



**Instituto Politécnico Nacional**  
**Centro de Investigación en Computación**

**Modelo de clasificación supervisada  
usando la función de Heaviside**

**Tesis**

**que para obtener el grado de  
Doctorado en Ciencias de la Computación**

**Presenta:**

**M. en C. Andrés García Floriano**

**Asesores:**

**Dr. Cornelio Yáñez Márquez**

**Dr. Oscar Camacho Nieto**



**Ciudad de México, junio de 2017**



# INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

## ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 09 del mes de mayo de 2017 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

**Centro de Investigación en Computación**  
para examinar la tesis titulada:

**“Modelo de clasificación supervisada usando la función de Heaviside”**

Presentada por el alumno:

**GARCÍA**  
Apellido paterno

**FLORIANO**  
Apellido materno

**ANDRÉS**  
Nombre(s)

Con registro: 


B	1	3	0	2	9	3
---	---	---	---	---	---	---

aspirante de: **DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISIÓN REVISORA


Director de Tesis

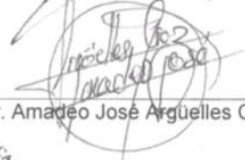
  
\_\_\_\_\_  
Dr. Cornelio Yáñez Márquez

  
\_\_\_\_\_  
Dr. Oscar Camacho Nieto

  
\_\_\_\_\_  
Dr. Sergio Suárez Guerra

  
\_\_\_\_\_  
Dr. Oleksiy Pogrebnyak

  
\_\_\_\_\_  
Dr. Itzamá López Yáñez

  
\_\_\_\_\_  
Dr. Amadeo José Argüelles Cruz



\_\_\_\_\_  
Dr. Marco Antonio Ramírez Salinas  
INSTITUTO POLITÉCNICO NACIONAL  
CENTRO DE INVESTIGACION  
EN COMPUTACION  
DIRECCION



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de México el día 09 del mes mayo del año 2017, el que suscribe M. en C. Andrés García Floriano alumno del Programa de Doctorado en Ciencias de la Computación con número de registro B130293, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Cornelio Yáñez Márquez y cede los derechos del trabajo intitulado Modelo de clasificación supervisada usando la función de Heaviside, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección [andreiosgf@gmail.com](mailto:andreiosgf@gmail.com). Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

---

M. en C. Andrés García Floriano

# Resumen

En este documento se presentan los fundamentos teóricos de un nuevo modelo de clasificación perteneciente al enfoque asociativo del reconocimiento de patrones: El Clasificador de Heaviside. Como su nombre lo indica, este clasificador sustenta dos fases, aprendizaje y clasificación, en la función de Heaviside.

Pese a que en este trabajo se sustenta de manera teórica el funcionamiento del clasificador de Heaviside, consideramos importante ilustrar la efectividad del clasificador de manera experimental. Por lo tanto, se incluyen los resultados de un estudio comparativo compuesto por dos rondas de pruebas. En la primera, el clasificador fue comparado contra 9 modelos destacados en el estado del arte del Reconocimiento de Patrones en 20 bancos de datos de los prestigiosos repositorios UCI y KEEL; en la segunda, el clasificador fue nuevamente comparado en contra de siete modelos en 20 bancos de datos. Durante la primera ronda, en el mejor de los casos el Clasificador de Heaviside obtuvo un 97.15% (con el banco de datos Wisconsin), mientras que en el peor de los casos obtuvo un 72.64% (con el banco de datos Wine); por otro lado, en la segunda ronda de pruebas a pesar de que el desempeño del clasificador no superó a sus demás competidores, sólo hubo dos casos en los que las diferencias de rendimientos fueron significativas, cabe destacar que en estas pruebas nuestro clasificador logró el 100% en uno de los bancos de datos, mientras que su peor desempeño fue de poco más del 50%. Es importante mencionar que los resultados obtenidos fueron sometidos a las pruebas no paramétricas de los signos, la de Wilcoxon, y la de Friedman, las cuales proporcionaron mayor certidumbre estadística a los resultados obtenidos en la comparación.

Tras analizar los resultados de las pruebas de significancia estadística hemos concluido que nuestro modelo supera a algunos modelos importantes del estado del arte, y en aquellos casos donde nuestro modelo es superado, es alentador observar que las diferencias no resultan significativas. También es importante mencionar que hemos propuesto algunas aplicaciones donde el clasificador es empleado exitosamente en problemas médicos y relacionados con educación, por lo que se puede considerar a este clasificador como una opción viable para resolver problemas de clasificación. Finalmente, es preciso hacer notar que actualmente contamos con dos publicaciones relacionadas con este clasificador: una en revista del padrón de Conacyt y otra JCR.

# Abstract

In this document the theoretical foundations of a novel classification model, which belongs to the associative approach to Pattern Recognition: The Heaviside's Classifier, are shown. As its name indicates, this classifier supports both phases, learning and classification, on Heaviside function.

This document shows, the theoretical foundations of a novel classification model, which belongs to the associative approach of Pattern Recognition: The Heaviside's Classifier. As its name indicates, this classifier supports two phases: learning and classification on Heaviside function.

Even though this work presents the theoretical support for the Heaviside classifier performance, we considered to be important to illustrate the effectiveness of the classifier in an experimental form. For this reason, a comparative study of its performance is included, which is composed by two rounds of tests are included. In the first round, the classifier was compared against nine models, featured in the state-of-the-art of pattern recognition, in 20 datasets of prestigious repositories UCI and KEEL. In the second round, the classifier was, once more, compared against seven models in 20 datasets. During the first round, the Heaviside classifier, in the best of the cases, the Heaviside classifier scored a 97.15% whilst in the worst case scored a 72.64%. On the other hand, in the second round of tests; although the performance of the classifier did not surpass other classifiers, there were only two cases where the differences were significant. It is worth mentioning that in these tests our classifier achieved 100% of performance, in one of the datasets, while its worst performance was a little above of 50%. It is important to remark that the obtained results, were subjected to the following non-parametric tests: Sign, Wilcoxon and Friedman, which provided statistical certainty to the comparison results obtained by all the classifiers.

After analyzing the results of the statistical significance test, we concluded that our model overcomes some featured state-of-art models, and in those cases where our model is surpassed, it is encouraging to note that differences were not significant. It is also worth mentioning that we have proposed some applications where the classifier is employed successfully in medical and educational issues; so, we can consider this classifier as feasible option to tackle classification problems. Finally, is important to highlight that, currently, we have published two papers: one in a Conacyt indexed journal and the other in a JCR journal.

# Agradecimientos

## A Dios:

Por ser mi fuente de fuerza e inspiración, por la oportunidad de llegar a este momento culminante.

## A mi madre, Carmen Floriano; a mis hermanas, Carmen y Karen y a mis cuñados, Miguel y José Luis.

Por todo su apoyo y compañía en estos años.

## A mi princesa Gina:

Por ser la fuente de inspiración que ha cambiado mi vida en este último año.

## Al Instituto Politécnico Nacional y al Centro de Investigación en Computación:

Por darme la oportunidad de continuar con mi preparación profesional y por todos los apoyos brindados.

## Al Consejo Nacional de Ciencia y Tecnología:

Por el sustento económico que ha hecho posible continuar con este trabajo.

## A mis directores de tesis:

Dr. Cornelio Yáñez y Dr. Oscar Camacho. Por su ilimitado e incondicional apoyo durante todo este proceso.

## A mis sinodales:

Dr. Sergio Suárez, Dr. Oleksiy Pogrebnyak, Dr. Amadeo Argüelles y Dr. Itzamá López. Por sus invaluable observaciones y comentarios que han mejorado y dado forma definitiva a esta tesis.

## A mis amigos y a los miembros del Grupo Alfa-Beta:

Cerón, Elías, Ferreira, Víctor, David, Mariana, José Luis, Dr. Cuauhtémoc López, Dr. Benjamín, Dra. Guohua Sun, Dr. Shi-Hai Dong, Dr. Octavio López, Dr. Mario Aldape y Dra. Abril Uriarte. Por el apoyo permanente y por todos aquellos momentos especiales de convivencia.

## A quien que me enseñó que la mayoría de las personas entran y salen de nuestra vida, como personajes de una novela, pero que sólo las realmente valiosas se quedan por siempre.

# Dedicatorias

*Dedico este trabajo a aquellas personas que se encuentran en este mismo camino; en particular a quienes, por momentos, pierden el ánimo de seguir adelante.*

*También dedico este trabajo a quienes, por diversas circunstancias, no pudieron llegar a este momento culminante.*

# Índice general

Resumen .....	4
Abstract .....	5
Agradecimientos.....	6
Dedicatorias.....	7
Índice de tablas.....	10
Índice de figuras .....	11
Capítulo 1. Introducción.....	13
1.1 Antecedentes.....	13
1.2 Justificación.....	17
1.3 Objetivos .....	18
1.3.1 Objetivo general .....	18
1.3.2 Objetivos particulares .....	18
1.4 Contribuciones.....	18
1.5 Organización del documento .....	19
Capítulo 2. Estado del Arte.....	21
2.1 Enfoque estadístico-probabilístico: clasificador Bayesiano .....	22
2.1.1 Clasificador Naïve Bayes .....	23
2.1.2 Aplicaciones del enfoque probabilístico estadístico .....	24
2.2 Clasificadores basados en métricas .....	25
2.2.1 Clasificador de la distancia mínima.....	26
2.2.2 Clasificador k-NN (k - Nearest Neighbor) .....	27
2.2.3 Algunos trabajos relevantes del enfoque basado en métricas.....	29
2.3 Árboles de decisión .....	30
2.3.1 Algunas aplicaciones de los árboles de decisión .....	31
2.4 Enfoque Neuronal.....	32
2.4.1 Aplicaciones del enfoque neuronal.....	34
2.5 Máquinas de Soporte Vectorial (SVM) .....	35
2.5.1 Aplicaciones de las SVM .....	37
2.6 Enfoque Asociativo .....	38



2.6.1 Clasificador Híbrido Asociativo con Traslación (CHAT).....	39
2.6.2 Clasificador Gamma .....	39
2.5.3 Aplicaciones del enfoque asociativo .....	42
Capítulo 3. Materiales y Métodos.....	45
3.1 Memorias asociativas .....	45
3.2 La función de Heaviside .....	47
3.3 Teoría de lattices y estructuras algebraicas ordenadas .....	48
3.4 Sistemas de numeración: representaciones de números enteros.....	49
Capítulo 4. Modelo propuesto .....	51
4.1 Hipótesis fundamentales acerca del modelo propuesto .....	51
4.2 Operación L para los modos HI y HS.....	52
4.2.1 Características particulares de la operación L .....	53
4.3 La operación C .....	59
4.3.1 Análisis general de la operación C para ambos modos del clasificador .....	60
4.3.2 ¿Cómo se usa la función de Heaviside en la operación $C^I$ del modo HI?.....	65
4.3.3 ¿Cómo se usa la función de Heaviside en la operación $C^S$ del modo HS? .....	67
4.4 Fases del clasificador de Heaviside en el modo HI .....	78
4.5 Fases del Clasificador de Heaviside en el modo HS .....	80
4.6 Algoritmo del clasificador de Heaviside .....	82
4.7 Análisis de la complejidad del Clasificador de Heaviside.....	85
4.7.1 Complejidad espacial.....	85
4.7.2 Complejidad temporal .....	86
4.7.3 Potenciales limitaciones del clasificador de Heaviside .....	89
4.8 Ejemplo de operación del clasificador de Heaviside en el modo HI .....	90
4.9 Ejemplo de operación del clasificador de Heaviside en el modo HS .....	95
Capítulo 5. Resultados experimentales y discusión.....	99
5.1 Modelos de clasificación seleccionados .....	99
5.2 Pruebas no paramétricas .....	101
5.2.1 Prueba de los signos .....	101
5.2.2 Prueba de Wilcoxon.....	101
5.2.3 Prueba de Friedman .....	102
5.3 Bancos de datos seleccionados .....	103

5.4 Resultados experimentales (primera ronda de pruebas) .....	106
5.5 Resultados de las pruebas de los signos, de Wilcoxon y de Friedman (primera ronda) .....	108
5.6 Resultados experimentales (segunda ronda).....	113
5.7 Resultados de las pruebas de los signos, de Wilcoxon y de Friedman (segunda ronda) .....	114
5.8 Pruebas intensivas del clasificador de Heaviside.....	120
5.8.1 Selección del valor adecuado para la base.....	122
5.9 Aplicaciones del Clasificador de Heaviside .....	122
5.9.1 Clasificación de recursos educativos para la generación inteligente de metadatos .....	122
5.9.2 Pre-diagnóstico de la Degeneración Macular relacionada con la Edad.....	123
5.10 Trabajos publicados.....	123
Capítulo 6. Conclusiones, Contribuciones y Trabajo a Futuro.....	125
6.1 Conclusiones.....	125
6.2 Contribuciones.....	126
6.3 Trabajo a futuro .....	127
Apéndices .....	129
A1. Demostración de las propiedades de la operación L.....	129
A2. Demostración de las propiedades de la operación C .....	135
A3. Demostración de los teoremas, lemas y corolarios correspondientes con el modo HI del clasificador de Heaviside.....	140
A4. Demostración de los teoremas, lemas y corolarios correspondientes con el modo HS del clasificador de Heaviside.....	144
A5. Planteamiento de una expresión alterna de la operación C.....	149
A6. Desempeños obtenidos en las pruebas intensivas del Clasificador de Heaviside.....	155
A7. Primera página del artículo publicado en la revista Nova Scientia.....	160
A8. Primera página del artículo publicado en la revista IRRODL .....	161
Referencias .....	162

## Índice de tablas

Tabla 1. Ejemplo de la operación L con $b=2$ .....	54
Tabla 2. Ejemplo de la operación L con $b=3$ .....	54
Tabla 3. Ejemplo de la operación L con $b=4$ .....	55
Tabla 4. Ejemplo de la operación L con $b=5$ .....	55

Tabla 5. Ejemplo de la operación L con $b=6$ .....	57
Tabla 6. Construcción de la operación C.....	61
Tabla 7. Construcción de la operación C a partir de la operación L.....	61
Tabla 8. Operación C considerando todos los posibles valores de $x, y$ .....	63
Tabla 9. Operación C para el modo HI del clasificador de Heaviside.....	64
Tabla 10. Evaluación de casos NI con $b=3$ , bajo la condición 1. ....	66
Tabla 11. Evaluación de casos NI con $b=3$ , bajo la condición 2. ....	66
Tabla 12. Evaluación de casos que no son NI para $b=3$ . ....	66
Tabla 13. Evaluación de pares NI para $b=3$ , bajo la condición 1. ....	69
Tabla 14. Evaluación de pares NI para $b=3$ bajo la condición 2. ....	69
Tabla 15. Evaluación de pares que no son NI para $b=3$ . ....	69
Tabla 16. Operación C para el modo HI con $b=2$ .....	71
Tabla 17. Operación C para el modo HI con $b=4$ .....	71
Tabla 18. Operación C para el modo HI con $b=5$ .....	72
Tabla 19. Operación C para el modo HS con $b=2$ . ....	74
Tabla 20. Operación C para el modo HS con $b=4$ .....	75
Tabla 21. Operación C para el modo HS con $b=5$ . ....	76
Tabla 22. Factores de desbalance de algunos de los bancos de datos seleccionados. ....	106
Tabla 23. Rendimiento de los modelos de clasificación en los Bancos de Datos seleccionados .....	106
Tabla 24. Diferencias entre los modelos empleados en los experimentos.....	108
Tabla 25. Significancias obtenidas en la prueba de los signos. ....	110
Tabla 26. Prueba de rangos con signo de Wilcoxon.....	110
Tabla 27. Estadísticos de contraste de la prueba de Wilcoxon.....	112
Tabla 28. Rangos obtenidos al aplicar la prueba de Friedman. ....	113
Tabla 29. Rendimiento de los modelos de clasificación en los bancos de datos seleccionados. ....	114
Tabla 30. Resultados de la prueba de los signos. ....	115
Tabla 31. Valores de significancia de la prueba de los signos.....	117
Tabla 32. Prueba de rangos con signo de Wilcoxon.....	117
Tabla 33. Estadísticos de contraste de la prueba de Wilcoxon.....	119
Tabla 34. Resultados de la prueba de Friedman. ....	120
Tabla 35. Pruebas intensivas con el banco de datos Iris Plant.....	121
Tabla 36. Pruebas intensivas con el banco de datos Haberman.....	121

## Índice de figuras

Figura 1. Principio básico de los clasificadores basados en métricas.....	26
Figura 2. Ejemplos de bancos de datos en donde el clasificador de la distancia mínima fallará. (a) Clases concéntricas. (b) Clases en forma encadenada. ....	28
Figura 3. Clasificación de un patrón desconocido (representado por la estrella) de acuerdo con el clasificador 1NN.....	28
Figura 4. Neuronas fuente: Quasar Jarosz at English Wikipedia, CC BY-SA 3.0, <a href="https://commons.wikimedia.org/w/index.php?curid=7616130">https://commons.wikimedia.org/w/index.php?curid=7616130</a> .....	32

Figura 5. Modelo de neurona artificial. ....	33
Figura 6. Función umbral. ....	34
Figura 7. Estructura de una Red Neuronal Artificial. ....	34
Figura 8. Arquitectura de una Red Neuronal Convolutiva. Fuente: By Aphex34 - Own work, CC BY-SA 4.0, <a href="https://commons.wikimedia.org/w/index.php?curid=45679374">https://commons.wikimedia.org/w/index.php?curid=45679374</a> .....	35
Figura 9. Funcionamiento básico de las SVM: (a) en patrones linealmente separables; (b) en patrones que no son linealmente separables.....	36
Figura 10. Funcionamiento general de una memoria asociativa.....	45
Figura 11. Método Propuesto para la clasificación inteligente de recursos educativos.....	123
Figura 12. Método para generar el banco de datos empleado en el pre-diagnóstico de la degeneración macular en imágenes de retina.....	124

# Capítulo 1. Introducción

En este documento mostraremos los fundamentos teóricos y algunas de las características de un nuevo modelo de clasificación supervisada de patrones llamado Clasificador de Heaviside. Dicho clasificador tiene como bases teóricas algunos conceptos de Memorias Asociativas, teoría de las *lattices*, teoría de las estructuras algebraicas ordenadas y teoría de los sistemas de numeración. Posteriormente se presentará la aplicación del modelo propuesto en diversos problemas de clasificación de patrones.

## 1.1 Antecedentes

Los seres humanos buscamos comprender la manera en la que funciona el mundo, y esto ha llevado a una interacción holística del hombre con la naturaleza; como parte de esta interacción se han descubierto y puesto en práctica formas de cambiar el ambiente en beneficio de la población. El proceso no ha sido fácil, y como ejemplo podemos mencionar el trabajo de los alquimistas quienes en la búsqueda de la llamada piedra filosofal tuvieron que enfrentarse a una gran cantidad de problemas (accidentes y persecución por autoridades religiosas principalmente). Sin embargo, estos trabajos sentaron las bases de la Química, la cual es sin duda una de las principales Ciencias Naturales [1].

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 [2]. A pesar de que las actividades anteriores son cotidianas para los seres humanos es muy complejo o a veces imposible programar a una computadora para que las realice; esto se debe a que se deberán modelar y programar todos los factores involucrados en el proceso. Este problema ha motivado el surgimiento de ciencias como la Inteligencia Artificial y el Reconocimiento de Patrones [2,3].

El reconocimiento de patrones es una ciencia que contempla cuatro tareas básicas: la recuperación, la clasificación de patrones, la regresión y el agrupamiento o *clustering*. La tarea de la clasificación, en la cual se centra esta tesis, 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, aunque por lo regular es posible que los patrones que forman las clases tengan varias similitudes. Esta ciencia, que surgió en los años 60 del siglo XX como resultado de investigaciones relacionadas con la estadística [3], tiene una gran cantidad de aplicaciones: desde la detección de actividades fraudulentas; los asistentes inteligentes como lo son Siri de Apple o Cortana de Microsoft; la predicción de producción de pozos petroleros; o incluso el pre-diagnóstico de enfermedades a través de datos o imágenes de los pacientes [4].

En la actualidad, muchos de los avances relacionados con el reconocimiento de patrones difieren con el enfoque clásico propuesto por la Inteligencia Artificial; en dicho enfoque se busca la solución de problemas mediante la emulación del funcionamiento del cerebro humano [5]. sin embargo, este enfoque ha demostrado no ser competente para atacar problemas de búsqueda, optimización y aprendizaje en sistemas que manejan grandes bases de datos relacionadas con cuestiones biológicas o comerciales; también ha sido incapaz de ofrecer buenos resultados en sistemas orientados a industrias especializadas como la farmacéutica, la aeroespacial o la de los energéticos. El fracaso de la IA tradicional dio paso al uso de métodos no tradicionales como los conjuntos rugosos [6], las redes neuronales artificiales [7], los algoritmos genéticos [8], las *belief networks* [9], la teoría del aprendizaje computacional [10] y la teoría del caos [11]. Estas teorías no convencionales se utilizaron exitosamente en aplicaciones como las mencionadas anteriormente y han dado paso al surgimiento de una nueva disciplina llamada *Computational Intelligence* [12] o Inteligencia Computacional.

La Inteligencia Computacional es considerada como la evolución de la IA. De acuerdo con James Bezdek, quien es considerado el padre de la Inteligencia Computacional, un sistema es computacionalmente inteligente cuando cumple con las siguientes características [13]:

1. Trabaja únicamente con datos numéricos,
2. Tiene componentes de reconocimiento de patrones,
3. No usa el conocimiento en términos de la Inteligencia Artificial,
4. Exhibe un comportamiento adaptativo,
5. Tiene alta tolerancia a los fallos,
6. Tiene velocidad de reacción similar a la de los seres humanos,
7. Su tasa de errores es similar a la de los humanos.

A partir de estas características es posible definir a la Inteligencia Computacional como 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 [12].

En la actualidad el Reconocimiento de Patrones tiene una gran cantidad de aplicaciones. Entre las aplicaciones que se han reportado en la literatura científica destacan el reconocimiento óptico de caracteres, el reconocimiento de voz y el reconocimiento de objetos en imágenes; aunque también hay trabajos relacionados con la biometría, la bioinformática, análisis de multimedia, reconocimiento de documentos y sistemas expertos, entre otros [14].

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. Un científico experto en el área del Reconocimiento de Patrones debe tener interacción constante con expertos de otras áreas [15].

Idealmente, un sistema reconocedor de patrones sería capaz de tomar decisiones a partir de datos recibidos del exterior [16], de manera que revolucionaría muchos procesos industriales o ayudaría a mejorar la calidad de vida de las personas. Sin embargo, la creación de un sistema reconocedor de patrones (para la clasificación) es una tarea bastante compleja, ya que el diseño y construcción de un sistema de este tipo debe pasar por un conjunto de etapas como las que se describen en [3,15,17].

1. Selección del área de interés y extracción de rasgos: con ayuda de un experto de dicha área se eligen aquellos rasgos o características que representan a los patrones que serán reconocidos por el sistema.
2. De entre los rasgos propuestos inicialmente se eligen aquellos que permiten distinguir, en la medida de lo posible, los grupos o clases en las que se encuentran divididos los patrones.
3. Una vez que se seleccionaron los rasgos adecuados, los patrones son representados en términos de los rasgos seleccionados y posteriormente se usarán para modelar las clases que se desean discriminar o reconocer con el sistema. A este proceso típicamente se le llama entrenamiento.
4. Se valida el entrenamiento del modelo de clasificación mediante la clasificación de patrones cuyas clases son conocidas, pero que no fueron incluidos en el entrenamiento; los patrones empleados en esta clase se conocen como conjunto de prueba.

Las etapas descritas anteriormente corresponden con un sistema de reconocimiento de patrones basado en el aprendizaje supervisado; en dicho aprendizaje el clasificador se construye a partir de patrones prototipo o de entrenamiento [2]. El aprendizaje supervisado se aplica en tres de las cuatro tareas básicas del reconocimiento de patrones: clasificación, regresión y recuperación.

La parte central del Reconocimiento de Patrones y en particular de la clasificación, es la creación de modelos capaces de clasificar los patrones que se les presentan con el mayor grado de certeza posible. En los últimos 50 años se ha propuesto un gran número de modelos de clasificación, los cuales están sustentados en conceptos diferentes. A continuación, se presentan algunas de las bases conceptuales que mayor impacto han tenido en el reconocimiento de patrones.

- Métricas [18].
- Teoría de probabilidad y estadística [16].
- Teoría de lenguajes y autómatas [2]. Actualmente este soporte conceptual es poco utilizado ya que, en la investigación bibliográfica realizada, no se encontraron trabajos recientes que lo utilicen.
- Teoría de los árboles de decisión [19–21]
- Modelos matemáticos de las neuronas [7,22].
- Teoría del aprendizaje estadístico de Vapnik y Chervonenkis [23,24].
- Memorias asociativas [22,25,26].

Se han propuesto muchos modelos de clasificación basados en los soportes teóricos enlistados anteriormente; sin embargo, sólo algunos de estos cumplen con dos características que deberían considerarse indispensables: sencillez y eficacia.

Tras analizar diversos trabajos pertenecientes a la mayoría de los enfoques presentados anteriormente, consideramos que uno de los pocos modelos que cumplen con estas dos características es el clasificador k-NN (k Nearest Neighbor, los k vecinos más cercanos); el cual está basado en métricas y en algunos fundamentos de los conjuntos parcialmente ordenados [27,28]. El algoritmo k-NN clasifica un patrón de prueba calculando la distancia entre éste y los patrones del conjunto de entrenamiento; posteriormente se ordenan las distancias obtenidas y se asigna al patrón de prueba la clase mayoritaria de los k patrones más cercanos. La efectividad del clasificador fue comprobada en el famoso proyecto Statlog [29], en el que Michie y sus colaboradores demostraron la superioridad de este modelo con respecto a otros en una gran cantidad de bancos de datos. Desafortunadamente, este modelo de clasificación tiene al menos dos desventajas considerables: la primera es su poca eficiencia cuando los conjuntos de aprendizaje son muy grandes, dado que el cálculo de la distancia entre un patrón de prueba y los patrones de aprendizaje aunado al ordenamiento de las distancias implica un consumo alto de recursos computacionales [18,30]; la otra desventaja del k-NN es su susceptibilidad a los efectos de la llamada maldición de la dimensionalidad [31]. El término maldición de la dimensionalidad hace referencia al hecho de que un clasificador tendrá un rendimiento bajo al utilizarse con patrones de alta dimensionalidad.

Hay un grupo de modelos de reconocimiento de patrones que cumple con las premisas de eficacia y simplicidad; estos modelos están agrupados en el enfoque asociativo de clasificación de patrones. Dicho enfoque comenzó en el año 2002 cuando se presentaron dos nuevos operadores que enriquecieron el estado del arte de las memorias asociativas: los operadores alfa y beta. A partir de este trabajo se formó el grupo de investigación Alfa-Beta, dentro del cual se han propuesto varios de trabajos tesis y publicaciones orientados a realizar tareas de clasificación, regresión o recuperación de patrones. El primer modelo de clasificación presentado dentro del enfoque asociativo es el clasificador híbrido de patrones basado en la Lernmatrix y el *Linear Associator* [32]. Posteriormente, el enfoque asociativo se ha utilizado para complementar modelos clásicos como las redes neuronales. En el trabajo del Dr. Amadeo Argüelles [33] se presenta un nuevo modelo de redes neuronales sin pesos basadas en las operaciones alfa, beta y otros operadores propuestos por el autor. Las ideas propuestas en el enfoque asociativo también se han utilizado en la formulación de modelos de SVM, como lo presentó el Dr. López Leyva en su tesis doctoral [34]. Cabe destacar que las SVM Alfa-Beta se han aplicado exitosamente en pre-diagnóstico de cáncer de mama [35].

Es importante mencionar que en el enfoque asociativo de reconocimiento de patrones también se han propuesto modelos de clasificación que han sido empleados exitosamente en la regresión de patrones; este es el caso del clasificador Gamma [36–39] el cual ha sido utilizado exitosamente en la predicción de series de tiempo y que actualmente se está utilizando en la predicción del éxito o fracaso de los estudiantes que recién ingresan a la universidad [40].



Finalmente, hay que mencionar que se propusieron trabajos importantes relacionados con la recuperación de patrones; se han diseñado modelos de memorias asociativas bidireccionales [41,42] y memorias asociativas difusas Alfa-Beta [43], las cuales difieren del modelo de memorias difusas presentado por Sussner [44]. Cabe destacar que todos los modelos pertenecientes al enfoque asociativo del reconocimiento de patrones han mostrado ser competitivos en comparación con otros modelos importantes del estado del arte; además cumplen con la premisa de estar basados en ideas simples, lo cual implica que estos modelos buscan consumir pocos recursos computacionales.

## 1.2 Justificación

El modelo de clasificación que se propone en este trabajo de tesis refuerza el enfoque asociativo del reconocimiento de patrones que se ha trabajado en CIC desde el año 2002. Es importante resaltar que todos los modelos de clasificación propuestos en el Grupo Alfa-Beta hasta ahora han estado caracterizados por su eficacia y por la simplicidad de las ideas en las que se basan; las comparaciones que se han hecho contra modelos tradicionales del Reconocimiento de Patrones muestran que los modelos asociativos, incluidas las pruebas realizadas con el clasificador de Heaviside, ofrecen resultados bastante competitivos. Además, en muchos casos la cantidad de recursos computacionales empleados por estos modelos es significativamente menor a modelos como el k-NN o las redes neuronales.

Las operaciones de aprendizaje y clasificación del clasificador de Heaviside fueron propuestas, inicialmente, de forma empírica a partir de los operadores alfa-beta. Debido a que en las primeras pruebas los resultados fueron satisfactorios, consideramos fundamental proporcionar el sustento matemático de su funcionamiento para determinar las condiciones en las que el modelo trabajará, pero sin omitir sus alcances y limitaciones. El proceso de formalización matemática del clasificador de Heaviside involucra conceptos de sistemas de numeración, teoría de lattices y teoría de estructuras algebraicas ordenadas, por lo que consideramos este proceso como primera justificación del presente trabajo de tesis. Cabe destacar que en el presente trabajo de tesis no solo se hará una discusión formal; también se presentarán algunas de las aplicaciones que se han dado al clasificador propuesto. Otro hecho que justifica el presente trabajo de tesis es que mediante el clasificador de Heaviside estamos proponiendo e ilustrando el uso de las funciones especiales para la creación de clasificadores inteligentes de patrones. Este trabajo de tesis no solo fortalece el enfoque asociativo del reconocimiento de patrones, también podría sentar las bases de una nueva tendencia a utilizar funciones especiales en la creación de modelos relacionados con las tareas de la clasificación, recuperación y regresión de patrones.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Desarrollar un nuevo modelo de clasificación supervisada de patrones cuyas fases de aprendizaje y de clasificación estén basadas en la función de Heaviside y en operaciones propias de las *lattices* y en estructuras algebraicas ordenadas.

### 1.3.2 Objetivos particulares

1. Analizar el estado del arte de los diferentes modelos de clasificación supervisada de patrones.
2. Diseñar el modelo de clasificación basado en la función de Heaviside, en operaciones propias de las *lattices* y en estructuras algebraicas ordenadas.
3. Demostrar de manera formal el rendimiento, las capacidades y limitaciones del modelo propuesto.
4. Aplicar el modelo diseñado en problemas de Minería de Datos, de *Machine Learning*, de Reconocimiento de Patrones, de Inteligencia Computacional, de predicción de Series de Tiempo y de otras ramas afines, a fin de determinar de manera empírica su funcionamiento, alcances y limitaciones.

## 1.4 Contribuciones

La principal aportación científica de esta tesis doctoral es el desarrollo de un modelo original para la clasificación inteligente y supervisada de patrones empleando la función de Heaviside y algunas operaciones propias de las *lattices* y en las estructuras algebraicas ordenadas.

Tras una extensa investigación documental realizada en el Laboratorio de Cómputo Inteligente del CIC al inicio de este trabajo Doctoral, observamos que no se había reportado en revistas indizadas ISI-JCR que esta función haya sido utilizada como sustento de algún modelo de clasificación inteligente de patrones. Es por eso que en este trabajo de tesis se propuso el uso de la función de Heaviside, ya que no ha sido trabajado hasta hoy para clasificar patrones, salvo por el modelo original del Perceptrón de Rosenblatt, quien la usó en sus orígenes, hacia 1957. Sin embargo, dado que el algoritmo de backpropagation para redes neuronales multicapa requiere que las funciones sean diferenciables en todo el dominio, la función de Heaviside fue reemplazada de manera ostensible por otro tipo de funciones diferenciables, especialmente las sigmoideas como la tangente hiperbólica y la función logística.

Es importante resaltar que el clasificador propuesto está sustentado en un modelo matemático que será fundamentado teóricamente mediante demostraciones de lemas y teoremas. Además, el modelo propuesto está diseñado de manera tal que es invariante a los sistemas numéricos utilizados por los patrones a clasificar.

Otra de las aportaciones que ofrece este trabajo de tesis es que el clasificador es del tipo *one-shot*; por lo tanto, no se enfrentará a los problemas de convergencia que pueden llegar a ocurrir en modelos tales como las redes neuronales. El problema de la convergencia ha llegado a limitar la utilidad de algunos de los modelos de reconocimiento de patrones más sofisticados, a la hora de atacar algunos problemas demasiado complejos.

Los experimentos que se realizaron con el clasificador de Heaviside, arrojaron resultados competitivos con relación a otros modelos del estado del arte; por lo tanto, es posible hipotetizar que, derivado de este clasificador, se puede llegar a pensar en un nuevo paradigma de clasificación de patrones basado en funciones especiales como la función de Heaviside y algunas operaciones propias de las *lattices* y en las estructuras algebraicas ordenadas.

Este trabajo ofrece un modelo de clasificación que, además de arrojar resultados competitivos en el rendimiento a la hora de clasificar patrones, será capaz de ejecutarse con un consumo relativamente bajo de recursos, dado que su complejidad no es muy alta. En caso de que el problema demande recursos computacionales considerables, queda la posibilidad de implementar el algoritmo del clasificador de Heaviside en FPGAs o en GPUs, lo cual permitirá que dicho modelo de clasificación pueda ser aplicado en problemas en donde se maneje una gran cantidad de información. Ello conlleva a una conclusión natural: este modelo que nace en el dominio de la Inteligencia Computacional, podrá aplicarse en otros ámbitos donde la demanda de recursos computacionales es grande, como minería de datos o la predicción de series de tiempo.

## 1.5 Organización del documento

En el presente capítulo se han presentado los antecedentes, la justificación, los objetivos (generales y particulares) y las contribuciones del presente trabajo de tesis. El resto del documento está organizado como se describe a continuación.

En el Capítulo 2 presentaremos el estado del arte de las áreas en la que se encuentra ubicada la presente tesis, la clasificación de patrones, el enfoque asociativo de clasificación de patrones y el uso de funciones especiales para la clasificación de patrones. Por otro lado, en el Capítulo 3 se presentarán las herramientas formales que nos han permitido construir el modelo de clasificación propuesto.

En el Capítulo 4 presentaremos detalladamente el modelo de clasificación propuesto, incluimos los fundamentos teóricos del modelo y sus propiedades. Los resultados de los experimentos realizados y su discusión, con base en los resultados de las pruebas de significancia estadística, se presentan en el Capítulo 5; mientras que en el Capítulo 6 se exponen las conclusiones y trabajos a futuro. Finalmente, presentamos una sección de apéndices o anexos donde se incluyen las demostraciones de las propiedades y teoremas relacionados con el clasificador de Heaviside, una formulación alterna de la operación  $C$ , los resultados de pruebas

intensivas de rendimiento del clasificador en donde se muestran el desempeño del mismo tras variar el parámetro de base, así como las primeras páginas de los artículos, relacionados con el clasificador, que fueron aceptados y publicados.

# Capítulo 2. Estado del Arte

En este capítulo presentaremos algunos de los modelos más importantes en el estado del arte de la clasificación de patrones. Como se mencionó en el capítulo anterior, 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 y las redes Bayesianas.
- Clasificadores basados en métricas. En este enfoque se aprovechan las propiedades de las métricas para realizar la clasificación de patrones; se destacan los clasificadores de distancia mínima (clasificador Euclidiano) y los clasificadores de los vecinos más cercanos (k-NN).
- 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 y el *Random Forest*.
- 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.
- Enfoque asociativo. Este enfoque surgido 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 [26], 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.

En algunos textos especializados en reconocimiento de patrones [2,45] también se menciona el enfoque sintáctico estructural, el cual reconoce patrones aplicando conceptos de la teoría de lenguajes y autómatas. Sin embargo, tras una extensa investigación realizada en repositorios de revistas científicas indizadas no encontramos trabajos recientes relacionados con este enfoque. Por lo tanto, decidimos no considerarlo para su análisis en este trabajo de tesis.

En las siguientes secciones mostraremos algunos de los modelos más importantes de los enfoques de reconocimiento de patrones que fueron enlistados anteriormente. Se presentan sus características principales y algunas de sus aplicaciones más recientes.

## 2.1 Enfoque estadístico-probabilístico: clasificador Bayesiano

El clasificador Bayesiano, o también llamado clasificador de máxima verosimilitud, es famoso en el reconocimiento de patrones porque está considerado como el clasificador óptimo [14,16] debido a que minimiza la probabilidad promedio del error. Este modelo de clasificación asume que la información de las clases está en forma de probabilidades previas o *a priori* y que se conoce la distribución de los patrones en las clases; esto implica que el clasificador usa la probabilidad posterior, o *a posteriori*, para asignar la clase a los patrones. Por lo tanto, un patrón será asignado a la clase  $C_i$  si esta clase es la que tiene una mayor probabilidad posterior; esto se puede lograr utilizando el teorema de Bayes el cual convierte la probabilidad *a priori* en *a posteriori* basándose en el patrón que será clasificado y en los valores de verosimilitud. A continuación, se presenta el teorema de Bayes [15].

Teorema 2.1 Teorema de Bayes. Sean los eventos  $A_1, A_2, \dots, A_n$  tales que forman una partición del espacio muestral  $X$  y sea  $B$  un evento dentro de ese mismo espacio, entonces:

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(B)} = \frac{P(B|A_i) \cdot P(A_i)}{\sum_{i=1}^n P(B|A_i) \cdot P(A_i)} \quad (2.1)$$

Donde  $P(A_i)$  define el conocimiento a priori del problema, la probabilidad  $P(B|A_i)$ , o la verosimilitud, define la probabilidad de que ocurra  $B$  dentro del espacio reducido del evento  $A_i$  y luego  $P(B)$  es la evidencia que indica la probabilidad de que ocurra  $B$  si se tiene todo el conocimiento a priori de toda la partición y la verosimilitud de cada evento  $A_k$ ; típicamente la evidencia es utilizada como un factor de normalización. Finalmente  $P(A_i|B)$  es la probabilidad de que ocurra el evento  $A_i$  dado  $B$ . Lo anterior puede expresarse de forma más simple:

$$posterior = \frac{a\ priori \times verosimilitud}{evidencia}$$

Para aplicar el teorema de Bayes es necesario que los eventos estén expresados en términos de variables aleatorias y distribuciones de probabilidad; por lo tanto, para conocer el valor de probabilidad *a posteriori* es necesario conocer las distribuciones de probabilidad del problema a resolver y todas las probabilidades *a priori*. Si se desea utilizar este teorema para construir un clasificador de patrones, es necesario que las clases y los patrones sean modelados como variables aleatorias. A partir de esta representación la clasificación se hace de la siguiente forma: Un patrón  $x$  el cual está representado por la variable aleatoria  $X$ , pertenece a la clase  $i$  representada por la variable aleatoria  $C_i$  si la probabilidad de que pertenezca a la clase  $C_i$  es mayor que la probabilidad de pertenecer a otras clases; esto se expresa de la siguiente forma:

$$X \in C_i \text{ si } P(C_i|X) > P(C_j|X) \forall i \neq j \quad (2.2)$$

Para que esta regla de decisión funcione con un patrón cuya clase es desconocida, es necesario disponer de todo el conocimiento *a priori* de todas las clases y sus correspondientes distribuciones de probabilidad, lo cual ocurre en muy pocas ocasiones [16].

A partir de la regla presentada en la expresión 2.2 es posible proponer el siguiente algoritmo para el clasificador Bayesiano [36].

Algoritmo 2.1 Clasificador de máxima verosimilitud o clasificador Bayesiano.

1. Obtener una muestra representativa de los objetos a clasificar.
2. Determinar cada una de las clases  $C_k$  que formarán parte del sistema.
3. Determinar, con base en la muestra y la cardinalidad de cada clase, las probabilidades  $P(C_k)$ .
4. Determinar los rasgos útiles que se van a utilizar para clasificar, y elaborar cada distribución de probabilidad  $P(X|C_k)$  la cual va a ser dependiente del número y naturaleza de cada rasgo de la variable aleatoria vectorial  $X$ .
5. Aplicar la siguiente regla de clasificación para un patrón desconocido de entrada  $X$ :

$$X \in C_i \text{ si } P(C_i|X) > P(C_j|X) \forall i \neq j \quad (2.3)$$

Donde  $P(C_i|X)$  se calcula de la siguiente forma:

$$P(C_i|X) = \frac{p(X|C_i)P(C_i)}{p(x)} = \frac{p(X|C_i)P(C_i)}{\sum_{k=1}^c p(X|C_k)P(C_k)} \quad (2.4)$$

En esta expresión el valor de la evidencia puede tomarse como un factor de escalamiento que garantiza que todas las probabilidades posteriores sumen 1 [16] por lo tanto, se podría omitir al momento de aplicar la regla de clasificación que se basa en la mayor de las probabilidades posteriores. Hay algunos autores [3] que plantean la posibilidad de trabajar la clasificación Bayesiana empleando funciones discriminantes  $g(x)$ , las cuales son funciones monótonas crecientes (como el logaritmo natural). Si se toman en cuenta estas funciones, la regla de clasificación quedaría de la siguiente manera:

$$X \in C_i \text{ si } g(P(C_i|X)) > g(P(C_j|X)) \forall i \neq j \quad (2.5)$$

A pesar de que este modelo de clasificación es considerado como el óptimo, pocas veces se cuenta con la estadística completa del problema de clasificación. Por lo tanto, se han propuesto modificaciones al modelo original que buscan solventar esta debilidad. A continuación, se presenta una de estas modificaciones, la cual es bastante popular en la comunidad científica del Reconocimiento de Patrones.

### 2.1.1 Clasificador Naïve Bayes

En términos generales el clasificador Naïve Bayes es un modelo probabilístico, basado en el teorema de Bayes, en donde se asume la independencia estadística entre las características o rasgos de los patrones; esta consideración simplifica los cálculos y es la razón por la cual a este

modelo se le considera como ingenuo o *naïve*<sup>1</sup> [14]. De no usarse esta consideración la cantidad de datos necesarios para construir un clasificador bayesiano sería muy grande; por ejemplo, si se tuvieran  $P$  puntos, muestras o patrones para construir una buena estimación de una función de densidad de probabilidad para una sola dimensión, se necesitarían  $P^n$  muestras para un patrón de  $n$  dimensiones. Por lo tanto, entre mayor sea la dimensión de los patrones, mayor será la necesidad de los datos.

Si retomamos la hipótesis de independencia estadística entre los  $n$  rasgos de un patrón  $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ , la verosimilitud se representa en la expresión 2.4 y la probabilidad *a posteriori* se expresa en la ecuación 2.6:

$$p(\mathbf{x}|C_i) = \prod_{j=1}^n p(x_j|C_i), i = 1, 2, \dots, c \quad (2.6)$$

Con este cambio se reduce drásticamente la necesidad de datos para el entrenamiento del clasificador; por ejemplo, para estimar  $n$  funciones de densidad de una sola dimensión sólo se necesitarán  $nP$  puntos o muestras. Finalmente, el clasificador Naïve Bayes clasificará a un patrón cuya clase se desconoce  $\tilde{\mathbf{x}} = (x_1, x_2, \dots, x_n)^t$  a una clase  $C_m$  de la siguiente forma (ec. 2.7) [3]:

$$C_m = \arg \max_{C_i} P(C_i) \prod_{j=1}^n p(x_j|C_i), i = 1, 2, \dots, n \quad (2.7)$$

A pesar de que esta suposición no suele ser real, se ha observado que este modelo ofrece buenos resultados al ser comparado con modelos como las redes neuronales o los árboles. Cabe destacar que además de ofrecer buenos resultados, este modelo es capaz de ejecutarse con relativa rapidez en grandes bancos de datos [3,14].

### 2.1.2 Aplicaciones del enfoque probabilístico estadístico

Debido a que el clasificador Bayesiano requiere la estadística completa del problema, es poco práctico implementarlo. Sin embargo, para superar este problema se han utilizado dos enfoques muy populares: las redes Bayesianas y el clasificador Naïve Bayes. Las redes Bayesianas han sido aplicadas exitosamente en la fusión de clasificadores [46], en la solución de problemas de clasificación multiclase [47]. En cuanto a las aplicaciones las Redes Bayesianas se han aplicado en diversas situaciones entre las que destacan la solución de problemas médicos [48] o el análisis de situaciones relacionadas con la seguridad pública [49]. Cabe destacar que en años recientes y con el auge del *Big Data* han surgido propuestas de adaptar estos modelos para bancos de datos de muchos patrones, como en el trabajo de Arias et al. [50] donde se comprueba que es posible

---

<sup>1</sup> La palabra inglesa *naive* se traduce como ingenuo, en los idiomas francés y alemán la palabra ingenuo se escribe *naïve*.



emplear Redes Bayesianas en la clasificación de bancos de datos de gran tamaño y con resultados de eficiencia y eficacia aceptables.

Hemos observado que el clasificador Naïve Bayes ha sido empleado en una gran diversidad de situaciones, como la construcción de redes Bayesianas [51], en la detección de fallos en procesos industriales [52]; en la clasificación de imágenes [53] y de objetos presentes en imágenes [54], en la construcción de sistemas expertos para la detección de infecciones agudas [55], en la detección de spam en mensajes de textos [56], en la clasificación de texto [57], en la predicción de fallos en procesos computacionales de alto desempeño [58], en tareas relacionadas con la pesca [59], en la clasificación de señales de electroencefalografía [60] y la selección de rasgos para la predicción para la detección oportuna de ataques epilépticos [61], en la detección de personas en señales de radar [62], en la clasificación de imágenes satelitales para ubicar plantas de energía solar [63], en la localización topológica [64] y semántica, basada en anotaciones, de robots [65], en la predicción de factores importantes para la producción de semiconductores [66]; además de que se ha utilizado para localizar patrones de proteínas en las células [67]. Finalmente, en los últimos años se han propuesto modificaciones al modelo original con la finalidad de mejorar su desempeño. En particular, se han mejorado sus resultados mediante la ponderación de los rasgos de los patrones la cual puede lograrse con métodos basados en propiedades de los vectores [60] y en la aplicación de metaheurísticas [68].

## 2.2 Clasificadores basados en métricas

Los clasificadores pertenecientes a este enfoque están basados en el concepto de métrica. La definición formal de métrica se presenta a continuación [69]:

**Definición 2.2 Métrica.** Sea  $X$  un conjunto no vacío. Una métrica en  $X$  es una función  $d: X \times X \rightarrow \mathbb{R}$  tal que para toda  $x, y, z \in X$  se tienen las siguientes propiedades:

- a) No negatividad:  $0 \leq d(x, y) < \infty$ ,
- b) Unicidad:  $d(x, y) = 0$  si y solo si  $x = y$
- c) Simetría:  $d(x, y) = d(y, x)$ ,
- d) Desigualdad del triángulo:  $d(x, z) \leq d(x, y) + d(y, z)$ .

En términos muy generales los clasificadores basados en métricas funcionan de la siguiente forma [36]:

**Algoritmo 2.2.** Algoritmo de clasificación de patrones basados en métricas.

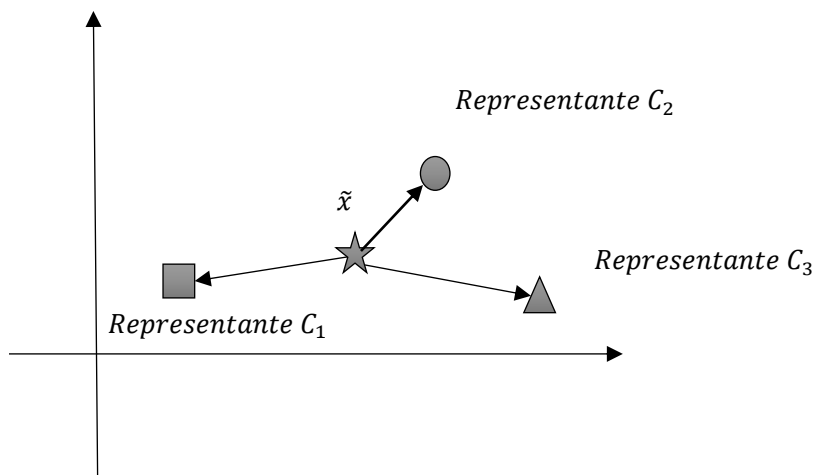
1. Elegir una muestra de patrones clasificada de antemano en  $n$  clases  $\{C_1, C_2, \dots, C_n\}$  y una métrica  $d$ .
2. Con base en la muestra y para cada clase  $C_i$  encontrar un patrón  $m_i$  que la represente mejor.

3. Si  $x$  es un patrón de dimensión  $n$  cuya clase se desconoce, este patrón quedará clasificado en la clase  $C_i$ , si se cumple lo siguiente:

$$\forall j, j \neq i, d(x, m_i) \leq d(x, m_j)$$

■

La idea principal de este clasificador se presenta en la figura 1.



**Figura 1. Principio básico de los clasificadores basados en métricas.**

Es importante mencionar que el éxito de este enfoque está sujeto a dos condiciones fundamentales:

- La elección de la métrica adecuada.
- La elección del patrón representante de cada clase.

La elección de la métrica es una cuestión muy seria ya que prácticamente existe una cantidad infinita de funciones que cumplen con las características presentadas en la definición 2.2; por lo tanto, se tiene que buscar aquella función que se adecue al problema de clasificación. La elección del patrón representante de cada clase dependerá del modelo en particular.

### 2.2.1 Clasificador de la distancia mínima

Este modelo de clasificación utiliza una estrategia muy peculiar: cada clase está representada por un solo patrón, por lo regular es el patrón medio o el centroide de todos los patrones que conforman la clase.

$$\mu_i = \frac{1}{N_i} \sum_{x^j \in C_i} x^j \text{ en donde } N_i \text{ es el número de patrones de la clase } C_i$$

Para clasificar un patrón de prueba  $\tilde{x}$  se calcularán las distancias entre  $\tilde{x}$  y todos los centroides. Si el  $\mu_k$  es más cercano a  $\tilde{x}$ , entonces  $\tilde{x}$  será asignado a la clase  $k$ , representada por el centroide  $\mu_k$ . El Clasificador Euclidiano es un caso particular de este modelo si utilizamos la distancia Euclidiana para calcular la distancia entre el patrón  $\tilde{x}$  y los centroides  $\mu_k$ .

La mayor ventaja del clasificador de la mínima distancia es su baja complejidad temporal. Sin embargo, este clasificador sólo ofrecerá resultados óptimos cuando las clases están normalmente distribuidas, es decir que se ajustan a la distribución normal; además, las clases deben tener una matriz de covarianza diagonal y se tienen las mismas varianzas en las diferentes direcciones, esto implica que las clases son isotrópicas [3,14,16].

Este clasificador maneja algunas estrategias para seleccionar los patrones representantes de las clases. Las más populares son las siguientes:

1. Seleccionar prototipos que pertenecen al conjunto de entrenamiento.
2. Generar prototipos, o patrones representantes, a partir de los patrones del conjunto de entrenamiento; estos prototipos pueden ser o no ser parte del conjunto original de entrenamiento.

La primera estrategia a su vez ofrece tres opciones diferentes:

1. Condensación: se elige un subconjunto de los mejores representantes de las clases.
2. Editar el conjunto de entrenamiento y remover los malos prototipos.
3. Agrupar los patrones mediante clustering y elegir los mejores representantes.

La segunda estrategia plantea sólo una opción diferente: elegir el patrón más cercano al representante de la clase que fue generado mediante el cálculo del centroide.

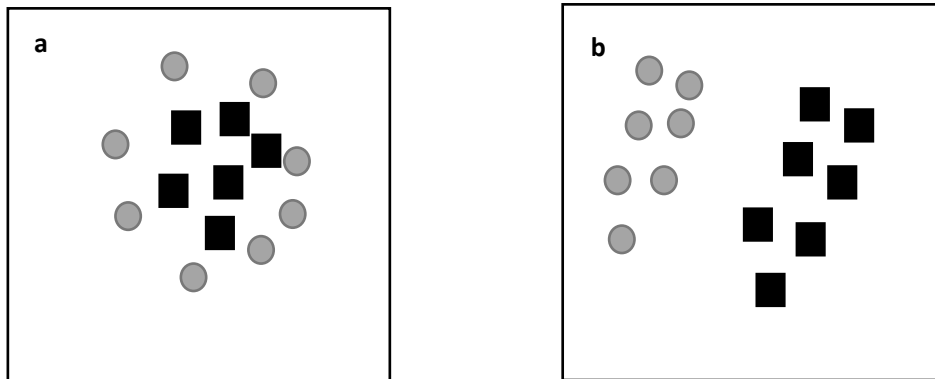
Este modelo de clasificación tendrá problemas cuando el centroide no es el mejor representante de la clase; esto ocurre cuando las clases tienen una estructura de cadena elongada o cuando algunas clases tienen una estructura concéntrica. Estos problemas se ilustran en la figura 2.

### 2.2.2 Clasificador k-NN (k - Nearest Neighbor)

Uno de los modelos de clasificación más simples en su base conceptual es el de los k Vecinos más Cercanos. Cuando la clasificación se hace con una gran cantidad de patrones, es posible demostrar que esta regla tiene una probabilidad de error menor que el doble del error óptimo [18].

A diferencia de otros clasificadores basados en la mínima distancia, este modelo utiliza algunos o todos los patrones del conjunto de entrenamiento para realizar la clasificación de los

patrones. La idea básica de este modelo es encontrar la similitud entre el patrón de prueba y cada patrón del conjunto de entrenamiento.



**Figura 2.** Ejemplos de bancos de datos en donde el clasificador de la distancia mínima fallará. (a) Clases concéntricas. (b) Clases en forma encadenada.

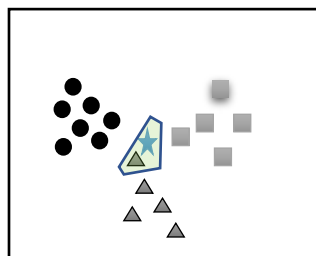
Existen dos variantes de este modelo. La primera, conocida como el clasificador 1-NN, asigna el patrón de prueba a la clase de su vecino más cercano. A continuación, se presenta el algoritmo del clasificador 1-NN (figura 3).

#### Algoritmo 2.2.2 Clasificador 1-NN

1. Seleccionar la métrica a emplear.
2. Calcular la distancia de un patrón desconocido  $\tilde{x}$  a cada uno de los patrones del conjunto fundamental o de entrenamiento.
3. Obtener la distancia mínima de entre las distancias calculadas.
4. Asignar a  $\tilde{x}$  la clase del patrón con la menor distancia.

A pesar de que este modelo es muy simple en su concepción e implementación, Michie comprobó empíricamente, en el proyecto Statlog, que este clasificador tiene buen desempeño en una gran cantidad de bancos de datos [29].

En años recientes, se ha empleado mucho el clasificador IB1 el cual se proporciona en la herramienta WEKA [70] y su funcionamiento es equivalente al del 1NN, con la excepción de que normaliza los rangos de los atributos, procesa los datos de forma incremental y tiene una regla básica para manejar valores perdidos [71].



**Figura 3.** Clasificación de un patrón desconocido (representado por la estrella) de acuerdo con el clasificador 1NN.

La segunda variante de este modelo, denominada como k-NN, presenta una diferencia muy sutil: en lugar de encontrar el vecino más cercano, se encuentran los k vecinos más cercanos. La clase mayoritaria entre los k vecinos es la clase que se asignará al patrón de prueba. El valor elegido para k es crucial; ya que, con el valor adecuado de k, el performance de este clasificador será mejor que el obtenido por el 1-NN.

Sin embargo, como se mencionó en el capítulo 1, este enfoque tiene problemas al momento de procesar una gran cantidad de patrones, además de que resiente los efectos de la llamada maldición de la dimensionalidad; por lo tanto, no se puede utilizar exitosamente en todos los bancos de datos; de igual forma, el rendimiento de los clasificadores basados en los vecinos más cercanos disminuye al momento de clasificar patrones atípicos o *outliers* [72].

### 2.2.3 Algunos trabajos relevantes del enfoque basado en métricas

El clasificador de mínima distancia sólo funciona adecuadamente con problemas con datos bien distribuidos; es por eso que se le han dado pocas aplicaciones ya que en la mayoría de problemas tiene un desempeño bajo. Una de las aplicaciones que se le han dado es en la construcción de clasificadores a partir de reglas obtenidas del conjunto de entrenamiento [73]. Por otro lado, se ha propuesto el uso de métricas para construir clasificadores basados en lattices difusas [74].

En cuanto a los clasificadores basados en los vecinos más cercanos (k-NN), hemos observado que a pesar de que este modelo tiene varios años de antigüedad, es uno de los más utilizados e incluso se emplea con frecuencia en estudios comparativos de clasificadores [75] o como una referencia directa para evaluar nuevos clasificadores. Sin embargo, constantemente se proponen cambios y mejoras de este modelo. Algunos autores han propuesto el uso de funciones kernel para mejorar el rendimiento de la clasificación en bancos de datos con baja o alta dimensionalidad [76]; este modelo también se ha aplicado en el monitoreo y detección del desgaste de componentes mecánicos [77]; de igual manera encontramos un sistema detector de fallas en circuitos analógicos en donde se proponen combinaciones del método de los k vecinos más cercanos y la *conventional Kernel density estimation* para estimar probabilidades acumulativas entre los k vecinos más cercanos al patrón de prueba, la clasificación se hace empleando un criterio de la mayor probabilidad acumulada [78].

Existen trabajos científicos en los que se busca mejorar la idea original del k-NN. En algunos se propone mejorar el rendimiento mediante el aprendizaje de la información de los patrones vecinos [79]; en algunos casos, el uso de la información de los patrones vecinos puede emplearse para construir clasificadores cuyo principio de clasificación es la distancia mínima a un conjunto de patrones medios generados a partir de los  $r$  patrones más cercanos por cada clase [80], esto permite que el clasificador k-NN aproveche de mayor forma la información de los vecinos más cercanos y de igual forma aprovechar otras funciones de distancia como la armónica [72]. También se han publicado artículos donde se ponderan los rasgos para disminuir sus efectos, independientemente de que los rasgos sean numéricos o categóricos [81]. Hay trabajos en los que

se propone fusionar este modelo con otros modelos de la Inteligencia Computacional como las Redes neuronales y los algoritmos genéticos [82]. En otros artículos se proponen solventar los problemas relacionados con la maldición de la dimensionalidad [83] o con la elección de la métrica [84]; inclusive este método de clasificación ha sido aplicado exitosamente para la clasificación de datos desbalanceados, esto es posible mediante la generación de vecindades dinámicas y la inducción de un sesgo, mediante el ajuste de la probabilidad posterior, en la clasificación de la clase “rara” (con menor número de patrones) [85]. Finalmente, es importante destacar que debido a la simplicidad de este modelo se han propuesto implementaciones en hardware que superan la velocidad de las implementaciones hechas con lenguajes de programación [86].

### 2.3 Árboles de decisión

Los árboles son modelos que mapean los valores de los rasgos del patrón a su etiqueta de clase. Ahora bien, la construcción de estos árboles es un problema que se puede resolver de manera recursiva. En primer lugar, se debe seleccionar uno de los atributos como nodo base y crear una rama por cada valor de los atributos. Este proceso tiene que repetirse de forma recursiva para cada una de las ramas, usando únicamente aquellas instancias que puedan llegar a esa rama. Si en todo momento los patrones en un nodo tienen el mismo valor de clase, entonces ya no se sigue explorando esa rama del árbol [87]. En este tipo de modelos la pregunta real es cómo determinar el nodo raíz a partir del conjunto de entrenamiento. En el árbol ID3 [19,87] la elección del mejor atributo se hace mediante la entropía; de manera que se elige aquel atributo que proporcione la mejor ganancia de la información. Este modelo de árboles de decisión está diseñado para trabajar con atributos categóricos, siendo poco útil al procesar datos numéricos.

Para remediar este problema, Ross Quinlan desarrolló una extensión del algoritmo ID3 que se conoce como C4.5 [20,88]. En este modelo, se construye un árbol de decisión de la misma forma que el ID3 a partir del concepto de entropía y ganancia. En este modelo es posible procesar datos numéricos y en cada nodo el algoritmo puede elegir el atributo que divide de mejor forma el conjunto de entrenamiento en sus diferentes clases. Como en el caso del ID3 el criterio de decisión es la ganancia normalizada de información, es decir diferencias en la entropía. Al igual que en el ID3 el procesamiento es recursivo. El árbol ID3 tiene algunos casos base dignos de considerar:

1. Si todas las muestras en un subconjunto recién creado tienen la misma clase, entonces se crea un nodo hoja con la etiqueta de clase.
2. El conjunto de entrenamiento está vacío.
3. La lista de atributos está vacía; en este caso se regresa la clase más frecuente.

Además de la creciente popularidad de los árboles de decisión, también ha llamado la atención de los métodos de ensambles de clasificadores [4], en los cuales es posible superar el rendimiento de un clasificador a partir de la combinación de varios clasificadores. A partir de este principio, Tin Ho propuso un primer llamado *Random Decision Forests* [89] y posteriormente

Breiman propuso una modificación basada en su método de Bagging [90] que se conoce como *Random Forest* o *Decision Forest* [21]. Por otro lado, el *Random Forest* se define como un método de ensamble para realizar tareas de clasificación y regresión de patrones mediante la construcción de varios árboles de aprendizaje durante la etapa de entrenamiento y determinando la clase a la que pertenece un patrón mediante una regla de votación o mediante la media de los valores predichos para las tareas de predicción [21]; la principal ventaja que ofrece este modelo contra otros modelos de árboles es que evita el sobrentrenamiento que se llega a producir en la etapa de aprendizaje. El algoritmo del *Random Forest* emplea el método de *Bagging* para crear los árboles de decisión a partir de un conjunto de entrenamiento  $E = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{NE}\}$ ; se aplica *Bagging* un total de B veces para seleccionar una muestra con reemplazo del conjunto de entrenamiento y crear árboles de decisión para esas muestras. Cuando se presenta un patrón  $\mathbf{x}^{Pk}$ ,  $\mathbf{x}^{Pk} \in P$  entonces este patrón se presenta a todos los árboles generados a partir del conjunto de entrenamiento y la clase se toma mediante una regla de votación. Breiman menciona que este proceso se puede mejorar, si se generan subconjuntos de rasgos aleatorios por cada árbol que se genere en la etapa de entrenamiento.

### 2.3.1 Algunas aplicaciones de los árboles de decisión

Los árboles de decisión han cobrado una gran popularidad en los últimos años, y esto se debe a que, a pesar de estar basados en un principio relativamente simple, logran muy buenos resultados en diversos problemas. En el caso del ID3 hemos encontrado aplicaciones recientes como la propuesta por Kale y Auti en donde se aplica este modelo para la creación de un sistema empleado para sugerir dietas infantiles [91]. En el caso de modelos más flexibles como el C4.5 hemos encontrado trabajos en los que se muestra su capacidad para tratar con patrones alterados por ruido [92,93], para la evaluación de la seguridad en redes eléctricas basadas en fotoceldas [94], análisis y clasificación de tráfico en *Botnets* [95], en la detección de síndromes que alteran el sueño [96], sistemas de predicción de sobrevivencia en pacientes gravemente heridos [97], entre otras más.

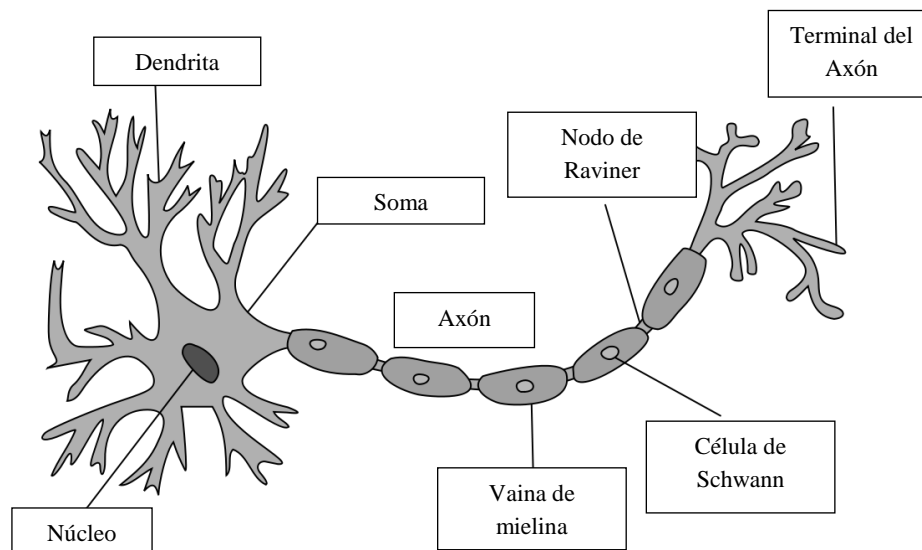
Para el caso del modelo *Random Forest* la gama de aplicaciones es considerable. Durante una exploración de trabajos recientes hemos encontrado que este modelo ha sido exitosamente aplicado en clasificación de escenas trabajando en conjunto con Redes Neuronales Convolucionales [98], en la predicción de la radiación solar [99], en la segmentación y ubicación de elementos estructurales del cerebro humano [100], en descubrimiento de medicamentos mediante la selección de descriptores moleculares [101], en la predicción del desempeño de índices de bolsas de valores [102], en la segmentación y clasificación de gestos [103], entre otras aplicaciones.

## 2.4 Enfoque Neuronal

Los clasificadores presentados hasta ahora son útiles para bancos de datos en donde las clases son linealmente separables. Desafortunadamente, esta situación ocurre muy pocas veces en los bancos de datos generados a partir de datos del mundo real, pues la mayoría de los bancos de datos están formados por clases de patrones que no son linealmente separables [14].

También es necesario tomar en cuenta que en algunos problemas no se pueden estimar las propiedades estadísticas de los patrones [3]; por lo tanto, es conveniente disponer de modelos que puedan ser entrenados únicamente con los patrones pertenecientes al conjunto de entrenamiento.

Las redes neuronales artificiales se iniciaron como un intento de estudiar el funcionamiento del cerebro humano [14]. El cerebro tiene millones de neuronas, las cuales se conectan entre sí utilizando transmisores que son activados mediante señales electromagnéticas; esta comunicación se hace mediante uniones llamadas sinapsis. Las entradas de una neurona son combinadas y si éstas rebasan un umbral la neurona se dispara y su salida es enviada a otras neuronas a través del axón. Las Redes Neuronales Artificiales buscan emular este comportamiento [12]. La fisiología básica de una neurona se presenta en la figura 4.



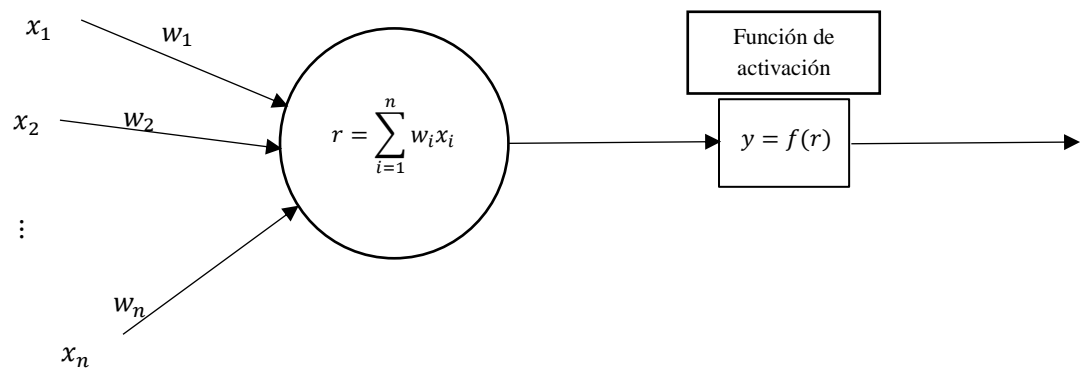
**Figura 4. Neurona** fuente: Quasar Jarosz at English Wikipedia, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=7616130>

La salida de una red neuronal artificial depende de los valores de sus entradas y de los pesos asociados a estos valores de entrada. La etapa de entrenamiento consiste en hacer que la red proporcione la salida correcta para todos los patrones del conjunto de entrenamiento, y se inicia creando pesos aleatorios para cada una de las entradas de la neurona. Posteriormente, se proporciona un patrón de entrada a la red, el cual se multiplica por los pesos asociados y se suman obteniéndose una salida. Si la salida es correcta no se hace nada a los pesos de la red; en cambio, si la salida es errónea se calcula el error y éste se usa para ajustar los pesos de la red. Este proceso se debe realizar para todos los patrones del conjunto de entrenamiento y termina hasta que todas



las salidas sean correctas o se rebase un límite máximo de iteraciones [3,14]. En resumen, el entrenamiento de una red neuronal implica cambiar los pesos de forma adecuada.

Las neuronas artificiales están modeladas para emular el funcionamiento de las neuronas del cerebro humano. Una neurona artificial está representada matemáticamente por dos módulos: un módulo lineal de activación o inhibición, y un módulo no lineal que limita los niveles de una señal dentro de una banda finita [12]. Las entradas de la neurona son ponderadas y sumadas; si esta suma excede un umbral entonces la neurona emite una señal. Por ejemplo, si se rebasa el umbral la neurona podría emitir un 1 y en caso de que no se rebasa la neurona emitirá un 0 (o bien se puede definir con 1 y -1) [3,12,16]. En la figura 5 se esquematiza el funcionamiento básico de una neurona artificial, la cual fue propuesta originalmente por McCulloch y Pitts [104].



**Figura 5. Modelo de neurona artificial.**

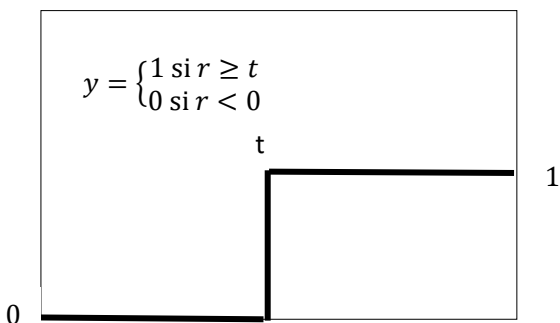
Si  $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$  es la entrada de la neurona con los pesos correspondientes  $\mathbf{w} = (w_1, w_2, \dots, w_n)^t$  la activación de la neurona se expresa de la siguiente expresión [3]:

$$r = w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (2.8)$$

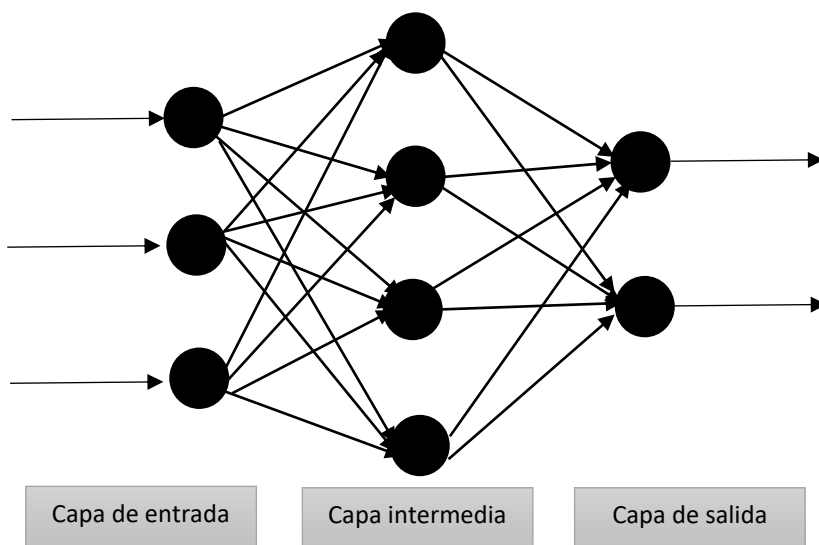
La estructura circular, donde se realiza la suma de las entradas por los pesos, en la figura 4 representa el cuerpo de la neurona, mientras que las entradas pueden ser vistas como las dendritas. La sinapsis de la neurona puede ser modelada como la función de activación mientras que la conexión entre el “cuerpo de la neurona” y la función de activación equivaldría al axón de la neurona [12].

La salida de la neurona es una función de activación. Una de las funciones de activación más antiguas y populares es la función umbral, la cual es una variante de la función escalón unitario [12,14]. La función se ilustra en la figura 6.

En términos generales, una red neuronal se compone por la unión de varias neuronas artificiales. En la figura 7 se presenta una red neuronal a partir de la conexión de varias neuronas artificiales.



**Figura 6. Función umbral.**



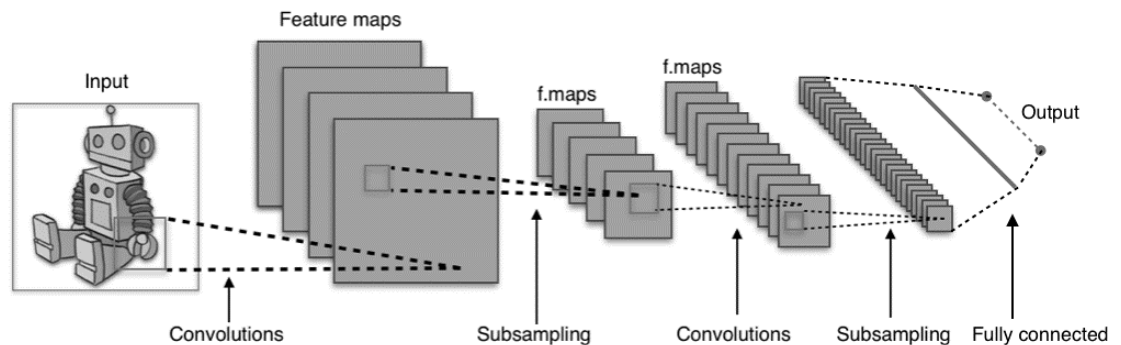
**Figura 7. Estructura de una Red Neuronal Artificial.**

### 2.4.1 Aplicaciones del enfoque neuronal

Las redes neuronales aún representan una línea de investigación vigente, ya que se siguen proponiendo formas de mejorar su desempeño en la tarea de la clasificación inteligente de patrones. En algunos casos se han propuesto redes con retardos en las entradas, las cuales facilitan la representación de sistemas dinámicos, no paramétricos y con incertidumbre [105]; además, se ha propuesto adaptar una red neuronal en sistemas de inferencia difusa [106]; mientras que en otros casos se ha hecho investigación en cuanto al uso de otras funciones de activación [107] o diseñando redes neuronales a partir de elementos de hardware como los memristores [108]. Es importante destacar que en últimos años ha cobrado relevancia el tema de las redes neuronales pulsantes, las cuales se basan en modelos biológicos modernos de las neuronas humanas, particularmente llaman la atención los modelos de redes neuronales pulsantes las cuales buscan aproximar el comportamiento de las reacciones electroquímicas que se producen en las neuronas [109–111].

Debido a su gran popularidad las redes neuronales se han aplicado en una gran cantidad de situaciones. Algunas de las aplicaciones más recientes son las que a continuación se enlistan: análisis de señales cerebrales [112]; reconocimiento facial [113]; problemas de optimización [114]; predicción en series de tiempo [110,115]; predicción de precios de servicios básicos [116]; en la predicción de la demanda intermitente de automóviles [117]; reconocimiento de voz [118]; en la remoción de ruido de señales [119] y en modelos implementados en hardware para la clasificación inteligente de gases [120].

En años recientes ha cobrado mucha relevancia el tema del *Deep Learning*. En el emblemático artículo de LeCun, Benio y Hinton [121] los autores comentan que el *Deep Learning* facilita la creación de modelos compuestos por muchas capas de procesamiento que permiten abstraer los datos a distintos niveles de abstracción. El *Deep Learning* propone, entre otros modelos similares, las redes neuronales convolucionales las cuales se pueden analizar como arreglos múltiples, lo cual facilita el procesamiento y reconocimiento de objetos en imágenes o del reconocimiento de voz y del lenguaje natural. Las redes neuronales convolucionales se basan en cuatro ideas clave que se pueden tomar de las señales naturales: conexiones locales, pesos compartidos, agrupación (*pooling*) y el uso de muchas capas. La arquitectura general de una red neuronal convolucional se presenta en la figura 8.



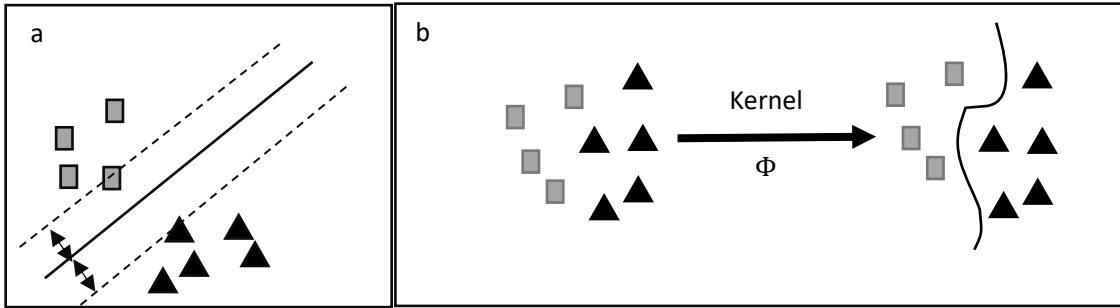
**Figura 8. Arquitectura de una Red Neuronal Convolucional. Fuente: By Aphex34 - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=45679374>**

Dado que el tema está en pleno auge, a pesar de que la idea no es reciente, se han propuesto una gran cantidad de aplicaciones, entre las que destacan: clasificación inteligente de escenas [98], identificación de las tos a partir del espectro de los datos acústicos [122] y segmentaciones de tumores cerebrales [123].

## 2.5 Máquinas de Soporte Vectorial (SVM)

Las SVMs son un modelo de reconocimiento de patrones que está basado en la teoría de aprendizaje estadístico la cual fue desarrollada por Vapnik, Chervonenkis y otros en los *Bell Laboratories* de AT&T [23]. La teoría de aprendizaje estadístico de Vapnik es diferente a la presentada en el enfoque probabilístico-estadístico, puesto que no está sujeta al teorema de Bayes.

Las SVM fueron desarrolladas como un clasificador binario que abstrae la frontera de decisión en un espacio multidimensional, y que construye un modelo lineal que separa los elementos de ambas clases. Posteriormente las SVM han sido adaptadas a problemas de clasificación con datos no separables [3,14,16] y finalmente se extendió para problemas de clasificación de más de dos clases. El funcionamiento básico de una SVM, para patrones linealmente separables, se ilustra en la figura 9 (a) y el funcionamiento de las SVM para patrones que no son linealmente separables se ilustra en la figura 9(b).



**Figura 9. Funcionamiento básico de las SVM: (a) en patrones linealmente separables; (b) en patrones que no son linealmente separables.**

La idea subyacente del algoritmo de las SVM es que los patrones pueden ser mapeados de forma no lineal en un espacio de alta dimensionalidad [24] donde es posible generar una superficie lineal de decisión que permita clasificar los patrones de manera óptima. Este hiperplano, o superficie de decisión, es generado a partir de un subconjunto de vectores de entrenamiento que se conoce como vectores de soporte [124] que regularmente son aquellos patrones más cercanos a la frontera de las clases; el hiperplano obtenido se considera como óptimo debido a que maximiza el margen existente entre los patrones pertenecientes a dos clases diferentes, independientemente de sus dimensiones y minimiza el error obtenido en tareas de clasificación y regresión.

La fase de entrenamiento de las SVM se plantea de la siguiente forma [125]: dado un conjunto de patrones de entrenamiento  $\mathbf{x}^i \in \mathbb{R}^n, i = 1, \dots, l$  donde  $l$  es el número de patrones de entrenamiento y  $n$  es la dimensión de los patrones y  $y^i \in \{-1, 1\}$  es la etiqueta de clase; resolver el problema de optimización presentado en la expresión 2.9.

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2.9)$$

$$\text{Sujeto a } y^i (w^T \phi(\mathbf{x}^i) + b) \geq 1 - \xi_i$$

El problema descrito anteriormente se interpreta como buscar el vector de pesos  $w$  y el escalar  $b$  que satisfagan cada una de las restricciones de la solución óptima; en el momento en que estas restricciones queden satisfechas los patrones serán linealmente separables. Dentro de los elementos que son importantes en esta formulación, tenemos  $\phi(\mathbf{x}^i)$  que es un mapeo no lineal para

representar el patrón  $\mathbf{x}^i$  a un espacio de mayor dimensionalidad; sin embargo, puede ocurrir el caso, poco probable, de que el patrón sea linealmente separable en su dimensión original, si este es el caso no se aplicará ninguna transformación. Por otro lado, y considerando que es poco probable que los patrones sean linealmente separables se agrega un conjunto de variables de holgura las cuales se encargan de penalizar los errores de clasificación de los patrones de entrenamiento; estas variables se asocian al parámetro C que se utiliza para controlar el sobre-entrenamiento [23,24,124,125].

Debido al número de patrones de entrenamiento o la alta dimensionalidad de éstos, es muy complicado resolver el problema de optimización presentado en la expresión 2.9; por lo tanto, se debe resolver el problema en términos de la formulación dual y mediante el método de los multiplicadores de Lagrange [24,126]. Al resolver el problema de optimización, el vector óptimo de pesos satisface la siguiente restricción presentada en la expresión 2.10:

$$\mathbf{w} = \sum_{i=1}^l y^i \alpha_i \phi(\mathbf{x}^i) \quad (2.10)$$

Una vez resuelto este problema de optimización concluye la etapa de entrenamiento de las SVM. La etapa de prueba [127] consiste en clasificar cada uno de los patrones de prueba  $\mathbf{x}$  en la siguiente función de decisión, expresión 2.11, que se generó al resolver el problema de optimización.

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \phi(\mathbf{x}) + b) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(\mathbf{x}^i, \mathbf{x}) + b\right) \quad (2.11)$$

Donde  $\alpha_i$  son los multiplicadores de Lagrange, y  $K(\mathbf{x}^i, \mathbf{x}) = \phi(\mathbf{x}^i)^T \phi(\mathbf{x})$  es la función Kernel que es la que se encarga de representar a los patrones en un espacio en donde sí sean linealmente separables para después continuar con el proceso de entrenamiento o clasificación de las SVM [128–130]. Este es el modelo base de las SVM también conocido como C-SVC (*C-Support Vector Classification*) [127] o Red de Soporte Vectorial. Cabe destacar que a pesar de que inicialmente las SVM se plantearon en problemas de clasificación binaria, posteriormente se amplió su alcance a más clases y se han propuesto métodos para tareas de regresión como lo es la  $\varepsilon$ -*Support Vector Regression* ( $\varepsilon$ -SVR) [127,131].

### 2.5.1 Aplicaciones de las SVM

Se han planteado diferentes modificaciones al modelo original propuesto por Vapnik; en esta sección presentamos algunas de esas propuestas. Hay algunos autores que han propuesto modelos de SVM para atacar problemas de clasificación multiclase a partir de la solución de sub-problemas de clasificación binaria; con este cambio se busca generalizar el performance de la clasificación a más de dos clases [132]; otros autores han publicado modificaciones que aparte de mejorar la capacidad de generalización, también buscan simplificar la formulación del modelo de

clasificación [133]. Hemos encontrado publicaciones en las que se reporta que es posible mejorar la generalización mediante algoritmos auxiliares que eliminan características de forma recursiva y que aparte buscan la optimización de parámetros [134]. Otros autores propusieron modificaciones que además de mejorar el performance del modelo también buscan mejorar la velocidad con la que se realiza el entrenamiento [135], en el procesamiento de bancos de datos grandes [136] y en otros casos se propusieron modificaciones orientadas a proporcionar mayor robustez al modelo con relación a perturbaciones aleatorias que pudieran sufrir los datos [137] o mucho ruido [138].

A pesar de que las SVM buscan maximizar el margen que separa a dos clases, también se ha considerado complementar esta maximización con la minimización de la dispersión de los patrones que forman las clases. Por lo tanto, la generación del hiperplano óptimo considera dos problemas: maximizar la separación entre las clases y minimizar la dispersión interna de las clases [128]. Algunas formas alternas de modificar el entrenamiento de las SVM involucra el uso de metaheurísticas [129].

No solo hemos encontrado publicaciones destinadas a modificar el modelo original de las SVM. También hemos encontrado trabajos en los que se aplican las SVM a problemas reales como los listados a continuación: riesgo en la asignación de créditos [139], detección de fallos en medicamentos [140], predicción de consumo eléctrico [141], predicción de momentos en los que podrían ocurrir accidentes graves en plantas nucleares [142], predicción de la rigidez de mezclas de asfalto y PET [143], en la estimación de anomalías en las aguas oceánicas como consecuencia del calentamiento global [130], en la creación de sistemas detectores de intrusiones [144], en la detección de síndromes en bancos de datos metagenómicos [145] control del desempeño de *Routers* para el ahorro de energía [146], reconocimiento de expresiones faciales [147], en teoría de sistemas de control [148], en la detección de fallos en baleros de maquinaria industrial [149], en la detección de tornillos aflojados [150], entre otras más.

## 2.6 Enfoque Asociativo

El enfoque asociativo del Reconocimiento de Patrones se inició en el año 2002 [26], cuando se presentó un nuevo modelo de Memoria Asociativa en donde las operaciones de aprendizaje y recuperación se realizan mediante dos nuevos operadores:  $\alpha$  y  $\beta$ . A partir de este trabajo se han propuesto más de una decena modelos de Reconocimiento de Patrones que han realizado, de forma exitosa, tareas de clasificación, regresión y recuperación de patrones. Es posible afirmar que se podrían hacer tareas de agrupamiento o clustering.

En las siguientes secciones presentaremos dos de los modelos más relevantes que se han creado en el grupo de investigación alfa beta: El Clasificador Híbrido Asociativo con Traslación (CHAT), el cual está formado por las mejores características de dos modelos primitivos de memorias asociativas; y finalmente se presentará el clasificador Gamma, el cual funciona mediante un operador de similitud basado en los operadores  $\alpha$  y  $\beta$ .

### 2.6.1 Clasificador Híbrido Asociativo con Traslación (CHAT)

El CHAT se base en los mejores principios de la *Lernmatrix* de Steinbuch y el *Linear Associator* de Anderson-Kohonen para realizar tareas de clasificación de patrones. El CHAT supera las graves desventajas que presentan, de forma individual, estos dos modelos de memorias asociativas: Por una parte, evita la saturación de la *Lernmatrix* y por otra parte solventa el problema del *Linear Associator* de requerir patrones ortonormales.

La versión inicial de este clasificador, el Clasificador Híbrido Asociativo, tenía graves problemas cuando la magnitud de los patrones de una clase es mayor que la magnitud de los patrones de las otras clases. Para resolver este problema, se hace una traslación de ejes previo a la fase de entrenamiento, esta versión modificada se denominó Clasificador Híbrido Asociativo con Traslación (CHAT). A continuación, se presenta el algoritmo del CHAT [32,36].

#### Algoritmo 2.5.1 Algoritmo del Clasificador Híbrido Asociativo con Traslación

1. Sea un conjunto fundamental de patrones de entrada, de dimensión  $n$ , con valores reales en sus componentes (a la manera del *Linear Associator*), que se aglutinan en  $m$  clases diferentes.
2. A cada uno de los patrones de entrada que pertenece a la clase  $k$  se le asigna el vector formado por ceros, excepto en la coordenada  $k$ -ésima, donde el valor es uno (a la manera de la *Lernmatrix*).
3. Se calcula el vector medio del conjunto fundamental de patrones.
4. Se toman las coordenadas del vector medio a manera de centro de un nuevo conjunto de ejes coordenados.
5. Se realiza la traslación de todos los patrones del conjunto fundamental.
6. La fase de aprendizaje es similar a la del *Linear Associator*.
7. La fase de recuperación es similar a la que usa la *Lernmatrix*.
8. Se traslada todo patrón a clasificar a los nuevos ejes.
9. Se procede a clasificar los patrones desconocidos.

### 2.6.2 Clasificador Gamma

Este modelo está basado en un nuevo operador llamado operador gamma de similitud; a su vez, este operador está basado en los operadores asociativos  $\alpha$  y  $\beta$ . El operador gamma de similitud indica si dos vectores o patrones son parecidos o no dado un grado de disimilitud  $\theta$ , por tal razón este operador es de similitud. El argumento  $\theta$  indica la tolerancia para que dos vectores sean considerados similares, al compararlos, a pesar de que son diferentes. A continuación, se presenta el algoritmo del clasificador Gamma tal como se describe en [36].

Sean dos patrones binarios  $\mathbf{x} \in A^n$  y  $\mathbf{y} \in A^m$  de dimensiones  $m$  y  $n$  respectivamente, con  $n, m \in \mathbb{Z}^+$  y con  $n \leq m$ ; y un número entero no negativo  $\theta$ . La salida es un número binario que se calcula de la siguiente forma:

$$\gamma_g(\mathbf{x}, \mathbf{y}, \theta) = \begin{cases} 1 & \text{Si } m - u[\alpha(\mathbf{x}, \mathbf{y}) \bmod 2] \leq \theta \\ 0 & \text{en otro caso} \end{cases}$$

Algoritmo 2.5.2 Sea el conjunto fundamental del clasificador Gamma de acuerdo con la definición del operador gamma de similitud. Al presentarse un patrón a clasificar  $\tilde{\mathbf{x}}$ , donde  $\tilde{\mathbf{x}}$  es un vector real  $n$ -dimensional  $\tilde{\mathbf{x}} \in \mathbb{R}^n$ , con  $n \in \mathbb{Z}^+$ , se realiza lo siguiente.

1. Codificar cada componente de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado, restando el valor menor a todos los valores por cada componente; asimismo, se obtiene un valor  $e_m = \bigvee_{i=1}^p x_j^i$  por cada componente. Con lo anterior se desplaza el rango de cada componente para que vaya de 0 a  $e_m$ . Así la componente  $x_j^i$  se transforma en un vector binario de dimensión  $e_m(j)$ .
2. Codificar cada componente del patrón a clasificar con el código Johnson-Möbius modificado, utilizando las mismas condiciones que se utilizaron para codificar las componentes de los patrones fundamentales. En caso de que alguna componente del patrón a clasificar sea mayor al  $e_m$  correspondiente ( $\tilde{x}_\xi > e_m(\xi)$ ), igualar esa componente a  $e_m(\xi)$  y guardar su valor anterior en la variable  $mgamma_\xi$ . Por otro lado, si alguna componente da un valor negativo una vez desplazada, igualar esa componente a 0 y asignar el valor  $e_m(\xi) + |\tilde{x}_\xi|$  a  $mgamma_\xi$ .
3. Calcular el parámetro de paro  $\rho$  y el parámetro de pausa  $\rho_0$ . Dependiendo del problema a tratar, algunas posibilidades sugeridas para estos parámetros son las siguientes.
  1.  $\rho = \bigwedge_{j=1}^n (\bigvee_{i=1}^p x_j^i)$
  2.  $\rho = \frac{1}{n} \sum_{j=1}^n (\bigvee_{i=1}^p x_j^i)$
  3.  $\rho = \bigvee_{j=1}^n (\bigvee_{i=1}^p x_j^i)$
  4.  $\rho_0 = \bigwedge_{j=1}^n (\bigvee_{i=1}^p x_j^i)$ , sobre todo si  $\rho = \bigvee_{j=1}^n (\bigvee_{i=1}^p x_j^i)$
  5.  $\rho_0 > \rho$  cuando se desea asignar forzosamente una clase conocida a patrones desconocidos.
4. Determinar el umbral de pausa  $u$ . Considerando que el valor de este umbral depende fuertemente de las características del problema y las propiedades del conjunto fundamental, se presentan las siguientes sugerencias como valores iniciales.
  1.  $u = 0$
  2.  $u = n$
5. Determinar los pesos de cada dimensión  $w_i \in \mathbb{R}^+ | i = 1, 2, \dots, n$ . Con la finalidad de afinar estos pesos a las características del problema y las propiedades del conjunto fundamental, se sugieren los siguientes rangos como valores iniciales empíricos.
  1. Dentro del rango  $[1.5, 2]$  a las dimensiones que sean puntualmente separables para todas las clases.



2. Dentro del rango [1,1.5] a las dimensiones que sean puntualmente separables para algunas clases o bien que sean puntualmente segmentables para todas las clases.
3. Dentro del rango [0.8, 1.2] a las dimensiones que sean puntualmente segmentables para todas o algunas clases.
4. Dentro del rango (0,0.5] a las dimensiones que sean puntualmente no separables.
6. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo  $x^\mu$ , se convierta en dos índices: uno para la clase (por ejemplo, para la clase  $i$ ) y otro para el orden que le corresponde al patrón dentro de esa clase (por ejemplo  $\omega$ ). Bajo estas condiciones ejemplificadas, la notación para el patrón  $x^\mu$  será ahora, con la transformación  $x^{i\omega}$ . Lo anterior se realiza para todos los patrones del conjunto fundamental.
7. Inicializar  $\theta$  a 0.
8. Realizar la operación  $\gamma_g(x_j^\mu, \tilde{x}_j, \theta)$  para cada componente de cada uno de los patrones fundamentales y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$ .
9. Calcular la suma ponderada inicial  $c_\mu^0$  de los resultados obtenidos en el paso 7, para cada patrón fundamental  $\mu = 1, 2, \dots, p$ :

$$c_\mu^0 = \sum_{j=1}^n w_j \cdot \gamma_g(x_j^\mu, \tilde{x}_j, \theta)$$

10. Si existe un máximo único, cuyo valor es además igual a  $n$ , asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$C_{\tilde{x}} = C_j \text{ tal que } \bigvee_{\mu}^p c_\mu^0 = c_j^0 = n$$

11. Realizar la operación  $\gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)$  para cada clase y para cada componente de cada uno de los patrones fundamentales que corresponden a esa clase, y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$ .
12. Calcular la suma ponderada  $c_i$  de los resultados obtenidos en 11, para cada clase  $i = 1, 2, \dots, m$ :

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n w_j \cdot \gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)}{k_i}$$

13. Si existe más de un máximo entre las sumas ponderadas por clase, incrementar  $\theta$  en 1 y repetir los pasos 11 y 12 hasta que exista un máximo único; o se cumpla con la condición de pausa  $\theta = \rho_0$ ; o se cumpla con la condición de paro:  $\theta \geq \rho$ .
14. Si se cumple con la condición de pausa  $\theta = \rho_0$ , se compara el valor máximo de las sumas ponderadas con el umbral de pausa.

1. Si  $\bigvee_{i=1}^m c_i \leq u$  entonces se asigna la clase desconocida al patrón a clasificar:

$$C_{\bar{x}} = C_0$$

2. Si  $\bigvee_{i=1}^m c_i > u$  entonces se continúa en el paso 11.
15. Si existe un máximo único, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$C_{\bar{x}} = C_j \text{ tal que } \bigvee_{i=1}^m c_i = c_j$$

16. En caso contrario: Si  $\lambda$  es el índice más pequeño de clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase  $C_\lambda$ .

En la siguiente sección presentamos algunas de las aplicaciones de estos modelos, las cuáles han logrado resultados importantes.

### 2.5.3 Aplicaciones del enfoque asociativo

El enfoque asociativo del Reconocimiento de Patrones es relativamente reciente; derivado de este enfoque se han propuesto varios modelos nuevos [32–34,36,42,43] los cuales ofrecen resultados competitivos. Una de las características principales de los modelos asociativos propuestos en el grupo alfa beta es que obtienen un *resubstitution error* de 0 (es decir recuperan o clasifican perfectamente el conjunto de entrenamiento).

El enfoque asociativo del RP ha sido aplicado exitosamente en diversos problemas. A continuación, enlistamos algunas de las aplicaciones sobresalientes:

- Reconocimiento de voz [151],
- Tratamiento de bancos de datos desbalanceados [152],
- Problemas médicos [153–156],
- Predicción de series de tiempo aplicadas en pozos petroleros [38,157].

En el caso particular del clasificador Gamma ha sido tan amplia la cantidad de aplicaciones de este modelo que incluso se han presentado trabajos en donde se han realizado modificaciones a su funcionamiento [39] empleando como base una técnica muy usual en la Inteligencia Computacional: el uso de las metaheurísticas como la evolución diferencial [158,159].

Es importante destacar el hecho de que no solo se han desarrollado modelos asociativos basados en las memorias asociativas alfa beta. Tras una investigación extensa encontramos algunos trabajos relacionados con modelos asociativos desarrollados en otras partes del mundo. A continuación, se presentan algunos trabajos recientes en el campo de las memorias asociativas.

Se han propuesto modelos asociativos basados en redes neuronales como el caso de los trabajos propuestos por Zhou *et al.* [160,161] donde se muestra cómo es posible crear memorias asociativas a partir de redes neuronales continuas recurrentes; por otro lado, además de sustentarse en las redes neuronales, han surgido modelos asociativos basados en la teoría del caos [162]; en otros trabajos encontramos propuestas de extender la teoría de las redes pulsantes para la generación de memorias asociativas [163]. También encontramos que algunos de los trabajos retoman el concepto de las memorias asociativas bidireccionales [164–167], pero desde el punto de vista de las redes neuronales; estas memorias aceptan retardos en las entradas o buscan la forma de mejorar su estabilidad.

Como parte de la investigación que realizamos sobre los modelos asociativos, también encontramos propuestas relacionadas con las memorias asociativas difusas como las propuestas por Sussner [44] o como aquellas que son consideradas caso particular de redes neuronales difusas [168–170], las cuales, incluso, pueden crearse a partir de métodos de aprendizaje no supervisado [171]. Estas memorias difusas tienen la capacidad de corregir errores o apoyar tareas de clasificación [168]. En otros trabajos, los autores han formulado memorias asociativas basadas en números complejos, las cuales se han utilizado exitosamente en la clasificación de imágenes [172]. Las memorias asociativas se han empleado exitosamente en la solución del juego del sudoku, dado que se utilizó una memoria asociativa basada en los principios de las redes neuronales [173] para encontrar patrones de Sudoku. A pesar de que la Memoria Hopfield ofrece un performance muy bajo, hay quienes la siguen utilizando como base para proponer nuevos modelos [174–176] e incluso el modelo de Hopfield ha servido de base para modelos asociativos que funcionan de manera paralela [177]; de igual forma los modelos asociativos, basados en producto tensor de Kronecker para detección automática de temas en textos [178].

Por otro lado, otros autores han publicado otros trabajos de investigación en los que el problema del diseño de las memorias asociativas es un problema relacionado con Hardware. Hu *et al.* [179] proponen una memoria asociativa basada en memristores de escala nanométrica; esta red neuronal asociativa es capaz de exhibir comportamientos inteligentes tales como la cognición y la recuperación, y “pérdida” de patrones, esto permite a los autores afirmar que es posible simular emociones básicas del ser humano e incluso “olvidar” patrones de datos que actúan como estímulos. Los autores destacan el hecho de que las neuronas de esta red asociativa pueden ajustar su comportamiento de acuerdo con el comportamiento de las neuronas pre-sinápticas y post-sinápticas a cada neurona. El procesamiento de emociones básicas también es abordado en trabajos como el propuesto por Kassab [180].

De todos los trabajos analizados, observamos que, en la mayoría de ellos, salvo el propuesto por Valle en [181], las memorias asociativas son vistas como un caso particular de las redes neuronales. De hecho, varios de los modelos fueron formulados a partir de las propiedades de los modelos del enfoque neuronal; incluso se han utilizado las novedosas redes pulsantes como base para la creación de memorias asociativas [182]; un caso interesante es el trabajo propuesto por Ge *et al.* [183] ya que en este se propone un modelo de Red Neuronal Asociativa o de Memoria Asociativa que explota propiedades visuales de los patrones, esto es posible mediante el uso de Kernels de Base Elíptica y mediante el concepto de *Network Potential Fields* los cuales se generan

a partir de combinaciones lineales de Kernels de base Elipsoidal para Memorias Auto-asociativas y para Kernels de Base Radial para Memorias Hetero-asociativas; los autores del modelo aseguran que los resultados obtenidos y el tiempo de ejecución son bastante buenos considerando el hecho de que como función de activación no se requiere una función derivable. Los modelos de memorias asociativas basadas en redes neuronales se pueden aplicar exitosamente en diversos campos, aunque uno de los más frecuentes es el de la visión por computadora [113,183].

Finalmente, es importante considerar que a pesar de que el enfoque neuronal ofrece modelos poderosos (incluidos los modelos de memorias asociativas), a menudo los algoritmos para entrenar las redes neuronales son iterativos y sujetos a condiciones de convergencia; mientras que en el enfoque asociativo del Reconocimiento de Patrones, y en particular para el grupo alfa beta, las memorias asociativas no están restringidas a las propiedades o a las restricciones de las redes neuronales; por lo tanto, es posible proponer modelos nuevos con una mayor flexibilidad y sin problemas de convergencia.

# Capítulo 3. Materiales y Métodos

En este capítulo describiremos las bases formales en las que sustentamos teóricamente el Clasificador de Heaviside.

En la primera sección presentaremos algunos conceptos fundamentales acerca de las memorias asociativas, ya que el clasificador propuesto es un clasificador asociativo. En la siguiente sección presentaremos la definición formal de la función de Heaviside, la cual es fundamental para nuestro modelo de clasificación. Considerando que el clasificador de Heaviside también está basado en las *lattices* y en las estructuras algebraicas ordenadas, en la sección 3 de este capítulo presentaremos algunas definiciones que han sido empleadas hasta ahora. Finalmente, y tomando en cuenta que el clasificador de Heaviside funciona con patrones cuyas componentes están en términos de un sistema numérico base  $b$ , con  $b > 1$ , en la sección 4 de este capítulo se presentarán algunos conceptos relacionados con los sistemas de numeración.

## 3.1 Memorias asociativas

En esta sección se presentan algunos conceptos básicos acerca de las memorias asociativas. El material está basado en [26].

El propósito fundamental de una memoria asociativa es recuperar patrones completos que pueden estar alterados con ruido aditivo, sustractivo o combinado, particularmente en el caso de los patrones binarios. De acuerdo con lo anterior, una memoria asociativa  $M$  puede formularse como el sistema esquematizado en la figura 10.



**Figura 10. Funcionamiento general de una memoria asociativa.**

En este sistema, un patrón de entrada está representado por un vector columna denotado por  $\mathbf{A}$ , mientras que el patrón de salida está denotado por  $\mathbf{D}$ .

Cada uno de los patrones de entrada forma una asociación correspondiente con el patrón de salida. La notación para una asociación es similar a la notación de que denota a una pareja ordenada; por ejemplo, los patrones  $\mathbf{A}, \mathbf{D}$  de la figura 7 formarían la asociación  $(\mathbf{A}, \mathbf{D})$ . Considerando lo anterior, tenemos que para el  $k$ -ésimo patrón perteneciente a un conjunto de patrones, en donde  $k$  es un entero positivo, la asociación quedaría de la siguiente forma  $(\mathbf{A}^k, \mathbf{D}^k)$ .

Una memoria asociativa  $M$  se representa mediante una matriz cuya componente  $ij$ -ésima es  $m_{ij}$ ; la matriz  $M$  se forma a partir de un conjunto finito de asociaciones conocidas de antemano, dicho conjunto por lo regular se denomina como el conjunto fundamental de patrones, o simplemente el conjunto fundamental (aunque en el contexto de la clasificación utilizaremos el término conjunto de entrenamiento).

Si denotamos con  $p$  la cardinalidad del conjunto fundamental, entonces éste se puede representar de la siguiente forma:

$$\{(A^k, D^k) | k = 1, 2, \dots, p\}$$

La naturaleza del conjunto fundamental proporciona un criterio importante para clasificar las memorias asociativas. Si se cumple que  $A^k = D^k \forall k \in \{1, 2, \dots, f\}$ , entonces se dice que la memoria es autoasociativa; de otro modo la memoria será heteroasociativa. De manera formal una memoria será heteroasociativa si  $\exists k \in \{1, 2, \dots, f\}$  para el que se cumple que  $A^k \neq D^k$ .

Si al presentarle a la memoria asociativa  $M$  un patrón alterado  $\tilde{A}^\omega$  como entrada, con  $\omega \in \{1, 2, \dots, f\}$ ,  $M$  responde con el patrón de salida  $D^\omega$  (el cual corresponde con el patrón original  $A^\omega$ ), entonces se dice que la recuperación de la memoria es correcta. Una memoria asociativa es correcta cuando recupera correctamente todos los patrones fundamentales.

Es importante mencionar que el planteamiento de un modelo de memoria asociativa requiere la especificación de dos conjuntos denominados  $E$  y  $G$ . La importancia de estos dos conjuntos radica en que las componentes de los patrones, tanto de entrada como de salida, estarán en términos del conjunto  $E$ , mientras que las entradas de la matriz  $M$  estarán en términos del conjunto  $G$ .

Las restricciones sobre  $E$  y  $G$  estarán dadas en términos del modelo asociativo que se esté trabajando. Por ejemplo, para las memorias asociativas alfa beta  $E = \{0, 1\}$  y  $G = \{0, 1, 2\}$ . Ya especificadas las características de los conjuntos  $E$  y  $G$ , ahora es necesario mencionar algunas consideraciones acerca de las dimensiones de los patrones tanto de entrada como de salida.

Sean  $m, n$  dos números enteros positivos. Se denota por  $n$  la dimensión de los patrones de entrada y por  $m$  la dimensión de los patrones de salida; los valores de  $m$  y  $n$  pueden ser iguales, lo cual es un requisito cuando se tiene una memoria autoasociativa. Por otro lado, si ocurre que  $m \neq n$ , es evidente que la memoria es heteroasociativa.

Cada patrón de entrada tiene  $n$  componentes, cuyos valores pertenecen al conjunto  $E$ , y cada vector columna que representa al patrón de salida posee  $m$  componentes cuyos valores también pertenecen al conjunto  $E$ . Es decir:

$$A^k \in E^n \text{ y } D^k \in E^m \forall k \in \{1, 2, \dots, f\}$$

Tomando en cuenta la expresión anterior, los patrones de entrada y salida se representan de la siguiente forma:

$$\mathbf{A}^k = (A_1^k, A_2^k, \dots, A_n^k)^t = \begin{pmatrix} A_1^k \\ A_2^k \\ \vdots \\ A_n^k \end{pmatrix}$$

$$\mathbf{D}^k = (D_1^k, D_2^k, \dots, D_m^k)^t = \begin{pmatrix} D_1^k \\ D_2^k \\ \vdots \\ D_m^k \end{pmatrix}$$

Empleando los elementos anteriores es posible plantear el problema de las memorias asociativas de la siguiente forma:

1. Fase de aprendizaje. Encontrar los operadores y una manera de generar una matriz  $M$  que almacene las  $p$  asociaciones del conjunto fundamental  $\{(\mathbf{A}^1, \mathbf{D}^1), (\mathbf{A}^2, \mathbf{D}^2), \dots, (\mathbf{A}^f, \mathbf{D}^f)\}$ , donde  $\mathbf{A}^k \in E^n$  y  $\mathbf{D}^k \in E^m \forall k \in \{1, 2, \dots, f\}$ . Si  $\exists k \in \{1, 2, \dots, f\}$  tal que  $\mathbf{A}^k \neq \mathbf{D}^k$ , se tendrá una memoria heteroasociativa; mientras que si  $m = n$  y  $\mathbf{A}^k = \mathbf{D}^k \forall k \in \{1, 2, \dots, f\}$ , la memoria será autoasociativa y las asociaciones quedarán en términos de  $\{(\mathbf{A}^1, \mathbf{A}^1), (\mathbf{A}^2, \mathbf{A}^2), \dots, (\mathbf{A}^f, \mathbf{A}^f)\}$ .
2. Fase de recuperación. Hallar los operadores adecuados y las condiciones suficientes para obtener el patrón de salida  $\mathbf{D}^k$ , cuando se opera la memoria  $M$  con el patrón fundamental de entrada  $\mathbf{A}^k$ ; lo anterior para todos los patrones del conjunto fundamental y para ambos modos: autoasociativo y heteroasociativo. Además, es importante que en esta fase se exhiba y caracterice el ruido que puede soportar la memoria en el patrón de entrada  $\tilde{\mathbf{A}}^\omega$ , para entregar una salida correcta  $\mathbf{D}^\omega$ . Para el caso de las memorias autoasociativas el objetivo es obtener el patrón  $\mathbf{A}^k$  cuando se opera la memoria  $M$  con el patrón fundamental de entrada  $\mathbf{A}^k$  o con el patrón fundamental alterado  $\tilde{\mathbf{A}}^\omega$ .

### 3.2 La función de Heaviside

Es importante destacar que la definición de función de Heaviside que se adopta en este trabajo aparece en el prestigioso *Handbook of Mathematical Functions* de Abramowitz y Stegun [184] y en el libro *The Fourier Transform and its Applications* de Ronald Bracewell [185].

**Definición 3.2.1.** Sea  $x$  un número real. La función de Heaviside de  $x$  se define mediante la siguiente expresión:

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

### 3.3 Teoría de *lattices* y estructuras algebraicas ordenadas

El material que se presentará en esta sección está basado en [27,28].

**Definición 3.3.1 (orden parcial).** Un orden parcial, o solamente un orden, de un conjunto no vacío  $E$  es una relación binaria  $\leq$  en  $E$  que es reflexiva, antisimétrica y transitiva  $\forall x, y, z \in E$ .

- 1) Reflexiva:  $\forall x \in E \ x \leq x$ ,
- 2) Antisimétrica:  $\forall x, y \in E$  Si  $x \leq y$  y  $y \leq x$  entonces  $x = y$ .
- 3) Transitiva:  $\forall x, y, z \in E$  Si  $x \leq y$  y  $y \leq z$  entonces  $x \leq z$ .

A partir de lo anterior se define el concepto de un conjunto ordenado.

**Definición 3.3.2 (conjunto ordenado).** Un conjunto ordenado  $(E; \leq)$  se define como aquel conjunto en el que está definido un orden  $\leq$ .

De igual forma el par  $(E; \leq)$  también es llamado conjunto parcialmente ordenado o *poset*. Si  $x \leq y$ , entonces  $x$  es menor o igual que  $y$  o  $y$  es mayor o igual que  $x$ ; también se puede decir que  $x$  está contenido en  $y$  o que  $y$  contiene a  $x$ . Si  $x \leq y$  pero  $x \neq y$  se dice que  $x < y$  o  $y > x$ . Si  $x \leq y$  o  $y \leq x$  se dice que  $x, y$  son comparables, de otra forma  $x$  y  $y$  son incomparables o paralelos, lo cual se denota con  $a \parallel b$ .

A partir de la definición anterior se deduce lo siguiente.

**Definición 3.3.3 (conjunto totalmente ordenado).** Un conjunto parcialmente ordenado  $(E; \leq)$  es totalmente ordenado si cada  $x, y \in E$  son comparables, esto es:

$$\forall x, y \ x \leq y \text{ o } y \leq x$$

Una parte fundamental del funcionamiento del clasificador de Heaviside está basada en los siguientes conceptos relacionados con los conjuntos ordenados: máximo, mínimo, supremo e ínfimo. A continuación, presentamos sus definiciones formales.

**Definición 3.3.4 (máximo y mínimo).** Sea  $(E; \leq)$  un conjunto parcialmente ordenado, entonces:

- i. 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:

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

- ii. 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 3.3.5 (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  $\forall y \in F y \leq x$ .

Partiendo de estas definiciones surgen los conceptos de ínfimo y supremo.

**Definición 3.3.6 (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.

A partir de las definiciones anteriores, surge el concepto de *lattice*.

**Definición 3.3.7 (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. Las *lattices* se denotan por  $(E; \wedge, \vee, \leq)$ .

### 3.4 Sistemas de numeración: representaciones 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. A continuación, presentaremos algunas definiciones relacionadas con la representación de números en una base  $b$  y en la conversión de números de base 10 a base  $b$ ; tales definiciones fueron tomadas de [186,187].

**Definición 3.4.1.** 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:

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

En donde  $b$  es un entero no negativo,  $a_0, a_1, \dots, a_k$  son enteros no negativos menores que  $b$  y  $a_k \neq 0$ . La representación de  $n$  dada en el teorema 3.4.1 se denomina expresión de  $n$  en términos de una base  $b$ . La expresión de  $n$  en base  $b$  se denota por  $(a_k a_{k-1} \dots a_1 a_0)$ . 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 el algoritmo que presentamos a continuación.

**Algoritmo 3.4.1 (conversión de base)**

1. Dividir  $n$  entre  $b$  para obtener el cociente y el residuo, esto es:

$$n = bq_0 + a_0, \quad 0 \leq a_0 < b$$

2. El resto  $a_0$ , es el dígito situado más a la derecha de la expresión en base  $b$  de  $n$ .
3. Se divide  $q_0$  entre  $b$  para obtener:

$$q_0 = bq_1 + a_1, \quad 0 \leq a_1 < b$$

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 cuando obtenemos un cociente igual a cero.

Por ejemplo, si se pide calcular la expresión en base 8 de  $(12345)_{10}$  se haría lo siguiente:

Dividir 12345 entre 8 para obtener:

$$12345 = 8 \cdot 1543 + 1$$

Dividiendo sucesivamente los cocientes por 8 se obtiene:

$$1543 = 8 \cdot 192 + 7$$

$$192 = 8 \cdot 24 + 0$$

$$24 = 8 \cdot 3 + 0$$

$$3 = 8 \cdot 0 + 3$$

Como los residuos obtenidos en los cocientes son los dígitos de  $(12345)_{10}$  en base 8, tenemos lo siguiente:

$$12345_{10} = 30071_8$$

# Capítulo 4. Modelo propuesto

En este capítulo se desarrollará el modelo matemático en el que se sustenta el clasificador propuesto en este trabajo de tesis; el clasificador es supervisado y del tipo *one-shot*, de manera que supera los inconvenientes de los modelos iterativos al no requerir convergencia.

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 corresponde con el ínfimo y ayudará en la clasificación de patrones contaminados con ruido sustractivo y el modo HS que corresponde con el supremo y será de utilidad en la clasificación de patrones alterados con ruido aditivo. Aunque, como se comentó en el capítulo anterior, estos tipos de ruido afectan principalmente a los patrones binarios.

El modelo propuesto en este trabajo basa su funcionamiento en dos operaciones originales: la operación L, la cual se empleará en la fase de aprendizaje; y la operación C, la cual se empleará en la fase de clasificación.

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. En las siguientes secciones se enunciarán las hipótesis que fundamentan el clasificador de Heaviside y posteriormente se presentarán las definiciones de las operaciones L y C; estas definiciones se complementarán con un análisis de sus principales propiedades. Cabe destacar que la teoría correspondiente con la operación L se aplica para ambos modos del clasificador, mientras que para la operación C se desarrollará la definición para cada uno de los modos del clasificador de Heaviside.

## 4.1 Hipótesis fundamentales acerca del modelo propuesto

El clasificador de Heaviside está sustentado en las cuatro hipótesis que se presentan a continuación

**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 repositorio de la UCI [188].

**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 que 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 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.

Por lo tanto, 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, representadas respecto de una base  $b$  de algún sistema numérico posicional.

Si  $B$  es el banco de vectores con componentes enteras no negativas expresadas en la base  $b$  que resulta de las 4 hipótesis, se aplica a  $B$  un método de validación tal como *10-Fold Cross-Validation* [87]. De aquí se obtienen dos conjuntos de patrones, mutuamente exclusivos: el conjunto de aprendizaje  $A$  y el conjunto de prueba  $P$ , cuya unión es  $B$ .

## 4.2 Operación L para los modos HI y HS

La operación L se define de la siguiente forma:

**Definición 4.2.1.-** 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. Entonces, para ambos modos del clasificador de Heaviside se define la operación de aprendizaje  $L$ .

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

Cabe mencionar que en la definición anterior se considera a la función de Heaviside como una forma de asegurar que los valores de las componentes de los patrones y de la base sean mayores que 0; sin embargo, a partir de la valiosa observación hecha por el Dr. Sergio Suárez Guerra, integrante del jurado de esta tesis, es posible escribir la operación L de la siguiente forma:

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

Sin que se alteren las propiedades básicas que se enlistan a continuación:

**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 la siguiente desigualdad:  $L(x, y) \leq L(y, x)$ .

**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 se cumple la siguiente desigualdad:  $L(x, z) \leq L(y, z)$

**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 se cumple la siguiente desigualdad:  $L(z, x) \geq L(z, y)$

**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 se cumple que:  $L[(x \vee y), z] = L(x, z) \vee L(y, z)$ .

**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 se cumple que:  $L[(x \wedge y), z] = L(x, z) \wedge L(y, z)$ .

#### 4.2.1 Características particulares de la operación L

En las tablas 1-5 presentamos el resultado de aplicar la operación  $L$  para las bases numéricas 2, 3, 4, 5 y 6; posteriormente presentaremos las observaciones hechas a partir de los resultados obtenidos.

Con objeto de simplificar la notación, en las tablas 1-5 representaremos la  $i$ -ésima y la  $j$ -ésima componentes del  $k$ -ésimo patrón del conjunto A de aprendizaje con las literales  $x$ ,  $y$ :

$$A_i^k = x$$

$$A_j^k = y$$

Por lo tanto, la operación L se expresa como:

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

Con  $b=2$  se tiene lo siguiente (Tabla 1):

**Tabla 1. Ejemplo de la operación L con  $b=2$ .**

x	y	$L(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

Total de valores: 4

Como puede observarse, esta tabla es exactamente igual al operador asociativo alfa propuesto en la tesis del Dr. Cornelio Yáñez [26].

Con  $b=3$  se tiene lo siguiente (Tabla 2):

**Tabla 2. Ejemplo de la operación L con  $b=3$ .**

x	y	$L(x, y)$
0	0	2
0	1	1
0	2	0
1	0	3
1	1	2
1	2	1
2	0	4
2	1	3
2	2	2

Total de valores: 9

Con  $b=4$  se tiene lo siguiente (Tabla 3):

**Tabla 3. Ejemplo de la operación L con  $b=4$ .**

x	y	$L(x, y)$
0	0	3
0	1	2
0	2	1
0	3	0
1	0	4
1	1	3
1	2	2
1	3	1
2	0	5
2	1	4
2	2	3
2	3	2
3	0	6
3	1	5
3	2	4
3	3	3

Total de valores: 16

Con  $b=5$  se tiene lo siguiente (Tabla 4):

**Tabla 4. Ejemplo de la operación L con  $b=5$ .**

x	y	$L(x, y)$
0	0	4
0	1	3

Total de valores: 25

0	2	2
0	3	1
0	4	0
1	0	5
1	1	4
1	2	3
1	3	2
1	4	1
2	0	6
2	1	5
2	2	4
2	3	3
2	4	2
3	0	7
3	1	6
3	2	5
3	3	4
3	4	3
4	0	8
4	1	7
4	2	6
4	3	5
4	4	4



Con  $b=6$  se tiene lo siguiente (Tabla 5):

**Tabla 5. Ejemplo de la operación L con  $b=6$ .**

x	y	$L(x, y)$
0	0	5
0	1	4
0	2	3
0	3	2
0	4	1
0	5	0
1	0	6
1	1	5
1	2	4
1	3	3
1	4	2
1	5	1
2	0	7
2	1	6
2	2	5
2	3	4
2	4	3
2	5	2
3	0	8
3	1	7
3	2	6
3	3	5
3	4	4

Total de valores: 36

3	5	3
4	0	9
4	1	8
4	2	7
4	3	6
4	4	5
4	5	4
5	0	10
5	1	9
5	2	8
5	3	7
5	4	6
5	5	5

A partir de los valores observados en las tablas previas, podemos enlistar las siguientes propiedades de la operación  $L$ :

1. En cada una de las tablas se observa que hay un total de  $b$  ciclos; dentro de cada ciclo se tienen otros  $b$  ciclos anidados, en donde los valores de  $x$  corresponden con el ciclo exterior y los valores de  $y$  corresponden con el ciclo interior.
2. Derivado del hecho anterior se deduce que el número total de elementos para  $L$  es de  $b^2$ .
3. El primer ciclo de cada tabla comienza con el valor de  $b-1$  (valores resaltados en gris claro) y decrece hasta terminar con 0; el segundo ciclo comienza con  $b$  y termina con 1; cada ciclo incrementa en una unidad los valores del ciclo anterior.
4. Los valores con los que inicia el último ciclo de cada tabla siguen un patrón inicialmente desconocido; sin embargo, tras aplicar un proceso inductivo observamos que el último ciclo comienza con  $b + (b - 2)$ ; analizando esta expresión se tiene que:

$$b + (b - 2) = 2b - 2 = 2(b - 1)$$

5. Al final, el último ciclo termina con  $b-1$  (valores sombreados con gris oscuro).
6. Para una base  $b$  dada, los valores de  $x$  y  $y$ , van de 0 a  $b-1$ .
7. Para una base  $b$  dada, los valores de  $L$  van de 0 hasta  $2(b-1)$ .

### 4.3 La operación C

A continuación, presentamos la definición formal de la operación C para cada uno de los modos del clasificador de Heaviside. Cabe destacar que la definición de C para el modo HI es diferente a la del modo HS; por lo tanto, diferenciaremos las operaciones de la siguiente manera:  $C^I$  corresponde a la operación C para el modo HI y  $C^S$  corresponde a la operación C para el modo HS.

**Definición 4.3.1.** Sean  $M_{ij}$  y  $P_j^k$  la  $ij$ -ésima componente de la memoria asociativa 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 la operación  $C^I$  para la clasificación de patrones:

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

**Definición 4.3.2.** Sean  $M_{ij}$  y  $P_j^k$  la  $ij$ -ésima componente de la memoria asociativa 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 HS del clasificador de Heaviside se define la operación  $C^S$  para la clasificación de patrones:

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

Para la operación  $C$ , en ambos modos del clasificador de Heaviside, tenemos las siguientes propiedades:

**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  $C(x, z) \leq C(y, z)$ .

4.3.1 Análisis general de la operación  $C$  para ambos modos del clasificador

En esta sección presentaremos algunas características generales de la operación  $C$ , las cuales son válidas para ambos modos del clasificador. Posteriormente analizaremos con detalle el planteamiento de la expresión que define a la operación  $C$  para el modo HI y para el modo HS del clasificador de Heaviside.

A fin de simplificar la notación y facilitar la explicación, proponemos lo siguiente para representar las dos versiones de la operación  $C$ :

$$M_{ij} = x \text{ y } P_j^k = y$$

$$C^I(x, y) = H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] \text{ Modo HI}$$

$$C^S(x, y) = H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] + (H[(b - 1) - (x + y)] + H[(x + 2) - (2b - y)]) \cdot (b - 1) \text{ Modo HS}$$

La definición de la operación  $C$  se obtuvo a partir de un riguroso proceso inductivo el cual se describirá a continuación. No obstante que este proceso puede ilustrarse con la tabla de la operación  $L$  para cualquier valor de  $b$ , lo haremos con  $b=3$  por razones de espacio y simplicidad. Considérese la siguiente tabla de la operación  $L$  para  $b=3$  (Tabla 2):

x	y	$L(x, y)$
0	0	2
0	1	1
0	2	0
1	0	3
1	1	2
1	2	1
2	0	4
2	1	3
2	2	2

Como definimos previamente, la operación  $C$  toma por argumento  $x$  un valor perteneciente a la memoria asociativa  $M$ , la cual fue obtenida a partir de la operación  $L$ ; mientras que por argumento  $y$  toma una componente del patrón de entrada  $P^k$  que se presenta al patrón a clasificar. Por lo tanto, es posible calcular la operación  $C$  a partir de la operación  $L$ , y para ello sólo basta con tomar como primer argumento el valor de  $L$  y el segundo argumento será el valor de  $y$ . Lo anterior se muestra en la tabla 6:

**Tabla 6. Construcción de la operación  $C$ .**

x	y	$L(x,y)$
0	0	2
0	1	1
0	2	0
1	0	3
1	1	2
1	2	1
2	0	4
2	1	3
2	2	2
$C(x,y)$	$y$	$x$
$C(M_{ij}, P_j^k)$	$P_j^k$	$M_{ij}$

Ahora la tabla donde se define  $C$  tiene estas columnas

Entonces con lo anterior es posible definir  $C$  de la siguiente manera (Tabla 7):

**Tabla 7. Construcción de la operación  $C$  a partir de la operación  $L$ .**

x	y	$C(x,y)$
2	0	0
1	1	0
0	2	0
3	0	1

2	1	1
1	2	1
4	0	2
3	1	2
2	2	2

Para fines de implementación y prueba no es posible definir  $C$  como una tabla, esto consumiría una gran cantidad de memoria de la computadora; por lo tanto, es necesario deducir una operación que establezca una relación entre  $x$ ,  $y$  y  $C(x, y)$ . Tras un extenso proceso inductivo con los elementos presentados en la tabla anterior se dedujo que la siguiente expresión:

$$(x + 1) - (b - y)$$

representa la relación existente entre los pares de valores  $x, y$  y su correspondiente  $C(x, y)$ . Sin embargo, es importante considerar que la tabla y la expresión formulada a partir de ella no contemplan todas las posibles combinaciones entre valores de  $x$  y  $y$ . Es necesario tomar en cuenta que para  $x$  los valores estarán en el rango de 0 hasta  $2(b-1)$ , los cuales corresponden con todos los valores posibles para  $L$ ; mientras que para  $y$  los valores estarán en el rango de 0 hasta  $b-1$ . Por lo tanto, el número total de elementos de la operación  $C$  será superior al de la operación  $L$ .

A continuación, presentamos el cálculo realizado para obtener el número de elementos de  $C$ .

$$\text{Rango de valores en } x: \text{ de } 0 \text{ hasta } 2(b-1) \therefore \text{Elementos}_x = 2b - 1$$

$$\text{Rango de valores en } y: \text{ de } 0 \text{ a } b-1 \therefore \text{Elementos}_y = b$$

$$\text{Elementos}_C = \text{Elementos}_x \times \text{Elementos}_y$$

$$\text{Elementos}_C = (2b - 1)(b) = \mathbf{2b^2 - b}$$

Por lo tanto, la operación  $C$  tiene  $2b^2 - b$  elementos; si consideramos las posibles combinaciones (ya ordenadas) entre elementos de  $x$  y  $y$ , la operación  $C$  se representará de la siguiente forma:

**Tabla 8. Operación  $C$  considerando todos los posibles valores de  $x, y$ .**

$x$	$y$	$C(x, y)$
0	0	
0	1	
0	2	0
1	0	
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1
3	1	2
3	2	
4	0	2
4	1	
4	2	

Ahora surge una cuestión interesante: hay pares de elementos para los cuales la operación  $C$  no está definida (los pares sombreados en gris); esto en primera instancia puede verse como un impedimento para la operación propuesta. Sin embargo, consideramos que estos pares serán catalogados como los casos NI (No Importa) ya que no deben influir negativamente en el proceso de la clasificación.

Para que lo anterior se cumpla debemos elegir valores de  $C$  de tal forma que no provoquen errores de clasificación. Recordemos que en el modo HI la etapa de aprendizaje se realizará con el operador ínfimo y la de clasificación con el operador supremo; por lo tanto, para los pares NI en la operación  $C$ , se debe devolver un valor mínimo que evite alteraciones en el patrón recuperado. Tomando en cuenta lo presentado hasta el momento, proponemos que  $C$  devuelva el valor de 0; esto se debe a que 0 es el mínimo valor numérico que se maneja en nuestro modelo de clasificación, y por ello nunca será mayor que el supremo. El análisis correspondiente para el

modo HS se presentará más adelante. La operación C, ya completa para el modo HI, se presenta en la tabla 9.

**Tabla 9. Operación C para el modo HI del clasificador de Heaviside.**

x	y	$C^I(x, y)$
0	0	<b>0</b>
0	1	<b>0</b>
0	2	0
1	0	<b>0</b>
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1
3	1	2
3	2	<b>0</b>
4	0	2
4	1	<b>0</b>
4	2	<b>0</b>

El número de pares o casos NI para la operación C, a partir de un valor de base b, se obtiene de la siguiente forma:

$$Elementos_C = 2b^2 - b \text{ Elementos totales de C}$$

$$Elementos_L = b^2 \text{ Elementos totales de L}$$

$$Total_{NI} = Elementos_C - Elementos_L \text{ Total de casos NI}$$

$$Total_{NI} = 2b^2 - b - b^2 = b^2 - b = b(b - 1)$$



Por lo tanto, se tendrán  $b^2 - b$  casos NI, esta cantidad de casos aplica para ambos modos del clasificador.

#### 4.3.2 ¿Cómo se usa la función de Heaviside en la operación $C^l$ del modo HI?

Por simplicidad y sólo con fines de ilustración, para el desarrollo de esta sección consideraremos el caso particular de  $b=3$ . Los casos NI ocurren bajo dos situaciones:

Cuando  $x + y < b - 1$  o cuando  $x + y \geq 2b - 1$

Dado que en el modo HI los casos NI tendrán un valor de 0, la operación  $C^l$  debe plantearse de tal forma que para los casos NI devuelva 0 y que en caso contrario se devuelva el valor obtenido al evaluar la expresión:

$$(x + 1) - (b - y)$$

Para lograr esto es necesario transformar las dos condiciones bajo las cuales ocurren los casos NI, de manera tal que al evaluarlas se obtenga un valor menor o igual que 0 si es que está ocurriendo un caso NI, de lo contrario devolverán un valor positivo. Para simplificar la evaluación de estas dos expresiones se colocarán como argumentos de sendas funciones de Heaviside. A continuación, presentamos el proceso de transformación realizado.

Condición 1:  $x + y < b - 1$ . Esta primera condición se debe transformar de tal forma que la expresión sea menor o igual que cero. Por otro lado:

$x + y < b - 1$  puede reescribirse como  $x + y \leq b - 2$

Dado que  $b - 2 < b - 1$ , al acomodar términos obtenemos lo siguiente:

$$x + y - b + 2 \leq 0$$

Lo anterior lo podemos re-escribir como:

$$(x + 2) - (b - y) \leq 0$$

Si al evaluar la expresión anterior, se tienen valores negativos o cero, entonces se tiene un caso NI; al colocar la expresión anterior como argumento de una función de Heaviside se tiene lo siguiente:

$$H[(x + 2) - (b - y)] = H[(M_{ij} + 2) - (b - P_j^k)]$$
 Si se ajusta a la notación original

Ahora bien, la condición 2 queda escrita de la siguiente manera:

$$x + y \geq 2b - 1$$

Como se indicó anteriormente, es necesario que la expresión quede escrita de forma tal que sea menor o igual que 0, para lograr esto se realiza lo siguiente:

$$0 \geq -x - y + (2b - 1)$$

Reescribiendo términos se tiene que:  $(2b - 1) - (x + y) \leq 0$

Si al evaluar la expresión anterior, se tienen valores negativos o cero, entonces se tiene un caso NI; se utilizará la función de Heaviside para que si se cae en un caso NI la expresión de  $C^l$  se haga 0:

$$H[(2b - 1) - (x + y)] = H[(2b - 1) - (M_{ij} + P_j^k)] \text{ Tras adecuar a la notación original}$$

Al probar estas expresiones con los valores NI para  $b=3$  se obtiene lo siguiente (Tablas 10 y 11):

**Tabla 10. Evaluación de casos NI con  $b=3$ , bajo la condición 1.**

x	y	$(x + 2) - (b - y)$	$H[(x + 2) - (b - y)]$
0	0	$2-3=-1$	0
0	1	$2-2=0$	0
1	0	$3-3=0$	0

**Tabla 11. Evaluación de casos NI con  $b=3$ , bajo la condición 2.**

x	y	$(2b - 1) - (x + y)$	$H[(2b - 1) - (x + y)]$
3	2	$5-5=0$	0
4	1	$5-5=0$	0
4	2	$5-6=-1$	0

Al probar estas expresiones con valores que no caen en la condición de NI se tiene lo siguiente (Tabla 12):

**Tabla 12. Evaluación de casos que no son NI para  $b=3$ .**

x	y	$(2b - 1) - (x + y)$	$H[(x + 2) - (b - y)]$	$(x + 2) - (b - y)$	$H[(2b - 1) - (x + y)]$
0	2	$5-2=3$	1	$2-1=1$	1
1	1	$5-2=3$	1	$3-2=1$	1
1	2	$5-3=2$	1	$3-1=2$	1

Con estas dos expresiones se asegura que si ocurre un caso NI la operación regresará 0; debido a que ambas expresiones deben devolver un 1. Para que no ocurra un caso NI es necesario combinar las expresiones, de manera que si una de las dos no se cumple la operación  $C^l$  devuelva 0. Tras analizar algunas opciones decidimos proponer la siguiente expresión compuesta:

$$H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)]$$

Con esta expresión se asegura que si  $x + y < b - 1$  ó  $x + y \geq 2b - 1$  la operación  $C^l$  regresará 0; es decir, el operador multiplicación se está usando como un operador lógico AND.

El resto de la operación  $C$  es el siguiente:

$$(x + 1) - (b - y) = (M_{ij} + 1) - (b - P_j^k)$$

De acuerdo con la notación propuesta inicialmente

Al agregar esta expresión compuesta a la operación  $C^l$  evitamos posibles errores de clasificación inducidos por casos NI. Al final la operación  $C^l$  se expresa de la siguiente forma:

$$C^l(x, y) = H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] \text{ o bien:}$$

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

#### 4.3.3 ¿Cómo se usa la función de Heaviside en la operación $C^S$ del modo HS?

Lo propuesto anteriormente resuelve el problema de los casos NI en el modo HI del clasificador; sin embargo, el modo HS tiene diferencias significativas, por lo que se debe analizar lo que ocurre para el modo HS del clasificador de Heaviside.

En primer lugar, los NI en el modo HS no pueden recibir un valor de 0, ya que contrario a lo que ocurre en el modo HI la etapa de aprendizaje se realiza con el operador supremo y la etapa de clasificación emplea el operador ínfimo; por lo tanto, la operación  $C^S$  ahora deberá regresar el máximo valor que se esté manejando en los patrones. De acuerdo con lo establecido para la operación  $L$ , el valor máximo que tendrán las componentes de los patrones es  $b-1$ ; por ello, consideramos que ese valor no interferirá con otros valores que se estén considerando en la etapa de clasificación.

Ahora bien, ese valor  $b-1$  debe agregarse en la expresión de  $C^l$  que se planteó para el modo HI:

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

Pero no se puede agregar de manera directa, mediante una simple operación aritmética; tienen que agregarse uno o más términos que hagan que  $C^S$  devuelva  $b-1$  únicamente cuando se caiga en casos NI. Considerando lo anterior, la operación  $C^S$  se plantea de la siguiente forma:

$$C^S(x, y) = H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] + \mathbb{X} \cdot (b - 1)$$

Ese término  $\mathbb{X}$  deberá valer 1 cuando se tenga un caso NI mientras que el resto de la operación valdrá 0 (de acuerdo con lo analizado en la sección anterior); de esta forma  $C^S$  devolverá  $b-1$ . Ahora, para hallar  $\mathbb{X}$  se debe analizar lo que ya se conoce en relación con el modo HS:

Los casos NI ocurren si:

$$x + y < b - 1 \text{ o } x + y \geq 2b - 1$$

Dado que en el modo HS los casos NI tendrán un valor de  $b-1$ , la operación  $C^S$  debe plantearse de tal forma que para los casos NI devuelva  $b-1$  y que en caso contrario se devuelva el valor obtenido al evaluar la expresión:

$$(x + 1) - (b - y)$$

Para lograr esto es necesario transformar las dos condiciones bajo las cuales ocurren los casos NI, de manera tal que al evaluarlas se obtenga un valor mayor que 0 si es que está ocurriendo un caso NI, y de lo contrario devuelvan negativo o 0. Para simplificar la evaluación de estas dos expresiones, decidimos colocarlas como argumentos de sendas funciones de Heaviside, de tal manera que si hay un caso NI las funciones de Heaviside devolverán uno y en caso contrario devolverán 0. A continuación, se presenta el proceso de transformación realizado.

Para la condición 1:  $x + y < b - 1$  se realiza lo siguiente:

$$0 < -x - y + (b - 1) \text{ Esto puede reescribirse como: } (b - 1) - (x + y) > 0$$

Si al evaluar la expresión anterior se obtienen valores positivos, entonces se tiene un caso NI; colocando esta expresión como argumento de una función de Heaviside se tiene lo siguiente (considerando la notación original):

$$H[(b - 1) - (x + y)] = H[(b - 1) + (M_{ij} + P_j^k)]$$

En cuanto a la condición 2:  $x + y \geq 2b - 1$ , debemos considerar que  $2b - 2 < 2b - 1 < 2b < 2b + 1 < 2b + 2$ , por lo tanto, la condición 2 puede reescribirse de la siguiente forma:

$$x + y > 2b - 2$$

Esto puede, a la vez, escribirse como:

$$x + y - 2b + 2 > 0$$

$$(x + 2) - (2b - y) > 0$$

Si al evaluar la expresión anterior, se obtienen valores positivos, entonces se tiene un caso NI; colocando esta expresión como argumento de una función de Heaviside quedaría de la siguiente forma:

$$H[(x + 2) - (2b - y)] = H[(M_{ij} + 2) - (2b - P_j^k)]$$

Al probar estas expresiones con los valores NI para  $b=3$  se obtiene lo siguiente (Tablas 13 y 14):

**Tabla 13. Evaluación de pares NI para  $b=3$ , bajo la condición 1.**

$x$	$y$	$(b - 1) - (x + y)$	$H[(b - 1) - (x + y)]$
0	0	$2-0=2$	1
0	1	$2-1=1$	1
1	0	$2-1=1$	1

**Tabla 14. Evaluación de pares NI para  $b=3$  bajo la condición 2.**

$x$	$y$	$(x + 2) - (2b - y)$	$H[(x + 2) - (2b - y)]$
3	2	$5-4=1$	1
4	1	$6-5=1$	1
4	2	$6-4=2$	1

En la Tabla 15 se presentan los resultados al probar estas expresiones con valores que no caen en la condición de NI.

**Tabla 15. Evaluación de pares que no son NI para  $b=3$ .**

$x$	$y$	$(b - 1) - (x + y)$	$H[(b - 1) - (x + y)]$	$(x + 2) - (2b - y)$	$H[(x + 2) - (2b - y)]$
0	2	$2-2=0$	0	$2-4=-2$	0
1	1	$2-2=0$	0	$3-5=-2$	0
1	2	$2-3=-1$	0	$3-4=-1$	0

Es importante considerar que no pueden ocurrir las dos condiciones de forma simultánea:

$$x + y < b - 1$$

$$x + y \geq 2b - 1$$

Para un par de valores  $x, y$  dados, sólo se cumplirá una u otra, pero no ambas. Ese “o” puede asociarse con un operador lógico OR EXCLUSIVO (XOR), de manera análoga a lo hecho con el AND para el modo HI. Esto a su vez se puede asociar con la operación aritmética suma. Por lo tanto, es posible formular la siguiente expresión:

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

De esta forma solo es necesario que alguno de los dos términos de la expresión se cumpla para que se tenga un 1 multiplicando a  $b-1$ .

En resumen, es posible definir el término  $\mathbb{X}$  de la siguiente manera:

$$\mathbb{X} = H[(b - 1) - (x + y)] + H[(x + 2) - (2b - y)]$$

Ahora bien, a partir de la tabla en la que se describe la operación  $C^S$  para  $b=3$  es posible deducir que la operación central, en la que se realiza la recuperación de los patrones, es la siguiente:

$$[(x + 1) - (b - y)]$$

Sin embargo, esta operación por sí sola no daría buenos resultados debido a los casos NI, es por eso que se han tenido que agregar los términos necesarios para tratar con los casos especiales.

Finalmente se reescribe la expresión de la operación  $C^S$  en el modo HS de la siguiente forma:

$$\begin{aligned} C^S(x, y) &= H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] + \mathbb{X} \cdot (b - 1) \\ &= \\ &= H[(x + 2) - (b - y)] \cdot H[(2b - 1) - (x + y)] \cdot [(x + 1) - (b - y)] \\ &\quad + (H[(b - 1) - (x + y)] + H[(x + 2) - (2b - y)]) \cdot (b - 1) \end{aligned}$$

El uso de la función de Heaviside permite que la implementación de estas operaciones sea de manera directa o bien empleando estructuras condicionales tales como la if-then-else.

A continuación, se presenta el desarrollo de la operación  $C$  para  $b=2$ ,  $b=4$  y  $b=5$  en los modos HI y HS. En las tablas 16-21 se resaltan en gris los casos NI.

Modo HI

Para  $b=2$

**Tabla 16. Operación  $C^I$  para el modo HI con  $b=2$ .**

$x$	$y$	$C^I(x, y)$
0	0	<b>0</b>
0	1	0
1	0	0
1	1	1
2	0	1
2	1	<b>0</b>

Como se puede observar, en la tabla 16 la operación  $C^I$  en modo HI es muy similar al operador beta propuesto en la tesis del Dr. Cornelio Yáñez [26]. La principal diferencia radica en que para el caso de  $x=2$  y  $y=1$  el valor es 0 en lugar de 1. En el caso de esta tesis se puso el valor de 0 ya que es un caso NI.

Para  $b=4$

**Tabla 17. Operación  $C^I$  para el modo HI con  $b=4$ .**

$x$	$y$	$C^I(x, y)$
0	0	<b>0</b>
0	1	<b>0</b>
0	2	<b>0</b>
0	3	0
1	0	<b>0</b>
1	1	<b>0</b>
1	2	0
1	3	1
2	0	<b>0</b>

2	1	0
2	2	1
2	3	2
3	0	0
3	1	1
3	2	2
3	3	3
4	0	1
4	1	2
4	2	3
4	3	<b>0</b>
5	0	2
5	1	3
5	2	<b>0</b>
5	3	<b>0</b>
6	0	3
6	1	<b>0</b>
6	2	<b>0</b>
6	3	<b>0</b>

Para  $b=5$

**Tabla 18. Operación  $C^I$  para el modo HI con  $b=5$ .**

$x$	$y$	$C^I(x, y)$
0	0	<b>0</b>
0	1	<b>0</b>
0	2	<b>0</b>



0	3	<b>0</b>
0	4	0
1	0	<b>0</b>
1	1	<b>0</b>
1	2	<b>0</b>
1	3	0
1	4	1
2	0	<b>0</b>
2	1	<b>0</b>
2	2	0
2	3	1
2	4	2
3	0	<b>0</b>
3	1	0
3	2	1
3	3	2
3	4	3
4	0	0
4	1	1
4	2	2
4	3	3
4	4	4
5	0	1
5	1	2
5	2	3

5	3	4
5	4	<b>0</b>
6	0	2
6	1	3
6	2	4
6	3	<b>0</b>
6	4	<b>0</b>
7	0	3
7	1	4
7	2	<b>0</b>
7	3	<b>0</b>
7	4	<b>0</b>
8	0	4
8	1	<b>0</b>
8	2	<b>0</b>
8	3	<b>0</b>
8	4	<b>0</b>

Modo HS

Para  $b=2$

**Tabla 19. Operación  $C^S$  para el modo HS con  $b=2$ .**

$x$	$y$	$C^S(x, y)$
0	0	<b>1</b>
0	1	0
1	0	0

1	1	1
2	0	1
2	1	<b>1</b>

Para el modo HS, también podemos observar que existen similitudes con el operador beta propuesto en la tesis del Dr. Yáñez. Sin embargo, la principal diferencia radica en que para  $x=0$  y  $y=0$ , el cual es un caso NI, se tiene procede a asignar el valor de 1.

Para  $b=4$

**Tabla 20. Operación  $C^S$  para el modo HS con  $b=4$ .**

$x$	$y$	$C^S(x, y)$
0	0	<b>3</b>
0	1	<b>3</b>
0	2	<b>3</b>
0	3	0
1	0	<b>3</b>
1	1	<b>3</b>
1	2	0
1	3	1
2	0	<b>3</b>
2	1	0
2	2	1
2	3	2
3	0	0
3	1	1
3	2	2
3	3	3

4	0	1
4	1	2
4	2	3
4	3	<b>3</b>
5	0	2
5	1	3
5	2	<b>3</b>
5	3	<b>3</b>
6	0	3
6	1	<b>3</b>
6	2	<b>3</b>
6	3	<b>3</b>

Para  $b=5$

**Tabla 21. Operación  $C^S$  para el modo HS con  $b=5$ .**

$x$	$y$	$C^S(x, y)$
0	0	<b>4</b>
0	1	<b>4</b>
0	2	<b>4</b>
0	3	<b>4</b>
0	4	0
1	0	<b>4</b>
1	1	<b>4</b>
1	2	<b>4</b>
1	3	0
1	4	1

2	0	<b>4</b>
2	1	<b>4</b>
2	2	0
2	3	1
2	4	2
3	0	<b>4</b>
3	1	0
3	2	1
3	3	2
3	4	3
4	0	0
4	1	1
4	2	2
4	3	3
4	4	4
5	0	1
5	1	2
5	2	3
5	3	4
5	4	<b>4</b>
6	0	2
6	1	3
6	2	4
6	3	<b>4</b>
6	4	<b>4</b>

7	0	3
7	1	4
7	2	4
7	3	4
7	4	4
8	0	4
8	1	4
8	2	4
8	3	4
8	4	4

#### 4.4 Fases del clasificador de Heaviside en el modo HI

A continuación, se presentan definiciones, lemas y teoremas que fundamentan teóricamente las etapas de aprendizaje y clasificación del clasificador de Heaviside en el modo HI.

**Definición 4.4.1.-** Sean  $A^k$  y  $A^d$  dos patrones que pertenecen al conjunto de aprendizaje  $A$  en un clasificador de Heaviside en ambos modos operativos. Se define el producto externo bajo  $L$  de ambos patrones, como la matriz cuya  $ij$ -ésima componente o elemento se calcula así:

$$M_k = A^k \boxtimes_L A^d$$

$$(M_k)_{ij} = L(A_i^k, A_j^d)$$

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

**Nota 1.** Si la dimensión de los patrones de  $A$  es  $n$ , entonces el producto externo bajo  $L$  es una matriz cuadrada de dimensiones  $n \times n$ .

**Nota 2.** El valor  $k=d$  es válido.

**Definición 4.4.2. Fase de aprendizaje del clasificador de Heaviside.** La  $ij$ -ésima componente o elemento de la memoria asociativa  $M$  del clasificador de Heaviside, 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.** El valor mínimo se toma con respecto al índice  $k$ , que va de 1 a  $|A|$ .

**Definición 4.4.3. Fase de clasificación del clasificador de Heaviside.** Sea  $M$  una memoria asociativa del 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  $R$ :

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

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

**Nota 2.** Por Definiciones 4.4.2 y 4.4.3, se tiene:  $R_i = \max[C^I(\min L(A_i^k, A_j^k), P_j^d)]$ .

A partir de las definiciones anteriores se establecen los siguientes resultados:

**Teorema 4.4.1.** Sea  $b \in \mathbb{Z}$  tal que  $b > 1$ , entonces  $\forall x \in \mathbb{Z}$  tal que  $0 \leq x \leq b - 1$  se cumple  $L(x, x) = b - 1$ .

**Corolario 4.4.1.1:**

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

**Corolario 4.4.1.2:**

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

**Nota.** Obsérvese que los corolarios 4.4.1.1 y 4.4.1.2 son las propiedades  $L1$  y  $L2$ ; es decir, el teorema 1 es una propiedad general de la operación  $L$ .

**Lema 4.4.1.** La memoria asociativa  $M$ , del clasificador de Heaviside, contiene sólo valores  $b-1$  en su diagonal principal, en ambos modos.

**Lema 4.4.2:**

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:  $C^I[L(x, y), y] = x$ .

**Lema 4.4.3.** Sea  $M$  la memoria asociativa del clasificador de Heaviside en modo operativo HI, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

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

**Lema 4.4.4.** Sea  $M$  la memoria asociativa del clasificador de Heaviside en modo operativo HI, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

$$R_i \geq A_i^d$$

**Teorema 4.4.2.** Un clasificador de Heaviside, basado en la asociativa  $M$ , en modo operativo HI, clasifica correctamente cualquier patrón  $A^d$  del conjunto de aprendizaje  $A$ .

En el teorema 4.4.2 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 es capaz de clasificar correctamente todos los patrones del conjunto de aprendizaje. Aunque esto pueda entenderse como algo no muy importante, en realidad no lo es ya que pocos modelos en el estado del arte del reconocimiento de patrones son capaces de clasificar correctamente el conjunto fundamental. Como se verá más adelante, esto también se aplica para el modo HS del clasificador de Heaviside.

#### 4.5 Fases del Clasificador de Heaviside en el modo HS

A continuación, se presentan definiciones, lemas y teoremas que fundamentan teóricamente las etapas de aprendizaje y clasificación del clasificador de Heaviside en el modo HS; se omiten aquellas que tienen que ver con la operación  $L$  ya que esta operación es la misma para ambos modos del clasificador.

**Definición 4.5.1.-** Sean  $A^k$  y  $A^d$  dos patrones que pertenecen al conjunto de aprendizaje  $A$  en un clasificador de Heaviside en ambos modos operativos. Se define el producto externo bajo  $L$  de ambos patrones, como la matriz cuya  $ij$ -ésima componente se calcula así:

$$M_k = A^k \boxtimes_L A^d$$

$$(M_k)_{ij} = L(A_i^k, A_j^d)$$

donde  $A_i^k$  y  $A_j^d$  son la  $i$ -ésima y la  $j$ -ésima componentes del  $k$ -ésimo y  $d$ -ésimo patrón del conjunto  $A$  de aprendizaje, respectivamente. Esta definición ya se presentó en la sección anterior. Sin embargo, consideramos importante incluir esta definición para definir de mejor forma las fases de aprendizaje y clasificación.

**Nota 1.** Si la dimensión de los patrones de  $A$  es  $n$ , entonces el producto externo bajo  $L$  es una matriz cuadrada de dimensiones  $n \times n$ .

**Nota 2.** El valor  $k=d$  es válido.



**Definición 4.5.2. Fase de aprendizaje del clasificador de Heaviside.** La  $ij$ -ésima componente de la memoria asociativa  $M$  del clasificador de Heaviside, en modo operativo HS dado un conjunto de aprendizaje  $A$ , se define por la siguiente expresión:

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

**Nota:** El valor máximo se toma con respecto al índice  $k$ , que va de 1 a  $|A|$ .

**Definición 4.5.3. Fase de clasificación del clasificador de Heaviside.** Sea  $M$  la memoria asociativa de un clasificador de Heaviside en modo operativo HS, 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  $R$ :

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

**Nota 1.** El valor mínimo se toma con respecto al índice  $j$ , que va de 1 a  $n$ .

**Nota 2.** Por Definiciones 4.5.1 y 4.5.2, se tiene:  $R_i = \min[C^S(\max L(A_i^k, A_j^k), P_j^d)]$ .

A partir de las definiciones anteriores se establecen los siguientes resultados:

**Teorema 4.5.1.** Sea  $b \in \mathbb{Z}$  tal que  $b > 1$ , entonces  $\forall x \in \mathbb{Z}$  tal que  $0 \leq x \leq b - 1$  se cumple  $L(x, x) = b - 1$ .

**Corolario 4.5.1.1:**

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

**Corolario 4.5.1.2:**

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

**Nota:** Obsérvese que los corolarios 4.5.1.1 y 4.5.1.2 son las propiedades  $L1$  y  $L2$ ; es decir, el teorema 1 es una propiedad general de la operación  $L$ .

**Lema 4.5.1.** La memoria asociativa  $M$ , del clasificador de Heaviside, contiene sólo valores  $b-1$  en su diagonal principal, en ambos modos.

**Lema 4.5.2:**

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:  $C^S[L(x, y), y] = x$ .

**Lema 4.5.3.** Sea  $M$  la memoria asociativa del clasificador de Heaviside en modo operativo HS, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

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

**Lema 4.5.4.** Sea  $M$  la memoria asociativa del clasificador de Heaviside en modo operativo HS, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

$$R_i \leq A_i^d$$

**Teorema 4.5.2.** Un clasificador de Heaviside, basado en la memoria  $M$ , en modo operativo HS, clasifica correctamente cualquier patrón  $A^d$  del conjunto de aprendizaje  $A$ .

En el teorema 4.5.2 el índice  $d$  se fijó arbitrariamente, por lo que la conclusión es válida para cualquier patrón del conjunto de aprendizaje  $A$ .

Como se mencionó anteriormente ambos modos del clasificador de Heaviside son capaces de recuperar íntegramente el conjunto fundamental de patrones. Es importante mencionar que las demostraciones de todos los lemas y teoremas presentados en esta sección están disponibles en los apéndices A1-A4 de esta tesis.

## 4.6 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 4.6.1 clasificador de Heaviside en modo operativo HI

### Etapas de preprocesamiento

1. Para cada patrón del banco de datos.
  1. Buscar si hay componentes negativas
  2. Si hay componentes negativas, buscar aquella que tenga el menor valor y restarla de las demás componentes del patrón.
  3. Si no hay componentes negativas el patrón no se modifica.
2. Buscar si existen componentes con punto decimal en el banco de datos.
3. 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  $d=0$ .
4. Por cada patrón del banco de datos
  1. Multiplicar todas las componentes por  $10^d$ .

### Etapa de transformación

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

### Etapa de entrenamiento

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

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

2. 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

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

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

2. Se obtendrá el patrón  $\mathbf{R}$ , de dimensión  $n$ , cuya  $i$ -ésima componente se expresa como:

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

3. Buscar en  $A$  el patrón recuperado  $\mathbf{R}$  o en su defecto el patrón más parecido<sup>2</sup> a  $\mathbf{R}$ .
4. Asignar a  $P^d$  la clase del patrón encontrado en  $A$  para  $\mathbf{R}$ .

---

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

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 4.6.2 clasificador de Heaviside en modo operativo HS

Etapa de preprocesamiento

1. Para cada patrón del banco de datos.
  1. Buscar si hay componentes negativas
  2. Si hay componentes negativas, buscar aquella que tenga el menor valor y restarla de las demás componentes del patrón.
  3. Si no hay componentes negativas el patrón no se modifica.
2. Buscar si existen componentes con punto decimal en el banco de datos.
3. 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  $d=0$ .
4. Por cada patrón del banco de datos
  1. Multiplicar todas las componentes por  $10^d$ .

Etapa de transformación

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

Etapa de entrenamiento

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

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

2. Aplicar el ínfimo a las matrices obtenidas en el paso 1 para obtener la memoria asociativa  $M$ .

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

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

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

## Etapa de clasificación

1. Por cada patrón  $P^d \in P$  hacer lo siguiente:
  1. 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

$$\mathbf{R} = M \mathbin{\text{\textcircled{C}}}_{CS} P^d$$

2. Se obtendrá el patrón  $\mathbf{R}$ , de dimensión  $n$ , cuya  $i$ -ésima componente se expresa como:

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

3. Buscar en  $A$  el patrón recuperado  $\mathbf{R}$  o en su defecto el patrón más parecido<sup>3</sup> a  $\mathbf{R}$ .
4. Asignar a  $P^d$  la clase del patrón encontrado en  $A$  para  $\mathbf{R}$ .

## 4.7 Análisis de la complejidad del Clasificador de Heaviside

En esta sección presentamos los análisis de complejidad que se han realizado para el algoritmo del clasificador de Heaviside. En primer lugar, se muestra la complejidad espacial y, posteriormente, se presenta el análisis de la complejidad temporal.

### 4.7.1 Complejidad espacial

El algoritmo del clasificador de Heaviside parte de las cuatro hipótesis mencionadas al inicio de este capítulo; para que éstas se cumplan, es necesario someter los bancos de datos a un proceso de transformación. En primer lugar, el banco de datos compuesto por  $p$  vectores de dimensión  $n$  se transforman las componentes de los patrones a componentes no negativas. En esta primera etapa, la cantidad de espacio requerido es la siguiente:

$$E_{DS} = 8pn \text{ bytes}$$

Asumiendo que los patrones tienen componentes decimales (las cuales requieren 8 bytes de almacenamiento en lenguajes como C o Java). En la siguiente etapa del proceso, las componentes de los patrones son transformadas a componentes enteras, por lo que la cantidad de espacio requerido cambiará de la siguiente forma:

$$E_{DS} = 4pn \text{ bytes}$$

---

<sup>3</sup> En este trabajo de tesis se utilizó la idea de asignar la clase del vecino más cercano.

Donde el valor de 4 se utiliza a partir de que los números enteros usualmente requieren 4 bytes en lenguajes como C o Java. Al transformar las componentes de los patrones del banco de datos a componentes enteras no negativas en términos de una base  $b$ , el banco de datos cambiará su tamaño a  $p$  vectores de dimensión  $nl$ , en donde  $l$  es el máximo número de dígitos en base  $b$  para representar la máxima componente del banco de datos. Por lo tanto, tras concluir el pre-procesamiento, necesitaremos almacenar:

$$E_{DS} = 4pnl \text{ bytes}$$

Dado que estamos utilizando componentes enteras.

Para la etapa de aprendizaje y clasificación, el clasificador de Heaviside trabajará con la memoria asociativa  $M$ , la cual requiere el siguiente espacio:

$$E_m = (4nl)^2 \text{ bytes}$$

Ya que se trata de una matriz cuadrada cuya dimensión es igual a la dimensión de los patrones. Por lo tanto, la cantidad de espacio requerida por el clasificador de Heaviside es la siguiente:

$$E = 4pnl + (4nl)^2 \text{ bytes.}$$

Por lo tanto, el algoritmo del clasificador de Heaviside tiene una complejidad espacial cuadrática respecto a la dimensión de los patrones que forman el banco de datos.

#### 4.7.2 Complejidad temporal

Para realizar este análisis, es necesario definir las siguientes variables:

$OE$ : operaciones elementales.

$p$ : número de patrones.

$n$ : dimensión de los patrones.

A continuación, presentamos el análisis del algoritmo del modo HI del clasificador. No haremos el análisis del modo HS ya que las principales diferencias radican en el orden en que se usan los operadores ínfimo y supremo y algunas operaciones elementales como el caso de la operación C.

##### Etapa de preprocesamiento

1. Para cada patrón del banco de datos. (1P)
  1. Buscar si hay componentes negativas (2P)
  2. Si hay componentes negativas, buscar aquella que tenga el menor valor y restarla de las demás componentes del patrón. (3P)

3. Si no hay componentes negativas el patrón no se modifica. (4P)
2. Buscar si existen componentes con punto decimal en el banco de datos. (5P)
3. 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  $d=0$ . (6P)
4. Por cada patrón del banco de datos (7P)
  1. Multiplicar todas las componentes por  $10^d$ .

(1P):  $p$  OE comparaciones.

(2P):  $pn$  OE comparaciones.

(3P): la probabilidad de casos es de 0.5, por lo tanto  $0.5pn$ .

(4P): Al no modificarse, se realizan 0OE.

(5P):  $pn$  OE comparaciones:

(6P):  $pn$  OE comparaciones + 1 asignación.

(7P):  $pn$  multiplicaciones:

Complejidad de la etapa de preprocesamiento:

$$\begin{aligned}
 p + pn + 0.5pn + pn + pn + pn + 1 &= p(1 + n + 0.5n + n + n + n) + 1 = p(1 + 4.5n) + 1 \\
 &= p + 4.5pn + 1 = 4.5pn + p + 1
 \end{aligned}$$

Por lo tanto, en esta etapa del algoritmo del clasificador, la complejidad es  $O(n)$ .

#### Etapa de transformación de los patrones

1. Elegir como base un número entero positivo mayor que 1. (1T)
2. Por cada patrón del banco de datos (2T).
  1. Transformar cada componente en términos de una base  $b$  elegida. (3T)
  2. Crear el patrón transformado a partir de la concatenación de las componentes transformadas en términos de la base  $b$ . (4T)

(1T): 1 OE asignación.

(2T):  $p$  OE comparaciones.

(3T):  $pnl$  OE divisiones, donde  $l$  es el número de divisiones requeridas para transformar la componente en términos de la base  $b$ .

(4T):  $pn$  OE asignaciones.

La complejidad es la siguiente:

$$1 + p + pnl + pn = 1 + p + (pl + p)n$$

Por lo tanto, la complejidad es  $O(n)$ .

### Etapa de entrenamiento

1. Por cada patrón de entrenamiento, de dimensión  $n$ ,  $\mathbf{A}^k \in A$   $k = 1, 2, \dots, |A|$  construir la matriz: (1E)

$$M_k = [A^k \boxtimes_L (A^k)^t]_{n \times n} \quad (2E)$$

2. Aplicar el ínfimo a las matrices obtenidas en el paso 1 para obtener la memoria asociativa  $M$ . (3E)

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

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

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

(1E):  $N_E$  OE comparaciones.

(2E):  $N_E n^2$  OE sumas, restas y comparaciones.

(3E):  $N_E n^2$  OE comparaciones

Por lo tanto, la complejidad de la fase de entrenamiento es la siguiente:

$$N_E + N_E n^2 + N_E n^2 = 2N_E n^2 + N_E = N_E(2n^2 + 1)$$

A partir de esto, es posible inferir que la complejidad de la fase de entrenamiento está en  $O(n^2)$ .

### Etapa de clasificación

1. Por cada patrón  $\mathbf{P}^d \in P$  hacer lo siguiente: (1C)
  1. 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 (2C)

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

2. Se obtendrá el patrón  $\mathbf{R}$ , de dimensión  $n$ , cuya  $i$ -ésima componente se expresa como:

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



3. Buscar en A el patrón recuperado  $\mathbf{R}$  o en su defecto el patrón más parecido a  $\mathbf{R}$  (4C).
4. Asignar a  $\mathbf{P}^d$  la clase del patrón encontrado en A para  $\mathbf{R}$  (5C).

(1C):  $N_p$  OE comparaciones:

(2C):  $N_p n^2$  OE sumas, restas y comparaciones.

(3C):  $N_p n^2$  OE comparaciones.

(4C):  $N_E n$  OE distancias y comparaciones.

(5C): 1 OE asignación.

La complejidad es la siguiente:

$$N_p + N_p n^2 + N_p n^2 + N_E n + 1 = N_p + 2N_p n^2 + N_E n + 1 = 2N_p n^2 + N_E n + N_p + 1$$

A partir de lo anterior, es posible inferir que la complejidad de la fase de prueba es de orden  $O(n^2)$ . A partir de lo analizado en las cuatro etapas del clasificador de Heaviside, podemos observar que su complejidad de tipo  $O(n^2)$ . Y también es posible afirmar que este cálculo es válido para el modo HS del clasificador de Heaviside; las únicas diferencias serán en las OE empleadas en las versiones diferentes de la operación C y en las comparaciones hechas al intercambiar el Ínfimo por el Supremo en el aprendizaje y el Supremo por el Ínfimo en la clasificación.

#### 4.7.3 Potenciales limitaciones del clasificador de Heaviside

A partir del análisis de complejidad realizado en la sección anterior, es posible inferir que el tiempo de ejecución del clasificador de Heaviside incrementará de manera significativa al momento de trabajar con bancos de datos de alta dimensionalidad. Por tal razón el uso de valores pequeños de base no es recomendado para manejar patrones cuyas componentes enteras son muy grandes.

Este hecho quedó comprobado de forma empírica al momento de realizar las dos rondas de pruebas descritas en el capítulo 5. En la mayoría de los experimentos, el uso de valores bajos de  $b$ , particularmente 2, se reflejaba en un tiempo considerable de procesamiento que estaba en el orden de los minutos hasta las horas; además de que el desempeño, por lo regular, no era el mejor. De hecho, en la mayoría de los bancos de datos que fueron utilizados en la fase de pruebas, los mejores resultados se obtuvieron con un valor de base igual al valor de la componente más grande del conjunto de entrenamiento más uno.

## 4.8 Ejemplo de operación del clasificador de Heaviside en el modo HI

A partir del siguiente banco de datos:

$$\mathbf{x}^1 = \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} \quad \mathbf{x}^2 = \begin{pmatrix} 6.0 \\ -2.2 \end{pmatrix} \quad \mathbf{x}^3 = \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix}$$

$$\mathbf{x}^4 = \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} \quad \mathbf{x}^5 = \begin{pmatrix} 6.0 \\ -2.4 \end{pmatrix} \quad \mathbf{x}^6 = \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix}$$

Distribuidos en las siguientes clases

$$C1 = \{\mathbf{x}^1, \mathbf{x}^4\} \quad C2 = \{\mathbf{x}^2, \mathbf{x}^5\} \quad C3 = \{\mathbf{x}^3, \mathbf{x}^6\}$$

Después de aplicar algún método de validación, el banco de datos queda dividido en los siguientes conjuntos:

Aprendizaje:

$$\mathbf{A}^1 = \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} \quad \mathbf{A}^2 = \begin{pmatrix} 6.0 \\ -2.2 \end{pmatrix} \quad \mathbf{A}^3 = \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix}$$

Prueba:

$$\mathbf{P}^1 = \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} \quad \mathbf{P}^2 = \begin{pmatrix} 6.0 \\ -2.4 \end{pmatrix} \quad \mathbf{P}^3 = \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix}$$

Lo primero que podemos hacer es transformar las componentes negativas existentes en los patrones.

$$\mathbf{A}^1 = \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} \text{ Sin componentes negativas.}$$

$$\mathbf{A}^2 = \begin{pmatrix} 6.0 \\ -2.2 \end{pmatrix} \text{ ¡Hay una componente negativa! Se restará la componente de menor valor.}$$

$$\mathbf{A}^2 = \begin{pmatrix} 6.0 \\ -2.2 \end{pmatrix} = \begin{pmatrix} 6.0 - (-2.2) \\ -2.2 - (-2.2) \end{pmatrix} = \begin{pmatrix} 8.2 \\ 0 \end{pmatrix}$$

$$\mathbf{A}^3 = \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix} \text{ Sin componentes negativas.}$$

$$\mathbf{P}^1 = \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} \text{ Sin componentes negativas.}$$

$$\mathbf{P}^2 = \begin{pmatrix} 6.0 \\ -2.4 \end{pmatrix} \text{ ¡Hay una componente negativa! Se restará la componente de menor valor.}$$

$$\mathbf{P}^2 = \begin{pmatrix} 6.0 - (-2.4) \\ -2.4 - (-2.4) \end{pmatrix} = \begin{pmatrix} 8.4 \\ 0 \end{pmatrix}$$

$$\mathbf{P}^3 = \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix} \text{ Sin componentes negativas.}$$

El máximo número de dígitos es 1; por lo tanto,  $d=1$  y los patrones se multiplicarán por  $10^d = 10^1 = 10$ .

$$\mathbf{A}^1 = \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} * 10 = \begin{pmatrix} 50 \\ 35 \end{pmatrix}$$

$$\mathbf{A}^2 = \begin{pmatrix} 8.2 \\ 0 \end{pmatrix} * 10 = \begin{pmatrix} 82 \\ 0 \end{pmatrix}$$

$$\mathbf{A}^3 = \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix} * 10 = \begin{pmatrix} 77 \\ 28 \end{pmatrix}$$

$$\mathbf{P}^1 = \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} * 10 = \begin{pmatrix} 48 \\ 35 \end{pmatrix}$$

$$\mathbf{P}^2 = \begin{pmatrix} 8.4 \\ 0 \end{pmatrix} * 10 = \begin{pmatrix} 84 \\ 0 \end{pmatrix}$$

$$\mathbf{P}^3 = \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix} * 10 = \begin{pmatrix} 74 \\ 28 \end{pmatrix}$$

En este caso utilizaremos un valor de  $b=8$ , por lo que todas las componentes de los patrones se transformarán a esta base.

$$\mathbf{A}^1 = \begin{pmatrix} 50 \\ 35 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix} \quad \mathbf{A}^2 = \begin{pmatrix} 82 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{A}^3 = \begin{pmatrix} 77 \\ 28 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

$$\mathbf{P}^1 = \begin{pmatrix} 48 \\ 35 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 0 \\ 0 \\ 4 \\ 3 \end{pmatrix} \quad \mathbf{P}^2 = \begin{pmatrix} 80 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{P}^3 = \begin{pmatrix} 74 \\ 28 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

A continuación, se presenta el proceso de entrenamiento en el modo HI. La primera etapa es el producto externo bajo  $L$  de  $\mathbf{A}^1$  con  $\mathbf{A}^1$ :

$$A^1 \boxtimes_L (A^1)^t = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix} \boxtimes_L (0 \ 6 \ 2 \ 0 \ 4 \ 3)$$

$$= \begin{pmatrix} L(0,0) & L(0,6) & L(0,2) & L(0,0) & L(0,4) & L(0,3) \\ L(6,0) & L(6,6) & L(6,2) & L(6,0) & L(6,4) & L(6,3) \\ L(2,0) & L(2,6) & L(2,2) & L(2,0) & L(2,4) & L(2,3) \\ L(0,0) & L(0,6) & L(0,2) & L(0,0) & L(0,4) & L(0,3) \\ L(4,0) & L(4,6) & L(4,2) & L(4,0) & L(4,4) & L(4,3) \\ L(3,0) & L(3,6) & L(3,2) & L(3,0) & L(3,4) & L(3,3) \end{pmatrix}$$

$$\text{Donde } M_1 = A^1 \boxtimes_L (A^1)^t = \begin{pmatrix} 7 & 1 & 5 & 7 & 3 & 4 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 9 & 3 & 7 & 9 & 5 & 6 \\ 7 & 1 & 5 & 7 & 3 & 4 \\ 11 & 5 & 9 & 11 & 7 & 8 \\ 10 & 4 & 8 & 10 & 6 & 7 \end{pmatrix}.$$

Para el patrón  $A^2$  tenemos que:

$$M_2 = A^2 \boxtimes_L (A^2)^t = \begin{pmatrix} 7 & 6 & 6 & 8 & 8 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \end{pmatrix}$$

Mientras que para  $A^3$  se tiene lo siguiente:

$$M_3 = A^3 \boxtimes_L (A^3)^t = \begin{pmatrix} 7 & 7 & 3 & 8 & 5 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 11 & 11 & 7 & 12 & 9 & 8 \\ 6 & 6 & 2 & 7 & 4 & 3 \\ 9 & 9 & 5 & 10 & 7 & 6 \\ 10 & 10 & 6 & 11 & 8 & 7 \end{pmatrix}$$

La fase de entrenamiento concluye calculando el ínfimo entre las componentes de las tres matrices anteriores; por lo tanto,  $M = M_1 \wedge M_2 \wedge M_3$ .

$$M = \begin{pmatrix} 7 & 1 & 5 & 7 & 3 & 4 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 9 & 3 & 7 & 9 & 5 & 6 \\ 7 & 1 & 5 & 7 & 3 & 4 \\ 11 & 5 & 9 & 11 & 7 & 8 \\ 10 & 4 & 8 & 10 & 6 & 7 \end{pmatrix} \wedge \begin{pmatrix} 7 & 6 & 6 & 8 & 8 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \end{pmatrix} \wedge \begin{pmatrix} 7 & 7 & 3 & 8 & 5 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 11 & 11 & 7 & 12 & 9 & 8 \\ 6 & 6 & 2 & 7 & 4 & 3 \\ 9 & 9 & 5 & 10 & 7 & 6 \\ 10 & 10 & 6 & 11 & 8 & 7 \end{pmatrix}$$

La memoria asociativa M, derivada del proceso de aprendizaje, queda de la siguiente manera:

$$M = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix}$$

A continuación, presentamos la clasificación del conjunto de aprendizaje:

$$M \Psi_C A^1 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C^I} \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix}$$

Al aplicar el producto definido bajo la operación C tenemos lo siguiente:

$$M \Psi_C A^1 = \begin{pmatrix} \vee(C^I(7,0) & C^I(1,6) & C^I(3,2) & C^I(7,0) & C^I(3,4) & C^I(4,3)) \\ \vee(C^I(7,0) & C^I(7,6) & C^I(3,2) & C^I(8,0) & C^I(5,4) & C^I(4,3)) \\ \vee(C^I(8,0) & C^I(3,6) & C^I(7,2) & C^I(9,0) & C^I(5,4) & C^I(6,3)) \\ \vee(C^I(6,0) & C^I(1,6) & C^I(2,2) & C^I(7,0) & C^I(3,4) & C^I(3,3)) \\ \vee(C^I(6,0) & C^I(5,6) & C^I(5,2) & C^I(7,0) & C^I(7,4) & C^I(6,3)) \\ \vee(C^I(6,0) & C^I(4,6) & C^I(5,2) & C^I(7,0) & C^I(6,4) & C^I(7,3)) \end{pmatrix}$$

$$M \Psi_C A^1 = \begin{pmatrix} \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 6 & 0 & 1 & 2 & 0) \\ \vee(1 & 2 & 2 & 2 & 2 & 2) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 4 & 0 & 0 & 4 & 2) \\ \vee(0 & 3 & 0 & 0 & 3 & 3) \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix} = A^1 \therefore A^1 \in C1$$

Por lo tanto, el primer patrón ha sido clasificado correctamente. Para el patrón  $A^2$  se tiene lo siguiente:

$$M \Psi_C A^2 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C^I} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$M \Psi_C A^2 = \begin{pmatrix} \vee(1 & 0 & 0 & 0 & 0 & 0) \\ \vee(1 & 2 & 0 & 1 & 0 & 0) \\ \vee(2 & 0 & 2 & 2 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = A^2 \therefore A^2 \in C2$$

Por lo tanto, el segundo patrón ha sido clasificado correctamente. Para el patrón  $A^3$  se tiene lo siguiente:

$$M \Psi_C A^3 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C'} \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

$$M \Psi_C A^3 = \begin{pmatrix} \vee(1 & 0 & 1 & 0 & 0 & 1) \\ \vee(1 & 1 & 1 & 1 & 1 & 1) \\ \vee(2 & 0 & 5 & 2 & 1 & 3) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 3 & 0 & 3 & 3) \\ \vee(0 & 0 & 3 & 0 & 2 & 4) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix} = A^3 \therefore A^3 \in C3$$

Por lo tanto, el tercer y último patrón del conjunto de aprendizaje ha sido clasificado correctamente. Con lo anterior, podemos observar que el clasificador de Heaviside en modo HI ha clasificado todos los patrones del conjunto de aprendizaje.

En cuanto a los patrones del conjunto de prueba, se realiza lo siguiente para el patrón  $P^1$ :

$$M \Psi_C P^1 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C'} \begin{pmatrix} 0 \\ 6 \\ 0 \\ 0 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 6 & 0 & 1 & 2 & 0) \\ \vee(1 & 2 & 0 & 2 & 2 & 2) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 3 & 0 & 0 & 4 & 2) \\ \vee(0 & 3 & 0 & 0 & 3 & 3) \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix}$$

$$= A^1 \therefore P^1 \in C1$$

Para el patrón  $P^2$  se realiza lo siguiente:

$$M \Psi_C P^2 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C'} \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \vee(1 & 0 & 0 & 0 & 0 & 0) \\ \vee(1 & 2 & 0 & 1 & 0 & 0) \\ \vee(2 & 0 & 0 & 2 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = A^2$$

$$\therefore P^2 \in C2$$

Finalmente, para el patrón  $P^3$  tenemos lo siguiente:

$$M \Psi_C P^3 = \begin{pmatrix} 7 & 1 & 3 & 7 & 3 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 8 & 3 & 7 & 9 & 5 & 6 \\ 6 & 1 & 2 & 7 & 3 & 3 \\ 6 & 5 & 5 & 7 & 7 & 6 \\ 6 & 4 & 5 & 7 & 6 & 7 \end{pmatrix} \Psi_{C'} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \vee(1 & 0 & 0 & 0 & 0 & 1) \\ \vee(1 & 1 & 0 & 1 & 1 & 1) \\ \vee(2 & 0 & 2 & 2 & 1 & 3) \\ \vee(0 & 0 & 0 & 0 & 0 & 0) \\ \vee(0 & 0 & 0 & 0 & 3 & 3) \\ \vee(0 & 0 & 0 & 0 & 2 & 4) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 3 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

El patrón resultante no corresponde con el patrón  $P^3$ ; sin embargo, si buscamos el patrón más similar a  $P^3$  obtendremos que ese patrón es  $A^3$  el cual pertenece a la clase C3, por lo que  $P^3$  se clasificará como C3.

#### 4.9 Ejemplo de operación del clasificador de Heaviside en el modo HS

En este ejemplo utilizaremos el banco de datos presentado en el ejemplo de la sección anterior

$$\begin{aligned} \mathbf{x}^1 &= \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} & \mathbf{x}^2 &= \begin{pmatrix} 6.0 \\ -2.2 \end{pmatrix} & \mathbf{x}^3 &= \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix} \\ \mathbf{x}^4 &= \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} & \mathbf{x}^5 &= \begin{pmatrix} 6.0 \\ -2.4 \end{pmatrix} & \mathbf{x}^6 &= \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix} \end{aligned}$$

Cuyos patrones están divididos en las siguientes clases:

$$C1 = \{\mathbf{x}^1, \mathbf{x}^4\} \quad C2 = \{\mathbf{x}^2, \mathbf{x}^5\} \quad C3 = \{\mathbf{x}^3, \mathbf{x}^6\}$$

Tras aplicar el método de validación, eliminar los valores negativos y convertirlo las componentes a valores enteros no negativos tenemos los siguientes patrones:

$$\mathbf{A}^1 = \begin{pmatrix} 5.0 \\ 3.5 \end{pmatrix} * 10 = \begin{pmatrix} 50 \\ 35 \end{pmatrix}$$

$$\mathbf{A}^2 = \begin{pmatrix} 8.2 \\ 0 \end{pmatrix} * 10 = \begin{pmatrix} 82 \\ 0 \end{pmatrix}$$

$$\mathbf{A}^3 = \begin{pmatrix} 7.7 \\ 2.8 \end{pmatrix} * 10 = \begin{pmatrix} 77 \\ 28 \end{pmatrix}$$

$$\mathbf{P}^1 = \begin{pmatrix} 4.8 \\ 3.5 \end{pmatrix} * 10 = \begin{pmatrix} 48 \\ 35 \end{pmatrix}$$

$$\mathbf{P}^2 = \begin{pmatrix} 8.4 \\ 0 \end{pmatrix} * 10 = \begin{pmatrix} 84 \\ 0 \end{pmatrix}$$

$$\mathbf{P}^3 = \begin{pmatrix} 7.4 \\ 2.8 \end{pmatrix} * 10 = \begin{pmatrix} 74 \\ 28 \end{pmatrix}$$

Tras transformar el banco de datos a base 8, tenemos lo siguiente:

$$\mathbf{A}^1 = \begin{pmatrix} 50 \\ 35 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix} \quad \mathbf{A}^2 = \begin{pmatrix} 82 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{A}^3 = \begin{pmatrix} 77 \\ 28 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

$$\mathbf{p}^1 = \begin{pmatrix} 48 \\ 35 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 0 \\ 4 \\ 3 \end{pmatrix} \quad \mathbf{p}^2 = \begin{pmatrix} 80 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{p}^3 = \begin{pmatrix} 74 \\ 28 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

A continuación, se presenta el entrenamiento del clasificador para el modo HS.

$$M_1 = \mathbf{A}^1 \boxtimes_L (\mathbf{A}^1)^t = \begin{pmatrix} 7 & 1 & 5 & 7 & 3 & 4 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 9 & 3 & 7 & 9 & 5 & 6 \\ 7 & 1 & 5 & 7 & 3 & 4 \\ 11 & 5 & 9 & 11 & 7 & 8 \\ 10 & 4 & 8 & 10 & 6 & 7 \end{pmatrix}.$$

$$M_2 = \mathbf{A}^2 \boxtimes_L (\mathbf{A}^2)^t = \begin{pmatrix} 7 & 6 & 6 & 8 & 8 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \end{pmatrix}$$

$$M_3 = \mathbf{A}^3 \boxtimes_L (\mathbf{A}^3)^t = \begin{pmatrix} 7 & 7 & 3 & 8 & 5 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 11 & 11 & 7 & 12 & 9 & 8 \\ 6 & 6 & 2 & 7 & 4 & 3 \\ 9 & 9 & 5 & 10 & 7 & 6 \\ 10 & 10 & 6 & 11 & 8 & 7 \end{pmatrix}$$

La fase de entrenamiento concluye calculando el ínfimo entre las componentes de las tres matrices anteriores; por lo tanto,  $M = M_1 \vee M_2 \vee M_3$ .

$$M = \begin{pmatrix} 7 & 1 & 5 & 7 & 3 & 4 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 9 & 3 & 7 & 9 & 5 & 6 \\ 7 & 1 & 5 & 7 & 3 & 4 \\ 11 & 5 & 9 & 11 & 7 & 8 \\ 10 & 4 & 8 & 10 & 6 & 7 \end{pmatrix} \wedge \begin{pmatrix} 7 & 6 & 6 & 8 & 8 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 8 & 7 & 7 & 9 & 9 & 9 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \\ 6 & 5 & 5 & 7 & 7 & 7 \end{pmatrix} \wedge \begin{pmatrix} 7 & 7 & 3 & 8 & 5 & 4 \\ 7 & 7 & 3 & 8 & 5 & 4 \\ 11 & 11 & 7 & 12 & 9 & 8 \\ 6 & 6 & 2 & 7 & 4 & 3 \\ 9 & 9 & 5 & 10 & 7 & 6 \\ 10 & 10 & 6 & 11 & 8 & 7 \end{pmatrix}$$

La memoria asociativa M, derivada del proceso de aprendizaje, queda de la siguiente manera:

$$M = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix}$$

Posteriormente, se realiza la clasificación del conjunto de aprendizaje:



$$M \Psi_C A^1 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix}$$

Al aplicar el producto definido bajo la operación C tenemos lo siguiente:

$$M \Psi_C A^1 = \begin{pmatrix} \wedge(C^S(7,0)) & C^S(7,6) & C^S(6,2) & C^S(8,0) & C^S(8,4) & C^S(9,3) \\ \wedge(C^S(13,0)) & C^S(7,6) & C^S(11,2) & C^S(13,0) & C^S(9,4) & C^S(10,3) \\ \wedge(C^S(11,0)) & C^S(11,6) & C^S(7,2) & C^S(12,0) & C^S(9,4) & C^S(9,3) \\ \wedge(C^S(7,0)) & C^S(6,6) & C^S(5,2) & C^S(7,0) & C^S(7,4) & C^S(7,3) \\ \wedge(C^S(11,0)) & C^S(9,6) & C^S(9,2) & C^S(11,0) & C^S(7,4) & C^S(8,3) \\ \wedge(C^S(10,0)) & C^S(10,6) & C^S(8,2) & C^S(11,0) & C^S(8,4) & C^S(7,3) \end{pmatrix}$$

$$M \Psi_C A^1 = \begin{pmatrix} \wedge(0 & 6 & 1 & 1 & 5 & 5) \\ \wedge(6 & 6 & 6 & 6 & 6 & 6) \\ \wedge(4 & 7 & 2 & 5 & 6 & 5) \\ \wedge(0 & 5 & 0 & 0 & 4 & 3) \\ \wedge(4 & 7 & 4 & 4 & 4 & 4) \\ \wedge(3 & 7 & 3 & 4 & 5 & 3) \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 2 \\ 0 \\ 4 \\ 3 \end{pmatrix} = A^1 \therefore A^1 \in C1$$

Por lo tanto, el primer patrón ha sido clasificado correctamente. Para el patrón  $A^2$  se tiene lo siguiente:

$$M \Psi_C A^2 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$M \Psi_C A^2 = \begin{pmatrix} \vee(1 & 2 & 1 & 1 & 1 & 2) \\ \vee(7 & 2 & 6 & 6 & 2 & 3) \\ \vee(6 & 6 & 2 & 5 & 2 & 2) \\ \vee(1 & 1 & 0 & 0 & 0 & 0) \\ \vee(5 & 4 & 4 & 4 & 0 & 1) \\ \vee(4 & 6 & 3 & 4 & 1 & 0) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} = A^2 \therefore A^2 \in C2$$

Por lo tanto, el segundo patrón ha sido clasificado correctamente. Para el patrón  $A^3$  se tiene lo siguiente:

$$M \Psi_C A^3 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix}$$

$$M \Psi_C A^3 = \begin{pmatrix} \vee(1 & 1 & 4 & 1 & 4 & 6) \\ \vee(7 & 1 & 7 & 6 & 5 & 7) \\ \vee(6 & 5 & 5 & 5 & 5 & 6) \\ \vee(1 & 0 & 3 & 0 & 3 & 4) \\ \vee(5 & 3 & 7 & 4 & 3 & 5) \\ \vee(4 & 4 & 6 & 4 & 4 & 4) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 5 \\ 0 \\ 3 \\ 4 \end{pmatrix} = A^3 \therefore A^3 \in C3$$

Por lo tanto, el tercer y último patrón del conjunto de aprendizaje ha sido clasificado correctamente. Con lo anterior, podemos observar que el clasificador de Heaviside en modo HS ha clasificado todos los patrones del conjunto de aprendizaje.

En cuanto a los patrones del conjunto de prueba, se realiza lo siguiente para el patrón  $P^1$ :

$$M \Psi_C P^1 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 0 \\ 6 \\ 0 \\ 0 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} \vee(0 & 6 & 1 & 1 & 5 & 5) \\ \vee(6 & 6 & 4 & 6 & 6 & 6) \\ \vee(4 & 7 & 0 & 5 & 6 & 5) \\ \vee(0 & 5 & 7 & 0 & 4 & 3) \\ \vee(4 & 7 & 2 & 4 & 4 & 4) \\ \vee(3 & 7 & 1 & 4 & 5 & 3) \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 0 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

Este patrón no es igual al patrón  $P^1$ , pero tras aplicar el método del vecino más cercano tenemos que  $P^1$  se clasifica como C1.

Para el patrón  $P^2$  se realiza lo siguiente:

$$M \Psi_C P^2 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \vee(1 & 2 & 7 & 1 & 1 & 2) \\ \vee(7 & 2 & 4 & 6 & 2 & 3) \\ \vee(5 & 6 & 0 & 5 & 2 & 2) \\ \vee(1 & 1 & 7 & 0 & 0 & 0) \\ \vee(5 & 6 & 2 & 4 & 0 & 1) \\ \vee(4 & 6 & 1 & 4 & 1 & 0) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ = A^2 \therefore P^2 \in C2$$

Finalmente, para el patrón  $P^3$  tenemos lo siguiente:

$$M \Psi_C P^3 = \begin{pmatrix} 7 & 7 & 6 & 8 & 8 & 9 \\ 13 & 7 & 11 & 13 & 9 & 10 \\ 11 & 11 & 7 & 12 & 9 & 9 \\ 7 & 6 & 5 & 7 & 7 & 7 \\ 11 & 9 & 9 & 11 & 7 & 8 \\ 10 & 10 & 8 & 11 & 8 & 7 \end{pmatrix} \Psi_{C^S} \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \vee(1 & 1 & 2 & 1 & 4 & 6) \\ \vee(7 & 1 & 9 & 6 & 5 & 7) \\ \vee(5 & 5 & 2 & 5 & 5 & 4) \\ \vee(1 & 0 & 0 & 0 & 3 & 4) \\ \vee(5 & 3 & 4 & 4 & 3 & 5) \\ \vee(4 & 4 & 3 & 4 & 4 & 4) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 3 \end{pmatrix}$$

El patrón resultante no corresponde con el patrón  $P^3$ ; sin embargo, si buscamos el patrón más similar a  $P^3$  obtendremos que ese patrón es  $A^3$  el cual pertenece a la clase C3, por lo que  $P^3$  se clasificará como C3.

# Capítulo 5. Resultados experimentales y discusión

En esta sección presentaremos los resultados obtenidos de diversas pruebas que se realizaron con el Clasificador de Heavisde. Cabe destacar que se realizaron dos rondas de pruebas con una cantidad considerable de bancos de datos de repositorios internacionales, y para fines de comparación se consideraron modelos que destacan en el campo de reconocimiento de patrones. Para brindar sustento a nuestros resultados hemos decidido someterlos a por lo menos una prueba de significancia estadística; en dicha prueba se rechazará o no la hipótesis nula que establece la igualdad de los resultados obtenidos por los diversos clasificadores.

Además de realizar pruebas con bancos de datos pertenecientes a repositorios como UCI [188] o KEEL [189], se propusieron e implementaron dos aplicaciones para nuestro modelo: una para la detección rápida de la Degeneración Macular y otra para la clasificación inteligente de recursos educativos. Finalmente, esta sección concluye con una relación de los trabajos científicos publicados o aceptados hasta el momento, en revistas indizadas.

## 5.1 Modelos de clasificación seleccionados

Se hizo un estudio comparativo entre el modelo de clasificación propuesto y otros nueve clasificadores reconocidos en el estado del arte del Reconocimiento de Patrones. Los modelos contra los que se comparó el Clasificador de Heavisde, en la primera ronda de pruebas, fueron los siguientes:

1. Memorias Morfológicas Min [25]: este modelo de memorias asociativas se basa en las dos operaciones básicas de la morfología matemática: erosión y dilatación.
2. Clasificador IB1 [18,71]: este clasificador asigna a los patrones de prueba la clase del vecino más cercano a partir de una función de distancia.
3. IBK con  $K=3$  [16,71]: 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 [16]: este modelo de clasificación está basado en la teoría de la probabilidad, pues se asigna la clase que sea considerada como la más probable a partir del patrón de entrada.
5. Perceptrón Multicapa [7,190]: uno de los modelos de redes neuronales *feedforward* más eficaces en donde se utiliza el algoritmo de entrenamiento *backpropagation*.
6. Máquinas de Soporte Vectorial [24]: técnica muy popular desarrollada por Vapnik hace dos décadas que busca establecer el margen óptimo entre dos clases.

7. SMO (*Sequential Minimal Optimization*) [191][115]: modelo empleado para resolver el problema de la programación cuadrática durante el entrenamiento de máquinas de soporte vectorial (SVMs).
8. Redes Neuronales de Base Radial [192]: tipo especial de redes neuronales que utilizan funciones de base radial como funciones de activación.
9. Función Logística [193]: modelo de regresión logística multinomial.

A partir de primeras experiencias al momento de trabajar en reconocimiento de patrones, se seleccionaron aquellos modelos que se consideraron representativos en el estado de arte. Es por eso que aparecen modelos destacados de las superestructuras predominantes en esta área: clasificadores basados en métricas, clasificadores basados en el enfoque probabilístico, clasificadores basados en el enfoque neuronal, máquinas de soporte vectorial y memorias asociativas.

Por otro lado, para la segunda ronda de pruebas consideramos adecuado incluir los dos modos del clasificador de Heaviside y compararlo con algunos de los modelos empleados en la primera ronda: IB1, Naïve Bayes, Perceptrón Multicapa y SVM; además agregamos los siguientes clasificadores:

1. AdaBoostM1 [194]: Un modelo clásico de metaclasificador basado en el método de Boosting; para esta prueba se eligió al modelo REPTree como clasificador base.
2. Bagging [90]: otro método clásico de metaclasificación que, en nuestro caso, utilizará como clasificador base el modelo REPTree.
3. RandomForest [21]: un método de clasificación basado en la combinación de varios árboles predictores que son generados a partir de vectores aleatorios que fueron muestreados de manera independiente.

La selección de clasificadores empleados en la segunda ronda de pruebas se hizo considerando aquellos clasificadores destacados en el estado del arte, pero también consideramos clasificadores que, a partir de nuestra experiencia, han llamado la atención debido a sus fundamentos teóricos y en buena medida a los resultados obtenidos con éstos. Particularmente nos referimos a los meta-clasificadores [4] y a un modelo basado en los árboles de decisión.

Para otorgar mayor certeza a los resultados de la comparación entre el clasificador de Heaviside y los clasificadores enlistados anteriormente y para determinar si realmente existen diferencias significativas, en términos estadísticos, entre nuestro modelo y los otros clasificadores elegimos un conjunto de técnicas estadísticas no paramétricas: prueba de los signos, prueba de Wilcoxon y prueba de Friedman.

## 5.2 Pruebas no paramétricas

A continuación, presentamos algunas de las características principales de estas pruebas de significancia estadística.

### 5.2.1 Prueba de los signos

Una forma muy popular para comparar el desempeño total de dos algoritmos es contar el número de casos en el que un algoritmo es el ganador absoluto. Esta cuenta también puede ser utilizada en estadística inferencial en la forma de una prueba de significancia conocida como la prueba de los signos [195,196]; el punto de esta prueba radica en que los dos algoritmos que están siendo comparados son equivalentes, como lo establece la hipótesis nula, si cada uno de ellos gana en  $n/2$  de  $n$  problemas (o bancos de datos en el contexto de la clasificación de patrones). El número de triunfos (de un algoritmo con respecto a otro) se distribuye de acuerdo con una distribución binomial; para un número grande de bancos de datos el número de triunfos está bajo la hipótesis nula distribuido de acuerdo con  $n\left(\frac{n}{2}, \sqrt{n}\right)$  lo que permite el uso de la prueba Z: si el número de triunfos es de al menos  $\frac{n}{2} + \frac{1.96\sqrt{n}}{2}$  entonces el algoritmo es significativamente mejor con  $p < 0.05$ .

### 5.2.2 Prueba de Wilcoxon

El Test de Wilcoxon [197] 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 [195]. 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, y 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:

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (5.1)$$

$$R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i) \quad (5.2)$$

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 [198], 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  $\alpha = 0.1$ , típicamente  $\alpha = 0.05$ ; esto 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 conmensurabilidad 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 esta prueba no se ve afectada por la presencia de outliers, o patrones atípicos [195].

### 5.2.3 Prueba de Friedman

Las pruebas anteriores se utilizan para hacer comparaciones entre los resultados de aplicar dos algoritmos en el mismo conjunto de bancos de datos, por lo que los algoritmos están relacionados o pareados. Si nosotros deseamos hacer una comparación que involucre a muchos algoritmos a la vez y, con base en la significancia obtenida, determinar cuál es el mejor, entonces podemos recurrir a la prueba de Friedman [199,200]. Esta prueba se utiliza para determinar si en un conjunto de  $k$  muestras, donde  $k \geq 2$ , hay al menos dos muestras (clasificadores) que representen poblaciones con diferentes valores de media. Por lo tanto, la prueba de Friedman es una prueba de comparaciones múltiples que busca encontrar diferencias significativas en el comportamiento de dos o más algoritmos. La hipótesis nula de esta prueba establece la igualdad de las medianas entre todas las poblaciones (en nuestro caso entre los resultados de los clasificadores con los distintos bancos de datos empleados); la hipótesis alternativa básicamente consiste en la negación de la hipótesis nula.

El primer paso de esta prueba es convertir los resultados en rangos; éstos son calculados de acuerdo con el siguiente procedimiento:

1. Reunir los resultados para cada par algoritmo-problema (banco de datos).

2. Para cada problema  $i$ , asignar un rango de 1 (mejor resultado) hasta  $k$  (peor resultado). Los rangos serán denotados como  $r_i^j$  ( $1 \leq j \leq k$ ).
3. Para cada algoritmo  $j$  obtener el promedio de todos los rangos obtenidos en todos los problemas para obtener el rango final  $R_j = \frac{1}{n} \sum_i r_i^j$ .

De esta forma cada uno de los algoritmos se obtiene un rango separadamente. El mejor algoritmo tendría un rango total de 1, el segundo de 2, y así sucesivamente. En el caso de empates se recomienda calcular rangos promediados. Bajo la hipótesis nula que establece el comportamiento similar de todos los algoritmos (por lo que los rangos  $R_j$  deberían ser iguales) la estadística Friedman  $F_f$  se calcula como:

$$F_f = \frac{12n}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (5.3)$$

La cual está sujeta a una distribución  $\chi^2$  con  $k-1$  grados de libertad, esto cuando  $k$  y  $n$  son lo suficientemente grandes (por regla  $n > 10$  y  $k > 5$ ).

### 5.3 Bancos de datos seleccionados

Para cumplir con la cantidad mínima de comparaciones u observaciones requeridas por las pruebas no paramétricas de significancia estadística, es decir al menos 20 bancos de datos para comparar 10 algoritmos; elegimos, en la primera ronda de pruebas, aquellos bancos de datos que, de acuerdo con la experiencia del autor fueron casos interesantes de analizar al momento de realizar los primeros experimentos con la propuesta de trabajos de tesis y al momento de cursar y apoyar en el trabajo de los cursos de Reconocimiento de Patrones y Memorias Asociativas. Cabe destacar que de entre los bancos de datos seleccionados, algunos fueron tomados del repositorio UCI [188], y otros más fueron extraídos del repositorio KEEL [189]. La lista se presenta a continuación:

1. Appendicitis: datos de pacientes para estimar si padecen apendicitis o no.
2. Australian: banco de datos relacionado con solicitudes para obtener tarjetas de crédito.
3. Banana: banco de datos formado a partir de objetos agrupados en *clusters* con forma de banana.
4. Bupa: banco de datos para el análisis 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: simulación del 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 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, en el cual se propone una distribución multivariada de patrones.
17. Segment-Test: Banco de datos utilizado en la determinación de regiones en imágenes digitales.
18. Vehículos: 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.

Con respecto a la segunda ronda de pruebas del clasificador, nuevamente partimos de la premisa de que al menos se requieren el doble de bancos de datos para comparar  $n$  clasificadores. Por lo tanto, con 20 bancos de datos podemos comparar hasta 10 clasificadores siguientes. Estos bancos de datos fueron seleccionados a partir de las experiencias del autor de esta tesis al momento de colaborar en la escritura de artículos y en la preparación de materiales de apoyo para cursos relacionados con Reconocimiento de Patrones y Memorias Asociativas. Es importante mencionar que se incluyeron bancos de datos empleados en la ronda anterior, estos bancos de datos se seleccionaron a partir de las complicaciones que inicialmente se tuvieron para analizarlos. Es necesario mencionar que al igual que en el caso anterior, estos bancos de datos pertenecen a los repositorios UCI [188] y KEEL [189]. La lista de Bancos de datos se presenta a continuación.

1. Balance: banco de datos creado para modelar resultados de experimentos relacionados con la psicología y las ciencias del conocimiento.
2. Banknote authentication: banco de datos creado a partir de imágenes obtenidas para la autenticación de billetes.
3. Breast Tissue: en este banco de datos se registran mediciones de impedancia que fueron hechas al tejido mamario de un conjunto de pacientes.
4. Cardiocotography: este banco de datos contiene mediciones del ritmo cardiaco fetal y rasgos de contracción uterina que han sido analizados y clasificados por expertos.



5. 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.
6. Columna vertebral: ya mencionado en la lista anterior.
7. 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.
8. Default of credit card clients: banco de datos creado para predecir la capacidad de un grupo de clientes Taiwaneses para hacer el pago de sus tarjetas de crédito.
9. 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.
10. Ionosphere: Banco de datos creado a partir de muestras, obtenidas mediante radar, de la ionósfera.
11. Led7Digit: Este banco de datos representa todos los dígitos que se pueden representar con un display de 7 segmentos.
12. Movement\_LIBRAS: Banco de datos compuesto por 24 clases; cada clase representa una letra en la LIBRAS (lengua brasileña de señas).
13. QSAR biodegradation: banco de datos creado para clasificar poco más de 1000 sustancias químicas en lista para biodegradación o no lista para biodegradación.
14. Ring: ya descrito en la sección anterior.
15. Seeds: banco de datos creado a partir de mediciones hechas en tres diferentes especies de trigo.
16. Spambase: Banco de datos creado a partir de la identificación de mensajes de correo no deseado (spam).
17. Texture: Banco de datos creado a partir de 11 diferentes tipos de textura presentes en imágenes digitales.
18. Twonorm: Banco de datos sintético cuyas clases se generan a partir de una distribución multinomial multivariable.
19. 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.
20. Yeast: banco de datos creado para predecir los sitios de localización de proteínas en células.

Cabe destacar que estos bancos de datos fueron elegidos porque sus componentes son numéricas y en muchos casos tienen componentes con valores racionales negativos, lo cual permitió experimentar con la etapa de pre-procesamiento que se describe en el algoritmo del clasificador de Heaviside. Por otro lado, debemos mencionar que los bancos de datos Cleveland y Spambase presentan valores perdidos y se realizó un tratamiento básico basado en la imputación de valor medio por clase. En cuanto al desbalance de clases, en la tabla 22 se presentan algunos de los factores de desbalance que llamaron nuestra atención al momento de seleccionar los bancos de datos para las dos rondas de pruebas.

**Tabla 22. Factores de desbalance de algunos de los bancos de datos seleccionados.**

Banco de datos	Factor de desbalance
Cleveland	12.62
Haberman	2.78
Led7Digits	10.97
New Thyrod	5.14
Pima Indians Diabetes	1.87
Segment	6.02
Vehículos	3.25
Wisconsin	1.86
Wine quality	29.17
Yeast	23.15

#### 5.4 Resultados experimentales (primera ronda de pruebas)

El clasificador de Heaviside, en su modo HI, fue implementado en Matlab, mientras que las Memorias Morfológicas Min fueron implementadas en Java; para el resto de los modelos se empleó la herramienta WEKA [70]. En la tabla 23 se presenta el performance promedio de los algoritmos (calculado con el método de validación *K-Fold Cross-Validation* estratificado con  $K=10$ ), en los veinte bancos de datos.

**Tabla 23. Rendimiento de los modelos de clasificación en los Bancos de Datos seleccionados**

Datasets	CH-HI	MM-Min	IB1	IB3	NB	MLP	SVMG	SMO	RBF	Log.
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	83.258%	83.129%	78.38%	71.29%	83.22%	85.48%	48.38%	74.51%	83.22%	<b>85.8%</b>
Glass	<b>91.9%</b>	<b>91.9%</b>	70.56%	71.96%	48.59%	67.75%	68.69%	56.07%	63.55%	64.01%
Haberman	70.34%	70.34%	68.3%	71.56%	<b>76.14%</b>	68.28%	73.52%	73.52%	73.85%	73.85%
Heart	78.51%	71.41%	74.81%	77.4%	<b>83.7%</b>	82.22%	55.55%	83.7%	83.7%	82.59%
Ionósfera	92.22%	92.22%	86.03%	84.9%	82.05%	92.02%	<b>93.44%</b>	87.75%	90.31%	88.03%
Iris Plant	96.033%	96.66%	95.33%	95.33%	96%	<b>97.33%</b>	96.66%	96%	95.33%	96%
Magic	81.14%	81.14%	80.93%	83.2%	72.68%	<b>85.87%</b>	65.88%	79.15%	78.42%	79.07%
New-Thyroid	91.77%	91.77%	<b>97.21%</b>	93.48%	96.74%	96.74%	75.81%	89.76%	95.34%	96.74%
Phoneme	<b>90.93%</b>	90.87%	90.19%	88.63%	76.03%	80.97%	83.77%	77.23%	77.7%	75.11%
Pima Indians diabetes	<b>96.98%</b>	<b>96.98%</b>	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%	<b>97.97%</b>	91.06%	50.46%	76.43%	97.82%	75.81%
Segment-test	94.56%	94.32%	<b>94.69%</b>	92.59%	86.41%	94.07%	41.11%	92.22%	88.39%	93.82%
Vehicle	87%	87%	97.66%	98.36%	90.77%	98.13%	<b>98.59%</b>	<b>98.59%</b>	<b>98.59%</b>	96.49%
Wine	72.64%	74.28%	94.94%	94.94%	97.19%	<b>97.19%</b>	44.38%	98.31%	98.31%	<b>97.19%</b>
Wisconsin	<b>97.15%</b>	<b>97.15%</b>	95.27%	96.42%	96.13%	95.71%	95.99%	96.71%	95.99%	96.56%

CH-HI: clasificador de Heaviside modo HI.

MM-Min: clasificador basado en las Memorias Morfológicas Min.

IB1: clasificador *Instance Based* con  $k=1$ .

IB3: clasificador *Instance Based* con  $k=3$ .

NB: clasificador *Naïve Bayes*.

MLP: clasificador Perceptrón Multicapa.

SVMG: Máquinas de Soporte Vectorial con kernel Gaussiano.

SMO: Máquinas de Soporte Vectorial con *Sequential Minimal Optimization*.

RBF: Redes Neuronales de Base Radial.

Log.: clasificador basado en la función logística.

## 5.5 Resultados de las pruebas de los signos, de Wilcoxon y de Friedman (primera ronda)

A partir de los resultados obtenidos en la primera ronda de experimentos se realizaron las pruebas no paramétricas de los signos, de Wilcoxon y de Friedman. En el caso de la prueba de los signos las diferencias obtenidas entre los algoritmos se presentan en la tabla 24.

**Tabla 24. Diferencias entre los modelos empleados en los experimentos.**

Clasificadores		N
Memorias morfológicas - C. de Heaviside	Diferencias negativas <sup>a,d,g,j,m,p,s,v,y</sup>	7
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	1
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	11
	Total	19
IB1 - C. de Heaviside	Diferencias negativas <sup>a,d,g,j,m,p,s,v,y</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	6
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
IBK K=3 - C. de Heaviside	Diferencias negativas <sup>a,d,g,j,m,p,s,v,y</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	6
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
Naïve Bayes - C. de Heaviside	Diferencias negativas <sup>a,d,g,j,m,p,s,v,y</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	6
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
Multilayer Perceptron - C. de Heaviside	Diferencias negativas <sup>a,d,g,j,m,p,s,v,y</sup>	10
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	8
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	1
	Total	19

SVM - C. de Heaviside	Diferencias	
	negativas <sup>a,d,g,j,m,p,s,v,y</sup>	16
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	3
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
SMO - C. de Heaviside	Diferencias	
	negativas <sup>a,d,g,j,m,p,s,v,y</sup>	15
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	4
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
RBF - C. de Heaviside	Diferencias	
	negativas <sup>a,d,g,j,m,p,s,v,y</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	6
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19
Logistic - C. de Heaviside	Diferencias	
	negativas <sup>a,d,g,j,m,p,s,v,y</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w,z</sup>	6
	Empates <sup>c,f,i,l,o,r,u,x,aa</sup>	0
	Total	19

- 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. Naïve Bayes < C. de Heaviside
- k. Naïve Bayes > C. de Heaviside
- l. Naïve 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

Mientras que las significancias derivadas de estos resultados se presentan a continuación en la tabla 25.

**Tabla 25. Significancias obtenidas en la prueba de los signos.**

	Memorias morfológicas - C. de Heaviside	IB1 - C. de Heaviside	IBK K=3 - C. de Heaviside	Naïve Bayes - C. de Heaviside	Multilayer Perceptron - C. de Heaviside	SVM - C. de Heaviside	SMO - C. de Heaviside	RBF - C. de Heaviside	Logistic - C. de Heaviside
Significación exacta (bilateral)	0.070*	0.167*	0.167*	0.167*	0.815*	<b>0.004*</b>	<b>0.019*</b>	0.167*	0.167*

\* Distribución binomial utilizada.

De acuerdo con los resultados obtenidos en esta prueba únicamente existen diferencias significativas entre nuestro modelo con respecto de las SVM y las SVM basadas en el método SMO. Sin embargo, cabe destacar el hecho de que la prueba de los signos es menos sensitiva que la de Wilcoxon, por lo que se decidió llevar a cabo ésta y reportar los resultados obtenidos.

Los resultados de la prueba de rangos con signo de Wilcoxon se presentan en la Tabla 26; mientras que en la Tabla 27 se presentan los valores estadísticos de contraste obtenidos.

**Tabla 26. Prueba de rangos con signo de Wilcoxon.**

		Rangos		
		N	Rango promedio	Suma de rangos
Memorias morfológicas - C. de Heaviside	Rangos negativos	7 <sup>a</sup>	4.43	31.00
	Rangos positivos	1 <sup>b</sup>	5.00	5.00
	Empates	11 <sup>c</sup>		
	Total	19		

IB1 - C. de Heaviside	Rangos negativos	13 <sup>d</sup>	10.31	134.00
	Rangos positivos	6 <sup>e</sup>	9.33	56.00
	Empates	0 <sup>f</sup>		
	Total	19		
IBK K=3 - C. de Heaviside	Rangos negativos	13 <sup>g</sup>	10.31	134.00
	Rangos positivos	6 <sup>h</sup>	9.33	56.00
	Empates	0 <sup>i</sup>		
	Total	19		
Naïve Bayes - C. de Heaviside	Rangos negativos	13 <sup>j</sup>	10.92	142.00
	Rangos positivos	6 <sup>k</sup>	8.00	48.00
	Empates	0 <sup>l</sup>		
	Total	19		
Multilayer Perceptron - C. de Heaviside	Rangos negativos	10 <sup>m</sup>	10.00	100.00
	Rangos positivos	8 <sup>n</sup>	8.88	71.00
	Empates	1 <sup>o</sup>		
	Total	19		
SVM - C. de Heaviside	Rangos negativos	16 <sup>p</sup>	10.94	175.00
	Rangos positivos	3 <sup>q</sup>	5.00	15.00
	Empates	0 <sup>r</sup>		
	Total	19		
SMO - C. de Heaviside	Rangos negativos	15 <sup>s</sup>	9.73	146.00
	Rangos positivos	4 <sup>t</sup>	11.00	44.00
	Empates	0 <sup>u</sup>		
	Total	19		
RBF - C. de Heaviside	Rangos negativos	13 <sup>v</sup>	9.46	123.00
	Rangos positivos	6 <sup>w</sup>	11.17	67.00
	Empates	0 <sup>x</sup>		
	Total	19		
Logistic - C. de Heaviside	Rangos negativos	13 <sup>y</sup>	10.15	132.00
	Rangos positivos	6 <sup>z</sup>	9.67	58.00
	Empates	0 <sup>aa</sup>		
	Total	19		

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. Naïve Bayes < C. de Heaviside
- k. Naïve Bayes > C. de Heaviside
- l. Naïve 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 27. Estadísticos de contraste de la prueba de Wilcoxon.**

Estadísticos de prueba									
	Memorias morfológicas - C. de Heaviside	IB1 - C. de Heaviside	IBK K=3 - C. de Heaviside	Naïve Bayes - C. de Heaviside	Multilayer Perceptron - C. de Heaviside	SVM - C. de Heaviside	SMO - C. de Heaviside	RBF - C. de Heaviside	Logistic - C. de Heaviside
Z	-1.820*	-1.569*	-1.569*	-1.891*	-.631*	-3.219*	-2.052*	-1.127*	-1.489*
Sig. asintótica (bilateral)	0.069	0.117	0.117	0.059	0.528	<b>0.001</b>	<b>0.040</b>	0.260	0.136

\*. Se basa en rangos positivos.

De acuerdo con los valores presentados en las tablas 26 y 27 es posible afirmar que el Clasificador de Heaviside supera a las Memorias Morfológicas Min, al clasificador Naïve Bayes, a las SVM, y al clasificador SMO con un 90% de confianza. Cabe destacar que con un 95% de confianza el clasificador de Heaviside supera a las SVM y al modelo SMO.



Finalmente, al realizar la prueba de Friedman se obtuvieron los rangos derivados del desempeño de los clasificadores en cada uno de los bancos de datos empleados. Los rangos se presentan en la tabla 28.

**Tabla 28. Rangos obtenidos al aplicar la prueba de Friedman.**

Rangos	
	Rango promedio
C. de Heaviside HI	3.63
Memorias morfológicas	4.68
IB1	6.18
IBK K=3	6.00
Naïve Bayes	6.30
Multilayer Perceptron	4.88
SVM	7.38
SMO	5.55
RBF	5.50
Logistic	4.93

Como puede observarse, el clasificador de Heaviside obtiene uno de los rangos más bajos, y además la significancia asintótica es de 0.01, por lo que es posible rechazar la hipótesis nula de igualdad de resultados entre los clasificadores. A partir de los resultados de las tres pruebas de significancia estadística, podemos concluir que en esta primera ronda de pruebas el clasificador de Heaviside obtiene los mejores resultados; ya que supera a dos modelos y tiene un comportamiento equivalente a los demás modelos, por lo que se puede considerar que sus resultados son bastante competitivos.

## 5.6 Resultados experimentales (segunda ronda)

Tras el éxito obtenido en la primera ronda de pruebas del clasificador de Heaviside consideramos adecuado probar nuestro clasificador tomando en cuenta otros bancos de datos y también algunos otros clasificadores destacados en el estado del arte del Reconocimiento de Patrones.

El clasificador de Heaviside, en ambos modos, fue implementado en Matlab; para el resto de los modelos se empleó la herramienta WEKA [70]. En la tabla 29 se presenta el performance promedio de los algoritmos (calculado con el método de validación K Fold Cross Validation con K=10), en los veinte bancos de datos.

## 5.7 Resultados de las pruebas de los signos, de Wilcoxon y de Friedman (segunda ronda)

A partir de los resultados obtenidos en la segunda ronda de experimentos se realizaron las pruebas no paramétricas de los signos, de Wilcoxon y de Friedman. En el caso de la prueba de los signos las diferencias obtenidas entre los algoritmos se presentan en la tabla 30.

**Tabla 29. Rendimiento de los modelos de clasificación en los bancos de datos seleccionados.**

Clasificador \ Banco de datos	CH-HI	CH-HS	ABM1	BGG	IB1	NB	MLP	RF	SVMG
Balance	86.6951%	86.4307%	81.92%	82.72%	84.8%	90.56%	<b>91.84%</b>	81.28%	89.44%
Banknote authentication	97.7850%	97.9301%	99.6356%	98.688%	99.8542%	84.2566%	99.9271%	99.2711%	<b>100%</b>
Breast Tissue	66.9485%	64.9254%	71.6981%	<b>73.5849%</b>	71.6981%	66.9811%	63.2075%	71.6981%	20.7547%
Cardiocotography	90.3919%	87.8234%	93.2393%	94.4497%	91.1571%	82.3612%	91.9567%	<b>94.92%</b>	80.0094%
Climate Model Simulation Crashes	<b>100%</b>	<b>100%</b>	94.2593%	92.2222%	89.4444%	95%	92.963%	93.3333%	91.4815%
Connectionist Bench	81.3357%	82.4114%	82.6923%	76.9231%	<b>86.5385%</b>	67.7885%	82.2115%	83.1731%	65.8654%
Default of credit card clients	76.7634%	76.69%	78.3767%	81.7%	73.02%	69.62%	<b>81.7833%</b>	81.7067%	77.9433%
Fertility	88%	88%	88%	88%	83%	88%	<b>90%</b>	86%	88%
Ionosphere	92.22%	91.1614%	92.3077%	91.1681%	86.3248%	82.6211%	91.1681%	92.8775%	<b>93.4473%</b>
Led7Digit	60.7415%	61.256%	71%	<b>72.8%</b>	69.8%	70.2%	70.6%	70.4%	71.8%
Movement_LIBRAS	85.2444%	86.33%	74.4444%	69.7222%	<b>85.8333%</b>	62.7778%	79.4444%	83.0556%	36.3889%
QSAR biodegradation	81.8352%	81.6501%	86.1611%	84.8341%	84.455%	75.9242%	86.7299%	<b>86.8246%</b>	84.9289%
Ring	90.43%	90.154%	94.3649%	93.7568%	75.05%	<b>97.97%</b>	91.06%	95.2973%	50.46%
Seeds	90.6666%	90.4285%	93.3333%	92.8571%	94.2857%	91.4286%	<b>95.2381%</b>	93.8095%	90.4762%
Spambase	82.4142%	81.2853%	92.8227%	94.0613%	90.8854%	79.5954%	91.4945%	<b>95.4753%</b>	84.4899%
Texture	98.1363%	93.6787%	97.5818%	95.1636%	99.1273%	77.5091%	<b>99.812%</b>	97.8727%	94.4727%
Twonorm	50.9307%	51.0968%	94.1757%	93.5496%	94.7568%	<b>97.8649%</b>	96.8378%	96.6216%	97.7297%
Vertebral column	83.258%	83.129%	81.2903%	82.5806%	78.3871%	83.2258%	<b>85.4839%</b>	83.5484%	48.3871%
Wine quality	60.2894%	60.0681%	64.2276%	62.9769%	63.9775%	54.7217%	61.4759%	<b>70.0438%</b>	58.2239%
Yeast	58.1771%	57.8129%	59.2992%	60.0404%	52.1563%	58.558%	58.2884%	<b>61.3881%</b>	42.9245%

---

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.

**Tabla 30. Resultados de la prueba de los signos.**

Frecuencias		N
Heaviside HS - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	13
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	5
	Empates <sup>c,f,i,l,o,r,u,x</sup>	2
	Total	20
AdaBoost M1 - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	5
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	14
	Empates <sup>c,f,i,l,o,r,u,x</sup>	1
	Total	20
Bagging - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	7
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	12
	Empates <sup>c,f,i,l,o,r,u,x</sup>	1
	Total	20
IB1 - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	8
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	12
	Empates <sup>c,f,i,l,o,r,u,x</sup>	0
	Total	20
Naïve Bayes - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	12
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	7
	Empates <sup>c,f,i,l,o,r,u,x</sup>	1
	Total	20

Multilayer Perceptron - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	4
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	16
	Empates <sup>c,f,i,l,o,r,u,x</sup>	0
	Total	20
Random Forest - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	5
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	15
	Empates <sup>c,f,i,l,o,r,u,x</sup>	0
	Total	20
SVM Gaussian K - Heaviside HI	Diferencias negativas <sup>a,d,g,j,m,p,s,v</sup>	11
	Diferencias positivas <sup>b,e,h,k,n,q,t,w</sup>	8
	Empates <sup>c,f,i,l,o,r,u,x</sup>	1
	Total	20

- a. Heaviside HS < Heaviside HI
- b. Heaviside HS > Heaviside HI
- c. Heaviside HS = Heaviside HI
- d. AdaBoost M1 < Heaviside HI
- e. AdaBoost M1 > Heaviside HI
- f. AdaBoost M1 = Heaviside HI
- g. Bagging < Heaviside HI
- h. Bagging > Heaviside HI
- i. Bagging = Heaviside HI
- j. IB1 < Heaviside HI
- k. IB1 > Heaviside HI
- l. IB1 = Heaviside HI
- m. Naïve Bayes < Heaviside HI
- n. Naïve Bayes > Heaviside HI
- o. Naïve Bayes = Heaviside HI
- p. Multilayer Perceptron < Heaviside HI
- q. Multilayer Perceptron > Heaviside HI
- r. Multilayer Perceptron = Heaviside HI
- s. Random Forest < Heaviside HI
- t. Random Forest > Heaviside HI
- u. Random Forest = Heaviside HI
- v. SVM Gaussian K < Heaviside HI
- w. SVM Gaussian K > Heaviside HI
- x. SVM Gaussian K = Heaviside HI

En esta tabla es posible apreciar que salvo el Naïve Bayes y las SVM, la mayoría de los modelos superan al clasificador de Heaviside en cuanto a los porcentajes obtenidos. Sin embargo, la validez de estas diferencias está en función de las significancias que se presentan en la tabla 31.

**Tabla 31. Valores de significancia de la prueba de los signos.**

	Heaviside HS - Heaviside HI	AdaBoost M1 - Heaviside HI	Bagging - Heaviside HI	IB1 - Heaviside HI	Naïve Bayes - Heaviside HI	Multilayer Perceptron - Heaviside HI	Random Forest - Heaviside HI	SVM Gaussian K - Heaviside HI
Significación exacta (bilateral)	0.096*	0.064*	0.359*	0.503*	0.359*	<b>0.012*</b>	<b>0.041*</b>	0.648*

\*. Distribución binomial utilizada.

De acuerdo con los valores de significancia obtenidos, únicamente hay diferencias significativas entre el desempeño de los clasificadores Perceptrón Multicapa (*Multilayer Perceptron*) y Random Forest respecto del clasificador de Heaviside en su modo HI; estas diferencias son a favor de los modelos ya mencionados y no a favor de nuestro clasificador. Sin embargo, dado que la prueba de los signos es menos sensitiva que la de Wilcoxon, se decidió llevar a cabo ésta y reportar los resultados obtenidos.

Los resultados de la prueba de rangos con signo de Wilcoxon se presentan en la Tabla 32; mientras que en la Tabla 33 se presentan los valores estadísticos de contraste obtenidos de la prueba de Wilcoxon.

**Tabla 32. Prueba de rangos con signo de Wilcoxon.**

		Rangos		
		N	Rango promedio	Suma de rangos
Heaviside HS - Heaviside HI	Rangos negativos	13 <sup>a</sup>	9.69	126.00
	Rangos positivos	5 <sup>b</sup>	9.00	45.00
	Empates	2 <sup>c</sup>		
	Total	20		
AdaBoost M1 - Heaviside HI	Rangos negativos	5 <sup>d</sup>	11.20	56.00
	Rangos positivos	14 <sup>e</sup>	9.57	134.00
	Empates	1 <sup>f</sup>		
	Total	20		

Bagging - Heaviside HI	Rangos negativos	7 <sup>g</sup>	9.43	66.00
	Rangos positivos	12 <sup>h</sup>	10.33	124.00
	Empates	1 <sup>i</sup>		
	Total	20		
IB1 - Heaviside HI	Rangos negativos	8 <sup>j</sup>	12.75	102.00
	Rangos positivos	12 <sup>k</sup>	9.00	108.00
	Empates	0 <sup>l</sup>		
	Total	20		
Naïve Bayes - Heaviside HI	Rangos negativos	12 <sup>m</sup>	11.00	132.00
	Rangos positivos	7 <sup>n</sup>	8.29	58.00
	Empates	1 <sup>o</sup>		
	Total	20		
Multilayer Perceptron - Heaviside HI	Rangos negativos	4 <sup>p</sup>	12.00	48.00
	Rangos positivos	16 <sup>q</sup>	10.13	162.00
	Empates	0 <sup>r</sup>		
	Total	20		
Random Forest - Heaviside HI	Rangos negativos	5 <sup>s</sup>	9.00	45.00
	Rangos positivos	15 <sup>t</sup>	11.00	165.00
	Empates	0 <sup>u</sup>		
	Total	20		
SVM Gaussian K - Heaviside HI	Rangos negativos	11 <sup>v</sup>	11.73	129.00
	Rangos positivos	8 <sup>w</sup>	7.63	61.00
	Empates	1 <sup>x</sup>		
	Total	20		

- a. Heaviside HS < Heaviside HI
- b. Heaviside HS > Heaviside HI
- c. Heaviside HS = Heaviside HI
- d. AdaBoost M1 < Heaviside HI
- e. AdaBoost M1 > Heaviside HI
- f. AdaBoost M1 = Heaviside HI
- g. Bagging < Heaviside HI
- h. Bagging > Heaviside HI
- i. Bagging = Heaviside HI
- j. IB1 < Heaviside HI
- k. IB1 > Heaviside HI
- l. IB1 = Heaviside HI
- m. Naïve Bayes < Heaviside HI
- n. Naïve Bayes > Heaviside HI

- o. Naïve Bayes = Heaviside HI
- p. Multilayer Perceptron < Heaviside HI
- q. Multilayer Perceptron > Heaviside HI
- r. Multilayer Perceptron = Heaviside HI
- s. Random Forest < Heaviside HI
- t. Random Forest > Heaviside HI
- u. Random Forest = Heaviside HI
- v. SVM Gaussian K < Heaviside HI
- w. SVM Gaussian K > Heaviside HI
- x. SVM Gaussian K = Heaviside HI

**Tabla 33. Estadísticos de contraste de la prueba de Wilcoxon.**

	Heaviside HS - Heaviside HI	AdaBoost M1 - Heaviside HI	Bagging - Heaviside HI	IB1 - Heaviside HI	Naïve Bayes - Heaviside HI	Multilayer Perceptron - Heaviside HI	Random Forest - Heaviside HI	SVM Gaussian K - Heaviside HI
Z	-1.764*	-1.569†	-1.167†	-0.112†	-1.489*	-2.128†	-2.240†	-1.368*
Sig. asintótica (bilateral)	0.078	0.117	0.243	0.911	0.136	<b>0.033</b>	<b>0.025</b>	0.171

\*. Se basa en rangos positivos.

†. Se basa en rangos negativos.

De acuerdo con los valores presentados en las tablas 32 y 33 es posible afirmar que el Clasificador de Heaviside en modo HI es superado con 95% de confianza por el Perceptrón Multicapa y el Random Forest. Mientras que el resto de los clasificadores no es posible rechazar la hipótesis nula de igualdad.

Para concluir las pruebas de significancia se llevó a cabo la prueba de Friedman. Los rangos obtenidos por cada clasificador a partir de cada uno de los bancos de datos empleados se presentan en la tabla 34.

Como se puede observar en esta tabla el Random Forest tiene el rango mínimo, seguido del perceptrón multicapa; el clasificador de Heaviside se ubica en sexta posición. Por otro lado, el valor de significancia asintótica es de 0, por lo que hay una evidencia altamente significativa para rechazar la hipótesis nula de igualdad de muestras (o de desempeños de los clasificadores).

A partir de los resultados obtenidos en las tres pruebas, es posible concluir que en esta segunda ronda el clasificador de Heaviside no fue el mejor; de hecho, es superado de forma

significativa por el perceptrón multicapa y por el algoritmo *Random Forest*, y tiene un desempeño equivalente al de los demás modelos, por lo que su comportamiento puede considerarse como competitivo. El hecho más importante de esta segunda ronda de pruebas, es que el clasificador *Random Forest* cumplió con las expectativas que nos orillaron a elegirlo; ya que, a partir de la experiencia adquirida durante la elaboración de artículos y materiales para cursos de Reconocimiento de Patrones, observamos que su comportamiento es bastante bueno en la mayoría de los bancos de datos en los que se utilizó.

**Tabla 34. Resultados de la prueba de Friedman.**

Rangos	
	Rango promedio
Heaviside HI	5.60
Heaviside HS	6.30
Ada Boost M1	4.08
Bagging	4.60
IB1	5.40
Naïve Bayes	6.18
Multilayer Perceptron	3.48
Random Forest	<b>3.25</b>
SVM Gaussian	6.13

## 5.8 Pruebas intensivas del clasificador de Heaviside

El clasificador de Heaviside trabaja con patrones cuyas componentes están representadas en términos de un sistema numérico de base  $b$ ; ahora bien, esto nos lleva a una gran cantidad de posibilidades relacionadas sobre las posibles bases en las que se pueden representar las componentes de los patrones. Por ejemplo, si tenemos un banco de datos en donde el valor máximo de las componentes, una vez concluida la fase de preprocesamiento, es 1000 entonces se tienen 999 valores posibles de base  $b$  (ya que las bases irían desde el 2 hasta el 1000). Por lo tanto, se cuenta con una gran cantidad de opciones, las cuales pueden aportar diferentes rendimientos de clasificación.

En esta sección presentaremos los resultados obtenidos tras probar todos los posibles valores de datos con algunos bancos de datos; cabe destacar que no fue posible realizar este análisis con una gran cantidad de bancos de datos ya que en muchos casos el rango de posibles bases abarca los miles o millones de casos y era necesario ejecutar todo el algoritmo del clasificador incluyendo el método de validación, que en este caso fue de *10-Fold Cross-Validation* estratificado [201].



El primer caso que analizamos fue el banco de datos de Iris Plant, el cual prácticamente es un banco de datos de rigor al momento de evaluar el desempeño de nuevos clasificadores. En la tabla 35. Se presenta un resumen de los resultados obtenidos.

**Tabla 35. Pruebas intensivas con el banco de datos Iris Plant.**

<b>Número de bases posibles</b>	<b>Desempeño mínimo</b>	<b>Base</b>	<b>Desempeño máximo</b>	<b>Base</b>
79	95.7%	6	96.033%	4

Para este banco de datos, que puede considerarse como uno pequeño, debido al número de patrones que lo conforman, el tiempo de ejecución fue de 1900.904s, es decir 31 minutos con 40 segundos, en un equipo con altos recursos computacionales.

Posteriormente, nos dimos a la tarea de analizar el banco de datos de Haberman; en la tabla 36 presentamos los resultados obtenidos.

**Tabla 36. Pruebas intensivas con el banco de datos Haberman.**

<b>Número de bases posibles</b>	<b>Desempeño mínimo</b>	<b>Base</b>	<b>Desempeño máximo</b>	<b>Base</b>
84	70.29%	11	71.467%	71

En este banco de datos compuesto por 306 patrones, el tiempo de ejecución fue de 5292.632 segundos, es decir 1 hora con 28 minutos y 12 segundos. A partir de los resultados obtenidos en las tablas 34 y 35 es posible observar que para el banco de datos Iris Plant se obtuvo el desempeño máximo con la base 4; por otro lado, en el banco de datos Haberman el desempeño máximo se obtuvo con la base 71, este resultado supera al resultado presentado en la tabla 22.

Posteriormente se seleccionaron los bancos de Titanic (2201 patrones y 3 rasgos) y banana (5300 patrones y 2 rasgos); cabe destacar que en el banco de datos de Titanic se tienen 63001 posibles valores de base, mientras que para el banco de datos de banana se tienen 42601 posibles valores de base. Con el banco de datos de Titanic se comenzó la prueba el día 22 de septiembre, mientras que con el banco de datos de banana la prueba se comenzó el día 25 de octubre. Para el día 30 de enero de 2017 la aplicación de Matlab colapsó y terminó abruptamente con la prueba de Titanic y la prueba con el banco de datos de banana se suspendió el día 31 de enero de 2017.

### 5.8.1 Selección del valor adecuado para la base

A pesar de que la cantidad de pruebas exhaustivas fue únicamente de dos, a lo largo de este trabajo de tesis el primer valor de  $b$  que se consideró al momento de hacer las pruebas fue  $b=mc+1$  donde  $mc$  es el valor de la componente más grande del conjunto de entrenamiento. La razón de esta elección es que el tiempo de ejecución será mínimo ya que los patrones tendrán su dimensión original. Posteriormente, se consideraron los valores de base que están entre 2 y 10 ya que se consideró adecuado probar con los dígitos base del sistema de numeración decimal. Cabe destacar que, durante las dos rondas de pruebas, en la mayoría de los casos, los mejores resultados se obtuvieron con el valor de base  $b=mc+1$ . Por lo tanto, en la mayoría de pruebas realizadas con el clasificador de Heaviside se han considerado los valores de base del 2 al 9 y el valor de  $b=mc+1$ .

Debido a que sólo fue posible completar dos pruebas exhaustivas, no es posible afirmar, empíricamente, que la mejor selección de base está entre los valores que nosotros consideramos en este trabajo; sin duda alguna, esta búsqueda está considerada dentro de los trabajos a futuro para esta tesis.

## 5.9 Aplicaciones del Clasificador de Heaviside

Una vez concluida la etapa de formulación y las primeras pruebas de funcionamiento del Clasificador propuesto en esta tesis, 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; mientras que la segunda aplicación está relacionada con la detección rápida de la Degeneración Macular Dependiente de la edad.

### 5.9.1 Clasificación de recursos educativos para la generación inteligente de metadatos

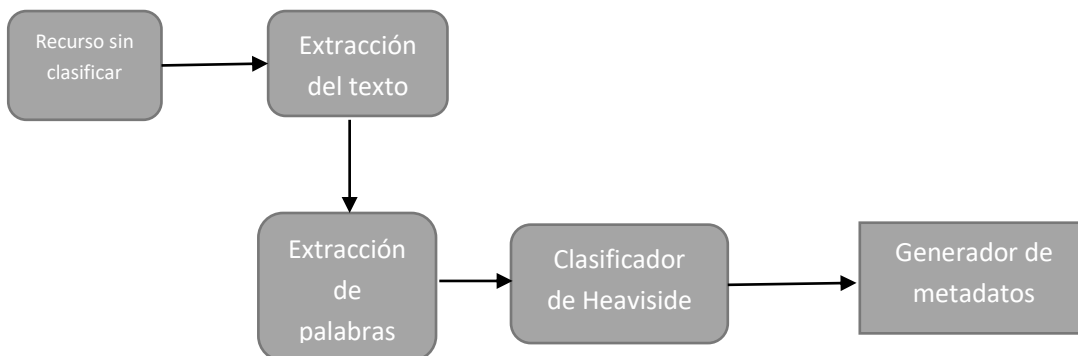
Hoy en día hay una gran cantidad de materiales de aprendizaje disponibles en Internet. Sitios como Udacity [202], Coursera [203] o las plataformas de aprendizaje de muchas instituciones educativas proporcionan una gran cantidad de información. Esta situación se complica al momento de buscar materiales de calidad sobre un tema específico. La solución para este problema es generar un conjunto de metadatos por cada material educativo que facilite su búsqueda; sin embargo, la generación manual de estos metadatos es una tarea que consume mucho tiempo y, por lo regular, está sujeta a muchos errores. Por lo tanto, la generación automática de metadatos se ha convertido en un área fértil de investigación; como resultado de esto se han propuesto diversos métodos para generar metadatos de materiales educativos. A partir del clasificador de Heaviside, propusimos un método para generar automáticamente metadatos para recursos educativos; el cual ya fue publicado en una revista indizada [204]. En la figura 11, se esquematiza el diseño de un sistema generador de metadatos a partir del algoritmo propuesto.

Este método fue probado en un banco de datos generado a partir de recursos educativos de asignaturas de los primeros semestres de carreras de Ingeniería y en los resultados el Clasificador de Heaviside obtuvo resultados competitivos en contra de otros modelos de clasificación.

### 5.9.2 Pre-diagnóstico de la Degeneración Macular relacionada con la Edad

Dado que se han estado buscando diferentes escenarios para la aplicación del clasificador de Heaviside, consideramos conveniente incursionar en un campo que ha sido muy explorado, incluso por el autor de este trabajo, y del cual se ha publicado una gran cantidad de trabajos de tesis y artículos; nos referimos al análisis de imágenes de retina y en el caso particular de una enfermedad que ha sido poco explorada: La Degeneración Macular relacionada con la Edad [205]. Dado que el pre-diagnóstico de esta enfermedad en imágenes de retina está basado en la detección de afecciones conocidas como drusas [206,207] fue necesario diseñar un método basado en algunas operaciones simples de procesamiento de imágenes y de morfología matemática, para posteriormente generar patrones de datos a partir del uso de momentos invariantes. A grandes rasgos, el método propuesto para generar un banco de datos a partir de las imágenes de retina disponibles se presenta en la figura 12.

Una vez generado el banco de datos es posible entrenar el clasificador de Heaviside. En los experimentos realizados nuestro modelo de clasificación superó a otros clasificadores como las Máquinas de Soporte Vectorial, a las Redes Bayesianas y árboles como el J48.

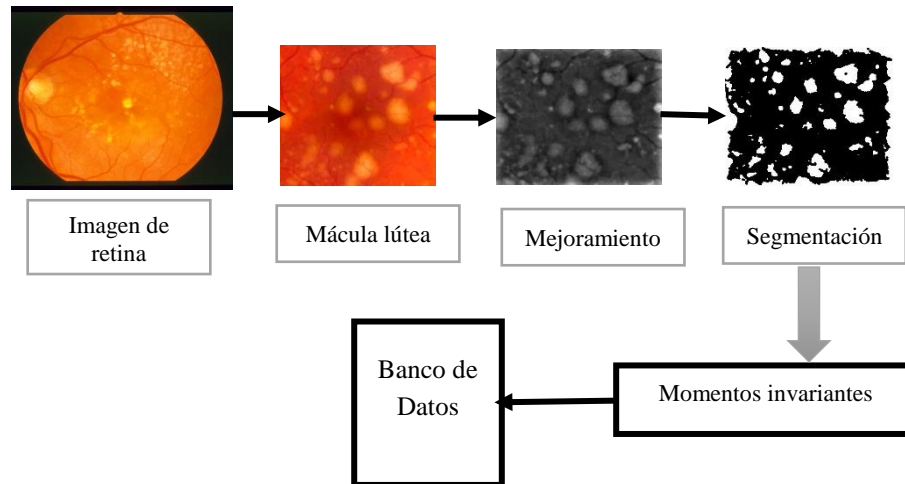


**Figura 11. Método Propuesto para la clasificación inteligente de recursos educativos.**

### 5.10 Trabajos publicados

Derivado del trabajo realizado en esta tesis, actualmente se tienen dos artículos publicados. El primero, se titula Clasificador de Heaviside [208] y está publicando en la revista Nova Scientia que pertenece al padrón de revistas Conacyt. Por otro lado, el artículo titulado Social Web Content

Enhancement in a Distance Learning Environment: Intelligent Metadata Generation for Resources [204], fue publicado en el volumen 18, número 1 de la revista *The International Review of Research in Open and Distributed Learning*, la cual es una revista JCR. El método publicado en este artículo se presentó de manera resumida en la sección 5.9.1.



**Figura 12. Método para generar el banco de datos empleado en el pre-diagnóstico de la degeneración macular en imágenes de retina.**

# Capítulo 6. Conclusiones, Contribuciones y Trabajo a Futuro

En este capítulo presentamos las conclusiones a las que hemos llegado tras las pruebas que hemos realizado con el Clasificador de Heaviside; posteriormente presentaremos las contribuciones que hemos logrado en este trabajo de tesis. Finalmente, enlistaremos los trabajos a futuro.

## 6.1 Conclusiones

En este trabajo de tesis proponemos un nuevo modelo para la clasificación supervisada de patrones que está basado en la función de Heaviside. Es muy importante resaltar que, tras un extenso análisis del estado del arte del reconocimiento de patrones, no se encontraron modelos de Inteligencia Computacional o de reconocimiento de patrones 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; por lo tanto, es posible afirmar que cumplimos el objetivo particular 1.

El modelo de clasificador propuesto sobresale por el hecho de que trabaja con patrones cuyas componentes están en términos de números enteros no negativos, los cuales pueden ser obtenidos a partir de patrones con valores reales mediante un proceso de transformación. Lo anterior permite que el modelo de clasificación propuesto trabaje con patrones representados en cualquier sistema numérico con base entera 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 y en operaciones propias de las lattices y las estructuras algebraicas ordenadas. La función de aprendizaje, la cual permite construir el modelo del clasificador se denomina L; mientras que la operación que permite realizar la clasificación de patrones se denomina C; cumpliéndose, de esta forma, el objetivo general de este trabajo de tesis y el objetivo particular 2.

Las operaciones de aprendizaje y clasificación de patrones propuestas en esta tesis están sustentadas en un conjunto de definiciones, lemas y teoremas que demuestran formalmente el rendimiento y las capacidades del clasificador de Heaviside en la clasificación de los patrones que pertenecen a los conjuntos de entrenamiento y de prueba; por lo tanto, se cumple el objetivo particular 3.

Después de probar el clasificador de Heaviside en varios bancos de datos relacionados con áreas como Inteligencia Computacional, *Machine Learning* y 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 en varios de los casos supera a otros modelos y en los otros casos ofrece un desempeño bastante competitivo. También es importante resaltar que la validez de los experimentos realizados fue sustentada mediante algunas pruebas no paramétricas de significancia estadística. A partir de los resultados de estas pruebas pudimos observar que a pesar de que nuestro clasificador fue superado por otros modelos en varios de los bancos de datos empleados, sobre todo en la segunda ronda de pruebas, las diferencias no fueron significativas.

No solo hicimos pruebas con bancos de datos de repositorios importantes como UCI o KEEL; como se mencionó en la sección 5.9 se propusieron al menos dos aplicaciones para nuestro clasificador, en dichas aplicaciones fue necesario diseñar un método para extraer rasgos útiles a partir de datos en bruto; 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. Por lo tanto, es posible afirmar que el objetivo particular 4 de esta tesis se cumple cabalmente.

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, cuando se elijan valores de base muy pequeños. Sin embargo, en las pruebas que hasta ahora hemos realizado, observamos que muchos de los mejores resultados se obtienen con valores altos de base, por lo que esta situación no ha limitado gravemente el rendimiento del clasificador.

## 6.2 Contribuciones

En este trabajo de tesis hemos propuesto un nuevo modelo de clasificación supervisada de patrones, el cual refuerza el enfoque asociativo del reconocimiento de patrones que se inició en el año 2002 en el Centro de Investigación en Computación. A continuación, enlistamos las contribuciones de esta tesis.

- Propusimos una operación original llamada L, la cual funciona para la etapa de entrenamiento o aprendizaje del clasificador de patrones.
- Hemos formulado y demostrado las propiedades más importantes de esta operación de aprendizaje.
- Propusimos una operación original llamada C, la cual funciona para la etapa de clasificación del clasificador de Heaviside.
- Hemos trabajado en la demostración de las propiedades más importantes de la operación C, para ambos modos del clasificador de Heaviside.

- Las operaciones L y C, generalizan el comportamiento de los operadores asociativos  $\alpha, \beta$  los cuáles fueron propuestos en las memorias asociativas Alfa-Beta.
- El modelo propuesto en esta tesis es del tipo *one-shot*, por lo que no tiene problemas de convergencia como las redes neuronales.
- El modelo propuesto puede aplicarse en diferentes contextos; hasta ahora se han aplicado en ámbitos como en la clasificación de textos académicos y en el pre-diagnóstico de la Degeneración Macular relacionada con la Edad.
- El modelo, en general, obtiene resultados competitivos; sin embargo, debemos recordar que por el teorema de *no free lunch* [209] no es posible tener un clasificador que sea el mejor para todos los bancos de datos, por lo que hubo casos en los que el clasificador de Heaviside tiene un desempeño deficiente.
- El algoritmo derivado del modelo de clasificación propuesto en esta tesis es fácil de programar y por lo tanto puede ser implementado en equipos con diferentes configuraciones de hardware.
  - Sin embargo, tras diferentes experimentos realizados hemos notado que se debe tener cuidado con la elección de la base en bancos formados por patrones cuyas componentes tienen una gran cantidad de dígitos decimales. Si se elige un valor pequeño para  $b$  entonces se van a generar conjuntos de entrenamiento y prueba de alta dimensionalidad, lo cual implica que las etapas de entrenamiento y clasificación requerirán mucho tiempo de cómputo.

### 6.3 Trabajo a futuro

Como se mencionó en secciones anteriores, hemos obtenido buenos resultados en este trabajo de tesis; sin embargo, consideramos que aún es posible realizar más investigación acerca de las posibles aplicaciones y limitantes del modelo. A continuación, presentamos los trabajos a futuro que consideramos se deben realizar para mejorar de forma significativa los resultados de esta tesis.

1. Revisar a fondo la formulación de las operaciones L y C en busca de formulaciones alternas que optimicen su funcionamiento.
2. Profundizar en el estudio de las propiedades, de ambas operaciones, del clasificador para potencializar sus alcances y reducir el efecto de sus limitaciones.
3. A partir de las hipótesis en las que se basa el clasificador de Heaviside, proponer al menos una regla para el manejo de valores perdidos en los bancos de datos, de manera que no sea necesario realizar alguna de las técnicas comunes de imputación.
4. Realizar pruebas en más bancos de datos, pertenecientes a repositorios públicos y privados, para comparar los resultados obtenidos con otros modelos famosos del estado del arte del reconocimiento de patrones y con otros modelos incipientes. En estas pruebas

se deben considerar más bancos de datos de alta complejidad; es decir, bancos de datos desbalanceados, con valores perdidos y con valores atípicos o *outliers*.

5. Continuar con la generación de escenarios de aplicación para el modelo propuesto en esta tesis; actualmente se han generado dos casos concretos de aplicación, pero creemos que es posible encontrar otros casos en donde podamos aplicar nuestro clasificador con resultados competitivos.
6. A partir de lo mencionado en los dos puntos anteriores, tenemos contemplada la publicación de algunos de éstos en congresos nacionales o internacionales.
7. Presentar los resultados más importantes de los experimentos del punto 3 y de las aplicaciones propuestas en el punto 4 en revistas JCR.
8. 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 compuestos por una gran cantidad de patrones o con una gran cantidad de rasgos.
9. Investigar sobre las formas en las que se podría encontrar el mejor valor de base para un determinado banco de datos; en la sección 5.6 mostramos que realizar pruebas intensivas con todos los posibles valores de base no es la mejor opción, ya que éstas pueden tomar una gran cantidad de tiempo, inclusive con equipos que cuentan con bastantes recursos de cómputo. Podemos considerar el trabajo a realizar en el punto anterior para que, a través de una implementación paralela continuemos investigando acerca de la elección del valor de base más adecuado.



# Apéndices

## A1. Demostración de las propiedades de la operación $L$

### Propiedad L1:

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

Demostración:

Por hipótesis y por la definición de H se tiene:

$$H(A_i^k + A_j^k + b) = H(0 + 0 + b) = 1 \text{ y } (A_i^k + b) - (A_j^k + 1) = (0 + b) - (0 + 1) = b - 1$$

Por lo tanto:

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

q.e.d.

### Propiedad L2:

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

Demostración:

Por hipótesis y por la definición de H se tiene:

$$H[(b - 1) + (b - 1) + b] = H(3b - 2) = 1, \text{ dado que } 3b - 2 > 0$$

Y además se cumple que:  $[(b - 1) + b] - [(b - 1) + 1] = 2b - 1 - b = b - 1$

Al realizar las operaciones, por L1 resulta lo siguiente:

$$\begin{aligned} L(b - 1, b - 1) &= [H((b - 1) + (b - 1) + b)] \cdot [((b - 1) + b) - ((b - 1) + 1)] \\ &= (1) \cdot (b - 1) = b - 1 = L(0, 0) \end{aligned}$$

q.e.d.

En general es demostrable, siguiendo la misma metodología que en L1 y L2, que si  $b \in \mathbb{Z}$  tal que  $b > 1, \forall x \in \mathbb{Z}$  tal que  $0 \leq x \leq b - 1$  se cumple  $L(x, x) = b - 1$ .

Este resultado puede verse como una generalización de las propiedades L1 y L2, pero también podría atacarse demostrando primero este resultado general como un teorema, de modo que las propiedades L1 y L2 queden como corolarios.

**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 la siguiente desigualdad:  $L(x, y) \leq L(y, x)$ .

Demostración:

Por hipótesis, se cumple que  $x \leq y$ , con  $0 \leq x \leq b - 1$  y  $0 \leq y \leq b - 1$ . Al realizar operaciones elementales a ambos lados del signo de desigualdad, se tiene:

$$x \leq y \tag{1}$$

$$x + x \leq y + y$$

$$x - y \leq y - x$$

$$x - y + (b - 1) \leq y - x + (b - 1)$$

$$(x + b) - (y + 1) \leq (y + b) - (x + 1)$$

$$(1) \cdot [(x + b) - (y + 1)] \leq (1) \cdot [(y + b) - (x + 1)]$$

Pero por definición de  $H$ , y dado que se cumple que  $x > 0$  y también  $y > 0$ , se cumplen las siguientes identidades:

$$1 = H(x + y + b) = H(y + x + b) \tag{2}$$

Al sustituir (2) en (1):

$$H(x + y + b) \cdot [(x + b) - (y + 1)] \leq H(y + x + b) \cdot [(y + b) - (x + 1)] \tag{3}$$

Sin embargo, por la definición de  $L$ , el lado izquierdo de (3) es ni más ni menos que  $L(x, y)$ , mientras que el lado derecho es  $L(y, x)$ , por ello se cumple:  $L(x, y) \leq L(y, x)$ .

q.e.d.

**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 se cumple la siguiente desigualdad:  $L(x, z) \leq L(y, z)$

Demostración:

Por hipótesis, se cumple que  $x \leq y$ , con  $0 \leq x \leq b - 1$  y  $0 \leq y \leq b - 1$ , al realizar operaciones elementales en ambos lados del signo de desigualdad se tiene:

$$x \leq y$$

$$x + z \leq y + z$$

$$x - z \leq y - z$$

$$x - z + b - 1 \leq y - z + b - 1$$

Agrupando términos tenemos que:

$$x + b - z - 1 \leq y + b - z - 1 \quad (1)$$

$$x + b - (z + 1) \leq y + b - (z + 1)$$

$$(1) \cdot [x + b - (z + 1)] \leq (1) \cdot [y + b - (z + 1)]$$

Por definición de H y dado que se cumple que  $x \geq 0, y \geq 0, z \geq 0$  y  $b > 1$ , se cumple la siguiente identidad:

$$H(x + z + b) = H(y + z + b) = 1 \quad (2)$$

Sustituyendo (2) en (1) tenemos:

$$H(x + z + b) \cdot [(x + b) - (z + 1)] \leq H(y + z + b) \cdot [(y + b) - (z + 1)] \quad (3)$$

Sin embargo, por la definición de  $L$ , el lado izquierdo de (3) corresponde a  $L(x, z)$ , mientras que el lado derecho, es  $L(y, z)$ , por lo tanto, se cumple que:  $L(x, z) \leq L(y, z)$

q.e.d.

### 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 se cumple la siguiente desigualdad:  $L(z, x) \geq L(z, y)$ .

Demostración:

Por hipótesis, se cumple que  $x \leq y$ , con  $0 \leq x \leq b - 1$  y  $0 \leq y \leq b - 1$ , al realizar operaciones elementales en ambos lados del signo de desigualdad se tiene:

$$x \leq y$$

$$-1(x \leq y)$$

$$-x \geq -y$$

Sumando  $z$  en ambos lados de la desigualdad:

$$z - x \geq z - y$$

$$z - x + b - 1 \geq z - y + b - 1$$

Agrupando términos:

$$z + b - z - 1 \geq z + b - y - 1 \quad (1)$$

$$(z + b) - (z + 1) \geq (z + b) - (y + 1)$$

$$(1) \cdot [(z + b) - (z + 1)] \geq (1) \cdot [(z + b) - (y + 1)]$$

Por definición de H y dado que se cumple que  $x \geq 0, y \geq 0, z \geq 0$  y  $b > 1$ , se cumple la siguiente identidad:

$$H(z + x + b) = H(z + y + b) = 1 \quad (2)$$

Sustituyendo (2) en (1):

$$H(z + x + b) \cdot [(z + b) - (x + 1)] \geq H(z + y + b) \cdot [(z + b) - (y + 1)] \quad (3)$$

Sin embargo, por la definición de  $L$ , el lado izquierdo de (3) corresponde a  $L(z, x)$ , mientras que el lado derecho, es  $L(z, y)$ , por lo tanto, se cumple que:  $L(z, x) \geq L(z, y)$

q.e.d

### 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 se cumple que:  $L[(x \vee y), z] = L(x, z) \vee L(y, z)$

Por definición de H se tiene que:

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

Por definición de H y dado que se cumple que  $x \geq 0, y \geq 0, z \geq 0$  y  $b > 1$ , se cumple la siguiente identidad:

$$H((x \vee y) + z + b) = 1$$

Por lo tanto, la expresión inicial se simplifica a lo siguiente:

$$(1) \cdot [((x \vee y) + b) - (z + 1)]$$

Si establecemos que  $u = b - (z + 1)$  y haciendo uso de la propiedad asociativa tenemos:

$$(1) \cdot [(x \vee y) + u] \quad (1)$$

Asumiendo que el operador máximo cumple con la siguiente propiedad distributiva con respecto de la suma:

$$(a \vee b) + c = (a + c) \vee (b + c)$$

Si eso lo aplicamos en (1), tendríamos lo siguiente:

$$(1) \cdot [(x + u) \vee (y + u)]$$

Lo cual se expresa como:

$$(1) \cdot (x + u) \vee (1) \cdot (y + u)$$

Sustituyendo  $u = b - (z + 1)$  y agrupando términos.

$$(1) \cdot ((x + b) - (z + 1)) \vee (1) \cdot ((y + b) - (z + 1))$$

(2)

Ahora bien, sabemos que se cumple la siguiente identidad:

$$H(x + z + b) = H(y + z + b) = 1$$

(3)

Sustituyendo (3) en (2):

$$H(x + z + b) \cdot [(x + b) - (z + 1)] \vee H(y + z + b) \cdot [(y + b) - (z + 1)]$$

(4)

Por definición de  $L$  es posible observar que el lado izquierdo de (4) corresponde con  $L(x, z)$ ; mientras que el lado derecho corresponde con  $L(y, z)$ . Por lo tanto, se cumple que

$$L[(x \vee y), z] = L(x, z) \vee L(y, z)$$

q.e.d

### 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 se cumple que:  $L[(x \wedge y), z] = L(x, z) \wedge L(y, z)$

Esta propiedad se demostrará de manera análoga a la anterior, considerando el operador  $\wedge$  en lugar del operador  $\vee$ .

Por definición de  $H$  se tiene que:

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

Por definición de  $H$  y dado que se cumple que  $x \geq 0, y \geq 0, z \geq 0$  y  $b > 1$ , se cumple la siguiente identidad:

$$H((x \wedge y) + z + b) = 1$$

Por lo tanto, la expresión inicial se simplifica a lo siguiente:

$$(1) \cdot [(x \wedge y) + b] - (z + 1)$$

Si establecemos que  $u = b - (z + 1)$  y haciendo uso de la propiedad asociativa tenemos:

$$(1) \cdot [(x \wedge y) + u] \tag{1}$$

Asumiendo que el operador ínfimo cumple con la siguiente propiedad distributiva con respecto de la suma:

$$(a \wedge b) + c = (a + c) \wedge (b + c)$$

Si eso lo aplicamos en (1), tendríamos lo siguiente:

$$(1) \cdot [(x + u) \wedge (y + u)]$$

Lo cual se expresa como:

$$(1) \cdot (x + u) \wedge (1) \cdot (y + u)$$

Sustituyendo  $u = b - (z + 1)$  y agrupando términos.

$$(1) \cdot ((x + b) - (z + 1)) \wedge (1) \cdot ((y + b) - (z + 1)) \tag{2}$$

Ahora bien, sabemos que se cumple la siguiente identidad:

$$H(x + z + b) = H(y + z + b) = 1 \tag{3}$$

Sustituyendo (3) en (2):

$$H(x + z + b) \cdot [(x + b) - (z + 1)] \wedge H(y + z + b) \cdot [(y + b) - (z + 1)] \tag{4}$$

Por definición de  $L$  es posible observar que el lado izquierdo de (4) corresponde con  $L(x, z)$ ; mientras que el lado derecho corresponde con  $L(y, z)$ . Por lo tanto, se cumple que

$$L[(x \vee y), z] = L(x, z) \wedge L(y, z)$$

q.e.d

Por su naturaleza y utilidad en la fase de clasificación de patrones, la operación  $C$  es mucho más complicada que la operación  $L$ .

## A2. Demostración de las propiedades de la operación $C$

En esta sección presentaremos las demostraciones correspondientes a las propiedades de la operación  $C$ ; es importante destacar que estas propiedades no consideran los casos NI que fueron presentados en la sección 4.3.

Por el momento hay una propiedad de  $C$ , que es de suma utilidad, la cual se enuncia y demuestra a continuación.

### Propiedad $C1$ (modo HI):

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

Demostración:

Por hipótesis y por las definiciones de  $H$  y de  $C$ , se tiene:

$$\begin{aligned} C^I(b, 0) &= H[(b+2) - (b-0)] \cdot H[(2b-1) - (b+0)] \cdot [(b+1) - (b-0)] \\ &= H[b+2-b] \cdot H[2b-1-b] \cdot [b+1-b] = H(2) \cdot H(b-1) \cdot (1) \\ &= (1) \cdot (1) \cdot (1) = 1 \end{aligned}$$

Es decir, ha quedado demostrado que  $C^I(b, 0) = 1$ .

q.e.d.

### Propiedad $C1$ (modo HS):

$$\begin{aligned} C^S(M_{ij}, P_j^k) &= H[(M_{ij}+2) - (b-P_j^k)] \cdot H[(2b-1) - (M_{ij}+P_j^k)] \\ &\quad \cdot [(M_{ij}+1) - (b-P_j^k)] + (H[(b-1) - (M_{ij}+P_j^k)] \\ &\quad + H[(M_{ij}+2) - (2b-P_j^k)]) \cdot (b-1) \end{aligned}$$

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

$$\begin{aligned} C^S(b, 0) &= H[(b+2) - (b-0)] \cdot H[(2b-1) - (b+0)] \cdot [(b+1) - (b-0)] \\ &\quad + (H[(b-1) - (b+0)] + H[(b+2) - (2b-0)]) \cdot (b-1) \\ &= H[b+2-b] \cdot H[2b-1-b] \cdot [b+1-b] \\ &\quad + (H[b-1-b] + H[b+2-2b]) \cdot (b-1) \\ &= H(2) \cdot H(b-1) \cdot (1) + (H(-1) + H(2-b)) \cdot (b-1) \\ &= (1) \cdot (1) \cdot (1) + (0+0) \cdot (b-1) = 1+0 = 1 \end{aligned}$$

Es decir, ha quedado demostrado que  $C^S(b, 0) = 1$ .

q.e.d.

### Propiedad $C2$ (modo HI)

En el modo HI tenemos la siguiente definición:

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

Sustituyendo valores

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

Realizando las operaciones:

$$C^I(b - 1, 0) = H[1] \cdot H[b] \cdot [0]$$

En ambos casos  $H(x)=1$ , entonces:

$$C^I(b - 1, 0) = [0]$$

q.e.d

### Propiedad C2 (modo HS)

Para el modo HS tenemos la siguiente definición:

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

Sustituyendo valores:

$$C^S(b - 1, 0) = H[(b - 1 + 2) - (b - 0)] \cdot H[(2b - 1) - (b - 1)] \cdot [(b - 1 + 1) - (b)] + (H[(b - 1) - (b - 1)] + H[(b - 1 + 2) - (2b)]) \cdot (b - 1)$$

Realizando las operaciones:

$$C^S(b - 1, 0) = H[1] \cdot H[b] \cdot [0] + (H[-2] + H[1 - b]) \cdot (b - 1)$$

Los dos primeros términos  $H(x)$  son iguales a 1, pero los otros dos son iguales a cero, por lo tanto:

$$C^S(b - 1, 0) = (1) \cdot (1) \cdot [0] + (0) \cdot (b - 1) = 0$$

q.e.d

### Propiedad C3 (modo HI)

En el modo HI se tiene que:



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

Sustituyendo valores:

$$C^I(0, b - 1) = H[(0 + 2) - (b - b + 1)] \cdot H[(2b - 1) - (b - 1)] \cdot [(0 + 1) - (b - b + 1)]$$

Realizando las operaciones:

$$C^I(0, b - 1) = H[1] \cdot H[b] \cdot [0]$$

En ambos casos  $H(x)=1$ , por lo tanto:

$$C^I(0, b - 1) = [0]$$

q.e.d.

### **Propiedad C3 (modo HS)**

A partir de la definición de la operación C:

$$C^S(0,0) = H[2 - b] \cdot H[2b - 1] \cdot [1 - b] + (H[(b - 1)] + H[2 - (2b - 0)]) \cdot (b - 1)$$

De acuerdo con las operaciones realizadas en la demostración anterior tenemos que:

$$C^S(0,0) = 0 + (H[(b - 1)] + H[2 - (2b - 0)]) \cdot (b - 1)$$

Al sustituir términos y realizar operaciones tenemos lo siguiente:

$$C^S(0,0) = (H[(b - 1)] + H[2 - 2b]) \cdot (b - 1)$$

El primer término de  $H(x)$  es 1, y el segundo término es igual a 0, por lo tanto:

$$C^S(0,0) = ((1) + (0)) \cdot (b - 1) = b - 1$$

q.e.d.

### **Propiedad C4 (modo HI)**

Para el modo HI partimos de la definición de la operación C:

$$C^I(b - 1, b - 1) = H[(b - 1 + 2) - (b - b + 1)] \cdot H[(2b - 1) - (2b - 2)] \cdot [(b) - (1)]$$

Al realizar las operaciones tenemos que:

$$C^I(b - 1, b - 1) = H[b] \cdot H[1] \cdot [(b) - (1)]$$

Ambos términos  $H(x)$  son iguales a 1, por lo tanto:

$$C^I(b - 1, b - 1) = b - 1$$

q.e.d

### Propiedad C4 (modo HI)

En el caso del modo HS partimos de la definición de C:

$$\begin{aligned} C^S(M_{ij}, P_j^k) &= H[(M_{ij} + 2) - (b - P_j^k)] \cdot H[(2b - 1) - (M_{ij} + P_j^k)] \\ &\quad \cdot [(M_{ij} + 1) - (b - P_j^k)] \\ &\quad + (H[(b - 1) - (M_{ij} + P_j^k)] + H[(M_{ij} + 2) - (2b - P_j^k)]) \cdot (b - 1) \end{aligned}$$

Al sustituir valores tenemos que:

$$\begin{aligned} C^S(b - 1, b - 1) &= H[(b - 1 + 2) - (b - b + 1)] \cdot H[(b - 1 + 2) - (2b - (b - 1))] \\ &\quad \cdot [(b) - (1)] + (H[(b - 1) - (2b - 2)] + H[(b - 1 + 2) - (2b - (2b - 1))]) \\ &\quad \cdot (b - 1) \end{aligned}$$

Al realizar operaciones:

$$C^S(b - 1, b - 1) = H[b] \cdot H[1] \cdot [(b) - (1)] + (H[1 - b] + H[0]) \cdot (b - 1)$$

Los dos primeros términos H(x) son iguales a 1, pero los segundos términos son iguales a 0; por lo tanto:

$$C^S(b - 1, b - 1) = b - 1$$

q.e.d

### Propiedad C5 (modo HI)

En el modo HI tenemos que:

$$x \leq y$$

Le sumamos z:

$$x + z \leq y + z$$

Restamos b-1:

$$x + z - (b - 1) \leq y + z - (b - 1)$$

Acomodando términos:

$$(x + 1) - (b - z) \leq (y + 1) - (b - z)$$

Para asegurarnos que  $x + y \geq b - 1$  y  $x + y < 2b - 1$  (para no tener casos NI), agregamos los términos:

$$\begin{aligned} & H[(x + 2) - (b - z)] \cdot H[(2b - 1) - (x + z)] \cdot [(x + 1) - (b - z)] \\ & \leq H[(y + 2) - (b - z)] \cdot H[(2b - 1) - (y + z)] \cdot [(y + 1) - (b - z)] \end{aligned}$$

Lo cual corresponde con:

$$C^I(x, z) \leq C^I(y, z)$$

q.e.d

### Propiedad C5 (modo HS)

Para el modo HS el planteamiento es el siguiente:

$$x \leq y$$

Le sumamos z:

$$x + z \leq y + z$$

Restamos b-1:

$$x + z - (b - 1) \leq y + z - (b - 1)$$

Acomodando términos:

$$(x + 1) - (b - z) \leq (y + 1) - (b - z)$$

Para asegurarnos que  $x + y \geq b - 1$  y  $x + y < 2b - 1$  (para no tener casos NI), agregamos los términos de manera tal que al obtener un caso NI se devuelva el valor de b-1:

$$\begin{aligned} & H[(x + 2) - (b - z)] \cdot H[(2b - 1) - (x + z)] \cdot [(x + 1) - (b - z)] \\ & \quad + (H[(b - 1) - (x + z)] + H[(x + 2) - (2b - z)]) \cdot (b - 1) \\ & \leq H[(y + 2) - (b - z)] \cdot H[(2b - 1) - (y + z)] \cdot [(y + 1) - (b - z)] \\ & \quad + (H[(b - 1) - (y + z)] + H[(y + 2) - (2b - z)]) \cdot (b - 1) \end{aligned}$$

Lo cual corresponde con la definición de la operación C, por lo tanto:

$$C^S(x, z) \leq C^S(y, z)$$

q.e.d

### A3. Demostración de los teoremas, lemas y corolarios correspondientes con el modo HI del clasificador de Heaviside

**Teorema 4.4.1.** Sea  $b \in \mathbb{Z}$  tal que  $b > 1$ , entonces  $\forall x \in \mathbb{Z}$  tal que  $0 \leq x \leq b - 1$  se cumple  $L(x, x) = b - 1$ .

Demostración:

Por hipótesis y por la definición 4.4.1 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 4.4.1.1:**

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

Demostración:

Dado que se cumple la hipótesis del Teorema 4.4.1, éste se puede aplicar directamente, haciendo  $x=0$ :

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

q.e.d.

#### **Corolario 4.4.1.2:**

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 4.4.1, 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 4.4.1.** La memoria asociativa  $M$  del clasificador de Heaviside en modo operativo HI contiene sólo valores  $b-1$  en su diagonal principal.

Demostración:

De acuerdo con la definición 4.4.2, 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 4.4.2.** 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:  $C^I[L(x, y), y] = x$  para el modo HI del clasificador de Heaviside.

Demostración:

Por definiciones 4.2.1 y 4.3.1:

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

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

Por otro lado:

$$\begin{aligned} H[(L(x, y) + 2) - (b - y)] &= H[(x + b - y - 1 + 2) - (b - y)] \\ &= H[x + b - y - 1 + 2 - b + y] = H[x + 1] = 1 \end{aligned}$$

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

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

Sustituyendo en la expresión de  $C$ :

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

q.e.d.

**Lema 4.4.3.** Sea  $M$  un clasificador de Heaviside en modo operativo HI, y sea  $\mathbf{A}^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

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

Demostración:

Por definición 4.4.1, 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  $C^l$  con  $A_j^d$  en ambos lados de la desigualdad y tomando el máximo con respecto al índice  $j$ :

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

Por nota 2 de la definición 4.4.3, se tiene que si  $\mathbf{P}^d = \mathbf{A}^d$ :

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

Y por transitividad con la expresión previa:

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

Por Lema 4.4.2, 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 4.4.4.** Sea  $M$  la memoria asociativa de un clasificador de Heaviside en modo operativo HI, y sea  $\mathbf{A}^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

$$R_i \geq A_i^d$$

Demostración:

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

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

Pero por Lema 4.4.1, 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 4.4.2:

$$R_i \geq A_i^d$$

q.e.d.

**Teorema 4.4.2.** Una memoria asociativa  $M$  de un clasificador de Heaviside en modo operativo HI, clasifica correctamente cualquier patrón  $\mathbf{A}^d$  del conjunto de aprendizaje  $A$ .

Demostración:

Por Lemas 4.4.3 y 4.4.4, 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  $\mathbf{R}^d = \mathbf{A}^d$ , por lo que el patrón recuperado es precisamente  $\mathbf{A}^d$ , lo cual significa que se le asigna su propia clase; es decir, es clasificado correctamente.

q.e.d

A4. Demostración de los teoremas, lemas y corolarios correspondientes con el modo HS del clasificador de Heaviside

**Teorema 4.5.1.** Sea  $b \in \mathbb{Z}$  tal que  $b > 1$ , entonces  $\forall x \in \mathbb{Z}$  tal que  $0 \leq x \leq b - 1$  se cumple  $L(x, x) = b - 1$ .

Demostración:

Por hipótesis y por la definición 4.5.1 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 4.5.1.1.** Sea  $b \in \mathbb{Z}$  tal que  $b > 1$ . Si  $A_i^k = A_j^k = 0$  entonces  $L(A_i^k, A_j^k) = b - 1$ .

Demostración:

Dado que se cumple la hipótesis del Teorema 4.5.1, éste se puede aplicar directamente, haciendo  $x=0$ :

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

q.e.d.

**Corolario 4.5.1.2.** 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 4.5.1, 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 4.5.1.** Una memoria asociativa  $M$  de clasificador de Heaviside en modo operativo HS contiene sólo valores  $b-1$  en su diagonal principal.

Demostración:

De acuerdo con la definición 4.5.2, las entradas de la diagonal principal de  $M$  se calculan con la siguiente expresión:

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

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  $M_{ii} = \max L(A_i^k, A_i^k) = L(A_i^k, A_i^k) = b - 1$  para todo valor de  $i$ .

q.e.d.

**Lema 4.5.2.** 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:  $C^S[L(x, y), y] = x$  para el modo HS del clasificador de Heaviside.

Demostración:

Por definiciones 4.2.1 y 4.3.1:

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

$$\begin{aligned} C^S[L(x, y), y] &= H[(L(x, y) + 2) - (b - y)] \cdot H[(2b - 1) - (L(x, y) + y)] \\ &\quad \cdot [(L(x, y) + 1) - (b - y)] \\ &\quad + (H[(b - 1) - (L(x, y) + y)] + H[(L(x, y) - 2) - (2b - y)]) \cdot (b - 1) \end{aligned}$$

$$\begin{aligned} C^S(M_{ij}, P_j^k) &= H[(M_{ij} + 2) - (b - P_j^k)] \cdot H[(2b - 1) - (M_{ij} + P_j^k)] \\ &\quad \cdot [(M_{ij} + 1) - (b - P_j^k)] + (H[(b - 1) - (M_{ij} + P_j^k)] \\ &\quad + H[(M_{ij} + 2) - (2b - P_j^k)]) \cdot (b - 1) \end{aligned}$$

Por otro lado:

$$\begin{aligned} H[(L(x, y) + 2) - (b - y)] &= H[(x + b - y - 1 + 2) - (b - y)] \\ &= H[x + b - y - 1 + 2 - b + y] = H[x + 1] = 1 \end{aligned}$$

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

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

$$\begin{aligned} H[(b - 1) - (L(x, y) + y)] &= H[b - 1 - ((x + b - y + 1) + y)] \\ &= H[b - 1 - x - b + y - 1 - y] = H[-2 - x] = 0 \end{aligned}$$

$$\begin{aligned} H[(L(x, y) - 2) - (2b - y)] &= H[((x + b - y - 1) - 2) - (2b - y)] \\ &= H[x + b - y - 1 - 2 - 2b + y] = H[x - b - 3] = H[x - (b + 3)] = 0 \end{aligned}$$

Debido a que como  $x$  vale desde 0 hasta  $b-1$  entonces:  $H[b - 1 - b - 3] = H[-4] = 0$

Sustituyendo en la expresión de  $C$ :

$$C^S[L(x, y), y] = (1) \cdot (1) \cdot (x) + (0 + 0) \cdot (b - 1) = x + (0 + 0) \cdot (b - 1) = x$$

q.e.d.

**Lema 4.5.3.** Sea  $M$  la memoria asociativa de un Clasificador de Heaviside en modo operativo HS, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

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

Demostración:

Por definición 4.5.1, se tiene que:

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

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

$$\max L(A_i^k, A_j^k) \geq L(A_i^d, A_j^d)$$

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

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

Por nota 2 de la definición 4.5.3, se tiene que si  $P^d = A^d$ :

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

Y por transitividad con la expresión previa:

$$R_i \geq \min C^S[L(A_i^d, A_j^d), A_j^d]$$

Por Lema 4.5.2, se tiene el resultado, porque min es con respecto al índice  $j$ , no al  $i$ :

$$R_i \geq A_i^d$$

q.e.d.

**Lema 4.5.4.** Sea  $M$  la memoria asociativa de clasificador de Heaviside en modo operativo HS, y sea  $A^d$  un patrón del conjunto de aprendizaje  $A$ , entonces:

$$R_i \leq A_i^d$$

Demostración:

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

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

Pero por Lema 4.5.1, para cualquier valor de  $k$  (incluyendo  $d$ ) se tiene:

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

Por lo que:

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

Y finalmente, por Lema 4.4.2:

$$R_i \leq A_i^d$$

q.e.d.

**Teorema 4.5.2.-** La memoria asociativa  $M$  del clasificador de Heaviside en modo operativo HI, clasifica correctamente cualquier patrón  $A^d$  del conjunto de aprendizaje  $A$ .

Demostración:

Por Lemas 4.5.3 y 4.5.4, para un valor de índice de patrón fijo  $d$ , se cumplen simultáneamente estas dos desigualdades:  $R_i \geq A_i^d$  y  $R_i \leq A_i^d$  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.

## A5. Planteamiento de una expresión alterna de la operación C

Durante la realización de este trabajo de tesis surgió la cuestión sobre si era posible unificar la operación del clasificador de forma que se tuviera una única para los modos HI y HS; de esta cuestión se plantearon algunas opciones y tras un análisis teórico se determinó que la siguiente operación podría usarse en ambos modos del clasificador de Heaviside.

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

Como puede observarse, esta operación es muy similar a la versión de la operación C para el modo HS del clasificador, aunque con la diferencia de que en la segunda parte de la misma se excluye la condición que evaluaba la primera condición para tener un caso NI, es decir si  $M_{ij} + P_j < b - 1$ , con esta simple modificación tenemos una formulación de la operación C que puede emplearse en ambos modos del clasificador de Heaviside.

Considérese la siguiente tabla donde se presenta el caso para  $b=3$

x	y	$C(x, y)$
0	0	
0	1	
0	2	0
1	0	
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1
3	1	2
3	2	
4	0	2
4	1	

4	2	
---	---	--

Los valores resaltados en gris corresponden con los casos NI. Ahora bien, para el modo HI del clasificador de Heaviside los casos NI quedarán evaluados de la siguiente forma:

x	y	$C(x,y)$
0	0	0
0	1	0
0	2	0
1	0	0
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1
3	1	2
3	2	0
4	0	2
4	1	0
4	2	0

Mientras que para el modo HS la operación C queda de la siguiente manera.

x	y	$C(x,y)$
0	0	2
0	1	2
0	2	0

1	0	2
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1
3	1	2
3	2	2
4	0	2
4	1	2
4	2	2

Finalmente, si aplicamos la formulación propuesta para ambos modos del clasificador tenemos lo siguiente:

x	y	$C(x, y)$
0	0	0
0	1	0
0	2	0
1	0	0
1	1	0
1	2	1
2	0	0
2	1	1
2	2	2
3	0	1

3	1	2
3	2	2
4	0	2
4	1	2
4	2	2

Como puede notarse, los valores resaltados en gris claro corresponden con lo que es la operación C en el modo HI, mientras que los valores resaltados en gris oscuro corresponden con la formulación de la operación C para el modo HS, de esta forma en una única operación se obtiene un comportamiento similar a lo planteado en las operaciones para los modos HI y HS.

¿Cómo se planteó esta expresión alterna?

La idea básica es dividir el resultado de la operación C para los casos NI. Por un lado, para los casos en los que se cumple la condición  $M_{ij} + P_j < b - 1$  se les asigna el valor de 0; por otra parte, aquellos casos en los que se cumpla la condición  $M_{ij} + P_j \geq 2b - 1$  recibirán el valor de b-1. Para lograr esta asignación, fue necesario conservar casi todas las partes de la expresión original de C para el modo HS:

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

Con la única excepción de la expresión  $H[(b - 1) - (M_{ij} + P_j^k)]$ , ya que esta es la que causaría que todos los valores que caen en la condición  $M_{ij} + P_j \geq 2b - 1$  recibieran el valor de b-1 en la operación C original. En un inicio también se consideró modificar la primera parte de la expresión original, es decir  $H[(M_{ij} + 2) - (b - P_j^k)] \cdot H[(2b - 1) - (M_{ij} + P_j^k)] \cdot [(M_{ij} + 1) - (b - P_j^k)]$ ; sin embargo, se descubrió que cualquier modificación alteraría el resultado original y de hecho podría implicar en la generación de resultados fuera del dominio de la operación L.

Pruebas de la expresión alterna

La validez de la expresión alterna fue comprobada de dos maneras: teórica y empírica. La comprobación empírica consistió en demostrar la validez de la propiedad C1; dicha comprobación se presenta a continuación.

**Propiedad C1:**

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



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

$$\begin{aligned}
 C(b, 0) &= H[(b + 2) - (b - 0)] \cdot H[(2b - 1) - (b + 0)] \cdot [(b + 1) - (b - 0)] \\
 &\quad + H[(b + 2) - (2b - 0)] \cdot (b - 1) \\
 &= H[b + 2 - b] \cdot H[2b - 1 - b] \cdot [b + 1 - b] + H[b + 2 - 2b] \cdot (b - 1) \\
 &= H(2) \cdot H(b - 1) \cdot (1) + H(2 - b) \cdot (b - 1) = (1) \cdot (1) \cdot (1) + (0) \cdot (b - 1) \\
 &= 1 + 0 = 1
 \end{aligned}$$

Es decir, ha quedado demostrado que  $C(b, 0) = 1$ .

q.e.d.

Una vez demostrada la equivalencia de la expresión alterna con la expresión original respecto a la propiedad C1 procedimos a realizar un análisis experimental donde comparamos el rendimiento del clasificador de Heaviside en sus modos HI y HS originales en contra del clasificador de Heaviside en sus modos HI y HS con la nueva expresión. Esta comparación fue hecha con algunos de los bancos de datos que pertenecen a repositorios famosos como UCI o KEEL y el método de validación elegido fue el *10-Fold Cross-Validation*. Los resultados obtenidos se presentan en la siguiente tabla.

**Tabla A 1. Comparación entre la operación C original y la expresión alterna para ambos modos del Clasificador de Heaviside.**

Dataset	Modo del Clasificador de Heaviside			
	HI Original	HS Original	HI Nuevo	HS Nuevo
Haberman	68.301	67.974	69.608	69.935
Banana	73.642	73.925	73.868	73.736
Titanic	78.101	76.238	78.237	63.971
Phoneme	90.063	90.026	90.063	90.026
Appendicitis	82.075	80.189	82.075	80.189
Iris	96.000	96.000	96.000	95.333
Heart	65.185	65.185	65.185	65.185
Magic	78.086	78.081	78.086	78.081
Australian	60.580	60.435	60.580	60.435
Ring	86.365	86.378	86.365	86.378
Ionosphere	89.459	87.749	89.459	87.749
Glass	98.131	98.598	98.131	98.598

Los resultados resaltados son aquellos en los que el clasificador tuvo un desempeño superior, como puede verse únicamente hay dos casos en los que la nueva formulación ofrece mejores resultados y por un margen muy estrecho. Sin embargo, la forma más adecuada de determinar si esta nueva expresión impacta en el rendimiento del clasificador es haciendo una

prueba de significancia estadística; al igual que en otros casos la prueba elegida es la prueba de Wilcoxon, cuyos resultados se presentan a continuación.

**Tabla A 2. Resultados de la prueba de Wilcoxon a los resultados de las dos versiones de la operación C.**

	HIModificado - HIN	HSModificado - HSN
Sig. asintótica (bilateral)	0.109	0.465

Para el caso de la comparación entre los resultados de la operación original del modo HI con respecto de los resultados de la expresión alterna el valor de la significancia asintótica es de 0.109, por lo que con un 95% de confianza NO es posible rechazar la hipótesis nula de igualdad de los resultados obtenidos. De igual forma, para el caso del modo HS la significancia es de 0.465, por lo que tampoco se rechaza la hipótesis nula igualdad de las muestras comparadas.

A partir de los resultados obtenidos anteriormente hemos decidido emplear la versión original de la operación C, ya que esta expresión alterna no modifica de forma contundente el rendimiento del clasificador de Heaviside.

## A6. Desempeños obtenidos en las pruebas intensivas del Clasificador de Heaviside

### Banco de datos Iris Plant

<b>Base</b>	<b>Desempeño</b>
1	95.833
2	96
3	95.867
4	96.033
5	95.867
6	95.9
7	95.767
8	95.9
9	95.967
10	95.933
11	95.967
12	95.933
13	96
14	95.9
15	95.933
16	95.967
17	95.867
18	95.933
19	95.967
20	95.933
21	95.867
22	95.867
23	95.867
24	95.9
25	95.9
26	95.833
27	95.9
28	95.867
29	95.867
30	95.9
31	95.833
32	96
33	95.933
34	95.867
35	95.9
36	95.833

<b>37</b>	95.933
<b>38</b>	95.9
<b>39</b>	95.867
<b>40</b>	95.867
<b>41</b>	95.9
<b>42</b>	95.933
<b>43</b>	96
<b>44</b>	95.9
<b>45</b>	95.933
<b>46</b>	95.933
<b>47</b>	95.967
<b>48</b>	95.9
<b>49</b>	95.867
<b>50</b>	95.933
<b>51</b>	95.8
<b>52</b>	95.7
<b>53</b>	95.867
<b>54</b>	95.9
<b>55</b>	95.9
<b>56</b>	95.9
<b>57</b>	96
<b>58</b>	95.967
<b>59</b>	95.867
<b>60</b>	95.867
<b>61</b>	95.933
<b>62</b>	95.967
<b>63</b>	95.8
<b>64</b>	95.833
<b>65</b>	95.767
<b>66</b>	95.967
<b>67</b>	95.967
<b>68</b>	95.9
<b>69</b>	95.933
<b>70</b>	95.933
<b>71</b>	95.9
<b>72</b>	95.967
<b>73</b>	95.933
<b>74</b>	95.833
<b>75</b>	95.9
<b>76</b>	95.867
<b>77</b>	95.933

<b>78</b>	95.867
<b>79</b>	95.767
<b>80</b>	95.9

Banco de datos Haberman

<b>Base</b>	<b>Desempeño</b>
<b>1</b>	70.963
<b>2</b>	70.557
<b>3</b>	70.716
<b>4</b>	70.766
<b>5</b>	71.21
<b>6</b>	70.938
<b>7</b>	71.054
<b>8</b>	70.962
<b>9</b>	71.036
<b>10</b>	71.029
<b>11</b>	70.287
<b>12</b>	70.738
<b>13</b>	70.819
<b>14</b>	70.744
<b>15</b>	70.671
<b>16</b>	71.13
<b>17</b>	70.993
<b>18</b>	70.96
<b>19</b>	70.623
<b>20</b>	70.953
<b>21</b>	70.696
<b>22</b>	70.793
<b>23</b>	70.919
<b>24</b>	70.9
<b>25</b>	71.329
<b>26</b>	71.057
<b>27</b>	70.823
<b>28</b>	70.796
<b>29</b>	71.087
<b>30</b>	70.634
<b>31</b>	71.168
<b>32</b>	71.077
<b>33</b>	70.754
<b>34</b>	71.17

<b>35</b>	70.809
<b>36</b>	71.142
<b>37</b>	70.87
<b>38</b>	71.205
<b>39</b>	70.287
<b>40</b>	70.791
<b>41</b>	70.882
<b>42</b>	70.869
<b>43</b>	71.162
<b>44</b>	70.823
<b>45</b>	70.293
<b>46</b>	70.903
<b>47</b>	70.47
<b>48</b>	70.585
<b>49</b>	70.762
<b>50</b>	71.137
<b>51</b>	71.041
<b>52</b>	70.789
<b>53</b>	70.515
<b>54</b>	70.678
<b>55</b>	70.96
<b>56</b>	70.71
<b>57</b>	70.533
<b>58</b>	70.958
<b>59</b>	70.554
<b>60</b>	71.191
<b>61</b>	70.697
<b>62</b>	70.99
<b>63</b>	70.884
<b>64</b>	70.972
<b>65</b>	71.173
<b>66</b>	70.741
<b>67</b>	70.915
<b>68</b>	70.713
<b>69</b>	71.218
<b>70</b>	70.935
<b>71</b>	71.467
<b>72</b>	70.868
<b>73</b>	71.079
<b>74</b>	70.949
<b>75</b>	71.024

<b>76</b>	70.881
<b>77</b>	70.848
<b>78</b>	70.901
<b>79</b>	70.973
<b>80</b>	70.938
<b>81</b>	71.142
<b>82</b>	70.799
<b>83</b>	70.737
<b>84</b>	70.69

A7. Primera página del artículo publicado en la revista Nova Scientia

García Floriano, A. et al.



*Revista Electrónica Nova Scientia*

**Clasificador de Heaviside  
Heaviside's Classifier**

**Andrés García Floriano<sup>1</sup>, Oscar Camacho Nieto<sup>2</sup> y Cornelio  
Yáñez Márquez<sup>3</sup>**

---

<sup>1</sup>Doctorado, Centro de Investigación en Computación del IPN, Ciudad de México

<sup>2</sup>Centro de Innovación y Desarrollo Tecnológico en Cómputo del IPN

<sup>3</sup>Laboratorio, Centro de Investigación en Computación del IPN

---

México

Cornelio Yáñez Márquez. E mail: [cyanez@cic.ipn.mx](mailto:cyanez@cic.ipn.mx)

© Universidad De La Salle Bajío (México)



## A8. Primera página del artículo publicado en la revista IRRODL

International Review of Research in Open and Distributed Learning  
Volume 18, Number 1

February – 2017

### Social Web Content Enhancement in a Distance Learning Environment: Intelligent Metadata Generation for Resources



Andrés García-Floriano<sup>1</sup>, Angel Ferreira-Santiago<sup>1</sup>, Cornelio Yáñez-Márquez<sup>1</sup>, Oscar Camacho-Nieto<sup>2</sup>, Mario Aldape-Pérez<sup>2</sup>, and Yenny Villuendas-Rey<sup>2</sup>

<sup>1</sup>Centro de Investigación en Computación, Instituto Politécnico Nacional, México, <sup>2</sup>Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, México

#### Abstract

Social networking potentially offers improved distance learning environments by enabling the exchange of resources between learners. The existence of properly classified content results in an enhanced distance learning experience in which appropriate materials can be retrieved efficiently; however, for this to happen, metadata needs to be present. As manual metadata generation is time-costly and often eschewed by the authors of the social web resources, automatic generation is a fertile area for research as several kinds of metadata, such as author or topic, can be generated or extracted from the contents of a document. In this paper we propose a novel metadata generation system aimed at automatically tagging distance learning resources. This system is based on a recently-created intelligent pattern classifier; specifically, it trains on a corpus of example documents and then predicts the topic of a new document based on its text content. Metadata is generated in order to achieve a better integration of the web resources with the social networks. Experimental results for a two-class problem are promising and encourage research geared towards applying this method to multiple topics.

*Keywords:* social networking, distance learning, social web content, metadata generation, intelligent classification

# Referencias

- [1] C. Cobb and H. Goldwhite, *Creations of Fire: Chemistry's Lively History from Alchemy to the Atomic Age*, Springer Science & Business Media, 2013.
- [2] J.P. Marques de Sá, *Pattern Recognition Concepts, Methods and Applications*, firstSpringer Verlag Berlin Heidelberg, Berlin, 2001.
- [3] S. Theodoridis and K. Koutroubas, *Pattern Recognition*, Elsevier, San Diego CA, 2009.
- [4] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 2<sup>nd</sup> Edition. John Wiley & Sons, Hoboken, NJ, 2014.
- [5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, USA, 2009.
- [6] Z. Pawlak, *Rough sets*, International Journal of Computer and Information Sciences 11 (1982), pp. 341–356.
- [7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning representations by back-propagating errors*, Nature 323 (1986), pp. 533–536.
- [8] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, USA, 1992.
- [9] G.E. Hinton, S. Osindero and Y.-W. Teh, *A Fast Learning Algorithm for Deep Belief Nets*, Neural Computation 18 (2006), pp. 1527–1554.
- [10] D. Angluin, *Computational Learning Theory: Survey and Selected Bibliography*, in Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing, 1992, pp. 351–369.
- [11] A.W. Hübler, G.C. Foster and K.C. Phelps, *Managing chaos: Thinking out of the box*, Complexity 12 (2007), pp. 10–13.
- [12] A. Konar, *Computational Intelligence: Principles, Techniques and Applications*, 1st Edition. Springer Berlin Heidelberg, Berlin, 2005.
- [13] J.C. Bezdek, *What is Computational Intelligence?*, in *Computational Intelligence Imitating Life*, IEEE Press, NY, 1994, pp. 1–12.
- [14] N.N. Murty and D. Susheela, *Pattern Recognition - An Algorithmic Approach*, firstSpringer, India, 2011.
- [15] J.L. Díaz de León Santiago, F.A. Sánchez Garfias and C. Yáñez Márquez, *Reconocimiento de Patrones: Enfoque Probabilístico-Estadístico*, Verde 83, Centro de Investigación en Computación del IPN, México, D.F., 2003.
- [16] R.O. Duda, P.E. Hart and D.E. Stork, *Pattern Classification*, 2nd Edition. Wiley, USA, 2001.
- [17] S. Raudys, *Statistical and Neural Classifiers - An Integrated Approach to Design*, 1st Edition. Advances in Computer Vision and Pattern Recognition. Springer Berlin Heidelberg, Berlin, 2001.
- [18] T. Cover and P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory 13 (1967), pp. 21–27.

- [19] L. Breiman, J. Friedman and C.J. Stone, *Classification and Regression Trees*, Chapman & Hall, 1983.
- [20] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [21] L. Breiman, *Random Forests*, *Machine Learning* 45 (2001), pp. 5–32.
- [22] J.J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, *PNAS* 79 (1982), pp. 2554–2558.
- [23] B.E. Boser, I.M. Guyon and V.N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992, pp. 144–152.
- [24] C. Cortes and V. Vapnik, *Support-vector networks*, *Machine Learning* 20 (1995), pp. 273–297.
- [25] G.X. Ritter, P. Sussner and J.L. Diza-de-Leon, *Morphological associative memories.*, *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 9 (1998), pp. 281–293.
- [26] C. Yáñez Márquez, *Memorias Asociativas Basadas en Relaciones de Orden y en operaciones binarias*, *Computación y Sistemas* 6 (2003), pp. 300–311.
- [27] T.S. Blyth, *Lattices and Ordered Algebraic Structures*, 1st edition. Springer, London, 2005.
- [28] S. Roman, *Lattices and Ordered Sets*, Springer, 2009.
- [29] D. Michie, D.J. Spiegelhalter, C.C. Taylor and J. Campbell, eds., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Upper Saddle River, NJ, USA, 1994.
- [30] P. Hart, *The condensed nearest neighbor rule*, *IEEE Transactions on Information Theory* 14 (1968), pp. 515–516.
- [31] V. Pestov, *Is the -NN classifier in high dimensions affected by the curse of dimensionality?*, *Computers & Mathematics with Applications* 65 (2013), pp. 1427–1437.
- [32] R. Santiago Montero, *Clasificador híbrido de patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen*, Tesis de Maestría en Ciencias de la Computación, Centro de Investigación en Computación, 2003.
- [33] A.J. Argüelles Cruz, *Redes Neuronales Alfa-Beta sin pesos: Teoría y factibilidad de implementación*, Tesis de Doctorado en Ciencias de la Computación, Centro de Investigación en Computación, 2007.
- [34] L.O. López Leyva, *Máquinas asociativas alfa-beta con soporte vectorial*, Tesis de Doctorado en Ciencias de la Computación, Centro de Investigación en Computación, 2008.
- [35] J.E. Velázquez-Cruz, I. López-Yáñez, A.J. Argüelles-Cruz and C. Yáñez-Márquez, *Risk Detection of Malignant Tumors in Mammograms using Unconventional Computing*, *Research in Computing Science* 78 (2014), pp. 55–66.
- [36] I. López Yáñez, *Teoría y aplicaciones del clasificador asociativo Gamma*, Centro de Investigación en Computación, 2011.
- [37] I. López-Yáñez, L. Sheremetov and C. Yáñez-Márquez, *Associative Model for the Forecasting of Time Series Based on the Gamma Classifier*, in *Pattern Recognition*, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, J.S. Rodríguez and G.S. di Baja, eds., Springer Berlin Heidelberg, 2013, pp. 304–313.

- [38] I. López-Yáñez, L. Sheremetov and C. Yáñez-Márquez, *A novel associative model for time series data mining*, Pattern Recognition Letters 41 (2014), pp. 23–33.
- [39] A. Ramirez, I. Lopez, Y. Villuendas and C. Yanez, *Evolutionary Improvement of Parameters in an Associative Classifier*, IEEE Latin America Transactions 13 (2015), pp. 1550–1555.
- [40] C. Fócil Arias, A.J. Argüelles Cruz and I. López Yáñez, *Unconventional Computing for Estimating Academic Performance*, Technological Trends in Computing 78 (2014), pp. 89–98.
- [41] C. Yáñez Márquez, M.E. Acevedo Mosqueda and L.O. López Leyva, *BAM Alfa Beta*, CIDETEC IPN, 2007.
- [42] M.E. Acevedo-Mosqueda, C. Yáñez-Márquez and I. López-Yáñez, *Alpha–Beta Bidirectional Associative Memories: Theory and Applications*, Neural Process. Lett. 26 (2007), pp. 1–40.
- [43] F.A. Sánchez Garfias, *Modelos Alfa-Beta difusos*, Tesis de Doctorado en Ciencias de la Computación, Centro de Investigación en Computación, 2009.
- [44] P. Sussner and M.E. Valle, *Implicative Fuzzy Associative Memories*, IEEE Transactions on Fuzzy Systems 14 (2006), pp. 793–807.
- [45] C. Yáñez Márquez, J.L. Díaz de León Santiago and F.A. Sánchez Garfias, *Reconocimiento de Patrones. Enfoque Sintáctico-Estructural*, IT 84, Centro de Investigación en Computación del IPN, México, D.F., 2003.
- [46] C. De Stefano, G. Folino, F. Fontanella and A. Scotto di Freca, *Using Bayesian networks for selecting classifiers in GP ensembles*, Information Sciences 258 (2014), pp. 200–216.
- [47] L. Enrique Sucar, C. Bielza, E.F. Morales, P. Hernandez-Leal, J.H. Zaragoza and P. Larrañaga, *Multi-label classification with Bayesian network-based chain classifiers*, Pattern Recognition Letters 41 (2014), pp. 14–22.
- [48] Y.-S. Chang, C.-T. Fan, W.-T. Lo, W.-C. Hung and S.-M. Yuan, *Mobile cloud-based depression diagnosis using an ontology and a Bayesian network*, Future Generation Computer Systems 43–44 (2015), pp. 87–98.
- [49] J.J. Dabrowski and J.P. de Villiers, *Maritime piracy situation modelling with dynamic Bayesian networks*, Information Fusion 23 (2015), pp. 116–130.
- [50] J. Arias, J.A. Gamez and J.M. Puerta, *Learning distributed discrete Bayesian Network Classifiers under MapReduce with Apache Spark*, Knowledge-Based Systems 117 (2017), pp. 16–26.
- [51] M.J. Flores, J.A. Gámez and A.M. Martínez, *Domains of competence of the semi-naive Bayesian network classifiers*, Information Sciences 260 (2014), pp. 120–148.
- [52] Y.-L. He, R. Wang, S. Kwong and X.-Z. Wang, *Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis*, Information Sciences 259 (2014), pp. 252–268.
- [53] L. Zhu, H. Jin, R. Zheng and X. Feng, *Effective naive Bayes nearest neighbor based image classification on GPU*, J Supercomput 68 (2013), pp. 820–848.
- [54] S.M. Darwish, *Combining firefly algorithm and Bayesian classifier: new direction for automatic multilabel image annotation*, IET Image Processing 10 (2016), pp. 763–772.

- [55] X. Liu, R. Lu, J. Ma, L. Chen and B. Qin, *Privacy-Preserving Patient-Centric Clinical Decision Support System on Naïve Bayesian Classification*, IEEE Journal of Biomedical and Health Informatics 20 (2016), pp. 655–668.
- [56] I. Ahmed, R. Ali, D. Guan, Y.-K. Lee, S. Lee and T. Chung, *Semi-supervised learning using frequent itemset and ensemble learning for SMS classification*, Expert Systems with Applications 42 (2015), pp. 1065–1073.
- [57] B. Tang, S. Kay and H. He, *Toward Optimal Feature Selection in Naive Bayes for Text Categorization*, IEEE Transactions on Knowledge and Data Engineering 28 (2016), pp. 2508–2521.
- [58] A. Bala and I. Chana, *Intelligent failure prediction models for scientific workflows*, Expert Systems with Applications 42 (2015), pp. 980–989.
- [59] J.A. Fernandes, X. Irigoien, J.A. Lozano, I. Inza, N. Goikoetxea and A. Pérez, *Evaluating machine-learning techniques for recruitment forecasting of seven North East Atlantic fish species*, Ecological Informatics 25 (2015), pp. 35–42.
- [60] M. Miao, H. Zeng, A. Wang, C. Zhao and F. Liu, *Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery EEG: An sparse regression and Weighted Naïve Bayesian Classifier-based approach*, Journal of Neuroscience Methods 278 (2017), pp. 13–24.
- [61] M. Behnam and H. Pourghassem, *Real-time seizure prediction using RLS filtering and interpolated histogram feature based on hybrid optimization algorithm of Bayesian classifier and Hunting search*, Computer Methods and Programs in Biomedicine 132 (2016), pp. 115–136.
- [62] R. Ricci and A. Balleri, *Recognition of humans based on radar micro-Doppler shape spectrum features*, Sonar Navigation IET Radar 9 (2015), pp. 1216–1223.
- [63] J. Alonso-Montesinos, M. Martínez-Durbán, J. del Sagrado, I.M. del Águila and F.J. Batlles, *The application of Bayesian network classifiers to cloud classification in satellite images*, Renewable Energy 97 (2016), pp. 155–161.
- [64] H. Cheng, H. Chen and Y. Liu, *Topological Indoor Localization and Navigation for Autonomous Mobile Robot*, IEEE Transactions on Automation Science and Engineering 12 (2015), pp. 729–738.
- [65] F. Rubio, J. Martínez-Gómez, M. Julia Flores and J.M. Puerta, *Comparison between Bayesian network classifiers and SVMs for semantic localization*, Expert Systems with Applications 64 (2016), pp. 434–443.
- [66] Z. Cao, X. Liu, J. Hao and M. Liu, *Simultaneous Prediction for Multiple Key Performance Indicators in Semiconductor Wafer Fabrication*, Chinese Journal of Electronics 25 (2016), pp. 1159–1165.
- [67] S. Sáez-Atienzar, J. Martínez-Gómez, J.I. Alonso-Barba, J.M. Puerta, M.F. Galindo, J. Jordán et al., *Automatic quantification of the subcellular localization of chimeric GFP protein supported by a two-level Naive Bayes classifier*, Expert Systems with Applications 42 (2015), pp. 1531–1537.
- [68] J. Wu, S. Pan, X. Zhu, Z. Cai, P. Zhang and C. Zhang, *Self-adaptive attribute weighting for Naive Bayes classification*, Expert Systems with Applications 42 (2015), pp. 1487–1502.

- [69] C. Heil, *A Short Introduction to Metric, Banach, and Hilbert Spaces*, 2014.
- [70] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutmann and I. Witten, *The WEKA Data Mining Software: An Update*, SIGKDD Explorations 11 (2009).
- [71] D.W. Aha, D. Kibler and M.K. Albert, *Instance-based learning algorithms*, Mach Learn 6 (1991), pp. 37–66.
- [72] Z. Pan, Y. Wang and W. Ku, *A new k-harmonic nearest neighbor classifier based on the multi-local means*, Expert Systems with Applications 67 (2017), pp. 115–125.
- [73] R. Meo, D. Bachar and D. Ienco, *LODE: A distance-based classifier built on ensembles of positive and negative observations*, Pattern Recognition 45 (2012), pp. 1409–1425.
- [74] Y. Jamshidi and H. Nezamabadi-pour, *Rule inducing by fuzzy lattice reasoning classifier based on metric distances (FLRC-MD)*, Applied Soft Computing 24 (2014), pp. 603–611.
- [75] D.R. Amancio, C.H. Comin, D. Casanova, G. Travieso, O.M. Bruno, F.A. Rodrigues et al., *A Systematic Comparison of Supervised Classifiers*, PLoS ONE 9 (2014), pp. e94137.
- [76] X. Fang, Y. Lu, Z. Li, L. Yu and Y. Chen, *Kernel representation-based nearest neighbor classifier*, Optik - International Journal for Light and Electron Optics 125 (2014), pp. 2320–2326.
- [77] F. Li, J. Wang, B. Tang and D. Tian, *Life grade recognition method based on supervised uncorrelated orthogonal locality preserving projection and K-nearest neighbor classifier*, Neurocomputing 138 (2014), pp. 271–282.
- [78] X. Tang and A. Xu, *Multi-class classification using kernel density estimation on K-nearest neighbours*, Electronics Letters 52 (2016), pp. 600–602.
- [79] Y. Lin, J. Li, M. Lin and J. Chen, *A new nearest neighbor classifier via fusing neighborhood information*, Neurocomputing 143 (2014), pp. 164–169.
- [80] Y. Mitani and Y. Hamamoto, *A local mean-based nonparametric classifier*, Pattern Recognition Letters 27 (2006), pp. 1151–1159.
- [81] J.A. Sáez, J. Derrac, J. Luengo and F. Herrera, *Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers*, Pattern Recognition 47 (2014), pp. 3941–3948.
- [82] N. Salari, S. Shohaimi, F. Najafi, M. Nallappan and I. Karishnarajah, *A Novel Hybrid Classification Model of Genetic Algorithms, Modified k-Nearest Neighbor and Developed Backpropagation Neural Network*, PLoS ONE 9 (2014), pp. e112987.
- [83] J. Chen, Y. Yan Tang, C.L. Philip Chen, B. Fang, Z. Shang and Y. Lin, *NNMap: A method to construct a good embedding for nearest neighbor classification*, Neurocomputing 152 (2015), pp. 97–108.
- [84] L. Jiao, Q. Pan and X. Feng, *Multi-hypothesis nearest-neighbor classifier based on class-conditional weighted distance metric*, Neurocomputing 151, Part 3 (2015), pp. 1468–1476.
- [85] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari and M. Cheriet, *KRNN: k Rare-class Nearest Neighbour classification*, Pattern Recognition 62 (2017), pp. 33–44.
- [86] H. Shu, R. Yu, W. Jiang and W. Yang, *Efficient Implementation of  $k$ -Nearest Neighbor Classifier Using Vote Count Circuit*, IEEE Transactions on Circuits and Systems II: Express Briefs 61 (2014), pp. 448–452.
- [87] I.H. Witten, E. Frank and M.A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, 3rd Edition. Morgan Kaufman Publishers, Burlington MA, USA, 2011.

- [88] J.R. Quinlan, *Improved use of continuous attributes in C4. 5*, Journal of artificial intelligence research 4 (1996), pp. 77–90.
- [89] T.K. Ho, Random decision forests, in Proceedings of 3rd International Conference on Document Analysis and Recognition, 1 (1995), pp. 278–282 vol.1.
- [90] L. Breiman, *Bagging Predictors*, Machine Learning 24 (1996), pp. 123–140.
- [91] A. Kale and N. Auti, *Automated Menu Planning Algorithm for Children: Food Recommendation by Dietary Management System using ID3 for Indian Food Database*, Procedia Computer Science 50 (2015), pp. 197–202.
- [92] C.J. Mantas, J. Abellán and J.G. Castellano, *Analysis of Credal-C4.5 for classification in noisy domains*, Expert Systems with Applications 61 (2016), pp. 314–326.
- [93] C.J. Mantas and J. Abellán, *Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data*, Expert Systems with Applications 41 (2014), pp. 4625–4637.
- [94] I.S. Saeh, M.W. Mustafa, Y.S. Mohammed and M. Almaktar, *Static Security classification and Evaluation classifier design in electric power grid with presence of PV power plants using C-4.5*, Renewable and Sustainable Energy Reviews 56 (2016), pp. 283–290.
- [95] F. Haddadi and A.N. Zincir-Heywood, *Benchmarking the Effect of Flow Exporters and Protocol Filters on Botnet Traffic Classification*, IEEE Systems Journal 10 (2016), pp. 1390–1401.
- [96] K. Polat, Ş. Yosunkaya and S. Güneş, *Comparison of Different Classifier Algorithms on the Automated Detection of Obstructive Sleep Apnea Syndrome*, Journal of Medical Systems 32 (2008), pp. 243–250.
- [97] J. Sanz, D. Paternain, M. Galar, J. Fernandez, D. Reyero and T. Belzunegui, *A new survival status prediction system for severe trauma patients based on a multiple classifier system*, Computer Methods and Programs in Biomedicine 142 (2017), pp. 1–8.
- [98] S. Bai, *Growing random forest on deep convolutional neural networks for scene categorization*, Expert Systems with Applications 71 (2017), pp. 279–287.
- [99] I.A. Ibrahim and T. Khatib, *A novel hybrid model for hourly global solar radiation prediction using random forests technique and firefly algorithm*, Energy Conversion and Management 138 (2017), pp. 413–425.
- [100] L. Zhang, Q. Wang, Y. Gao, H. Li, G. Wu and D. Shen, *Concatenated spatially-localized random forests for hippocampus labeling in adult and infant MR brain images*, Neurocomputing 229 (2017), pp. 3–12.
- [101] G. Cano, J. Garcia-Rodriguez, A. Garcia-Garcia, H. Perez-Sanchez, J.A. Benediktsson, A. Thapa et al., *Automatic selection of molecular descriptors using random forest: Application to drug discovery*, Expert Systems with Applications 72 (2017), pp. 151–159.
- [102] C. Krauss, X.A. Do and N. Huck, *Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500*, European Journal of Operational Research 259 (2017), pp. 689–702.
- [103] A. Joshi, C. Monnier, M. Betke and S. Sclaroff, *Comparing random forest approaches to segmenting and classifying gestures*, Image and Vision Computing 58 (2017), pp. 86–95.
- [104] W.S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics 5 (1943), pp. 115–133.

- [105] M. Alfaro-Ponce, A. Argüelles and I. Chairez, *Continuous neural identifier for uncertain nonlinear systems with time delays in the input signal*, Neural Networks 60 (2014), pp. 53–66.
- [106] J. Wang, C.-H. Wang and C.L.P. Chen, *The Bounded Capacity of Fuzzy Neural Networks (FNNs) Via a New Fully Connected Neural Fuzzy Inference System (F-CONFIS) With Its Applications*, IEEE Transactions on Fuzzy Systems 22 (2014), pp. 1373–1386.
- [107] X. Nie and W.X. Zheng, *Multistability of neural networks with discontinuous non-monotonic piecewise linear activation functions and time-varying delays*, Neural Networks 65 (2015), pp. 65–79.
- [108] D. Liu, S. Zhu and W. Chang, *Input-to-state stability of memristor-based complex-valued neural networks with time delays*, Neurocomputing 221 (2017), pp. 159–167.
- [109] A. Woodward, T. Froese and T. Ikegami, *Neural coordination can be enhanced by occasional interruption of normal firing patterns: A self-optimizing spiking neural network model*, Neural Networks 62 (2015), pp. 39–46.
- [110] D. Reid, A.J. Hussain and H. Tawfik, *Financial Time Series Prediction Using Spiking Neural Networks*, PLoS ONE 9 (2014), pp. e103656.
- [111] Q. Yu, H. Tang, K.C. Tan and H. Yu, *A brain-inspired spiking neural network model with temporal encoding and learning*, Neurocomputing 138 (2014), pp. 3–13.
- [112] N. Bertolino, S. Ferraro, A. Nigri, M.G. Bruzzone, F. Ghielmetti and The Coma Research Centre (CRC), *A Neural Network Approach to fMRI Binocular Visual Rivalry Task Analysis*, PLoS ONE 9 (2014), pp. e105206.
- [113] B.J.T. Fernandes, G.D.C. Cavalcanti and T.I. Ren, *Constructive Autoassociative Neural Network for Facial Recognition*, PLoS ONE 9 (2014), pp. e115967.
- [114] Q. Liu, T. Huang and J. Wang, *One-Layer Continuous-and Discrete-Time Projection Neural Networks for Solving Variational Inequalities and Related Optimization Problems*, IEEE Transactions on Neural Networks and Learning Systems 25 (2014), pp. 1308–1318.
- [115] M. Rocio Cogollo and J.D. Velasquez, *Methodological Advances in Artificial Neural Networks for Time Series Forecasting*, IEEE Latin America Transactions 12 (2014), pp. 764–771.
- [116] H. Mosbah and M. El-hawary, *Hourly Electricity Price Forecasting for the Next Month Using Multilayer Neural Network*, Canadian Journal of Electrical and Computer Engineering 39 (2016), pp. 283–291.
- [117] F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos and S. Gucci, *Single-hidden layer neural networks for forecasting intermittent demand*, International Journal of Production Economics 183, Part A (2017), pp. 116–128.
- [118] T.N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. Mohamed, G. Dahl et al., *Deep Convolutional Neural Networks for Large-scale Speech Tasks*, Neural Networks 64 (2015), pp. 39–48.
- [119] N. Vuković and Z. Miljković, *Robust sequential learning of feedforward neural networks in the presence of heavy-tailed noise*, Neural Networks 63 (2015), pp. 31–47.
- [120] X. Zhai, A.A.S. Ali, A. Amira and F. Bensaali, *MLP Neural Network Based Gas Classification System on Zynq SoC*, IEEE Access 4 (2016), pp. 8138–8146.
- [121] Y. LeCun, Y. Bengio and G. Hinton, *Deep learning*, Nature 521 (2015), pp. 436–444.



- [122] J. Amoh and K. Odame, *Deep Neural Networks for Identifying Cough Sounds*, IEEE Transactions on Biomedical Circuits and Systems 10 (2016), pp. 1003–1011.
- [123] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio et al., *Brain tumor segmentation with Deep Neural Networks*, Medical Image Analysis 35 (2017), pp. 18–31.
- [124] H. Li, X. Tang, R. Wang, F. Lin, Z. Liu and K. Cheng, *Comparative Study on Theoretical and Machine Learning Methods for Acquiring Compressed Liquid Densities of 1,1,1,2,3,3,3-Heptafluoropropane (R227ea) via Song and Mason Equation, Support Vector Machine, and Artificial Neural Networks*, Applied Sciences 6 (2016), pp. 25.
- [125] V.N. Vapnik, *Statistical Learning Theory*, 1st Edition. Wiley-Interscience, 1998.
- [126] S.I. Gass and M.C. Fu, eds., *Lagrange Multipliers*, in *Encyclopedia of Operations Research and Management Science*, Springer US, 2013, pp. 845–845.
- [127] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology 2 (2011), pp. 27:1–27:27.
- [128] J. Tao, S. Wang and W. Hu, *Minimum class spread constrained support vector machine*, Neurocomputing 151, Part 1 (2015), pp. 481–500.
- [129] Y. Zhang and P. Zhang, *Machine training and parameter settings with social emotional optimization algorithm for support vector machine*, Pattern Recognition Letters 54 (2015), pp. 36–42.
- [130] H. Su, X. Wu, X.-H. Yan and A. Kidwell, *Estimation of subsurface temperature anomaly in the Indian Ocean during recent global surface warming hiatus from satellite measurements: A support vector machine approach*, Remote Sensing of Environment 160 (2015), pp. 63–71.
- [131] B. Schölkopf, A.J. Smola, R.C. Williamson and P.L. Bartlett, *New Support Vector Algorithms*, Neural Computation 12 (2000), pp. 1207–1245.
- [132] P. Songsiri, T. Phetkaew and B. Kijirikul, *Enhancement of multi-class support vector machine construction from binary learners using generalization performance*, Neurocomputing 151, Part 1 (2015), pp. 434–448.
- [133] J.A. Nasiri, N. Moghadam Charkari and S. Jalili, *Least squares twin multi-class classification support vector machine*, Pattern Recognition 48 (2015), pp. 984–992.
- [134] M. Fu, Y. Tian and F. Wu, *Step-wise support vector machines for classification of overlapping samples*, Neurocomputing 155 (2015), pp. 159–166.
- [135] X. Peng, L. Kong and D. Chen, *Improvements on twin parametric-margin support vector machine*, Neurocomputing 151, Part 2 (2015), pp. 857–863.
- [136] S. Nan, L. Sun, B. Chen, Z. Lin and K.A. Toh, *Density-Dependent Quantized Least Squares Support Vector Machine for Large Data Sets*, IEEE Transactions on Neural Networks and Learning Systems 28 (2017), pp. 94–106.
- [137] N. Couellan and W. Wang, *Uncertainty-safe large scale support vector machines*, Computational Statistics & Data Analysis 109 (2017), pp. 215–230.
- [138] B. Fan, X. Lu and H.X. Li, *Probabilistic Inference-Based Least Squares Support Vector Machine for Modeling Under Noisy Environment*, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (2016), pp. 1703–1710.

- [139] P. Danenas and G. Garsva, *Selection of Support Vector Machines based classifiers for credit risk domain*, Expert Systems with Applications 42 (2015), pp. 3194–3204.
- [140] S. Kang, P. Kang, T. Ko, S. Cho, S. Rhee and K.-S. Yu, *An efficient and effective ensemble of support vector machines for anti-diabetic drug failure prediction*, Expert Systems with Applications 42 (2015), pp. 4265–4273.
- [141] F. Kaytez, M.C. Taplamacioglu, E. Cam and F. Hardalac, *Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines*, International Journal of Electrical Power & Energy Systems 67 (2015), pp. 431–438.
- [142] S.G. Kim, Y.G. No and P.H. Seong, *Prediction of severe accident occurrence time using support vector machines*, Nuclear Engineering and Technology 47 (2015), pp. 74–84.
- [143] M. Soltani, T.B. Moghaddam, M.R. Karim, S. Shamshirband and C. Sudheer, *Stiffness performance of polyethylene terephthalate modified asphalt mixtures estimation using support vector machine-firefly algorithm*, Measurement 63 (2015), pp. 232–239.
- [144] P. Nskh, M.N. Varma and R.R. Naik, *Principle component analysis based intrusion detection system using support vector machine*, in 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), 2016, pp. 1344–1350.
- [145] X. Xu, A. Wu, X. Zhang, M. Su, T. Jiang and Z.-M. Yuan, *MetaDP: a comprehensive web server for disease prediction of 16S rRNA metagenomic datasets*, Biophysics Reports (2017), pp. 1–10.
- [146] H. Kawase, Y. Mori, H. Hasegawa and K. c Sato, *Dynamic Router Performance Control Utilizing Support Vector Machines for Energy Consumption Reduction*, IEEE Transactions on Network and Service Management 13 (2016), pp. 860–870.
- [147] Y.D. Zhang, Z.J. Yang, H.M. Lu, X.X. Zhou, P. Phillips, Q.M. Liu et al., *Facial Emotion Recognition Based on Biorthogonal Wavelet Entropy, Fuzzy Support Vector Machine, and Stratified Cross Validation*, IEEE Access 4 (2016), pp. 8375–8385.
- [148] M. Rasmussen, J. Rieger and K.N. Webster, *Approximation of reachable sets using optimal control and support vector machines*, Journal of Computational and Applied Mathematics 311 (2017), pp. 68–83.
- [149] J. Zheng, H. Pan and J. Cheng, *Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines*, Mechanical Systems and Signal Processing 85 (2017), pp. 746–759.
- [150] Y.-J. Cha, K. You and W. Choi, *Vision-based detection of loosened bolts using the Hough transform and support vector machines*, Automation in Construction 71, Part 2 (2016), pp. 181–188.
- [151] S. Villarreal, J. Francisco, C. Yáñez Márquez and S. Suárez Guerra, *Reconocimiento automático de voz emotiva con memorias asociativas Alfa-Beta SVM*, Polibits (2011), pp. 19–23.
- [152] L. Cleofas-Sánchez, V. García, R. Martín-Félez, R.M. Valdovinos, J.S. Sánchez and O. Camacho-Nieto, *Hybrid Associative Memories for Imbalanced Data Classification: An Experimental Study*, in *Pattern Recognition*, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, J.S. Rodríguez and G.S. di Baja, eds., Springer Berlin Heidelberg, 2013, pp. 325–334.

- [153] M. Aldape-Pérez, C. Yáñez-Márquez, O. Camacho-Nieto, I. López-Yáñez and A.-J. Argüelles-Cruz, *Collaborative learning based on associative models: Application to pattern classification in medical datasets*, Computers in Human Behavior (2014).
- [154] A.V. Uriarte-Arcia, I. López-Yáñez and C. Yáñez-Márquez, *One-Hot Vector Hybrid Associative Classifier for Medical Data Classification*, PLoS ONE 9 (2014), pp. e95715.
- [155] I. López Yáñez, R. Flores Carapia, C. Yáñez Márquez and O. Camacho Nieto, *Automatic detection of cranial fractures in radiological images using a pattern classifier*, Revista Facultad de Ingeniería Universidad de Antioquia (2011), pp. 29–40.
- [156] M. Aldape-Pérez, C. Yáñez-Márquez, O. Camacho-Nieto and A. J. Argüelles-Cruz, *An associative memory approach to medical decision support systems*, Computer Methods and Programs in Biomedicine 106 (2012), pp. 287–307.
- [157] C. Yáñez-Márquez, I. López-Yáñez and G. de la L.S. Morales, *Analysis and Prediction of Air Quality Data with the Gamma Classifier*, in *Progress in Pattern Recognition, Image Analysis and Applications*, J. Ruiz-Shulcloper and W.G. Kropatsch, eds., Springer Berlin Heidelberg, 2008, pp. 651–658.
- [158] R. Storn and K. Price, *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, Journal of Global Optimization 11 (1997), pp. 341–359.
- [159] B. Xing and W.-J. Gao, *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Vol. 62, Intelligent Systems Reference Library. 1<sup>st</sup> Edition. Springer International Publishing, 2014.
- [160] C. Zhou, X. Zeng, H. Jiang and L. Han, *A generalized bipolar auto-associative memory model based on discrete recurrent neural networks*, Neurocomputing 162 (2015), pp. 201–208.
- [161] C. Zhou, X. Zeng, J. Yu and H. Jiang, *A unified associative memory model based on external inputs of continuous recurrent neural networks*, Neurocomputing 186 (2016), pp. 44–53.
- [162] A.K.O. Tiba, A.F.R. Araujo and M.N. Rabelo, *Hopf Bifurcation in a Chaotic Associative Memory*, Neurocomputing 152 (2015), pp. 109–120.
- [163] J. Hu, H. Tang, K.C. Tan and S.B. Gee, *A Spiking Neural Network Model for Associative Memory Using Temporal Codes*, in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*, H. Handa, H. Ishibuchi, Y.-S. Ong and K.C. Tan, eds., Springer International Publishing, 2015, pp. 561–572.
- [164] Q. Zhu, R. Rakkiyappan and A. Chandrasekar, *Stochastic stability of Markovian jump BAM neural networks with leakage delays and impulse control*, Neurocomputing 136 (2014), pp. 136–151.
- [165] G. Sun and Y. Zhang, *Exponential stability of impulsive discrete-time stochastic BAM neural networks with time-varying delay*, Neurocomputing 131 (2014), pp. 323–330.
- [166] A. Arunkumar, R. Sakthivel, K. Mathiyalagan and S. Marshal Anthoni, *Robust state estimation for discrete-time BAM neural networks with time-varying delay*, Neurocomputing 131 (2014), pp. 171–178.

- [167] A. Wu, Z. Zeng and X. Song, *Global Mittag–Leffler stabilization of fractional-order bidirectional associative memory neural networks*, *Neurocomputing* 177 (2016), pp. 489–496.
- [168] E. Esmi, P. Sussner, H. Bustince Sola and J. Fernandez, *O-Fuzzy Associative Memories (FAMs)*, *IEEE Transactions on Fuzzy Systems* 23 (2015), pp. 313–326.
- [169] E. Esmi, P. Sussner and S. Sandri, An introduction to tunable equivalence fuzzy associative memories, in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 1604–1611.
- [170] E. Esmi, P. Sussner and S. Sandri, *Tunable equivalence fuzzy associative memories*, *Fuzzy Sets and Systems* 292 (2016), pp. 242–260.
- [171] C. Zhong, W. Pedrycz, Z. Li, D. Wang and L. Li, *Fuzzy associative memories: A design through fuzzy clustering*, *Neurocomputing* 173, Part 3 (2016), pp. 1154–1162.
- [172] P. Zheng, *Threshold Complex-Valued Neural Associative Memory*, *IEEE Transactions on Neural Networks and Learning Systems* 25 (2014), pp. 1714–1718.
- [173] J.-M. Wu, P.-H. Hsu and C.-Y. Liou, *Sudoku associative memory*, *Neural Networks* 57 (2014), pp. 112–127.
- [174] A.A. Frolov, D. Husek, P.Y. Polyakov and V. Snasel, *New BFA method based on attractor neural network and likelihood maximization*, *Neurocomputing* 132 (2014), pp. 14–29.
- [175] H. Zhang, Y. Hou, J. Zhao, L. Wang, T. Xi and Y. Li, *Automatic welding quality classification for the spot welding based on the Hopfield associative memory neural network and Chernoff face description of the electrode displacement signal features*, *Mechanical Systems and Signal Processing* 85 (2017), pp. 1035–1043.
- [176] J. Yang, L. Wang, Y. Wang and T. Guo, *A novel memristive Hopfield neural network with application in associative memory*, *Neurocomputing* 227(2017), pp. 142–148.
- [177] E. Agliari, A. Barra, A. Galluzzi, F. Guerra, D. Tantari and F. Tavani, *Hierarchical neural networks perform both serial and parallel processing*, *Neural Networks* 66 (2015), pp. 22–35.
- [178] Á. Cabana, E. Mizraji and J.C. Valle-Lisboa, *A neural model that implements probabilistic topics*, *Neurocomputing* 171 (2016), pp. 1099–1107.
- [179] X. Hu, S. Duan, G. Chen and L. Chen, *Modeling affections with memristor-based associative memory neural networks*, *Neurocomputing* 223 (2017), pp. 129–137.
- [180] R. Kassab and F. Alexandre, *A Modular Network Architecture Resolving Memory Interference Through Inhibition*, in *Computational Intelligence*, Springer International Publishing, 2015, pp. 407–422.
- [181] M.E. Valle, *A Robust Subspace Projection Autoassociative Memory Based on the M-Estimation Method*, *IEEE Transactions on Neural Networks and Learning Systems* 25 (2014), pp. 1372–1377.
- [182] N.K. Kasabov, *NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data*, *Neural Networks* 52 (2014), pp. 62–76.
- [183] S.S. Ge, M. Li and T.H. Lee, *Dynamic saliency-driven associative memories based on network potential field*, *Pattern Recognition* 60 (2016), pp. 669–680.
- [184] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*, 1st Edition. National Bureau of Standards, USA, 1964.

- [185] R. Bracewell, *Heaviside's Unit Step Function*, in *The Fourier Transform and Its Applications*, 2000, pp. 61–65.
- [186] K.H. Rosen, *Discrete Mathematics and Its Applications*, 4th edition. McGraw-Hill Inc. US, Boston, 1999.
- [187] T. Andreescu and D. Andrica, *Number Theory: Structures, Examples, and Problems*, 1<sup>st</sup> Edition. Birkhäuser, Boston, 2009.
- [188] *UCI Machine Learning Repository*. Available at <http://archive.ics.uci.edu/ml>.
- [189] J. Alcalá-Fernández, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez et al., *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework*, *Journal of Multiple-Valued Logic and Soft Computing* 17 (2011), pp. 255–287.
- [190] P. Werbos, *New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD Thesis, Harvard, 1974.
- [191] J.C. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, in *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998, pp. 41–65.
- [192] D.S. Broomhead and D. Lowe, *Multivariable Functional Interpolation and Adaptive Networks*, *Complex Systems* 2 (1988), pp. 321–355.
- [193] J. Friedman, T. Hastie and R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent*, *J Stat Softw* 33 (2010), pp. 1–22.
- [194] Y. Freund and R.E. Schapire, *Experiments with a new boosting algorithm*, in *Proceedings of the 13th International in Machine Learning*, 96 (1996), pp. 148–156.
- [195] J. Derrac, S. García, D. Molina and F. Herrera, *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 (2011), pp. 3–18.
- [196] D.J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, 5<sup>th</sup> Edition. Chapman and Hall/CRC, Boca Raton, 2011.
- [197] F. Wilcoxon, *Individual comparisons by ranking methods*, *Biometrics* 1 (1945), pp. 80–83.
- [198] J.H. Zar, *Biostatistical Analysis*, 4th Edition. Prentice Hall, Upper Saddle River, N.J, 1998.
- [199] M. Friedman, *The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance*, *Journal of the American Statistical Association* 32 (1937), pp. 675–701.
- [200] M. Friedman, *A Comparison of Alternative Tests of Significance for the Problem of  $m$  Rankings*, *Ann. Math. Statist.* 11 (1940), pp. 86–92.
- [201] T.-T. Wong, *Performance evaluation of classification algorithms by  $k$ -fold and leave-one-out cross validation*, *Pattern Recognition* 48 (2015), pp. 2839–2846.
- [202] *Udacity - Free Online Classes & Nanodegrees*. Available at [/](http://www.udacity.com).
- [203] *Coursera | Online Courses From Top Universities. Join for Free*. Available at <https://es.coursera.org/>.
- [204] A. García-Florian, Á. Ferreira-Santiago, C. Yáñez-Márquez, O. Camacho-Nieto, M. Aldape-Pérez and Y. Villuendas-Rey, *Social Web Content Enhancement in a Distance Learning Environment: Intelligent Metadata Generation for Resources*, *The International Review of Research in Open and Distributed Learning* 18 (2017), pp. 161–176.

- [205] U.R. Acharya, M.R.K. Mookiah, J.E.W. Koh, J.H. Tan, K. Noronha, S.V. Bhandary et al., *Novel risk index for the identification of age-related macular degeneration using radon transform and DWT features*, *Computers in Biology and Medicine* 73 (2016), pp. 131–140.
- [206] D. Mittal and K. Kumari, *Automated detection and segmentation of drusen in retinal fundus images*, *Computers & Electrical Engineering* 47 (2015), pp. 82–95.
- [207] A. García Floriano, *Sistema Integral de análisis para la prevención de ceguera*, Thesis, Instituto Politécnico Nacional. Centro de Investigación en Computación, 2011.
- [208] A. García Floriano, O. Camacho Nieto and C. Yáñez Márquez, *Clasificador de Heaviside*, *Nova Scientia* 7 (2015), pp. 365–397.
- [209] D.H. Wolpert and W.G. Macready, *No free lunch theorems for optimization*, *IEEE Transactions on Evolutionary Computation* 1 (1997), pp. 67–82.