hart y Stork, 2001, p. 52) (este modelo de clasificación está basado en la teoría de la probabilidad, se asigna la clase que sea considerada como la más probable a partir del patrón de entrada),
5. Perceptrón Multicapa (Rumelhart, Hinton, y Williams, 1986, p. 533) (uno de los modelos de redes neuronales *feedforward* más conocidos en donde se utiliza el algoritmo de entrenamiento conocido como *backpropagation*),
6. Máquinas de Soporte Vectorial (Cortes y Vapnik, 1995, p. 274) (técnica muy popular desarrollada por Vapnik hace dos décadas que busca establecer el margen óptimo entre dos clases),
7. SMO (Platt, 1998, p. 44) (Sequential Minimal Optimization: modelo empleado para resolver el problema de la programación cuadrática durante el entrenamiento de máquinas de soporte vectorial),
8. Redes Neuronales de Base Radial (Broomhead y Lowe, 1988, p. 326) (tipo especial de redes neuronales que utilizan funciones de base radial como funciones de activación),
9. Función Logística (Tatsuoka, 1985, p. 55) (modelo de regresión logística multinomial).

Para realizar la comparación entre el clasificador de Heaviside y los clasificadores enlistados anteriormente, se eligió el uso de una técnica estadística no paramétrica denominada Test de Wilcoxon. A continuación se presentan algunas de las características principales de esta prueba.

Prueba no paramétrica de Wilcoxon

El Test de Wilcoxon (Wilcoxon, 1945, p. 80) se utiliza para comparar diferencias de medias en dos muestras relacionadas. Sin embargo, un caso particular de esta problemática es la comparación de los resultados de la ejecución de dos algoritmos sobre el mismo conjunto de bancos de datos. El test determina, dentro de un intervalo de confianza, si se acepta o no se acepta la hipótesis de igualdad de medias. Si no se acepta, se puede decir que existen diferencias

significativas entre los algoritmos a comparar (Derrac, García, Molina, y Herrera, 2011, p. 6). El test consiste en lo siguiente:

Sea d_i la diferencia del desempeño de dos algoritmos en alguno de n posibles problemas, dichas diferencias son evaluadas de acuerdo con sus valores absolutos. Sea R^+ la suma de los rankings o rangos en los cuales el desempeño del primer algoritmo es mayor al desempeño del segundo y sea R^- la suma de los rangos en los cuales el segundo algoritmo tiene mejores resultados que el primero; los rangos correspondientes para $d_i = 0$ se dividen uniformemente entre las sumas, se ignorará uno de los rangos en caso de que la cantidad de sumas sea impar. Las sumas R^+ y R^- se calculan de la siguiente forma:

$$R^+ = \sum_{d_i > 0} rank(d_i) + \frac{1}{2} \sum_{d_i = 0} rank(d_i)$$

$$R^- = \sum_{d_i < 0} rank(d_i) + \frac{1}{2} \sum_{d_i = 0} rank(d_i)$$

Sea T la menor de las sumas $T = \min(R^+, R^-)$, si T es menor que la distribución de Wilcoxon para n grados de libertad (Zar, 1998, p. 165), la hipótesis nula de igualdad es rechazada y por lo tanto un algoritmo dado es significativamente diferente que el otro con un valor p asociado.

Para considerar como válido el resultado de la prueba de Wilcoxon es importante tomar en cuenta que los valores de significancia estadística deben estar entre $\alpha = 0.01$ y $\alpha = 0.1$; el valor $\alpha = 0.05$ permite establecer un intervalo de confianza entre el 90% y 99%, típicamente 95%. Si el valor de la probabilidad arrojado (p) es menor que alfa, se dice que se rechaza la hipótesis nula (porque cae en la zona de rechazo del test), y por ende se acepta la hipótesis alternativa (que es la que plantea que sí hay diferencias).

La prueba de Wilcoxon asume la commensurabilidad de las diferencias, pero sólo de forma cualitativa: es decir, las diferencias grandes entre los desempeños cuentan más; sin embargo, las magnitudes absolutas siguen siendo ignoradas. Desde el punto de vista estadístico, la prueba es

más segura dado que no asume la distribución normal, además de que no se ve afectada por la presencia de outliers, o patrones atípicos (Derrac et al., 2011, p. 7).

Para cumplir con la cantidad mínima de comparaciones u observaciones requeridas por las pruebas de Wilcoxon, se eligieron los siguientes bancos de datos, algunos de los cuales fueron tomados del repositorio UCI (Lichman, 2013), y otros más fueron extraídos del repositorio KEEL (Alcalá-Fernandez et al., 2011, p. 256):

1. Appendicitis (Datos de pacientes para estimar si padece apendicitis o no),
2. Australian (banco de datos relacionado con solicitudes para tarjetas de crédito).
3. Banana (banco de datos formado a partir de objetos agrupados en grupos con forma de banana).
4. Bupa (banco de datos que analiza desórdenes en el hígado),
5. Cleveland (banco de datos relacionado con enfermedades cardíacas),
6. Columna Vertebral (banco de datos de características biomecánicas para determinar posibles problemas en la columna vertebral),
7. Glass (banco de datos para la clasificación de seis tipos de vidrio que pueden ser hallados en una escena de crimen),
8. Haberman (datos de un estudio que se realizó a pacientes que siguen con vida después de una cirugía para cáncer de mama),
9. Heart (detecta la presencia o ausencia de males cardíacos).
10. Ionosphere: (determinar si una señal transmitida es buena o mala).
11. Iris Plant (el mundialmente conocido banco de datos de la Iris Plant el cual contiene mediciones de tres tipos diferentes de plantas de Iris),
12. Magic (simular el registro de partículas gamma en un telescopio a partir de un conjunto de imágenes).
13. New Thyroid (banco de datos utilizado en la clasificación de pacientes normales, que padecen hipotiroidismo o que padecen hipertiroidismo),
14. Phoneme (banco de datos para distinguir los sonidos nasales y orales).
15. Pima Indians Diabetes (banco de datos relacionados con la incidencia de la diabetes en mujeres jóvenes de la tribu indígena de los Pima),

16. Ring (banco de datos artificial, propone una distribución multivariada de patrones).
17. Segment-Test (Banco de datos utilizado en la determinación de regiones en imágenes digitales),
18. Unbalanced (Una versión desbalanceada de un banco de datos que describe formas de vehículos),
19. Wine (banco de datos relacionado con el reconocimiento de tres tipos diferentes de vinos),
20. Winsconsin (Banco de datos relacionado con el diagnóstico del cáncer de mama).

El clasificador de Heaviside fue implementado en Matlab, mientras que las Memorias Morfológicas Min fueron implementadas en Java y para el resto de los modelos se empleó la herramienta Weka (Hall et al., 2009, p. 10). En la tabla 3 se presenta el desempeño promedio de los algoritmos (calculado con el método de validación K Fold Cross Validation con K=10), en los veinte conjuntos de datos.

Tabla 3. Desempeño de los modelos de clasificación en los Bancos de Datos seleccionados.

Datasets	C. de Heaviside	Memorias morfológicas min	IB1	IBK K=3	Naive Bayes	Multilayer Perceptron	SVM	SMO	RBF	Logistic
Appendicitis	88.57	84.02	82.07	83.96	85.84	85.84	80.18	87.73	83.96	86.79
Australian	76.81	76.81	80.72	85.79	76.95	84.34	55.51	85.5	84.06	86.81
Banana	78.37	78.37	87.24	88.39	61.11	77.09	90.24	55.16	74.26	56.07
Bupa	96.84	96.84	62.89	61.73	55.36	71.59	59.42	58.26	64.34	68.11
Cleveland	96.12	96.12	57.79	55.77	54.78	53.13	54.12	60.06	57.09	61.38
Columna Vertebral	96.77	95.48	78.38	71.29	83.22	85.48	48.38	74.51	83.22	85.8
Glass	91.9	91.9	70.56	71.96	48.59	67.75	68.69	56.07	63.55	64.01
Haberman	76.53	70.34	68.3	71.56	76.14	68.28	73.52	73.52	73.85	73.85
Heart	78.51	71.41	74.81	77.4	83.7	82.22	55.55	83.7	83.7	82.59
Ionosfera	92.22	92.22	86.03	84.9	82.05	92.02	93.44	87.75	90.31	88.03
Iris plant	97.33	96.66	95.33	95.33	96	97.33	96.66	96	95.33	96
Magic	81.14	81.14	80.93	83.2	72.68	85.87	65.88	79.15	78.42	79.07
New-Thyroid	91.77	91.77	97.21	93.48	96.74	96.74	75.81	89.76	95.34	96.74
Phoneme	90.93	90.87	90.19	88.63	76.03	80.97	83.77	77.23	77.7	75.11

Pima Indians diabetes	96.98	96.98	70.18	72.65	76.3	75.39	65.1	77.34	75.39	77.29
Ring	90.43	90.43	75.05	71.75	97.97	91.06	50.46	76.43	97.82	75.81
Segment-test	94.56	94.32	94.69	92.59	86.41	94.07	41.11	92.22	88.39	93.82
Unbalanced	87	87	97.66	98.36	90.77	98.13	98.59	98.59	98.59	96.49
Wine	72.64	74.28	94.94	94.94	97.19	97.19	44.38	98.31	98.31	97.19
Wisconsin	97.15	97.15	95.27	96.42	96.13	95.71	95.99	96.71	95.99	96.56

A partir de estos valores se realizó el test de Wilcoxon. Los resultados de la prueba de rangos con signo de Wilcoxon se presentan en la Tabla 4; mientras que en la Tabla 5 se presentan los valores estadísticos de contraste obtenidos de la prueba de Wilcoxon.

Tabla 4. Prueba de rangos con signo de Wilcoxon.

		N	Rango promedio	Suma de rangos
Memorias morfológicas vs C. de Heaviside	Rangos negativos Rangos positivos Empates Total	7 ^a 1 ^b 12 ^c 20	4,43 5,00	31,00 5,00
IB1 vs C. de Heaviside	Rangos negativos Rangos positivos Empates Total	14 ^d 6 ^e 0 ^f 20	10,86 9,67	152,00 58,00
IBK K=3 vs C. de Heaviside	Rangos negativos Rangos positivos Empates Total	14 ^g 6 ^h 0 ⁱ 20	10,64 10,17	149,00 61,00
Naive Bayes vs C. de Heaviside	Rangos negativos Rangos positivos Empates Total	14 ^j 6 ^k 0 ^l 20	11,57 8,00	162,00 48,00
Multilayer Perceptron vs C. de Heaviside	Rangos negativos Rangos positivos Empates Total	12 ^m 7 ⁿ 1 ^o 20	10,25 9,57	123,00 67,00

SVM vs C. de Heaviside	Rangos negativos	17 ^p	11,29	192,00
	Rangos positivos	3 ^q	6,00	18,00
	Empates	0 ^r		
	Total	20		
SMO vs C. de Heaviside	Rangos negativos	16 ^s	10,19	163,00
	Rangos positivos	4 ^t	11,75	47,00
	Empates	0 ^u		
	Total	20		
RBF vs C. de Heaviside	Rangos negativos	14 ^v	10,14	142,00
	Rangos positivos	6 ^w	11,33	68,00
	Empates	0 ^x		
	Total	20		
Logistic - C. de Heaviside	Rangos negativos	15 ^y	10,40	156,00
	Rangos positivos	5 ^z	10,80	54,00
	Empates	0 ^{aa}		
	Total	20		

- a. Memorias morfológicas < C. de Heaviside
- b. Memorias morfológicas > C. de Heaviside
- c. Memorias morfológicas = C. de Heaviside
- d. IB1 < C. de Heaviside
- e. IB1 > C. de Heaviside
- f. IB1 = C. de Heaviside
- g. IBK K=3 < C. de Heaviside
- h. IBK K=3 > C. de Heaviside
- i. IBK K=3 = C. de Heaviside
- j. Naive Bayes < C. de Heaviside
- k. Naive Bayes > C. de Heaviside
- l. Naive Bayes = C. de Heaviside
- m. Multilayer Perceptron < C. de Heaviside
- n. Multilayer Perceptron > C. de Heaviside
- o. Multilayer Perceptron = C. de Heaviside
- p. SVM < C. de Heaviside
- q. SVM > C. de Heaviside
- r. SVM = C. de Heaviside
- s. SMO < C. de Heaviside
- t. SMO > C. de Heaviside
- u. SMO = C. de Heaviside
- v. RBF < C. de Heaviside
- w. RBF > C. de Heaviside
- x. RBF = C. de Heaviside

- y. Logistic < C. de Heaviside
 z. Logistic > C. de Heaviside
 aa. Logistic = C. de Heaviside

Tabla 5. Estadísticos de contraste de la prueba de Wilcoxon^a.

	MM Min	IB1	IBK K=3	NB	MLP	SVM -	SMO -	RBF -	Logisti c -
Z	-1,820 ^b	-1,755 ^b	-1,643 ^b	-2,128 ^b	-1,127 ^b	-3,248 ^b	-2,165 ^b	-1,381 ^b	-1,904 ^b
Sig. asintót. (bilateral)	,069	,079	,100	,033	,260	,001	,030	,167	,057

a. Prueba de los rangos con signo de Wilcoxon

b. Basado en los rangos positivos.

De acuerdo con los valores presentados en las tablas 3, 4 y 5, es posible afirmar que el Clasificador de Heaviside supera a las Memorias Morfológicas Min, al clasificador 1NN, al clasificador Naive Bayes, a las SVM, al clasificador SMO y a la función logística con un 90% de confianza. Cabe destacar que con un 95% de confianza el clasificador de Heaviside supera al Naive Bayes, a las SVM y al modelo SMO.

Discusión y conclusiones

En este trabajo se ha presentado un nuevo modelo de clasificación supervisada de patrones, el cual está basado en la función de Heaviside; es muy importante mencionar que tras una extensa investigación, en el estado del arte no se encontraron modelos inteligentes de clasificación de patrones cuya base teórica esté dada por la función de Heaviside.

El modelo de clasificador propuesto destaca por el hecho de que trabaja con patrones enteros no negativos, los cuales pueden ser obtenidos a partir de patrones con valores reales mediante un proceso de transformación. Por ello, es posible representar las componentes de los patrones en cualquier sistema numérico con base mayor que 1.

De igual forma el modelo propuesto en este trabajo, presenta dos operaciones originales, las cuales están basadas en la función de Heaviside: la operación L de aprendizaje, y la operación C

de clasificación de patrones. En este artículo se presentaron los fundamentos matemáticos de ambas operaciones y algunas de sus propiedades más importantes.

Después de probar el Clasificador de Heaviside en varios bancos de datos y de comparar sus resultados con los obtenidos por algunos de los modelos más importantes de clasificación de patrones, se destaca el hecho de que el Clasificador de Heaviside es bastante competitivo. También es importante resaltar que se sustentó la validez de los experimentos realizados mediante una prueba no paramétrica.

Además, se está trabajando en ajustes al algoritmo que podrían incrementar de manera importante su rendimiento y performance. Esto posibilitaría su aplicación en problemas de clasificación de patrones donde se requiere un elevado rendimiento y una gran eficacia. Se podría mejorar la eficiencia a través de la implementación del algoritmo en FPGA.

Agradecimientos

Los autores desean agradecer al Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP y CIC), al CONACyT y al SNI por su apoyo en el desarrollo y publicación de este trabajo. Merece una mención especial la Dra. Yenny Villuendas-Rey, profesora adscrita a la Facultad de Informática de la Universidad de Ciego de Ávila, Cuba, por su asesoría y valioso apoyo en los temas relacionados con la prueba de Wilcoxon.

Referencias

- Abramowitz, M., e Irene A. Stegun. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Nueva York: Dower Publications.
- Acevedo-Mosqueda, María Elena, Cornelio Yáñez-Márquez e Itzamá López-Yáñez. (2007). Alpha-Beta Bidirectional Associative Memories: Theory and Applications. *Neural Processing Letters* 26(1): 1-40. doi:10.1007/s11063-007-9040-2
- Alcalá-Fernandez, J., A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, y F. Herrera. (2011). 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): 255–287.

- Aldape-Pérez, Mario, Cornelio Yáñez-Márquez, Oscar Camacho-Nieto, y Amadeo J. Argüelles-Cruz. (2012). An associative memory approach to medical decision support systems. *Computer Methods and Programs in Biomedicine* 106: 287-307.
- Amari, S. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers* C-21(11): 1197-1206.
- Anderson, J. A. (1972). A simple neural network generating an interactive memory, *Mathematical Biosciences* 14: 197-220.
- Anderson, J. A., y Eduard Rosenfeld. (Eds.) (1990). *Neurocomputing: Foundations of Research*, Cambridge: MIT Press.
- Argüelles Cruz, Amadeo José. (2007). *Redes Neuronales Alfa-Beta sin pesos: Teoría y factibilidad de implementación*. Tesis de Doctorado, Centro de Investigación en Computación.
- Blyth, T. S. (2005). *Lattices and Ordered Algebraic Structures*. Germany: Springer.
- Bracewell, R. (2000). Heaviside's Unit Step Function. En *The Fourier Transform and Its Applications*, 61-65. New York: McGraw-Hill.
- Broomhead, D. S. y David Lowe. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems* 2: 321-355.
- Cortes, C. y V. N. Vapnik. (1995). Support Vector Network. *Machine Learning* 20: 273-297.
- Cover, T.M. y P. E. Hart. (1967). Nearest Pattern Classification. *IEEE Transactions on Information Theory* XIII (1): 21-27.
- Derrac, J., S. García, D. Molina y F. Herrera. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1(1): 3-18.
- Du, Ke-Lin, y N. N. Swami. (2014). *Neural Networks and Statistical Learning*. Londres: Springer.
- Duda, Richard, Peter Hart, y David Stork. (2000). *Pattern Classification*. Wiley.
- Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, e I. H. Witten. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations Newsletter* 11(1): 10-18.
- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79: 2554-2558.
- Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers* C-21(4): 353-359.
- Lichman, M. (2013). *UCI Machine Learning Repository*. <http://archive.ics.uci.edu/ml> (1 de abril de 2014).
- López-Yáñez, I., A. J. Argüelles-Cruz, O. Camacho-Nieto y Cornelio Yáñez-Márquez. (2011). Pollutants time series prediction using the Gamma classifier. *International Journal of Computational Intelligence Systems* 4(4): 680-711.
- López Yáñez, Itzamá, Leonid Sheremetov, y Cornelio Yáñez Márquez. (2014). A novel associative model for time series data mining. *Pattern Recognition Letters* 41: 23-33.
- Marques de Sá, J. (2001). *Pattern recognition: concepts, methods and applications*. Springer.

- Nakano, K. (1972). Associatron-A model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics SMC-2(3)*: 380-388.
- Platt, J. C. (1998). Fast Training of Support Vector Machines Using Sequential Minimal Optimization. En *Advances in Kernel Methods - Support Vector Learning*, 41–65. MIT Press.
- Ritter, G. X., P. Sussner y J. L. Diaz-de-Leon. (1998). Morphological associative memories. *IEEE Transactions on Neural Networks 9(2)*: 281–293.
- Rumelhart, D. E., G. E. Hinton y R. J. Williams. (1986). Learning representations by back-propagating errors. *Nature 323(6088)*: 533–536.
- Sánchez Garfias, Flavio Arturo. (2009). Modelos Asociativos Alfa-Beta Difusos. Tesis Doctoral, Centro de Investigación en Computación.
- Steinbuch, K. (1961). Die Lernmatrix, *Kybernetik 1(1)*: 36-45.
- Tatsuoka, K. K. (1985). A Probabilistic Model for Diagnosing Misconceptions By The Pattern Classification Approach. *Journal of Educational and Behavioral Statistics 10(1)*: 55–73.
- Theodoridis, S. , y Konstantinos Koutrombas. (2009). *Pattern Recognition* . London, UK.: Academic Press.
- Uriarte-Arcia, A.V., López-Yáñez, I., y Yáñez-Márquez, Cornelio (2014). One-Hot Vector Hybrid Associative Classifier for Medical Data Classification. *PLoS ONE 9(4)*: e95715. eISSN: 1932-6203.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics 1*: 80–83.
- Willshaw, D., O. Buneman y H. Longuet-Higgins. (1969). Non-holographic associative memory. *Nature 222*: 960-962.
- Witten , I. H., y E. Frank. (2011). *Data mining: practical machine learning tools and techniques*. Amsterdam: Elsevier.
- Yáñez Márquez, Cornelio. (2002). Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios. Tesis Doctoral, Centro de Investigación en Computacion.
- Zar, J. H. (1998). *Biostatistical Analysis*. Upper Saddle River, N.J: Prentice Hall.

Anexo

A continuación se muestra el algoritmo (en pseudocódigo) del Clasificador de Heaviside en su modo HI, el cual tiene tres etapas principales: preprocesamiento, transformación de los patrones, entrenamiento (aprendizaje) y clasificación, sin contar la función principal que es en donde se establece la secuencia de las operaciones a realizar.

Función Principal()

```

B=leerBancodeDatos()
B=Preprocesamiento(B)
Base b=elegirBaseB()
B=TransformarBaseB(B, base b)
[A,P]=dividirBancoDatos(B) //Aplicar método de validación de RP
M=Entrenamiento(Conjunto de aprendizaje A)
Clasificación(Modelo de clasificador M, Conjunto de prueba P)

```

Fin

Función Preprocesamiento(Bando de Datos B)

Para cada patrón $x \in B$ hacer

Min=minimo(x) //Obtener componente mínima de cada patrón

Si $x < 0$

Para $i=1:dimensión(x)-1$ //No contar la etiqueta de clase

$x_i = x_i - Min$

Fin

Fin

Fin

Si hayComponentesDecimales(B)

d=buscarMaximoNúmeroDecimales(B);

Fin

Por cada patrón $x \in B$ hacer

Para $i=1: \text{dimensión}(x)-1$

$$x_i = x_i * 10^d$$

Fin

Fin

Regresar B

Fin

Función TransformarBaseB (Banco de Datos B, base b)

Por cada patrón $x \in B$ hacer

Para $i=1: \text{columnas}(B)-1$ hacer //recorrer las componentes del patrón

$$x_i = \text{Transformar}(x_i, b) //\text{Convertir la componente a base b}$$

Fin

Fin

Fin

Función Entrenamiento (Conjunto de aprendizaje A)

M=null

$tamA = |A| //\text{Calcular el número de patrones en A}$

Por cada patrón $A^k \in A$ hacer

$$M_k = L(A^k, (A^k)^t)_{n \times n} //\text{Matriz de nxn}$$

Fin

$$M = \bigwedge_{k=1}^{tamA} L(A^k, (A^k)^t)$$

Regresar M

Fin

Función Clasificación (Modelo de clasificador M, Conjunto de prueba P)

Por cada patrón $P^d \in P$ hacer

Calcular $R = M \uplus_C P^d$ donde $R_i = (M \uplus_C P^d)_i = \bigvee_{j=1}^n C(m_{ij}, P_j^d)$

Buscar en A el patrón recuperado R o en su defecto un patrón cercano a R.

Asignar a P^d la clase del patrón encontrado en A para R.

Fin

Fin

El modo HS del Clasificador de Heaviside puede obtenerse aplicando dualidad en las funciones de entrenamiento y de clasificación.