INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

ENFOQUE ASOCIATIVO PARA LA SELECCIÓN DE RASGOS

TESIS

QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

M. EN C. MARIO ALDAPE PÉREZ

DIRECTORES DE TESIS:

DR. OSCAR CAMACHO NIETO

DR. CORNELIO YÁÑEZ MÁRQUEZ

MÉXICO, D.F. MAYO DE 2011

INSTITUTO POLITÉCNICO NACIONAL

SIP-14

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 10:00 horas del día 1 del mes de Julio de 2010 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"ENFOQUE ASOCIATIVO PARA LA SELECCIÓN DE RASGOS"

Presentada por el alumno:

ALDAPE	**PÉREZ**	**MARIO**
Apellido paterno	Materno	nombre(s)

Con registro:

B	0	7	1	2	2	8

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

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

LA COMISIÓN REVISORA

Presidente

Dr. Sergio Suárez Guerra

Secretario

Dra. Nareli Cruz Cortés

Primer vocal
(Director de Tesis)

Dr. Oscar Camacho Nieto

Segundo vocal
(Director de Tesis)

Dr. Cornelio Yáñez Márquez

Tercer vocal

Dra. María Elena Acevedo Mosqueda

EL PRESIDENTE DEL COLEGIO

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
DIRECCION

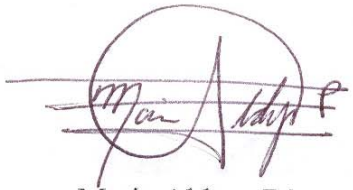
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México D.F. el día 13 del mes Mayo del año 2011, el (la) que suscribe Mario Aldape Pérez alumno (a) del Programa de Doctorado en Ciencias de la Computación con número de registro B071228, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Oscar Camacho Nieto y cede los derechos del trabajo intitulado Enfoque Asociativo para la Selección de Rasgos, 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 maldape@ieee.org. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Mario Aldape Pérez

Nombre y firma

Resumen

En este trabajo de tesis se presenta el Enfoque Asociativo para la Selección de Rasgos, que constituye un nuevo modelo para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental, el cual surge al tomar elementos de dos ramas importantes del reconocimiento de patrones. Por un lado, se toma como punto de partida el modelo de Clasificación Hibrida con Enmascaramiento (*HCM* por sus siglas en inglés) y por otro lado, el concepto de verosimilitud, tomado de la Teoría de Decisión Bayesiana. El nuevo modelo exhibe un desempeño experimental competitivo, al ser comparado con otros importantes clasificadores de patrones descritos en la literatura actual.

Abstract

In this thesis an Associative Memory Approach for Feature Selection is introduced. The proposed model constitutes a new model for data dimensionality reduction on pattern recognition tasks. This model has its foundations in two major areas of pattern recognition theory, namely: Associative Memory and Bayesian Theory. On one side, the proposed model is based on the Hybrid Classification and Masking algorithm *HCM* and on the other it is based on likelihood ratio, taken from Bayesian Theory. This model shows competitive performance when compared against other important classification methods of the current literature.

Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

En este trabajo de tesis se presenta el Enfoque Asociativo para la Selección de Rasgos, que constituye un nuevo modelo para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental, el cual surge al tomar elementos de dos ramas importantes del reconocimiento de patrones. Por un lado, se toma como punto de partida el modelo de Clasificación Hibrida con Enmascaramiento (*HCM* por sus siglas en inglés) y por otro lado, el concepto de verosimilitud, tomado de la Teoría de Decisión Bayesiana. El nuevo modelo exhibe un desempeño experimental competitivo, al ser comparado con otros importantes clasificadores de patrones descritos en la literatura actual.

1.1. Antecedentes

Un hecho innegable por todo ser humano es que con el paso del tiempo las fronteras de la ciencia incesantemente han sido redefinidas; en gran medida esto ha sido posible gracias a los constantes adelantos tanto en las capacidades de almacenamiento como de procesamiento de los equipos de cómputo. Sin duda alguna los avances en la integración de distintos materiales semiconductores [1], así como el aprovechamiento de diversos materiales ferromagnéticos se han hecho patentes en la fabricación de dispositivos avanzados de almacenamiento no volátil de datos [2].

En el año de 1991 se afirmó que la cantidad de información almacenada se duplicaría cada veinte meses [3]; dicha aseveración atrajo la atención de importantes grupos

de investigación interesados en el posicionamiento de nanopartículas de acero viajando dentro de conductos herméticos [4], que a la postre permitiría el surgimiento de dispositivos de almacenamiento masivo de datos basados en nanotubos de carbón de paredes múltiples [5], los cuales pueden alcanzar densidades de memoria superiores a los 200 Gb/in^2 [6]. Si bien es cierto que con tales recursos de almacenamiento es posible afrontar nuevos retos tecnológicos y científicos, también es cierto que existen complicaciones intrínsecas del procesamiento y análisis de datos altamente dimensionales y sus distribuciones [7].

Desde el punto de vista teórico, las propiedades de los espacios altamente dimensionales son diferentes y de comportamiento inesperado a lo que usualmente se observa en espacios dimensionalmente pequeños [8]. El fenómeno de espacio vacío, así como otros comportamientos extraños son ejemplos típicos de la "Maldición de la Dimensionalidad" [9]. El término de la "Maldición de la Dimensionalidad" aparentemente fue acuñado por Bellman [10] en el contexto de la optimización de una función de varias variables mediante la exploración exhaustiva de un espacio discreto, donde el número de muestras necesarias para optimizar una función de varias variables en cierto dominio crece exponencialmente con el número de dimensiones [11]. Al crecer la dimensión de los datos, el número de muestras disponibles para el aprendizaje se torna exponencialmente disperso [12], es decir, al incrementarse la dimensionalidad de los datos, surge la necesidad de contar con un mayor número de muestras disponibles para mantener el mismo coeficiente de densidad de muestreo con respecto a una representación de los datos de dimensión menor [13]. Otro problema que aparece en espacios altamente dimensionales es el fenómeno de concentración [14] (también conocido como la ley geométrica de los grandes números) el cual tiene que ver con el poco poder discriminatorio de una métrica conforme la dimensionalidad de los datos crece. Desde el punto de vista práctico, el fenómeno de concentración hace que el problema de clasificación de patrones basado en la búsqueda del vecino más cercano (*Nearest-Neighbor Search*) sea un problema difícil de resolver en espacios altamente dimensionales, debido a que la distancia Euclidiana entre cualesquiera dos vectores altamente dimensionales es aproximadamente constante [15].

A pesar de las complicaciones intrínsecas del procesamiento y análisis de datos altamente dimensionales y sus distribuciones, actualmente cada vez son más las áreas del

conocimiento humano que requieren procesar importantes volúmenes de datos altamente dimensionales en aras de un mayor grado de confianza en los resultados [16]. La Inteligencia Artificial como rama de las Ciencias de la Computación y la Ingeniería no es la excepción; ésta no solo se centra en la solución de problemas basados en la eficiencia predictiva de situaciones relacionadas con la detección, reconocimiento y clasificación de patrones [17], sino también en otras como el control y planeación de rutas para navegación de robótica móvil [18], reconocimiento de caracteres escritos a mano [19], procesamiento de arreglos de sensores [20], análisis de datos multivariables [21], clasificación de textos [22], clasificación de secuencias genéticas [23] e identificación de biomarcadores [24] entre muchas otras.

Un problema siempre presente al cual se enfrentan los científicos de la Inteligencia Artificial es la búsqueda de reglas o mecanismos (algoritmos de inducción) que permitan discriminar aquellos rasgos que lejos de mejorar la precisión predictiva, elevan los costos computacionales (complejidad). En el aprendizaje automático (*Machine Learning*), el algoritmo de inducción generalmente tiene que ser capaz de asignar correctamente una etiqueta o identificador (clase) sobre un conjunto de datos (rasgos) contenidos en un determinado vector (instancia). Desde esta perspectiva se podría pensar que el mayor problema que enfrenta un algoritmo de inducción en situaciones relacionadas con la clasificación de patrones consiste en pasar de un espacio de rasgos a un espacio de clases [25]; sin embargo, el verdadero problema radica en saber el número necesario de instancias y las dimensiones de éstas que permitan incrementar la precisión predictiva sobre instancias no conocidas [26].

Si bien es cierto que uno de los objetivos que persigue la Inteligencia Artificial es la creación de entidades autónomas capaces de tomar decisiones basadas en el procesamiento de importantes volúmenes de datos altamente dimensionales [16], también es cierto que no toda la información contenida en los datos es igualmente significativa [17], es decir, existe información redundante o irrelevante que dificulta el procesamiento de la misma [27]. Cuando se analizan los datos de manera multivariable aparecen dos características estrechamente relacionadas con la cantidad de información que puede suministrar una variable, a saber: irrelevancia y redundancia. La primera es donde las variables no necesariamente están fuertemente relacionadas con la información de interés para llevar a cabo procesos de aprendizaje o de extracción de información; por consiguiente éstas pueden ser eliminadas sin afectar el

desempeño de dichos procesos [28]. La otra tiene que ver con la cantidad de información que proporciona una variable cuando ésta se encuentra en función de alguna otra variable, es decir, cuando la información de una variable puede ser obtenida a partir de otra variable, podemos eliminar alguna de ellas sin afectar el desempeño de los procesos de aprendizaje o de extracción de información que se estén llevando a cabo [29]. Aun en el caso en el que todas las variables fueran relevantes, las complicaciones intrínsecas del procesamiento y análisis de datos altamente dimensionales y sus distribuciones hacen que la reducción dimensional de los datos sea una necesidad imperiosa para llevar a cabo procesos de aprendizaje o de extracción de información de manera exitosa [30]. Para el caso de clasificación de patrones altamente dimensionales, emerge la dificultad de hacer predicciones sobre instancias desconocidas mediante una hipótesis construida a partir de un número limitado de instancias de aprendizaje [27]. El número de variables o rasgos presentes en una instancia es un factor crucial que determina el tamaño del espacio de hipótesis [17]. Entre mas características tenga el patrón a clasificar, el espacio de hipótesis será más grande. Cabe señalar que el incremento lineal en el número de características que conforman un patrón, se traduce en un crecimiento exponencial del espacio de hipótesis [16]; por ejemplo, en un problema de clasificación de patrones donde solo se tengan dos clases posibles y los patrones de entrenamiento tengan N características, el espacio de hipótesis es tan grande como 2^{2^N} [14].

Las técnicas de reducción dimensional de los datos surgen no solo para simplificar el espacio de hipótesis mediante la identificación de características redundantes e irrelevantes en los datos, sino también para incrementar la precisión predictiva sobre instancias no conocidas, así como para dotar al algoritmo de aprendizaje de mayor estabilidad y capacidad de generalización [31]. Existen dos principales enfoques para la reducción dimensional de los datos: Extracción de Rasgos y Selección de Rasgos; mientras que el primero de éstos busca pasar de un espacio multidimensional de rasgos a uno menor realizando transformaciones sobre los datos, el segundo busca la obtención de un subconjunto de rasgos que no solo reduce dimensionalmente el problema, sino que además mejora la precisión predictiva sobre instancias desconocidas; de ahí que esta técnica también sea considerada como una alternativa combinatoria de optimización y tema central sobre el cual versará esta tesis.

1.2. Objetivo

Crear e implementar un modelo de Selección de Rasgos, que surja al tomar elementos de dos ramas importantes del reconocimiento de patrones. Por un lado, se toma como punto de partida el modelo de Clasificación Hibrida con Enmascaramiento (*HCM* por sus siglas en inglés) y por otro lado, el concepto de verosimilitud, tomado de la Teoría de Decisión Bayesiana. El nuevo modelo deberá exhibir un desempeño experimental competitivo, al ser comparado con otros importantes métodos de Selección de Rasgos descritos en la literatura actual.

1.3. Aportaciones

- Un nuevo modelo de Selección de Rasgos, llamado Enfoque Asociativo para la Selección de Rasgos, que exhibe un desempeño experimental competitivo, al ser comparado con otros importantes métodos de Selección de Rasgos descritos en la literatura actual.
- Análisis experimental del nuevo modelo, al aplicarlo en bases de datos conocidas.
- Aplicaciones en el ámbito de conjuntos de datos altamente dimensionales.

1.4. Organización del Documento

En este capítulo se han presentado: los antecedentes, el objetivo del presente trabajo de tesis y las aportaciones del mismo. El resto del documento de tesis está organizado de la siguiente manera:

En el Capítulo 2 se presentan los conceptos básicos de las Memorias Asociativas, así como el estado del arte de los modelos más representativos de Memorias Asociativas previos a las Alfa-Beta.

El Capítulo 3 inicia con una introducción a la Selección de Rasgos y algunos criterios aplicables para reducir dimensionalmente un conjunto de datos, sigue con los métodos de estimación de error mas comúnmente utilizados en el ámbito de clasificación de patrones, continúa con los fundamentos de la Teoría de Decisión Bayesiana y termina con una intro-

ducción al modelo de Clasificación Hibrida con Enmascaramiento (*HCM* por sus siglas en inglés).

El Capítulo 4 es la parte medular de este trabajo. Es aquí, donde se introduce formalmente el Enfoque Asociativo para la Selección de Rasgos, así como el algoritmo desarrollado para reducir dimensionalmente un conjunto de datos. El contenido del Capítulo incluye las definiciones matemáticas que sustentan al nuevo modelo.

Los resultados experimentales, así como la discusión de los mismos, se presentan en el Capítulo 5.

En el Capítulo 6 se presentan tanto las conclusiones como las recomendaciones para trabajos futuros, y termina con las referencias bibliográficas consultadas para la elaboración del presente trabajo de tesis.

Capítulo 2

Estado del Arte

Este capítulo consta de dos secciones: en la primera se presentan los conceptos básicos relacionados con las memorias asociativas, y en la segunda se incluye el estado del arte de los modelos más representativos de memorias asociativas previos a las Alfa-Beta.

2.1. Conceptos Básicos

Aun cuando los primeros modelos de memorias asociativas surgieron hace algunas décadas, no es sino hasta los años setenta cuando se vuelven el foco de atención de importantes grupos de investigación. Los prolíficos trabajos científicos de los años ochenta las posicionaron como entidades capaces de modelar no solo fenómenos biológicos asociativos, sino también como elementos fundamentales en los algoritmos concernientes a la teoría del reconocimiento de patrones y a sus aplicaciones.

El propósito fundamental de una memoria asociativa es recuperar correctamente patrones completos a partir de patrones de entrada, los cuales pueden estar alterados con algún tipo de ruido: ésta es la característica más atractiva de las memorias asociativas.

Los conceptos presentados en esta sección se han tomado de las referencias que, a nuestro juicio, son las más representativas [32, 33, 34, 35] y [36].

Una **Memoria Asociativa M** puede formularse como un sistema de entrada y

salida, idea que se esquematiza a continuación:

$$\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$$

En este esquema, los patrones de entrada y salida están representados por vectores columna denotados por $\mathbf{x}$ y $\mathbf{y}$, respectivamente.

Cada uno de los patrones de entrada forma una asociación con el correspondiente patrón de salida, la cual es similar a una pareja ordenada; por ejemplo, los patrones $\mathbf{x}$ y $\mathbf{y}$ del esquema anterior forman la asociación $(\mathbf{x}, \mathbf{y})$.

A continuación se propone una notación que se usará en la descripción de los conceptos básicos sobre memorias asociativas, y en los capítulos subsecuentes de esta tesis.

Los patrones de entrada y salida se denotarán con las letras negrillas, $\mathbf{x}$ y $\mathbf{y}$, agregándoles números naturales como superíndices para efectos de discriminación simbólica. Por ejemplo, a un patrón de entrada $\mathbf{x}^1$ le corresponderá el patrón de salida $\mathbf{y}^1$, y ambos formarán la asociación $\left(\mathbf{x}^1, \mathbf{y}^1\right)$; del mismo modo, para un número entero positivo k específico, la asociación correspondiente será $\left(\mathbf{x}^k, \mathbf{y}^k\right)$.

La Memoria Asociativa $\mathbf{M}$ se representa mediante una matriz, la cual se genera a partir de un conjunto finito de asociaciones conocidas de antemano: este es el **conjunto fundamental de aprendizaje**, o simplemente **conjunto fundamental**. El conjunto fundamental se representa de la siguiente manera:

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\} \tag{2.1}$$

donde p es un número entero positivo que representa la cardinalidad del conjunto fundamental.

A los patrones que conforman las asociaciones del conjunto fundamental se les llama **patrones fundamentales**. La naturaleza del conjunto fundamental proporciona un importante criterio para clasificar las memorias asociativas:

- Una memoria es **Autoasociativa** si se cumple que $\mathbf{x}^\mu = \mathbf{y}^\mu \ \forall \mu \in \{1, 2, ..., p\}$, por lo que uno de los requisitos que se debe de cumplir es que $n = m$.
- Una memoria **Heteroasociativa** es aquella en donde $\exists \mu \in \{1, 2, ..., p\}$ para el que se cumple que $\mathbf{x}^\mu \neq \mathbf{y}^\mu$. Nótese que puede haber memorias heteroasociativas con $n = m$.

En los problemas donde intervienen las memorias asociativas, se consideran dos fases importantes: La fase de aprendizaje, que es donde se genera la memoria asociativa a partir de las p asociaciones del conjunto fundamental, idea que se esquematiza a continuación:

$$\mathbf{x} \rightarrow \mathbf{M} \leftarrow \mathbf{y} \tag{2.2}$$

y la fase de recuperación que es donde la memoria asociativa opera sobre un patrón de entrada, idea que se esquematiza a continuación:

$$\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y} \tag{2.3}$$

A fin de especificar las componentes de los patrones, se requiere la notación para dos conjuntos a los que llamaremos arbitrariamente A y B. Las componentes de los vectores columna que representan a los patrones, tanto de entrada como de salida, serán elementos del conjunto A, y las entradas de la matriz $\mathbf{M}$ serán elementos del conjunto B.

No hay requisitos previos ni limitaciones respecto de la elección de estos dos conjuntos, por lo que no necesariamente deben ser diferentes o poseer características especiales. Esto significa que el número de posibilidades para escoger A y B es infinito.

Por convención, cada vector columna que representa a un patrón de entrada tendrá n componentes cuyos valores pertenecen al conjunto A, y cada vector columna que representa a un patrón de salida tendrá m componentes cuyos valores pertenecen también al conjunto A; es decir:

$$\mathbf{x}^\mu \in A^n \text{ y } \mathbf{y}^\mu \in A^m \ \forall \mu \in \{1, 2, ..., p\} \tag{2.4}$$

La j-ésima componente de un vector columna se indicará con la misma letra del vector, pero sin negrilla, colocando a j como subíndice ($j \in \{1, 2, ..., n\}$ o $j \in \{1, 2, ..., m\}$ según corresponda). La j-ésima componente del vector columna $\mathbf{x}^\mu$ se representa por: x_j^μ.

$$\mathbf{x}^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \qquad \mathbf{y}^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m \tag{2.5}$$

Al usar el superíndice t para indicar el transpuesto de un vector, se obtienen las siguientes expresiones para los vectores columna que representan a los patrones fundamentales de entrada y de salida, respectivamente:

$$\mathbf{x}^{\mu} = (x_1^{\mu}, x_2^{\mu}, ..., x_n^{\mu})^t = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \vdots \\ x_n^{\mu} \end{pmatrix} \in A^n \tag{2.6}$$

$$\mathbf{y}^{\mu} = (y_1^{\mu}, y_2^{\mu}, ..., y_m^{\mu})^t = \begin{pmatrix} y_1^{\mu} \\ y_2^{\mu} \\ \vdots \\ y_m^{\mu} \end{pmatrix} \in A^m \tag{2.7}$$

Con los conceptos básicos ya descritos y con la notación anterior, es posible expresar las dos fases de una memoria asociativa:

1. **Fase de Aprendizaje** (Generación de la memoria asociativa). Encontrar los operadores adecuados y una manera de generar una matriz $\mathbf{M}$ que almacene las p asociaciones del conjunto fundamental $\left\{\left(\mathbf{x}^1, \mathbf{y}^1\right), \left(\mathbf{x}^2, \mathbf{y}^2\right), ..., \left(\mathbf{x}^p, \mathbf{y}^p\right)\right\}$, donde $\mathbf{x}^{\mu} \in A^n$ y $\mathbf{y}^{\mu} \in A^m$ $\forall \mu \in \{1, 2, ..., p\}$. Si $\exists \mu \in \{1, 2, ..., p\}$ tal que $\mathbf{x}^{\mu} \neq \mathbf{y}^{\mu}$, la memoria será heteroasociativa; si $m = n$ y $\mathbf{x}^{\mu} = \mathbf{y}^{\mu}$ $\forall \mu \in \{1, 2, ..., p\}$, la memoria será autoasociativa.

2. **Fase de Recuperación** (Operación de la memoria asociativa). Hallar los operadores adecuados y las condiciones suficientes para obtener el patrón fundamental de salida $\mathbf{y}^{\mu}$, cuando se opera la memoria $\mathbf{M}$ con el patrón fundamental de entrada $\mathbf{x}^{\mu}$; lo anterior para todos los elementos del conjunto fundamental y para ambos modos: autoasociativo y heteroasociativo.

Definición 2.1 *Se dice que una memoria asociativa* $\mathbf{M}$ *exhibe recuperación correcta si al presentarle como entrada, en la fase de recuperación, un patrón* $\mathbf{x}^{\omega}$ *con* $\omega \in \{1, 2, ..., p\}$*, ésta responde con el correspondiente patrón fundamental de salida* $\mathbf{y}^{\omega}$*.*

2.2. Memorias Asociativas

A continuación, en esta sección haremos un breve recorrido por los modelos de memorias asociativas más representativos, los cuales sirvieron de base para la creación de modelos matemáticos que sustentan el diseño y operación de memorias asociativas más complejas. Para cada modelo se describe su fase de aprendizaje y su fase de recuperación.

Se incluyen cuatro modelos clásicos basados en el anillo de los números racionales con las operaciones de multiplicación y adición: *Lernmatrix*, *Correlograph*, *Linear Associator* y la Memoria Hopfield; además, se presentan tres modelos basados en paradigmas diferentes a la suma de productos, a saber: Memorias Asociativas Morfológicas, Memorias Asociativas Alfa-Beta y Memorias Asociativas Mediana.

2.2.1. *Lernmatrix* de Steinbuch

Karl Steinbuch fue uno de los primeros investigadores en desarrollar un método para codificar información en arreglos cuadriculados conocidos como *crossbar*. La importancia de la *Lernmatrix* se evidencia en una afirmación que hace Kohonen en su artículo de 1972, donde apunta que las matrices de correlación, base fundamental de su innovador trabajo, vinieron a sustituir a la *Lernmatrix* de Steinbuch [37].

La *Lernmatrix* es una memoria heteroasociativa que puede funcionar como un clasificador de patrones binarios si se escogen adecuadamente los patrones de salida; es un sistema de entrada y salida que al operar acepta como entrada un patrón binario y produce como salida la clase que le corresponde (de entre p clases diferentes), codificada ésta con un método que en la literatura se le ha llamado *one-hot* [38].

La codificación *one-hot* funciona así: para representar la clase $k \in \{1, 2, ..., p\}$, se asignan a las componentes del vector de salida $\mathbf{y}^\mu$ los siguientes valores: $y_k^\mu = 1$, y $y_j^\mu = 0$ para $j = 1, 2, ..., k-1, k+1, ..., p$.

Algoritmo de la *Lernmatrix*

- **Fase de Aprendizaje.** Se genera el esquema (*crossbar*) al incorporar la pareja de patrones de entrenamiento. Cada uno de los componentes m_{ij} de $\mathbf{M}$, la *Lernmatrix*

de Steinbuch, tiene valor cero al inicio, y se actualiza de acuerdo con la regla $m_{ij} = m_{ij} + \Delta m_{ij}$, donde:

$$\Delta m_{ij} = \begin{cases} +\varepsilon & \text{si} \quad y_i^\mu = 1 = x_j^\mu \\ -\varepsilon & \text{si} \quad y_i^\mu = 1 \text{ y } x_j^\mu = 0 \\ 0 & \text{en otro caso} \end{cases} \tag{2.8}$$

donde ε es una constante positiva escogida previamente: es usual que ε sea igual a 1.

- **Fase de Recuperación.** La i-ésima coordenada y_i^ω del vector de clase $\mathbf{y}^\omega \in A^p$ se obtiene como lo indica la siguiente expresión, donde $\bigvee$ es el operador máximo:

$$y_i^\omega = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^n m_{ij}.x_j^\omega = \bigvee_{h=1}^p \left[\sum_{j=1}^n m_{hj}.x_j^\omega\right] \\ 0 & \text{en otro caso} \end{cases} \tag{2.9}$$

2.2.2. *Correlograph* de Willshaw, Buneman y Longuet-Higgins

El *Correlograph* es un dispositivo óptico elemental capaz de funcionar como una memoria asociativa [39]. En palabras de los autores "el sistema es tan simple, que podría ser construido en cualquier laboratorio escolar de física elemental".

Algoritmo del *Correlograph*

- **Fase de Aprendizaje.** La *red asociativa* se genera al incorporar la pareja de patrones de entrenamiento $(\mathbf{x}^\mu, \mathbf{y}^\mu) \in A^n \times A^m$. Cada uno de los componentes m_{ij} de la *red asociativa* $\mathbf{M}$ tiene valor cero al inicio, y se actualiza de acuerdo con la siguiente regla:

$$m_{ij} = \begin{cases} 1 & \text{si} \quad y_i^\mu = 1 = x_j^\mu \\ \text{valor anterior} & \text{en otro caso} \end{cases} \tag{2.10}$$

- **Fase de Recuperación.** Se le presenta a la *red asociativa* $\mathbf{M}$ un vector de entrada $\mathbf{x}^\omega \in A^n$. Se realiza el producto de la matriz $\mathbf{M}$ por el vector $\mathbf{x}^\omega$ y se ejecuta una operación de umbralizado, de acuerdo con la siguiente expresión:

$$y_i^\omega = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^n m_{ij}.x_j^\omega \geq u \\ 0 & \text{en otro caso} \end{cases} \tag{2.11}$$

donde u es el valor de umbral. Una estimación aproximada del valor de umbral u se puede lograr con la ayuda de un número indicador mencionado en el artículo de Willshaw et al. de 1969: $\log_2 n$ [39].

2.2.3. *Linear Associator* de Anderson-Kohonen

El *Linear Associator* tiene su origen en los trabajos pioneros de 1972 publicados por Anderson y Kohonen [40]. Para presentar el Linear Associator consideremos de nuevo el conjunto fundamental:

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\} \text{ con } A = \{0, 1\}, \mathbf{x}^\mu \in A^n \text{ y } \mathbf{y}^\mu \in A^m \tag{2.12}$$

Algoritmo del *Linear Associator*

- **Fase de Aprendizaje.**

  1. Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se encuentra la matriz $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$ de dimensiones $m \times n$.
  2. Se suman las p matrices para obtener la memoria $\mathbf{M}$

  $$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \tag{2.13}$$

  de manera que la ij-ésima componente de la memoria $\mathbf{M}$ se expresa así:

  $$m_{ij} = \sum_{\mu=1}^{p} y_i^\mu x_j^\mu \tag{2.14}$$

- **Fase de Recuperación.** Esta fase consiste en presentarle a la memoria $\mathbf{M}$ un patrón de entrada $\mathbf{x}^\omega$, donde $\omega \in \{1, 2, ..., p\}$ y realizar la operación

$$\mathbf{M} \cdot \mathbf{x}^\omega = \left[\sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t \right] \cdot \mathbf{x}^\omega \tag{2.15}$$

2.2.4. La Memoria Asociativa Hopfield

En el modelo que originalmente propuso Hopfield [41], cada neurona x_i tiene dos posibles estados, a la manera de las neuronas de McCulloch-Pitts: $x_i = 0$ y $x_i = 1$; sin embargo, Hopfield observa que para un nivel dado de exactitud en la recuperación de patrones,

la capacidad de almacenamiento de información de la memoria se puede incrementar por un factor de 2, si se escogen como posibles estados de las neuronas los valores $x_i = -1$ y $x_i = 1$ en lugar de los valores originales $x_i = 0$ y $x_i = 1$.

Al utilizar el conjunto $\{-1, 1\}$ y el valor de umbral cero, la fase de aprendizaje para la Memoria Hopfield será similar, en cierta forma, a la fase de aprendizaje del *Linear Associator*. La intensidad de la fuerza de conexión de la neurona x_i a la neurona x_j se representa por el valor de m_{ij}, y se considera que hay simetría, es decir, $m_{ij} = m_{ji}$. Si x_i no está conectada con x_j entonces $m_{ij} = 0$; en particular, no hay conexiones recurrentes de una neurona consigo misma, lo cual significa que $m_{ii} = 0$. El estado instantáneo del sistema está completamente especificado por el vector columna de dimensión n cuyas componentes son los valores de las n neuronas.

La Memoria Hopfield es autoasociativa, simétrica, con ceros en la diagonal principal. En virtud de que la memoria es autoasociativa, el conjunto fundamental para la Memoria Hopfield es:

$$\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, ..., p\} \text{ con } \mathbf{x}^\mu \in A^n \text{ y } A = \{-1, 1\} \tag{2.16}$$

Algoritmo de la Memoria Asociativa Hopfield

- **Fase de Aprendizaje.** La fase de aprendizaje para la Memoria Hopfield es similar a la fase de aprendizaje del *Linear Associator*, con una ligera diferencia relacionada con la diagonal principal en ceros, como se muestra en la siguiente regla para obtener la ij-ésima componente de la Memoria Hopfield $\mathbf{M}$:

$$m_{ij} = \begin{cases} \sum_{\mu=1}^{p} x_i^\mu x_j^\mu & \text{si} \quad i \neq j \\ 0 & \text{si} \quad i = j \end{cases} \tag{2.17}$$

- **Fase de Recuperación.** Si se le presenta un patrón de entrada $\widetilde{\mathbf{x}}$ a la Memoria Hopfield, ésta cambiará su estado con el tiempo, de modo que cada neurona x_i ajuste su valor de acuerdo con el resultado que arroje la comparación de la cantidad $\sum_{j=1}^{n} m_{ij}.x_j$ con un valor de umbral, el cual normalmente se coloca en cero. Se representa el estado de la Memoria Hopfield en el tiempo t por $\mathbf{x}(t)$; entonces $x_i(t)$ representa el valor de

la neurona x_i en el tiempo t y $x_i(t+1)$ el valor de x_i en el tiempo siguiente $(t+1)$. Dado un vector columna de entrada $\widetilde{\mathbf{x}}$, la fase de recuperación consta de tres pasos:

1. Para $t = 0$, se hace $\mathbf{x}(t) = \widetilde{\mathbf{x}}$; es decir, $x_i(0) = \widetilde{x}_i$, $\forall i \in \{1, 2, ..., n\}$
2. $\forall i \in \{1, 2, ..., n\}$ se calcula $x_i(t+1)$ de acuerdo con la condición siguiente:

$$x_i(t+1) = \begin{cases} +1 & \text{si} \quad \sum_{j=1}^{n} m_{ij}.x_j(t) > 0 \\ x_i(t) & \text{si} \quad \sum_{j=1}^{n} m_{ij}.x_j(t) = 0 \\ -1 & \text{si} \quad \sum_{j=1}^{n} m_{ij}.x_j(t) < 0 \end{cases} \tag{2.18}$$

3. Se compara $x_i(t+1)$ con $x_i(t)$ $\forall i \in \{1, 2, ..., n\}$. Si $\mathbf{x}(t+1) = \mathbf{x}(t)$ el proceso termina y el vector recuperado es $\mathbf{x}(0) = \widetilde{\mathbf{x}}$. De otro modo, el proceso continúa de la siguiente manera: los pasos 2 y 3 se iteran tantas veces como sea necesario hasta llegar a un valor $t = \tau$ para el cual $x_i(\tau + 1) = x_i(\tau)$ $\forall i \in \{1, 2, ..., n\}$; el proceso termina y el patrón recuperado es $\mathbf{x}(\tau)$.

En el artículo original de 1982 [41], Hopfield había estimado empíricamente que su memoria tenía una capacidad de recuperar $0.15n$ patrones, y en el trabajo de Abu-Mostafa & St. Jacques [42] se estableció formalmente que una cota superior para el número de vectores de estado arbitrarios estables en una Memoria Hopfield es n.

2.2.5. Memorias Asociativas Morfológicas

La diferencia fundamental entre las memorias asociativas clásicas (*Lernmatrix*, *Correlograph*, *Linear Associator* y Memoria Asociativa Hopfield) y las Memorias Asociativas Morfológicas [43] radica en los fundamentos operacionales de éstas últimas, que son las operaciones morfológicas de *dilatación* y *erosión*; el nombre de las Memorias Asociativas Morfológicas está inspirado precisamente en estas dos operaciones básicas.

Estas memorias rompieron con el esquema utilizado a través de los años en los modelos de memorias asociativas clásicas, que utilizan operaciones convencionales entre vectores y matrices para la fase de aprendizaje y suma de productos para la recuperación de patrones. Las Memorias Asociativas Morfológicas cambian los productos por sumas y las

sumas por máximos o mínimos en ambas fases, tanto de aprendizaje como de recuperación de patrones [44].

Hay dos tipos de Memorias Asociativas Morfológicas: las memorias *max*, simbolizadas con $\mathbf{M}$, y las memorias *min*, cuyo símbolo es $\mathbf{W}$; en cada uno de los dos tipos, las memorias pueden funcionar en ambos modos: heteroasociativo y autoasociativo. Se definen dos nuevos productos matriciales:

1. El *producto máximo* entre $\mathbf{D}$ y $\mathbf{H}$, denotado por $\mathbf{C} = \mathbf{D} \bigtriangledown \mathbf{H}$, es una matriz $[c_{ij}]_{m\times n}$ cuya ij-ésima componente c_{ij} es

$$c_{ij} = \bigvee_{k=1}^{r} (d_{ik} + h_{kj}) \tag{2.19}$$

2. El *producto mínimo* entre $\mathbf{D}$ y $\mathbf{H}$, denotado por $\mathbf{C} = \mathbf{D} \boldsymbol{\Delta} \mathbf{H}$, es una matriz $[c_{ij}]_{m\times n}$ cuya ij-ésima componente c_{ij} es

$$c_{ij} = \bigwedge_{k=1}^{r} (d_{ik} + h_{kj}) \tag{2.20}$$

Los productos máximo y mínimo contienen a los operadores $\bigvee$ (máximo) y $\bigwedge$(mínimo), los cuales están íntimamente ligados con los conceptos de las dos operaciones básicas de la morfología matemática: *dilatación* y *erosión*, respectivamente.

Algoritmo de las Memorias Heteroasociativas Morfológicas *max*

- **Fase de Aprendizaje.**

  1. Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se usa el producto mínimo para crear la matriz $\mathbf{y}^\mu \boldsymbol{\Delta} (-\mathbf{x}^\mu)^t$ de dimensiones $m \times n$, donde el negado transpuesto del patrón de entrada $\mathbf{x}^\mu$ se define como $(-\mathbf{x}^\mu)^t = (-x_1^\mu, -x_2^\mu, ..., -x_p^\mu)$
  2. Se aplica el operador máximo $\bigvee$ a las p matrices para obtener la memoria $\mathbf{M}$.

$$\mathbf{M} = \bigvee_{\mu=1}^{p} \left[\mathbf{y}^\mu \boldsymbol{\Delta} (-\mathbf{x}^\mu)^t\right] \tag{2.21}$$

- **Fase de Recuperación.** Esta fase consiste en realizar el producto mínimo $\mathbf{\Delta}$ de la memoria $\mathbf{M}$ con el patrón de entrada $\mathbf{x}^\omega$, donde $\omega \in \{1, 2, ..., p\}$, para obtener un vector columna $\mathbf{y}$ de dimensión m:

$$\mathbf{y} = \mathbf{M} \ \mathbf{\Delta} \ \mathbf{x}^\omega \tag{2.22}$$

Las fases de aprendizaje y de recuperación de las Memorias Heteroasociativas Morfológicas *min* se obtienen por dualidad.

Algoritmo de las Memorias Autoasociativas Morfológicas *max*

Para este tipo de memorias se utilizan los mismos algoritmos descritos anteriormente y que son aplicados a las Memorias Heteroasociativas Morfológicas; lo único que cambia es el conjunto fundamental. Para este caso, se considera el siguiente conjunto fundamental:

$$\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, ..., p\} \text{ con } \mathbf{x}^\mu \in A^n \text{ y } A = \{0, 1\} \tag{2.23}$$

- **Fase de Aprendizaje.**

  1. Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{x}^\mu)$ se usa el producto mínimo para crear la matriz $\mathbf{x}^\mu \ \mathbf{\Delta} \ (-\mathbf{x}^\mu)^t$ de dimensiones $n \times n$.
  2. Se aplica el operador máximo $\bigvee$ a las p matrices para obtener la memoria $\mathbf{M}$.

$$\mathbf{M} = \bigvee_{\mu=1}^{p} \left[\mathbf{x}^\mu \mathbf{\Delta} (-\mathbf{x}^\mu)^t\right] \tag{2.24}$$

- **Fase de Recuperación.** Esta fase consiste en realizar el producto mínimo $\mathbf{\Delta}$ de la memoria $\mathbf{M}$ con el patrón de entrada $\mathbf{x}^\omega$, donde $\omega \in \{1, 2, ..., p\}$, para obtener un vector columna $\mathbf{x}$ de dimensión n:

$$\mathbf{x}^\mu = \mathbf{M} \ \mathbf{\Delta} \ \mathbf{x}^\omega \tag{2.25}$$

Las fases de aprendizaje y de recuperación de las Memorias Autoasociativas Morfológicas *min* se obtienen por dualidad.

x	y	$\alpha(x,y)$
0	0	1
0	1	0
1	0	2
1	1	1

Tabla 2.1: Operación binaria Alfa

2.2.6. Memorias Asociativas Alfa-Beta

Las Memorias Asociativas Alfa-Beta utilizan máximos y mínimos, y dos operaciones binarias originales α y β de las cuales heredan el nombre [34]. Para la definición de las operaciones binarias α y β se deben especificar los conjuntos A y B, los cuales son:

$$A = \{0, 1\} \text{ y } B = \{0, 1, 2\} \tag{2.26}$$

La operación binaria $\alpha : A \times A \to B$ se define como se muestra en la Tabla 2.1, asimismo, la operación binaria $\beta : B \times A \to A$ se define como se muestra en la Tabla 2.2. Los conjuntos A y B, las operaciones binarias α y β junto con los operadores $\bigvee$ (máximo) y $\bigwedge$(mínimo) usuales conforman el sistema algebraico $(A, B, \alpha, \beta, \bigvee, \bigwedge)$ en el que están inmersas las Memorias Asociativas Alfa-Beta. Se definen cuatro nuevos productos matriciales:

1. Operación α máx: $P_{m\times r}\nabla_\alpha Q_{r\times n} = \left[f_{ij}^\alpha\right]_{m\times n}$, donde $f_{ij}^\alpha = \vee_{k=1}^r \alpha(p_{ik}, q_{kj})$
2. Operación β máx: $P_{m\times r}\nabla_\beta Q_{r\times n} = \left[f_{ij}^\beta\right]_{m\times n}$, donde $f_{ij}^\beta = \vee_{k=1}^r \beta(p_{ik}, q_{kj})$
3. Operación α mín: $P_{m\times r}\Delta_\alpha Q_{r\times n} = \left[h_{ij}^\alpha\right]_{m\times n}$, donde $h_{ij}^\alpha = \wedge_{k=1}^r \alpha(p_{ik}, q_{kj})$
4. Operación β mín: $P_{m\times r}\Delta_\beta Q_{r\times n} = \left[h_{ij}^\beta\right]_{m\times n}$, donde $h_{ij}^\beta = \wedge_{k=1}^r \beta(p_{ik}, q_{kj})$

Algoritmo de las Memorias Heteroasociativas Alfa-Beta *max*

Se tienen dos tipos de Memorias Heteroasociativas Alfa-Beta: tipo *max*, denotadas por $\mathbf{M}$ y tipo *min*, denotadas por $\mathbf{W}$. En la generación de ambos tipos de memorias se usará el operador $\otimes$ el cual tiene la siguiente forma:

$$\left[\mathbf{y}^\mu \otimes (\mathbf{x}^\mu)^t\right]_{ij} = \alpha\left(y_i^\mu, x_j^\mu\right); \mu \in \{1, 2, ..., p\}, \ i \in \{1, 2, ..., m\}, \ j \in \{1, 2, ..., n\} \tag{2.27}$$

x	y	$\beta(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Tabla 2.2: Operación binaria Beta

- **Fase de Aprendizaje.**

  1. Para cada $\mu \in \{1, 2, ..., p\}$, a partir de la pareja $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se construye la matriz

  $$\left[\mathbf{y}^\mu \otimes (\mathbf{x}^\mu)^t\right]_{m\times n} \tag{2.28}$$

  2. Se aplica el operador binario máximo $\bigvee$ a las matrices obtenidas en el paso 1:

  $$\mathbf{M} = \bigvee_{\mu=1}^{p} \left[\mathbf{y}^\mu \otimes (\mathbf{x}^\mu)^t\right] \tag{2.29}$$

- **Fase de Recuperación.** Se presenta un patrón $\mathbf{x}^\omega$, con $\omega \in \{1, 2, ..., p\}$, a la Memoria Heteroasociativa Alfa-Beta tipo *max* y se realiza la operación Δ_β:

  $$\mathbf{M} \cdot \Delta_\beta \ \mathbf{x}^\omega \tag{2.30}$$

  Dado que las dimensiones de la matriz $\mathbf{M}$ son de $m \times n$ y $\mathbf{x}^\omega$ es un vector columna de dimensión n, el resultado de la operación anterior debe ser un vector columna de dimensión m.

Algoritmo de las Memorias Autoasociativas Alfa-Beta *max*

Si a una memoria heteroasociativa se le impone la condición de que $\mathbf{y}^\mu = \mathbf{x}^\mu \ \forall$ $\mu \in \{1, 2, ..., p\}$, entonces deja de ser heteroasociativa y ahora se le denomina autoasociativa. Para este caso, se considera el siguiente conjunto fundamental:

$$\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, ..., p\} \ \text{ con } \mathbf{x}^\mu \in A^n \text{ y } A = \{0, 1\} \tag{2.31}$$

- **Fase de Aprendizaje.**

1. Para cada $\mu \in \{1, 2, ..., p\}$, a partir de la pareja $(\mathbf{x}^\mu, \mathbf{x}^\mu)$ se construye la matriz

$$\left[\mathbf{x}^\mu \otimes (\mathbf{x}^\mu)^t\right]_{n\times n} \tag{2.32}$$

2. Se aplica el operador binario máximo $\bigvee$ a las matrices obtenidas en el paso 1:

$$\mathbf{M} = \bigvee_{\mu=1}^{p} \left[\mathbf{x}^\mu \otimes (\mathbf{x}^\mu)^t\right] \tag{2.33}$$

- **Fase de Recuperación.** Se presenta un patrón $\mathbf{x}^\omega$, con $\omega \in \{1, 2, ..., p\}$, a la Memoria Autoasociativa Alfa-Beta tipo *max* y se realiza la operación Δ_β:

$$\mathbf{M} \cdot \Delta_\beta \ \mathbf{x}^\omega \tag{2.34}$$

Dado que las dimensiones de la matriz $\mathbf{M}$ son de $n \times n$ y $\mathbf{x}^\omega$ es un vector columna de dimensión n, el resultado de la operación anterior debe ser un vector columna de dimensión n.

Cabe mencionar que por la forma como se lleva a cabo la fase de aprendizaje en las Memorias Autoasociativas Alfa-Beta y de acuerdo con el Teorema 4.28 (Numeración tal como aparece en), es posible afirmar que: **una Memoria Autoasociativa Alfa-Beta recupera de manera correcta el conjunto fundamental completo.**

2.2.7. Memorias Asociativas Mediana

Las Memorias Asociativas Mediana [45] utilizan los operadores A y B, definidos de la siguiente forma:

$$\begin{aligned} A(x, y) &= x - y \\ B(x, y) &= x + y \end{aligned} \tag{2.35}$$

Las operaciones utilizadas se describen a continuación:

Sean $P = [p_{ij}]_{m\times r}$ y $Q = [q_{ij}]_{r\times n}$ dos matrices.

1. Operación $\lozenge_A$: $P_{m\times r} \ \lozenge_A \ Q_{r\times n} = \left[f_{ij}^A\right]_{m\times n}$ donde $f_{ij}^A = med\ _{k=1}^{r}\ A\ (p_{ik}, q_{kj})$
2. Operación $\lozenge_B$: $P_{m\times r} \ \lozenge_B \ Q_{r\times n} = \left[f_{ij}^B\right]_{m\times n}$ donde $f_{ij}^B = med\ _{k=1}^{r}\ B\ (p_{ik}, q_{kj})$

Algoritmo de las Memorias Asociativas Mediana

- **Fase de Aprendizaje.**

  1. Para cada $\xi \in \{1, 2, ..., p\}$, a partir de la pareja $(\mathbf{x}^\xi, \mathbf{y}^\xi)$ se construye la matriz

  $$\left[\mathbf{y}^\xi \ \diamondsuit_A \ (\mathbf{x}^\xi)^t\right]_{m \times n} \tag{2.36}$$

  2. Se aplica el operador mediana a las matrices obtenidas en el paso 1 para obtener la matriz $\mathbf{M}$:

  $$\mathbf{M} = med \ {}^{p}_{\xi=1} \left[\mathbf{y}^\xi \ \diamondsuit_A \ (\mathbf{x}^\xi)^t\right] \tag{2.37}$$

- **Fase de Recuperación.** Se presenta un patrón $\mathbf{x}^\omega$, con $\omega \in \{1, 2, ..., p\}$, a la Memoria Asociativa Mediana y se realiza la siguiente operación:

  $$\mathbf{M} \cdot \diamondsuit_B \ \mathbf{x}^\omega \tag{2.38}$$

  Dado que las dimensiones de la matriz $\mathbf{M}$ son de $m \times n$ y $\mathbf{x}^\omega$ es un vector columna de dimensión n, el resultado de la operación anterior debe ser un vector columna de dimensión m.

Capítulo 3

Materiales y Métodos

3.1. Selección de Rasgos

La Selección de Rasgos se ha vuelto el foco de atención de muchos trabajos de investigación; principalmente en áreas del conocimiento humano donde las instancias que conforman el conjunto de datos (entrenamiento/prueba) consisten en cientos o miles de rasgos. Algunos ejemplos clásicos de su aplicación son: la selección de características en microarreglos genéticos [23], la predicción de ocurrencia de enfermedades en el ser humano [46], el control y planeación de rutas para navegación de robótica móvil [18], el reconocimiento de caracteres escritos a mano [19], la clasificación tanto de textos [22], así como de secuencias genéticas [47] e identificación de biomarcadores [48] entre muchas otros.

Diversos son los aspectos que deben ser observados para llevar a cabo procesos de Aprendizaje Automático (*Machine Learning*); sin embargo, la discriminación de información redundante y la preservación de información relevante son piezas clave para lograr la reducción dimensional de los datos. La mayoría de los autores coinciden en que la Selección de Rasgos (*Feature Selection*) puede ser dividida en dos grandes tareas: decidir que rasgos son los que mejor describen el contexto y seleccionar cual es la mejor combinación de éstos que mejore la precisión predictiva; sin embargo, las discrepancias ocurren por los criterios usados para definir tanto la relevancia como la redundancia en los datos.

3.1.1. Algunas Definiciones

En 1991 Almuallim y Dietterich coincidieron en que la relevancia de la información debe ser considerada booleana, libre de ruido y definida en términos de distribuciones de probabilidad, que permitan obtener una estimación confiable que sugiera la eliminación de rasgos claramente identificables [49].

Definición 3.1 *Un rasgo X_i se dice que es relevante en un concepto C, si aparece en toda la representación del concepto C y varía sistemáticamente para cada categoría o clase.*

Definición 3.2 *Un rasgo X_i se dice que es relevante si existe un x_i y y para el cual $p(X_i = x_i) > 0$, tal que*

$$p(Y = y \mid X_i = x_i) \neq p(Y = y) \tag{3.1}$$

Bajo ésta definición X_i es relevante, si al conocer su valor se produce un cambio en la asignación de la etiqueta de clase. En otras palabras, si Y es condicionalmente dependiente de X_i.

Definición 3.3 *Sea $S_i = \{X_1, ..., X_{i-1}, ..., X_{i+1}, ..., X_m\}$ el conjunto de todos los rasgos sin tomar en cuenta a X_i. Se denota s_i como el valor de todos los rasgos en S_i. Un rasgo X_i se dice que es relevante, si existe un x_i, y y s_i para el cual $p(X_i = x_i) > 0$, tal que $p(Y = y \ , \ S_i = s_i \mid X_i = x_i) \neq p(Y = y \ , \ S_i = s_i)$*

Definición 3.4 *Un rasgo X_i se dice que es relevante si existe un x_i, y y s_i para el cual $p(X_i = x_i \ , \ S_i = s_i) > 0$, tal que*

$$p(Y = y \mid \ X_i = x_i \ , \ S_i = s_i) \neq p(Y = y \mid S_i = s_i) \tag{3.2}$$

Bajo ésta definición X_i es relevante, si la probabilidad de la etiqueta de clase (dados todos los rasgos) puede cambiar cuando se elimina el conocimiento derivado del valor de X_i.

Sería deseable que con las definiciones antes mencionadas se pudiera identificar la información relevante en un conjunto de instancias determinadas; sin embargo, en situaciones donde existe alta correlación en los datos que describen un problema, las definiciones

anteriores no son suficientes; en consecuencia surgió la necesidad de definir dos grados de relevancia: relevancia fuerte y relevancia débil [11].

Para poder diferenciar entre relevancia fuerte y relevancia débil, es necesario considerar que la relevancia debe ser definida en términos de un Clasificador Bayesiano, basado en el principio matemático de que la mayoría de los sucesos son dependientes y que la probabilidad de que un suceso ocurra en el futuro puede ser deducida de las ocurrencias anteriores de dicho evento [50].

Relevancia Fuerte Un rasgo X_i es fuertemente relevante si al eliminar la información que aporta éste, se observa un decremento en la precisión predictiva del modelo en cuestión, es decir, la simple remoción de la información que proporciona este rasgo, deteriora el desempeño del proceso de clasificación. En términos de funciones de probabilidad condicional para la asignación de la etiqueta de clase, se puede enunciar lo siguiente:

Definición 3.5 *Un rasgo X_i se dice que es fuertemente relevante si existe un x_i, y y s_i para el cual $p\left(X_i = x_i \ , \ S_i = s_i\right) > 0$, tal que*

$$p\left(Y = y \mid \ X_i = x_i \ , \ S_i = s_i\right) \neq p\left(Y = y \mid S_i = s_i\right) \tag{3.3}$$

Relevancia Débil Un rasgo X_i es débilmente relevante, si no es fuertemente relevante y existe un subconjunto de rasgos, S, para el cual el desempeño del modelo en cuestión es inferior que el alcanzado por el subconjunto de rasgos dado por $S \cup \{X\}$; en otras palabras, la precisión predictiva alcanzada por el subconjunto de rasgos S, es menor que la precisión predictiva alcanzada cuando se considera la información contenida en el rasgo . En términos de funciones de probabilidad condicional para la asignación de la etiqueta de clase, se puede enunciar que un rasgo es débilmente relevante si existe un subconjunto de características para el cual existe alguna , y y con y cumple con la siguiente expresión.

Definición 3.6 *Dadas ambas definiciones para los dos grados de relevancia posibles, se dice que un rasgo es relevante, ya sea fuerte o débilmente relevante; de lo contrario se dice que es irrelevante.*

Los métodos de ordenación jerárquica de características, así como los algoritmos de selección de rasgos pueden ser, grosso modo, divididos en dos tipos. El primero de ellos (*Filters*) incluye a los algoritmos de selección de rasgos que son independientes del algoritmo predictor. Típicamente estos algoritmos buscan identificar aquellos rasgos que tienen poca probabilidad de ser útiles para el análisis de los datos. Los métodos de filtrado se basan en la evaluación de alguna métrica calculada directamente sobre los datos sin recibir retroalimentación del algoritmo predictor; consecuentemente, los algoritmos de filtrado usualmente son computacionalmente menos costosos que los métodos de envoltura (*Wrappers*).

Los métodos del segundo tipo funcionan como una envoltura alrededor del algoritmo predictor que entrega en cada iteración un subconjunto diferente de características y recibe la retroalimentación del desempeño alcanzado. Generalmente dicha retroalimentación está en términos de la precisión predictiva alcanzada por el algoritmo predictor usando un subconjunto específico de características. El enfoque de envoltura busca mejorar el desempeño de un predictor específico a través de la búsqueda del subconjunto óptimo de características.

Nota 3.1 *Se entiende por subconjunto óptimo de características como el subconjunto de características de menor cardinalidad que maximiza la precisión predictiva del algoritmo de inducción [51]*

3.1.2. Métodos *Filters*

Los métodos de filtrado son algunas de las técnicas más simples para llevar a cabo procesos de selección de rasgos y pueden ser utilizados como herramientas primarias de reducción dimensional de los datos antes de aplicar técnicas más complejas. La mayoría de los métodos de filtrado estiman la utilidad o significado que tiene una variable, independientemente de otras variables y del algoritmo predictor que se vaya a utilizar. La idea subyacente de la metodología *Filters* para llevar a cabo procesos de selección de rasgos se muestra en la Figura 3.1.

Un filtro de relevancia de los rasgos puede ser visto como una función que entrega un índice $I(S/D)$ que estima, dado un conjunto de datos D, qué tan relevante es el subcon-

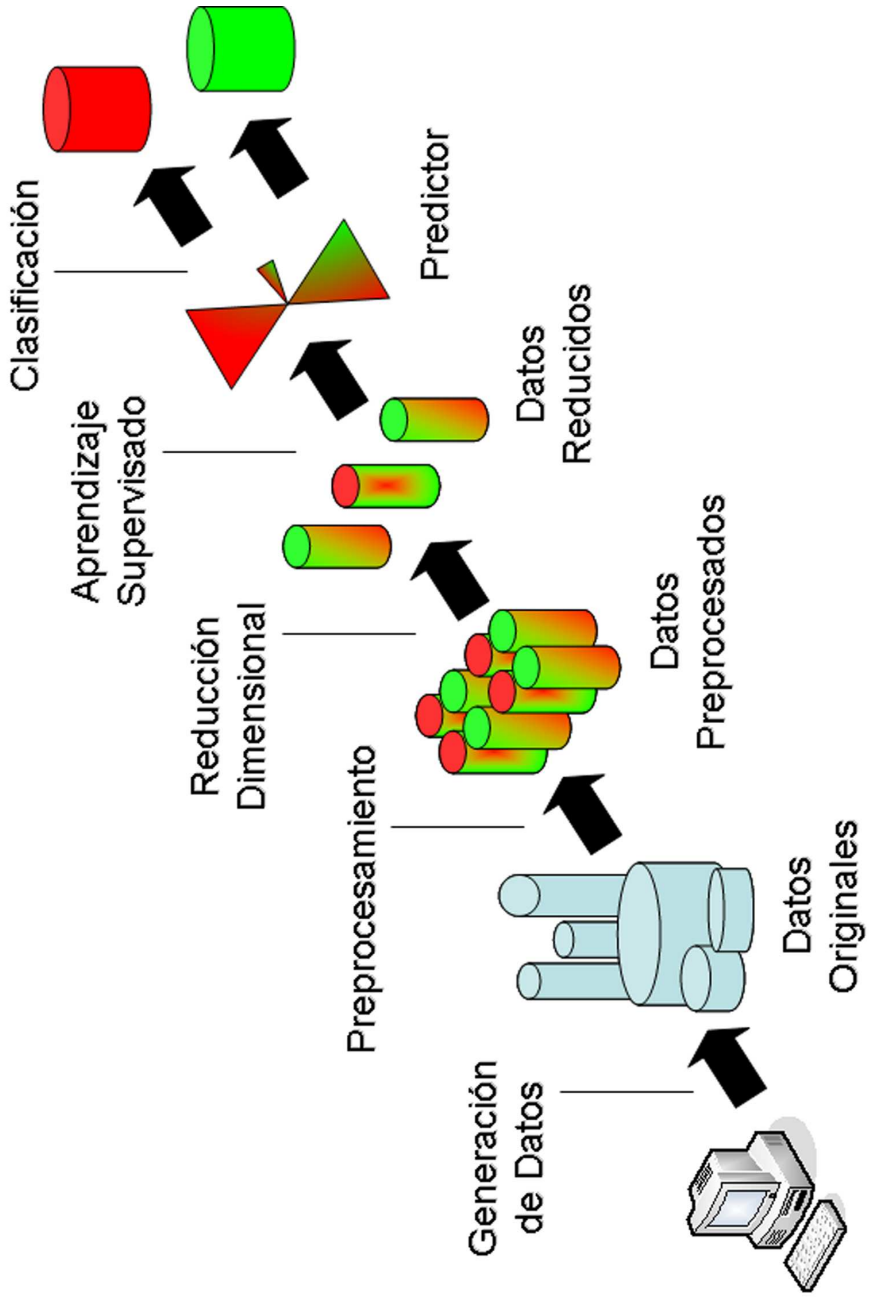


Figura 3.1: Metodología *Filters* para llevar a cabo procesos de selección de rasgos.

junto de características S para llevar a cabo la tarea Y. Dado que el conjunto de datos D y la tarea Y son generalmente fijas, el índice de relevancia variará únicamente en función del subconjunto de características seleccionadas S; consecuentemente, este índice puede ser escrito como $I(S)$.

Cuando el índice de relevancia se calcula de manera independiente para cada uno de los rasgos, es posible establecer un ordenamiento jerárquico de los mismos tomando en cuenta la relevancia de cada uno de éstos para llevar a cabo una tarea específica.

Desde la perspectiva de rasgos independientes, es suficiente considerar que aquellos rasgos cuyos índices de relevancia se encuentren por debajo de algún valor de referencia previamente establecido, serán filtrados.

3.1.3. Métodos *Wrappers*

Para llevar a cabo de manera eficaz procesos de aprendizaje supervisado, los especialistas de las ciencias de la computación no sólo deben decidir qué algoritmo es conveniente para abordar un problema en particular, sino que además deberán determinar los valores adecuados de los parámetros del mismo. Generalmente algunos de los parámetros del algoritmo seleccionado pueden determinarse ya sea tomando en cuenta información contextual del problema a resolver o aprovechando el conocimiento previo de expertos; sin embargo, cuando el algoritmo propuesto requiere de una gran cantidad de parámetros, el problema de la selección del algoritmo óptimo no puede ser visto simplemente como un problema de selección de modelos basado en una sola función objetivo. Desde una perspectiva teórica, cada valor posible de cada uno de los parámetros del algoritmo de inducción produce un modelo diferente; consecuentemente, la selección del algoritmo óptimo para abordar un problema en particular requerirá tanto de un primer mecanismo de búsqueda en el espacio de parámetros del modelo mismo, así como de un mecanismo adicional de búsqueda en el espacio de características de las instancias del problema particular.

Supongamos que tenemos un conjunto de datos cualquiera (Iris Dataset) [52], un algoritmo de inducción dado (C4.5) [51] y pretendemos llevar a cabo un proceso de aprendizaje supervisado. Inicialmente tendríamos que saber si los valores por defecto de los parámetros del algoritmo de inducción son los adecuados para resolver eficazmente el prob-

lema en cuestión. Después tendríamos que verificar si al modificar los valores por defecto de los parámetros del algoritmo de inducción, éste brinda un mejor desempeño como resultado de la incorporación tanto de información contextual del problema a resolver, así como del conocimiento previo de expertos. Posteriormente tendríamos que ejecutar repetidamente (sin cambiar los elementos del conjunto de entrenamiento) el algoritmo de inducción con diferentes valores para los parámetros, almacenando el desempeño alcanzado con cada uno de los diferentes valores de los parámetros. Finalmente tendríamos que seleccionar aquellos valores de los parámetros del algoritmo de inducción con los que se haya alcanzado el mejor desempeño; usualmente el desempeño del algoritmo de inducción se mide en términos de la precisión predictiva del mismo. La idea subyacente de la metodología *Wrapper* para el ajuste de parámetros se muestra en la Figura 3.2.

Una vez que se han encontrado los valores adecuados de los parámetros del algoritmo de inducción, podemos enfocarnos en el mecanismo de búsqueda en el espacio de características de las instancias del problema a resolver.

La metodología *Wrapper* para la selección de características considera el algoritmo de inducción como una "caja negra" cuyos parámetros han sido previamente determinados [51].

Existen dos componentes cruciales para llevar a cabo la selección de características o rasgos bajo el enfoque *Wrapper*, a saber: el componente de búsqueda y el componente de evaluación. El primero de éstos se encargará de proponer (en cada iteración) un subconjunto diferente de características o rasgos, mientras que el segundo de éstos se encargará de estimar el desempeño alcanzado por dicho subconjunto; usualmente el desempeño se mide en términos de la precisión predictiva del algoritmo de inducción [51].

Esta metodología consiste en llevar a cabo un proceso iterativo alrededor del algoritmo de inducción, donde en cada iteración se ejecutará tanto el componente de búsqueda, así como el componente de evaluación; consecuentemente, en cada iteración se obtendrá un subconjunto diferente de características o rasgos, así como el desempeño alcanzado por el algoritmo de inducción, utilizando dicho subconjunto de características o rasgos.

El procedimiento antes descrito se ejecutará repetidamente hasta que se hayan evaluado todos los posibles subconjuntos de características o rasgos, tal como se ilustra en

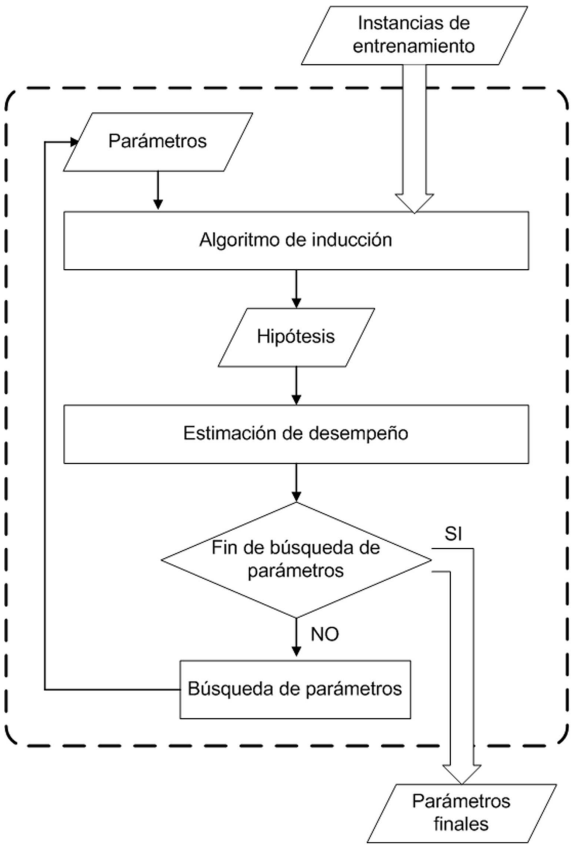


Figura 3.2: Metodología *Wrapper* para el ajuste de parámetros.

la Figura 3.3.

Una vez que se hayan evaluado todos los posibles subconjuntos de características o rasgos, si sucede que dos diferentes subconjuntos de características o rasgos proporcionan el mismo desempeño, claramente se escogerá el subconjunto de características o rasgos de menor cardinalidad, es decir, se escogerá aquel subconjunto de características o rasgos que reduzca mayormente la dimensionalidad del problema a resolver.

Nota 3.2 *El subconjunto óptimo de características se obtiene como resultado de la evaluación del desempeño alcanzado por cada uno de los posibles subconjuntos de características o rasgos; es decir, para encontrar el subconjunto óptimo de características tienen que efectuarse* $(2^f - 1)$ *estimaciones de precisión predictiva, siendo* $\mathbf{f}$ *el número de rasgos presentes en*

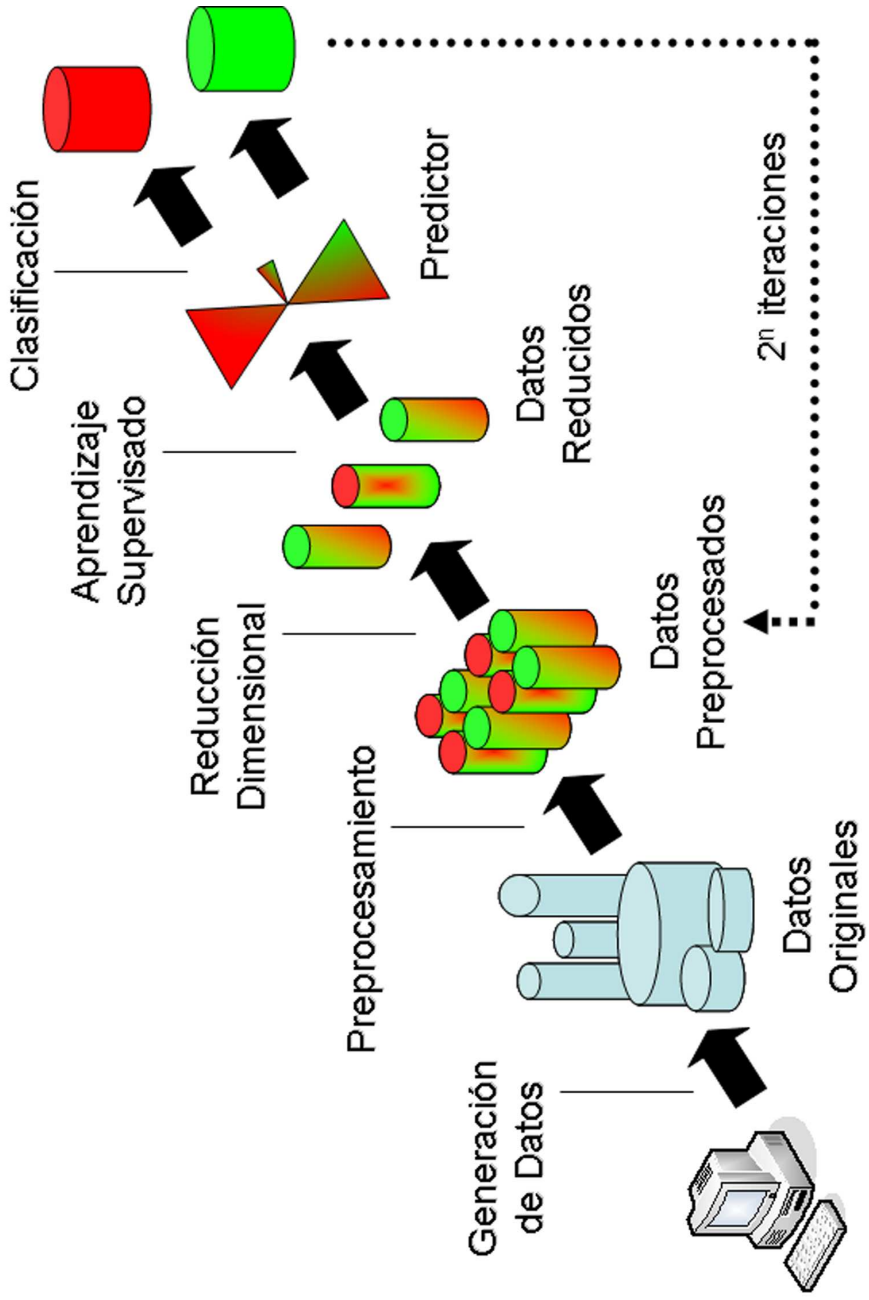


Figura 3.3: Metodología *Wrappers* para llevar a cabo procesos de selección de rasgos.

el conjunto original de datos. Claramente cuando $\mathbf{f}$ *es grande, la búsqueda del subconjunto óptimo de características implica costos computacionales prohibitivos*

3.2. Métodos de Estimación

Los métodos de validación cruzada son comúnmente aplicados cuando se busca obtener una estimación confiable del comportamiento de algún modelo propuesto en presencia de instancias no conocidas, es decir, se pretende saber si el subconjunto de rasgos seleccionado es el adecuado y si la precisión predictiva sobre instancias desconocidas es aceptable.

3.2.1. Hold Out Cross Validation

Estrictamente hablando, el método de Hold Out Cross Validation no debería ser considerado como técnica de validación cruzada porque los datos realmente nunca son cruzados; sin embargo, tomando los elementos al azar y repitiendo el procedimiento algunas veces, la estimación del error total, es aceptable [7]. Una ventaja de esta técnica es que su aplicación no requiere altos costos computacionales. Este método consiste en tomar el conjunto completo de datos disponibles y dividirlo en dos subconjuntos; el primero de ellos aplicable a la fase de entrenamiento y el segundo para la fase de prueba. Generalmente se consideran dos terceras partes para el conjunto de entrenamiento y la otra tercera parte para el conjunto de prueba.

Cabe mencionar que cuando se tienen pocas instancias disponibles o existe alta dispersión en los datos, puede presentarse alta varianza en la estimación del error total [11]; en consecuencia, es recomendable aplicar criterios estadísticos adicionales ya que el resultado de la estimación del desempeño alcanzado depende fuertemente de las instancias que hayan sido seleccionadas para formar el conjunto de entrenamiento [53].

3.2.2. K-Fold Cross Validation

Una alternativa para minimizar los efectos de la dispersión de los datos y las debilidades inherentes a la técnica de Hold Out Cross Validation, es el método de K-Fold Cross

Validation. En este método el conjunto de datos disponibles es dividido en K particiones que dan lugar a K subconjuntos mutuamente excluyentes. Para cada una de las K estimaciones de error, uno de los K subconjuntos es usado como conjunto de prueba y los otros $K-1$ restantes son agrupados para formar el conjunto de entrenamiento.

El procedimiento es repetido K veces para asegurar que los K subconjuntos han sido utilizados en la fase de prueba, de modo tal, que se tienen K estimaciones de error E_i, que serán promediadas para obtener el error total de predicción del modelo en cuestión.

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \tag{3.4}$$

Una de las ventajas que tiene este método es que aún cuando se disponga de pocas instancias para ambas fases (entrenamiento/prueba), la estimación total del error E, es confiable debido a que todos los patrones son considerados en la fase de prueba; en otras palabras, todos los patrones se consideran una vez en la fase de prueba y $K-1$ veces en la fase de entrenamiento. Cabe mencionar que la varianza en la estimación del error de predicción disminuye conforme el valor de K aumenta [54].

La desventaja evidente de este método es que la fase de entrenamiento tiene que ejecutarse en K ocasiones, lo que implica K veces más tiempo de cómputo que la técnica de Hold Out Cross Validation.

3.2.3. Leave-One-Out Cross Validation

Este método de validación cruzada puede verse como la versión de K-Fold Cross Validation cuando K es igual al número N de instancias disponibles. Esto implica que para cada una de las N veces que se realice la fase de aprendizaje, será necesario considerar $N-1$ instancias para entrenamiento y solo una instancia para prueba, obteniendo N estimaciones parciales de error. Análogamente, se puede calcular el error total en la precisión predictiva usando la siguiente expresión:

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i \tag{3.5}$$

Una de las ventajas de este método es que el problema de alta varianza en la estimación del error total E, derivado ya sea de un número reducido de instancias disponibles o de

la dispersión en los datos se vuelve casi despreciable [55]. Esto no debiera sorprender, ya que conforme crece el número de particiones, decrece el tamaño de cada una de éstas, haciendo que el número de instancias involucradas en la fase de aprendizaje sea menor, y en consecuencia la varianza tiende a disminuir [54].

Contrario a lo que se pudiera pensar, la precisión predictiva de un clasificador no gira en torno a una mayor cantidad de instancias procesadas en la fase de aprendizaje. Algunos trabajos de investigación [56],[53],[57] sugieren que entre mayor sea la cantidad de instancias conocidas durante el entrenamiento del clasificador, pueden surgir comportamientos no deseados en la fase de recuperación (overfitting); en otras palabras, únicamente se observará comportamiento adecuado con los patrones conocidos, pero no se asegura que la precisión predictiva sobre instancias desconocidas sea la misma [58].

Aún en situaciones donde no se tienen suficientes patrones para llevar a cabo las dos fases (entrenamiento/prueba) con diferentes instancias, el método de validación cruzada más comúnmente aplicado en la estimación del desempeño de clasificadores de patrones es el de K-Fold Cross Validation con un valor de K=10 o mayor [59].

3.3. Teoría de Decisión Bayesiana

El contenido de la presente sección está basado en las referencias [60], [61] y [62], en algunas de éstas se especifica que a partir de la definición de probabilidad condicional, surge uno de los teoremas más importantes en la teoría de la probabilidad, y en particular, en el enfoque probabilístico-estadístico de reconocimiento de patrones: el Teorema de Bayes. La importancia de éste radica en que es la base del Clasificador Bayesiano. Es notable la afirmación debida a Duda y Hart, autores de uno de los textos más utilizados a nivel mundial en los cursos de reconocimiento de patrones, respecto del Clasificador Bayesiano al presentar la Teoría de la Decisión Beyesiana:

> "Empezamos [con la Teoría de la Decisión Beyesiana] considerando el caso ideal en el cual la estructura estadística inherente a las categorías es perfectamente conocida. Mientras que este tipo de situación raramente se presenta en la práctica, [esta teoría] nos permite determinar el clasificador óptimo (Bayesiano) contra el cual podemos comparar todos los demás clasificadores"... ([60], pp. 17).

Teorema 3.1 ***Teorema de Bayes.*** *Sean los eventos $A_1, A_2, ..., A_n$ una partición del espacio muestral X y sea B un evento dentro del mismo espacio, entonces:*

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{k=1}^{n} P(A_k)P(B|A_k)} \tag{3.6}$$

El Teorema de Bayes es muy importante porque permite cambiar el sentido de la probabilidad condicional; es especialmente útil cuando es más fácil calcular la probabilidad de B dado A_i que de A_i dado B, y se puede explicar con palabras de la siguiente forma:

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

donde $P(A_i)$ define el conocimiento *a priori* del problema, es decir, la probabilidad absoluta de que suceda A_i antes de saber cualquier cosa sobre B; la probablidad posterior $P(B|A_i)$ define la *verosimilitud*, es decir, qué tan probable es que suceda el evento B dentro del espacio de trabajo reducido de A_i; luego, $P(B) = \sum_{k=1}^{n} P(A_k)P(B|A_k)$ es la *evidencia*, lo que indica cuál es la probabilidad de que ocurra B si se tiene todo el conocimiento *a priori* de toda la partición y la verosimilitud de B dentro de cada espacio de trabajo A_k (usualmente sólo se ve como un factor de normalización); finalmente, $P(A_i|B)$ es el conocimiento posterior o consecuente de que suceda el evento A_i dado que ocurrió B.

Usualmente, los eventos en el Teorema de Bayes están expresados en términos de variables aleatorias y distribuciones de probabilidad, por lo que para conocer la probabilidad absoluta $P(A_i|B)$ es necesario conocer todas las distribuciones de probabilidad asociadas al problema a resolver y todas las probabilidades *a priori*.

Lo anterior puede ser útil para reconocer patrones si las clases y los patrones se modelan como eventos o variables aleatorias. La idea general es la siguiente: un patrón (evento representado por una variable aleatoria vectorial X) pertenece a la clase i (evento representado por la variable aleatoria C_i) si su probabilidad de pertenecer a esa clase es más grande que la probabilidad de pertenecer a las demás clases.

El Clasificador Bayesiano toma ventaja de las probabilidades condicionales al sustituir, en el proceso de clasificación, el Teorema de Bayes en la siguiente regla:

$$X \in C_i, \text{ si } P(C_i \mid X) > P(C_j \mid X)\ \forall i \neq j \tag{3.7}$$

cuya interpretación lógica es la siguiente: si se conoce que el patrón X ocurrió (es decir que fue presentado al sistema), se calcula la probabilidad de que ocurra C_k $\forall k = 1, 2, ..., n$ y se clasifica en la clase C_i si dicha probabilidad es la mayor de todas, es decir, si la probabilidad de pertenencia de X a C_i es mayor a cualquier otra C_j.

Al usar el Teorema de Bayes en 3.7, y tomando en cuenta que las probabilidades siempre son positivas, queda lo siguiente:

$$P(C_i \mid X) > P(C_j \mid X)$$

$$\frac{P(C_i)\ p(X \mid C_i)}{P(X)} > \frac{P(C_j)\ p(X \mid C_j)}{P(X)}$$

$$P(C_i)\ p(X \mid C_i) > P(C_j)\ p(X \mid C_j) \tag{3.8}$$

Ahora bien, dado que ln (la función logaritmo natural) es una función monótona creciente, es decir del tipo: $f(x) < f(y) \Leftrightarrow x < y$, se puede hacer la siguiente sustitución en la expresión 3.8:

$$\ln\left(P(C_i)\ p(X \mid C_i)\right) > \ln\left(P(C_j)\ p(X \mid C_j)\right)$$

$$\ln(P(C_i)) + \ln(p(X \mid C_i)) > \ln(P(C_j)) + \ln(p(X \mid C_j))$$

$$d_i > d_j, \text{ con } d_k = \ln(P(C_k)) + \ln(p(X \mid C_k)) \tag{3.9}$$

donde d_k define una función discriminante para el clasificador.

Así pues, que un patrón desconocido sea clasificado en una clase C_i en particular, implica tener todo el conocimiento *a priori* de cada clase C_i y su distribución de probabilidad correspondiente, lo que raramente sucede en la práctica [63].

Tomando en cuenta lo anterior, el algoritmo para diseñar un Clasificador Bayesiano queda como sigue:

Algoritmo 3.1 *Algoritmo del Clasificador Bayesiano*

1. Obtener una muestra representativa S 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 en 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 $p(X \mid C_k)$ la cual va a ser dependiente del número y naturaleza de cada rasgo de la variable aleatoria vectorial X.

5. Aplicar la siguiente regla para clasificar un patrón desconocido de entrada X, :

$$X \in C_i, \text{ si } d_i > d_j \ \forall i \neq j, \text{ con } d_k = \ln(P(C_k)) + \ln(p(X \mid C_k)) \tag{3.10}$$

Como se puede apreciar en los párrafos anteriores, este clasificador, si bien es muy robusto, también tiene la desventaja de que se debe tener una estadística muy amplia y completa sobre todas las variables aleatorias que forman parte del sistema. Entre más grande sea la muestra, y por tanto las mediciones estadísticas sobre ella, más confiable serán los resultados del clasificador, lo cual de alguna manera significa haber hecho el proceso de clasificación a mano durante mucho tiempo para poder tener una buena respuesta. Dado que esta situación pocas veces se presenta en la práctica, el uso del Clasificador Bayesiano se ve limitado, forzando a los investigadores en este campo a establecer condicines artificiales a las probabilidades condicionales de modo que sea funcional su uso.

3.4. Clasificador Híbrido Asociativo con Traslación

El Clasificador Híbrido Asociativo con Traslación (CHAT) [64] surge a partir de las limitaciones que presenta el Clasificador Híbrido Asociativo (CHA) [65] al llevar a cabo tareas de clasificación de patrones. El CHA está basado en dos memorias asociativas pioneras: La *Lernmatrix* de Steinbuch [37] y el *Linear Associator* de Anderson-Kohonen [40].

El CHA puede aceptar valores reales en las componentes de sus vectores de entrada (a diferencia de la *Lernmatrix*, que sólo acepta valores binarios, 1 y 0) y a sus vectores de entrada no se les exige que sean ortonormales (los vectores de entrada del *Linear Associator* deber ser ortonormales para alcanzar recuperación correcta en todo su conjunto fundamental).

Cabe mencionar que aun cuando el CHA supera ampliamente las limitaciones de las memorias asociativas antes mencionadas, hay algunos casos en los que el CHA presenta problemas para llevar a cabo tareas de clasificación de patrones, es decir, la etiqueta de clase recuperada es ambigua. Con la finalidad de superar las limitaciones del CHA se aplicó el concepto de traslación del conjunto fundamental, dando paso al surgimiento del CHAT.

Definición 3.7 *Sean* $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^p$ *elementos del conjunto de patrones de entrada; se define* $\overline{\mathbf{x}}$ *el vector medio de todos ellos, calculado de acuerdo con la siguiente expresión:*

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{\mu=1}^{p} \mathbf{x}^{\mu} \tag{3.11}$$

Definición 3.8 *Sean* $\mathbf{x}^{1'}, \mathbf{x}^{2'}, ..., \mathbf{x}^{p'}$ *elementos del conjunto de patrones de entrada trasladados, obtenidos de acuerdo con la siguiente expresión:*

$$\mathbf{x}^{\mu'} = [\mathbf{x}^{\mu} - \overline{\mathbf{x}}] \;\; \forall \mu \in \{1, 2, ..., p\} \tag{3.12}$$

Algoritmo 3.2 *Algoritmo del Clasificador Híbrido Asociativo con Traslación*

1. Sea $\{\mathbf{x}^{\mu} \mid \mu = 1, 2, \ldots, p\}$ un conjunto de patrones de entrada de dimensión n con valores reales en sus componentes (a la manera del *Linear Associator*), agrupados en m clases diferentes.

2. A cada uno de los patrones de entrada que pertenece a la clase k se le asigna un vector formado por ceros, excepto en la coordenada k-ésima, donde el valor es uno (a la manera de la *Lernmatrix*).

3. Se calcula el vector medio del conjunto de patrones de entrada de acuerdo con la expresión (3.11) de la definición 3.7.

4. Se realiza la traslación de todos los patrones de entrada del conjunto fundamental de acuerdo con la expresión (3.12) de la definición 3.8.

5. Se aplica la fase de aprendizaje, que es similar a la del *Linear Associator*, explicada en la Sección 2.2.3.

6. Se aplica la fase de recuperación, que es similar a la de la *Lernmatrix*, explicada en la Sección 2.2.1.

■

A continuación se muestra un ejemplo en el que se observa el funcionamiento del CHAT.

Ejemplo 3.1 *Sean* $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \ldots, p\}$ *los elementos que definen el conjunto fundamental de patrones. Se tienen 2 asociaciones de patrones agrupados en dos diferentes clases.*

Sea $\mathbf{x}^1$un patrón de entrada del conjunto fundamental perteneciente a la clase 1 y sea $\mathbf{x}^2$ un patrón de entrada del conjunto fundamental perteneciente a la clase 2.

$$\mathbf{x}^1 = \begin{pmatrix} 2.0 \\ 3.0 \\ 6.0 \end{pmatrix}; \mathbf{x}^2 = \begin{pmatrix} 6.0 \\ 8.0 \\ 10.0 \end{pmatrix}$$

Lo anterior significa, de acuerdo con el paso 2 del algoritmo, que los correspondientes patrones de salida son los siguientes: $\mathbf{y}^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $\mathbf{y}^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Primeramente se calcula el vector medio del conjunto de patrones de entrada, de acuerdo con la expresión (3.11) de la definición 3.7. Esto es:

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{\mu=1}^{p} \mathbf{x}^\mu = \frac{1}{2} \left[\mathbf{x}^1 + \mathbf{x}^2 \right]$$

$$\overline{\mathbf{x}} = \frac{1}{2} \left[\begin{pmatrix} 2.0 \\ 3.0 \\ 6.0 \end{pmatrix} + \begin{pmatrix} 6.0 \\ 8.0 \\ 10.0 \end{pmatrix} \right] = \begin{pmatrix} 4.0 \\ 5.5 \\ 8.0 \end{pmatrix}$$

Una vez que se tiene el vector medio $\overline{\mathbf{x}}$, se realiza la traslación de todos los patrones de entrada del conjunto fundamental de acuerdo con la expresión (3.12) de la definición 3.8. Una vez trasladados todos los vectores que conforman el conjunto fundamental, se obtiene

un nuevo conjunto de vectores trasladados, denotado como: $\left\{(\mathbf{x}^{\mu'}, \mathbf{y}^{\mu'}) \mid \mu = 1, 2, \ldots, p\right\}$.

$$\mathbf{x}^{1'} = \begin{pmatrix} 2.0 \\ 3.0 \\ 6.0 \end{pmatrix} - \begin{pmatrix} 4.0 \\ 5.5 \\ 8.0 \end{pmatrix} = \begin{pmatrix} -2.0 \\ -2.5 \\ -2.0 \end{pmatrix}$$

$$\mathbf{x}^{2'} = \begin{pmatrix} 6.0 \\ 8.0 \\ 10.0 \end{pmatrix} - \begin{pmatrix} 4.0 \\ 5.5 \\ 8.0 \end{pmatrix} = \begin{pmatrix} 2.0 \\ 2.5 \\ 2.0 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{1'} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $\mathbf{y}^{2'} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Para llevar a cabo la fase de aprendizaje, de acuerdo con el paso 5 del algoritmo, se calculan los términos $\mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t$, usando la expresión (2.13). El clasificador que se obtiene es el siguiente:

$$\mathbf{C} = \sum_{\mu=1}^{2} \mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = \begin{pmatrix} -2.0 & -2.5 & -2.0 \\ 2.0 & 2.5 & 2.0 \end{pmatrix}$$

El paso 6 del algoritmo indica que la fase de recuperación se debe llevar a cabo de acuerdo con la expresión (2.9).

$$\mathbf{C} \cdot \left(\mathbf{x}^{1'}\right)^t = \begin{pmatrix} -2.0 & -2.5 & -2.0 \\ 2.0 & 2.5 & 2.0 \end{pmatrix} \cdot \begin{pmatrix} -2.0 \\ -2.5 \\ -2.0 \end{pmatrix} = \begin{pmatrix} 14.25 \\ -14.25 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \text{Clase 1}$$

$$\mathbf{C} \cdot \left(\mathbf{x}^{2'}\right)^t = \begin{pmatrix} -2.0 & -2.5 & -2.0 \\ 2.0 & 2.5 & 2.0 \end{pmatrix} \cdot \begin{pmatrix} 2.0 \\ 2.5 \\ 2.0 \end{pmatrix} = \begin{pmatrix} -14.25 \\ 14.25 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \text{Clase 2}$$

A continuación se muestra el funcionamiento del CHAT cuando se presentan patrones que NO pertenecen al conjunto fundamental, es decir, cuando se presentan patrones que no fueron considerados durante la fase de aprendizaje.

Sea $\mathbf{x}^3$un patrón de entrada desconocido perteneciente a la clase 1 y sea $\mathbf{x}^4$ un

patrón de entrada desconocido perteneciente a la clase 2.

$$\mathbf{x}^3 = \begin{pmatrix} 1.9 \\ 3.8 \\ 5.5 \end{pmatrix}; \mathbf{x}^4 = \begin{pmatrix} 6.4 \\ 7.2 \\ 9.7 \end{pmatrix}$$

Al trasladar los patrones desconocidos, de acuerdo con el paso 4 tenemos los siguientes patrones trasladados:

$$\mathbf{x}^{3'} = \begin{pmatrix} 1.9 \\ 3.8 \\ 5.5 \end{pmatrix} - \begin{pmatrix} 4.0 \\ 5.5 \\ 8.0 \end{pmatrix} = \begin{pmatrix} -2.1 \\ -1.7 \\ -2.5 \end{pmatrix}$$

$$\mathbf{x}^{4'} = \begin{pmatrix} 6.4 \\ 7.2 \\ 9.7 \end{pmatrix} - \begin{pmatrix} 4.0 \\ 5.5 \\ 8.0 \end{pmatrix} = \begin{pmatrix} 2.4 \\ 1.7 \\ 1.7 \end{pmatrix}$$

El paso 6 del algoritmo indica que la fase de recuperación se debe llevar a cabo de acuerdo con la expresión (2.9). Esto es:

$$\mathbf{C} \cdot \left(\mathbf{x}^{3'}\right)^t = \begin{pmatrix} -2.0 & -2.5 & -2.0 \\ 2.0 & 2.5 & 2.0 \end{pmatrix} \cdot \begin{pmatrix} -2.1 \\ -1.7 \\ -2.5 \end{pmatrix} = \begin{pmatrix} 13.45 \\ -13.45 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \text{Clase 1}$$

$$\mathbf{C} \cdot \left(\mathbf{x}^{4'}\right)^t = \begin{pmatrix} -2.0 & -2.5 & -2.0 \\ 2.0 & 2.5 & 2.0 \end{pmatrix} \cdot \begin{pmatrix} 2.4 \\ 1.7 \\ 1.7 \end{pmatrix} = \begin{pmatrix} -12.45 \\ 12.45 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \text{Clase 2}$$

Como se pudo observar, el CHAT tiene la capacidad de clasificar correctamente patrones que NO pertenecen al conjunto fundamental. El autor del CHAT describe en su trabajo de tesis [64] una serie de experimentos donde se muestra el desempeño alcanzado por este algoritmo en el ámbito del reconocimiento de patrones.

3.5. Clasificador Híbrido con Enmascaramiento

El algoritmo de Clasificación Híbrida con Enmascaramiento (*HCM* por sus siglas en inglés) permite obtener una representación equivalente (dimensionalmente menor) de

un conjunto fundamental de patrones [66]. El punto neurálgico sobre el cual descansa este algoritmo, es la obtención de un vector de enmascaramiento que representa un subconjunto de características que conservan información relevante para fines de clasificación de patrones.

3.5.1. Fase de Aprendizaje

Consiste en encontrar los operadores adecuados y una manera de generar una Memoria Asociativa $\mathbf{M}$ que almacene las p asociaciones del conjunto fundamental, donde $\mathbf{x}^\mu \in A^n$ y $\mathbf{y}^\mu \in A^m \; \forall \mu \in \{1, 2, \ldots, p\}$. Para tales fines, se construye una Memoria Asociativa $\mathbf{M}$ en dos etapas:

1. Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, obtenga una matriz de dimensión $m \times n$ efectuando la siguiente operación $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, ..., x_n^\mu) = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix} \tag{3.13}$$

2. Sumando cada una de las p matrices obtenidas en el paso anterior, obtenga una Memoria Asociativa $\mathbf{M}$, aplicando la siguiente expresión:

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \tag{3.14}$$

de este modo, la ij-ésima componente de la Memoria Asociativa $\mathbf{M}$ se expresa de la siguiente manera:

$$m_{ij} = \sum_{\mu=1}^{p} y_i^\mu x_j^\mu \tag{3.15}$$

3.5.2. Fase de Clasificación

Consiste en encontrar la clase a la cual pertenece un patrón de entrada $\mathbf{x}^\omega$dado. Encontrar la clase significa obtener el vector de salida $\mathbf{y}^\omega \in A^m$ correspondiente al vector

de entrada $\mathbf{x}^{\omega} \in A^n$ desconocido. El desempeño del modelo se mide en términos de la tasa de error; de este modo, la precisión predictiva se obtiene tomando en cuenta el número de patrones desconocidos clasificados de manera correcta. La i-ésima componente del vector de salida y_i^{ω} recuperado, está dada por la siguiente expresión:

$$y_i^{\omega} = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^{n} m_{ij}.x_j^{\omega} = \bigvee_{h=1}^{m} \left[\sum_{j=1}^{n} m_{hj}.x_j^{\omega} \right] \\ 0 & \text{en otro caso} \end{cases} \tag{3.16}$$

donde $\bigvee$ es el operador máximo [34].

3.5.3. Selección de Características Relevantes

Definición 3.9 *Sea* $\mathbf{f}$ *el número de rasgos presentes en el conjunto original de datos*

Definición 3.10 *Sea* $\mathbf{r}$ *un índice; tal que* $r \in \{1, 2, ..., (2^f - 1)\}$

Definición 3.11 *Sea* $\mathbf{e}^r$ *el* r*-ésimo vector de enmascaramiento de dimensión* n*, representado de la siguiente forma:*

$$\mathbf{e}^r = \begin{pmatrix} e_1^r \\ e_2^r \\ \vdots \\ e_n^r \end{pmatrix} \in B^n \tag{3.17}$$

donde $B = \{0, 1\}$*, y* e_n^r *es el bit menos significativo (LSB, por sus siglas en inglés)*

Definición 3.12 *Sea* $\dashv$ *una nueva operación llamada* $IntToVector$ *la cual toma un valor entero* $r \in \{1, 2, ..., (2^f - 1)\}$ *y entrega un vector columna* $\mathbf{e}^r$ *con el valor entero* r *expresado en forma binaria*

Definición 3.13 *Sea* $\|$ *una nueva operación llamada* $MagVector$ *la cual toma un vector columna* $\mathbf{e}^r$ *de dimensión* n *y entrega un valor entero positivo de acuerdo con la siguiente expresión:*

$$\| \mathbf{e}^r = \sum_{j=1}^{n} \left(e_j^r \wedge 1 \right) \tag{3.18}$$

donde $\wedge$ *es el operador lógico* AND

De este modo, al incorporar el r-ésimo vector de enmascaramiento $\mathbf{e}^r$ en la expresión (3.16), la etapa de clasificación se lleva a cabo aplicando la siguiente expresión:

$$y_i^\omega = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^{n} m_{ij}.\left(x_j^\omega.e_j^r\right) = \bigvee_{h=1}^{m}\left[\sum_{j=1}^{n} m_{hj}.\left(x_j^\omega.e_j^r\right)\right] \\ 0 & \text{en otro caso} \end{cases} \tag{3.19}$$

donde $r \in \{1, 2, ..., (2^f - 1)\}$

3.5.4. Procedimiento de Selección de Características

Definición 3.14 *Sea n la dimensión de cada uno de los patrones que conforman el conjunto fundamental, agrupados en m clases diferentes*

Definición 3.15 *Sea k la clase a la cual pertenece cada uno de los patrones de entrada, con $k \in \{1, 2, ..., m\}$ representada por un vector columna cuyas componentes son asignadas por $y_k^\mu = 1$ y $y_j^\mu = 0$ para cada $j = 1, 2..., k-1, k+1, ...m$*

1. Crear un clasificador utilizando las expresiones (3.13), (3.14) y (3.15)

2. Aplicar la operación $IntToVector$ para obtener el r-ésimo vector de enmascaramiento, como se indica en la expresión (3.17)

3. Obtener la r-ésima estimación de precisión predictiva aplicando la expresión (3.19)

4. Almacenar dos parámetros (la r-ésima estimación de precisión predictiva y el r-ésimo vector de enmascaramiento)

5. Comparar la r-ésima estimación de precisión predictiva contra la $(r-1)$-ésima estimación de precisión predictiva y almacenar la mejor, es decir, almacenar la estimación de precisión predictiva con menor tasa de error

6. Comparar el r-ésimo vector de enmascaramiento contra el $(r-1)$-ésimo vector de enmascaramiento, aplicando la operación $MagVector$ como se indica en la expresión (3.18) y almacenar el mejor, es decir, almacenar el vector de enmascaramiento con la menor magnitud

7. Obtener el subconjunto de características indicado por el valor del r-ésimo vector columna de enmascaramiento $\mathbf{e}^r$

8. Verificar si el valor de r ha alcanzado el número máximo de iteraciones posibles, es decir, verificar si $r = (2^f - 1)$

   NO Continuar con el paso 2

   SI Continuar con el paso 9

9. Fin de ejecución

■

Nota 3.3 *El subconjunto óptimo de características es aquel que maximiza la precisión predictiva utilizando el menor número de rasgos [51]*

Nota 3.4 *El subconjunto óptimo de características se obtiene como resultado de la evaluación del desempeño alcanzado por cada uno de los vectores columna de enmascaramiento posibles; es decir, para encontrar el subconjunto óptimo de características tienen que efectuarse* $(2^f - 1)$ *estimaciones de precisión predictiva, siendo* $\mathbf{f}$ *el número de rasgos presentes en el conjunto original de datos. Claramente cuando* $\mathbf{f}$ *es grande, la búsqueda del subconjunto óptimo de características implica costos computacionales prohibitivos*

Los autores del algoritmo HCM para la reducción dimensional de los datos, describen en su trabajo [66] una serie de experimentos donde se muestra el desempeño alcanzado por este algoritmo en el ámbito del reconocimiento de patrones.

Capítulo 4

Modelo Propuesto

Este capítulo es el más relevante del presente trabajo de tesis. Para la creación, diseño y fundamentación del Enfoque Asociativo para la Selección de Rasgos, que constituye un nuevo modelo para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental, se tomará como punto de partida el modelo de Clasificación Hibrida con Enmascaramiento (*HCM* por sus siglas en inglés) y por otro lado, el concepto de verosimilitud, tomado de la Teoría de Decisión Bayesiana.

Se asume que se tiene un problema de clasificación de patrones, donde el conjunto fundamental es de la forma $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \ldots, p\}$ con $\mathbf{x}^\mu \in \mathbb{R}^n$, $\mathbf{y}^\mu \in A^m$ siendo $n, m, p \in \mathbb{Z}^+$ y $A = \{0, 1\}$.

4.1. Fase de Aprendizaje

Consiste en encontrar los operadores adecuados y una manera de generar una Memoria Asociativa $\mathbf{M}$ que almacene las p asociaciones del conjunto fundamental.

Definición 4.1 *Sean* $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^p$ *elementos del conjunto de patrones de entrada; se define* $\overline{\mathbf{x}}$ *el vector medio de todos ellos, calculado de acuerdo con la siguiente expresión:*

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{\mu=1}^{p} \mathbf{x}^\mu \tag{4.1}$$

Nota 4.1 *Como resultado del cálculo del vector medio de las instancias que conforman el conjunto fundamental, se obtiene un vector n-dimensional que representa el origen del espacio de características que describen el problema a resolver.*

Definición 4.2 *Sean* $\mathbf{x}^{1'}, \mathbf{x}^{2'}, ..., \mathbf{x}^{p'}$ *elementos del conjunto de patrones de entrada trasladados, obtenidos de acuerdo con la siguiente expresión:*

$$\mathbf{x}^{\mu'} = [\mathbf{x}^{\mu} - \overline{\mathbf{x}}] \ \ \forall \mu \in \{1, 2, ..., p\} \tag{4.2}$$

Nota 4.2 *El objetivo que se persigue al trasladar el conjunto fundamental con respecto al vector medio* $\overline{\mathbf{x}}$*, es la representación de las instancias que conforman el conjunto fundamental en un espacio n-dimensional, donde las instancias pertenecientes a una clase se encontrarán ubicadas diametralmente opuestas a las instancias pertenecientes a la otra clase y el punto medio de dicho diámetro está definido por el vector medio* $\overline{\mathbf{x}}$*.*

Definición 4.3 *Sea* $m \in \mathbb{Z}^+$ *el número de clases diferentes, y sea* k *la clase a la cual pertenece cada uno de los patrones de entrada trasladados, con* $k \in \{1, 2, ..., m\}$*.*

Definición 4.4 *Sean* $\mathbf{y}^{1'}, \mathbf{y}^{2'}, ..., \mathbf{y}^{p'}$ *elementos del conjunto de patrones de salida trasladados, donde la* i*-ésima componente de cada uno de los* p *patrones de salida trasladados es obtenida de acuerdo con la siguiente expresión:*

$$y_i^{\mu'} = \begin{cases} 1 & si \quad i = k \\ 0 & si \quad i = 1, 2, ..., k-1, k+1, ...m \end{cases} \quad , \forall \mu \in \{1, 2, ..., p\} \tag{4.3}$$

Nota 4.3 *A cada uno de los* p *patrones de entrada trasladados* $\mathbf{x}^{\mu'} \in A^n$ *que pertenece a la clase* k*, se le asigna un vector de salida trasladado* $\mathbf{y}^{\mu'} \in A^m \ \forall \mu \in \{1, 2, \ldots, p\}$ *de acuerdo con la expresión (4.3) de la definición 4.4.*

4.1.1. Construcción de la Memoria Asociativa

Una vez trasladado todo el conjunto fundamental de patrones, se construye en dos etapas una Memoria Asociativa $\mathbf{M}$ que almacene las p asociaciones del conjunto fundamental trasladado.

1. Para cada una de las p asociaciones del conjunto fundamental trasladado $(\mathbf{x}^{\mu'}, \mathbf{y}^{\mu'})$, obtenga una matriz de dimensiones $m \times n$, de acuerdo con la siguiente expresión:

$$\mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = \begin{pmatrix} y_1^{\mu'} \\ y_2^{\mu'} \\ \vdots \\ y_m^{\mu'} \end{pmatrix} \cdot \left(x_1^{\mu'}, x_2^{\mu'}, ..., x_n^{\mu'}\right) \tag{4.4}$$

$$\mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = \begin{pmatrix} y_1^{\mu'} x_1^{\mu'} & y_1^{\mu'} x_2^{\mu'} & \cdots & y_1^{\mu'} x_j^{\mu'} & \cdots & y_1^{\mu'} x_n^{\mu'} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^{\mu'} x_1^{\mu'} & y_i^{\mu'} x_2^{\mu'} & \cdots & y_i^{\mu'} x_j^{\mu'} & \cdots & y_i^{\mu'} x_n^{\mu'} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^{\mu'} x_1^{\mu'} & y_m^{\mu'} x_2^{\mu'} & \cdots & y_m^{\mu'} x_j^{\mu'} & \cdots & y_m^{\mu'} x_n^{\mu'} \end{pmatrix} \tag{4.5}$$

2. Sumando cada una de las p matrices obtenidas en el paso anterior, obtenga una Memoria Asociativa $\mathbf{M}$, de acuerdo con la siguiente expresión:

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = [m_{ij}]_{m \times n} \tag{4.6}$$

de este modo, la ij-ésima componente de la Memoria Asociativa $\mathbf{M}$ se expresa de la siguiente manera:

$$m_{ij} = \sum_{\mu=1}^{p} y_i^{\mu'} x_j^{\mu'} \tag{4.7}$$

4.2. Selección de Rasgos

Consiste en encontrar los operadores adecuados y una manera de identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado.

Definición 4.5 *Sea E_{ij}^{μ} el error de clasificación de la μ-ésima instancia; calculado a partir del conjunto de aprendizaje, de acuerdo con la siguiente expresión:*

$$E_{ij}^{\mu} = \begin{cases} 1 & si \quad m_{ij} \cdot x_j^{\mu'} \cdot y_i^{\mu'} < 0 \\ 0 & en\ otro\ caso \end{cases} \tag{4.8}$$

$\forall \mu \in \{1, 2, ..., p\}$, $\forall i \in \{1, 2, ..., m\}$ y $\forall j \in \{1, 2, ..., n\}$.

Definición 4.6 *Sea ECA_j el error de clasificación acumulado del j-ésimo rasgo; calculado a partir del conjunto de aprendizaje, de acuerdo con la siguiente expresión:*

$$ECA_j = \sum_{j=1}^{n} E_{ij}^{\mu} \tag{4.9}$$

$\forall \mu \in \{1, 2, ..., p\}$ *y* $\forall i \in \{1, 2, ..., m\}$.

Definición 4.7 *Sea θ un valor de referencia; calculado a partir del error de clasificación acumulado del j-ésimo rasgo, de acuerdo con la siguiente expresión:*

$$\theta = \frac{1}{n} \sum_{j=1}^{n} \left(1 - \left[\frac{1}{p} ECA_j \right] \right) \tag{4.10}$$

Definición 4.8 *Sea $\mathbf{C}$ un vector de restricciones de dimensión n, cuya j-ésima componente se obtiene de acuerdo con la siguiente expresión:*

$$C_j = \begin{cases} 1 & si \quad \left(1 - \left[\frac{1}{p} ECA_j \right] \right) > \theta \\ 0 & en\ otro\ caso \end{cases} \quad , \forall j \in \{1, 2, ..., n\} \tag{4.11}$$

Definición 4.9 *Sea $\mathbf{C}^{abs}$ el vector de restricción absoluta, cuyas n componentes son todas de valor* 0.

Definición 4.10 *Sea $\mathbf{C}^{null}$ el vector de restricción nula, cuyas n componentes son todas de valor* 1.

4.3. Fase de Clasificación

Consiste en encontrar la clase a la cual pertenece un patrón de entrada $\mathbf{x}^{\omega}$dado. Encontrar la clase significa obtener el vector de salida $\mathbf{y}^{\omega} \in A^m$ correspondiente al vector de entrada $\mathbf{x}^{\omega} \in \mathbb{R}^n$ desconocido.

Nota 4.4 *Para cada nuevo patrón de entrada desconocido $\mathbf{x}^{\omega}$ que se desee clasificar, es necesario llevar a cabo el proceso de traslación, de acuerdo con la expresión (4.1) de la definición 4.1; por consiguiente, obtendremos un patrón de entrada desconocido trasladado $\mathbf{x}^{\omega'}$.*

Dado un patrón desconocido trasladado $\mathbf{x}^{\omega'} \in \mathbb{R}^n$, obtener el patrón de salida $\mathbf{y}^{\omega'} \in A^m$ correspondiente al patrón desconocido trasladado $\mathbf{x}^{\omega'}$; donde la i-ésima componente del patrón de salida $y_i^{\omega'}$ recuperado, está dada por la siguiente expresión:

$$y_i^{\omega'} = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^{n} m_{ij} \cdot \left(x_j^{\omega'} \cdot C_j\right) = \bigvee_{h=1}^{k} \left[\sum_{j=1}^{n} m_{hj} \cdot \left(x_j^{\omega'} \cdot C_j\right)\right] \\ 0 & \text{en otro caso} \end{cases} \tag{4.12}$$

donde $\bigvee$ es el operador máximo.

Nota 4.5 *Al tratar de clasificar un patrón de entrada $\mathbf{x}^{\omega}$ dado, aplicando el vector de restricción absoluta $\mathbf{C}^{abs}$ en la expresión (4.1), el patrón de salida $\mathbf{y}^{\omega}$ recuperado es totalmente ambiguo, es decir, no es posible determinar a que clase pertenece el patrón de entrada $\mathbf{x}^{\omega}$; por consiguiente, el vector de restricción absoluta $\mathbf{C}^{abs}$ no forma parte de la solución. No obstante, el vector de restricción absoluta $\mathbf{C}^{abs}$ establece el punto de partida de los algoritmos de reducción dimensional basados en búsquedas hacia adelante [67].*

4.4. Algoritmo Principal

Sean $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^p$ elementos del conjunto de patrones de entrada, de dimensión n, con valores reales en sus componentes; obtener un vector de restricciones que permita identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado, aplicando los pasos que siguen:

1. Se calcula el vector medio del conjunto de patrones de entrada de acuerdo con la expresión (4.1) de la definición 4.1.

2. Se toman las coordenadas del vector medio como origen del conjunto de patrones de entrada.

3. Se realiza la traslación de todos los patrones de entrada del conjunto fundamental de acuerdo con la expresión (4.2) de la definición 4.2.

4. A cada uno de los patrones de entrada trasladados que pertenece a la clase k, se le asigna un vector de salida de dimensión m, de acuerdo con la expresión (4.3) de la definición 4.4.

5. Se construye la Memoria Asociativa **M** de acuerdo con las expresiones (4.4) y (4.6).

6. Se calcula el error de clasificación acumulado de acuerdo con la expresión (4.9) de la definición 4.6, para obtener el valor de referencia θ de acuerdo con la expresión (4.10) de la definición 4.7.

7. Se obtiene el vector de restricciones **C** de acuerdo con la expresión (4.11) de la definición 4.8.

8. Se aplica la fase de Clasificación.

9. Se estima el desempeño alcanzado.

10. FIN de ejecución.

■

Nota 4.6 *El desempeño del modelo propuesto se mide en términos de la tasa de error de clasificación, es decir, la estimación de la precisión predictiva se obtiene tomando en cuenta el número de patrones desconocidos clasificados de manera correcta.*

A continuación se ilustra el proceso de reducción dimensional de los datos mediante un ejemplo.

Ejemplo 4.1 *Sean* $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \ldots, p\}$ *los elementos que definen el conjunto fundamental de patrones. Se tienen 8 asociaciones de patrones agrupados en dos diferentes clases con igual número de patrones de entrada en cada una de las clases.*

Sean $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ y $\mathbf{x}^4$ patrones de entrada del conjunto fundamental, pertenecientes a la clase 1

$$\mathbf{x}^1 = \begin{pmatrix} 1.40 \\ 1.70 \\ 0.69 \\ -1.10 \\ 0.69 \\ 2.14 \\ -0.30 \\ -0.27 \end{pmatrix}; \mathbf{x}^2 = \begin{pmatrix} 1.50 \\ -1.32 \\ 0.66 \\ 1.40 \\ 0.66 \\ -0.76 \\ -0.40 \\ -0.27 \end{pmatrix}; \mathbf{x}^3 = \begin{pmatrix} 1.20 \\ -1.27 \\ 0.56 \\ -1.20 \\ 0.56 \\ 1.10 \\ -0.50 \\ -0.27 \end{pmatrix}; \mathbf{x}^4 = \begin{pmatrix} 1.50 \\ 1.46 \\ -0.05 \\ -1.20 \\ 0.18 \\ -0.30 \\ -0.60 \\ -0.27 \end{pmatrix}$$

y sean $\mathbf{y}^1 = \mathbf{y}^2 = \mathbf{y}^3 = \mathbf{y}^4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ sus correspondientes etiquetas de clase.

Sean $\mathbf{x}^5, \mathbf{x}^6, \mathbf{x}^7$ y $\mathbf{x}^8$ patrones de entrada del conjunto fundamental, pertenecientes a la clase 2

$$\mathbf{x}^5 = \begin{pmatrix} -1.20 \\ -1.10 \\ -0.42 \\ 1.40 \\ 0.28 \\ -0.54 \\ 0.07 \\ -0.02 \end{pmatrix}; \mathbf{x}^6 = \begin{pmatrix} -1.20 \\ -1.58 \\ -0.44 \\ 0.21 \\ 0.58 \\ -0.74 \\ 0.13 \\ -0.27 \end{pmatrix}; \mathbf{x}^7 = \begin{pmatrix} 1.40 \\ -1.10 \\ -0.28 \\ 1.20 \\ -0.42 \\ -1.20 \\ 0.13 \\ -0.28 \end{pmatrix}; \mathbf{x}^8 = \begin{pmatrix} 1.10 \\ 1.60 \\ -0.44 \\ 1.40 \\ -0.42 \\ -0.84 \\ 0.13 \\ -0.29 \end{pmatrix}$$

y sean $\mathbf{y}^5 = \mathbf{y}^6 = \mathbf{y}^7 = \mathbf{y}^8 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ sus correspondientes etiquetas de clase.

Primeramente se calcula el vector medio $\overline{\mathbf{x}}$ del conjunto de patrones de entrada de acuerdo con la expresión (4.1) de la definición 4.1. Esto es:

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{\mu=1}^{p} \mathbf{x}^{\mu} = \frac{1}{8} \left[\mathbf{x}^1 + \mathbf{x}^2 + \ldots + \mathbf{x}^8 \right]$$

$$\overline{\mathbf{x}} = \frac{1}{8}\left[\begin{pmatrix} 1.40 \\ 1.70 \\ 0.69 \\ -1.10 \\ 0.69 \\ 2.14 \\ -0.30 \\ -0.27 \end{pmatrix} + \begin{pmatrix} 1.50 \\ -1.32 \\ 0.66 \\ 1.40 \\ 0.66 \\ -0.76 \\ -0.40 \\ -0.27 \end{pmatrix} + \ldots + \begin{pmatrix} 1.10 \\ 1.60 \\ -0.44 \\ 1.40 \\ -0.42 \\ -0.84 \\ 0.13 \\ -0.29 \end{pmatrix}\right]$$

$$\overline{\mathbf{x}} = \begin{pmatrix} 0.712 \\ -0.201 \\ 0.035 \\ 0.263 \\ 0.263 \\ -0.142 \\ -0.167 \\ -0.242 \end{pmatrix}$$

Una vez que se tiene el vector medio $\overline{\mathbf{x}}$, se realiza la traslación de todos los patrones de entrada del conjunto fundamental de acuerdo con la expresión (4.2) de la definición 4.2.Una vez trasladados todos los vectores que conforman el conjunto fundamental, se obtiene un nuevo conjunto de vectores trasladados, denotado como: $\left\{(\mathbf{x}^{\mu'}, \mathbf{y}^{\mu'}) \mid \mu = 1, 2, \ldots, p\right\}$, donde los patrones pertenecientes a la clase 1 son los siguientes:

$$\mathbf{x}^{1'} = \begin{pmatrix} 0.69 \\ 1.90 \\ 0.66 \\ -1.36 \\ 0.43 \\ 2.28 \\ -0.13 \\ -0.03 \end{pmatrix}; \mathbf{x}^{2'} = \begin{pmatrix} 0.79 \\ -1.12 \\ 0.63 \\ 1.14 \\ 0.40 \\ -0.62 \\ -0.23 \\ -0.03 \end{pmatrix}; \mathbf{x}^{3'} = \begin{pmatrix} 0.49 \\ -1.07 \\ 0.52 \\ -1.46 \\ 0.29 \\ 1.24 \\ -0.33 \\ -0.03 \end{pmatrix}; \mathbf{x}^{4'} = \begin{pmatrix} 0.79 \\ 1.66 \\ -0.09 \\ -1.46 \\ -0.08 \\ -0.16 \\ -0.43 \\ -0.03 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{1'} = \mathbf{y}^{2'} = \mathbf{y}^{3'} = \mathbf{y}^{4'} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, obtenidas de acuerdo con la expresión (4.3) de la definición 4.4.

Del mismo modo que con los patrones de pertenecientes a la clase 1, se aplican los pasos 2, 3 y 4 del algoritmo propuesto sobre los patrones pertenecientes a la clase 2. Esto es:

$$\mathbf{x}^{5'} = \begin{pmatrix} -1.91 \\ -0.90 \\ -0.45 \\ 1.14 \\ 0.01 \\ -0.40 \\ 0.24 \\ 0.22 \end{pmatrix}; \mathbf{x}^{6'} = \begin{pmatrix} -1.91 \\ -1.38 \\ -0.48 \\ -0.05 \\ 0.32 \\ -0.60 \\ 0.30 \\ -0.03 \end{pmatrix}; \mathbf{x}^{7'} = \begin{pmatrix} 0.69 \\ -0.90 \\ -0.31 \\ 0.94 \\ -0.68 \\ -1.06 \\ 0.30 \\ -0.04 \end{pmatrix}; \mathbf{x}^{8'} = \begin{pmatrix} 0.39 \\ 1.80 \\ -0.47 \\ 1.14 \\ -0.68 \\ -0.70 \\ 0.30 \\ -0.05 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{5'} = \mathbf{y}^{6'} = \mathbf{y}^{7'} = \mathbf{y}^{8'} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, obtenidas de acuerdo con la expresión (4.3) de la definición 4.4.

Tal como se indica en el paso 5 del algoritmo propuesto, se toma el conjunto fundamental de patrones trasladados para obtener las p matrices requeridas en la fase de aprendizaje.

A continuación se muestra el resultado del aprendizaje de algunas asociaciones de patrones trasladados.

$$\mathbf{y}^{1'} \cdot \left(\mathbf{x}^{1'}\right)^t = \begin{pmatrix} 0.69 & 1.90 & 0.66 & -1.36 & 0.43 & 2.28 & -0.13 & -0.03 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{y}^{2'} \cdot \left(\mathbf{x}^{2'}\right)^t = \begin{pmatrix} 0.79 & -1.12 & 0.63 & 1.14 & 0.40 & -0.62 & -0.23 & -0.03 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\vdots$$

$$\mathbf{y}^{7'} \cdot \left(\mathbf{x}^{7'}\right)^t = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.69 & -0.90 & -0.31 & 0.94 & -0.68 & -1.06 & 0.30 & -0.04 \end{pmatrix}$$

$$\mathbf{y}^{8'} \cdot \left(\mathbf{x}^{8'}\right)^t = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.39 & 1.80 & -0.47 & 1.14 & -0.68 & -0.70 & 0.30 & -0.05 \end{pmatrix}$$

Al aplicar las expresiones (4.4) y (4.6) sobre las p matrices obtenidas en el paso 5 del algoritmo propuesto, se obtiene la Memoria Asociativa $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} 2.750 & 1.375 & 1.718 & -3.155 & 1.032 & 2.752 & -1.129 & -0.117 \\ -2.750 & -1.375 & -1.718 & 3.155 & -1.032 & -2.752 & 1.129 & 0.117 \end{pmatrix}$$

Una vez que se tiene la Memoria Asociativa $\mathbf{M}$, continuamos con el cálculo del error de clasificación acumulado del j-ésimo rasgo, aplicando la expresión (4.9) de la definición 4.6.

Para esclarecer el cálculo del error de clasificación acumulado del j-ésimo rasgo ECA_j, primeramente obtendremos el error de clasificación de la μ-ésima instancia para la i-ésima clase en el j-ésimo rasgo, es decir, obtendremos E_{ij}^{μ}, aplicando la expresión (4.8) de la definición 4.5. A continuación se muestra el resumen de resultados del cálculo del error de clasificación acumulado del j-ésimo rasgo, cuando $j = 1$.

			m_{ij}	$x_j^{\mu'}$	$y_i^{\mu'}$				E_{ij}^{μ}
$j=1$	$i=1$	$\mu=1$	2.750	0.69	1	$<$	0	$\rightarrow$	0
		$\mu=2$	2.750	0.79	1	$<$	0	$\rightarrow$	0
		$\mu=3$	2.750	0.49	1	$<$	0	$\rightarrow$	0
		$\mu=4$	2.750	0.79	1	$<$	0	$\rightarrow$	0
		$\mu=5$	2.750	−1.91	0	$<$	0	$\rightarrow$	0
		$\mu=6$	2.750	−1.91	0	$<$	0	$\rightarrow$	0
		$\mu=7$	2.750	0.69	0	$<$	0	$\rightarrow$	0
		$\mu=8$	2.750	0.39	0	$<$	0	$\rightarrow$	0
							ECA_j	$=$	0
$j=1$	$i=2$	$\mu=1$	−2.750	0.69	0	$<$	0	$\rightarrow$	0
		$\mu=2$	−2.750	0.79	0	$<$	0	$\rightarrow$	0
		$\mu=3$	−2.750	0.49	0	$<$	0	$\rightarrow$	0
		$\mu=4$	−2.750	0.79	0	$<$	0	$\rightarrow$	0
		$\mu=5$	−2.750	−1.91	1	$<$	0	$\rightarrow$	0
		$\mu=6$	−2.750	−1.91	1	$<$	0	$\rightarrow$	0
		$\mu=7$	−2.750	0.69	1	$\not<$	0	$\rightarrow$	1
		$\mu=8$	−2.750	0.39	1	$\not<$	0	$\rightarrow$	1
							ECA_j	$=$	2

De acuerdo con la expresión (4.8) de la definición 4.5, se puede apreciar que en el primer rasgo ($j = 1$) únicamente se tienen errores de clasificación para la segunda clase ($i = 2$) en la séptima y octava instancias de aprendizaje ($\mu = 7$ y $\mu = 8$); consecuentemente, el error de clasificación acumulado del j-ésimo rasgo ECA_j es igual con 2. Del mismo modo se continúa con el cálculo del error de clasificación acumulado del j-ésimo rasgo ECA_j para cada uno de los j rasgos.

A continuación se muestran los resultados del cálculo del error de clasificación acumulado del j-ésimo rasgo ECA_j para cada uno de los j rasgos; donde m_{ij} es el contenido de la Memoria Asociativa $\mathbf{M}$ para la i-ésima clase en el j-ésimo rasgo.

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
m_{1j}	2.750	1.375	1.718	−3.155	1.032	2.752	−1.129	−0.117
m_{2j}	−2.750	−1.375	−1.718	3.155	−1.032	−2.752	1.129	0.117
ECA_j	2	3	1	2	3	2	0	3

Una vez que se tiene el valor del error de clasificación acumulado del j-ésimo rasgo ECA_j para cada uno de los j rasgos, se obtiene el valor de referencia θ de acuerdo con la expresión (4.10) de la definición 4.7. Esto es:

$$\theta = \frac{1}{8}\left[\begin{array}{l} \left[1-\frac{1}{8}2\right]+\left[1-\frac{1}{8}3\right]+\left[1-\frac{1}{8}1\right]+\left[1-\frac{1}{8}2\right]+\ldots \\ \ldots+\left[1-\frac{1}{8}3\right]+\left[1-\frac{1}{8}2\right]+\left[1-\frac{1}{8}0\right]+\left[1-\frac{1}{8}3\right] \end{array}\right]$$

$$\theta = \frac{1}{8}\left[\frac{48}{8}\right]$$

$$\theta = 0.75$$

Una vez que se tiene el valor de referencia θ, se continúa con la construcción del vector de restricciones $\mathbf{C}$ de dimensión n.

Para obtener cada una de las n componentes del vector de restricciones $\mathbf{C}$, tomaremos cada uno de los j valores del error de clasificación acumulado y lo compararemos contra el valor de referencia θ, de acuerdo con la expresión (4.11) de la definición 4.8.

Para esclarecer el procedimiento mediante el cual se obtiene el vector de restricciones $\mathbf{C}$ de dimensión n, tomaremos el valor del error de clasificación acumulado del primer rasgo ECA_1 y lo compararemos contra el valor de referencia θ. Aplicando $ECA_1 = 2$ y $\theta = 0.75$ en la expresión (4.11) de la definición 4.8, tenemos que $\left[1-\frac{1}{8}2\right] \not> 0.75$, lo cual implica que la primera componente del vector de restricciones $\mathbf{C}$ es igual con 0, es decir, $C_1 = 0$. Del mismo modo se continúa con cada uno de los j valores del error de clasificación acumulado para obtener el vector de restricciones $\mathbf{C}$.

A continuación se muestran los resultados del cálculo de cada una de las componentes del vector de restricciones $\mathbf{C}$.

Para $j = 1$ tenemos que $ECA_1 = 2$, esto es $\left[1-\frac{1}{8}2\right] \not> 0.75 \rightarrow C_1 = 0$

Para $j = 2$ tenemos que $ECA_2 = 3$, esto es $\left[1 - \frac{1}{8}3\right] \not> 0.75 \rightarrow C_2 = 0$
Para $j = 3$ tenemos que $ECA_3 = 1$, esto es $\left[1 - \frac{1}{8}1\right] > 0.75 \rightarrow C_3 = 1$
Para $j = 4$ tenemos que $ECA_4 = 2$, esto es $\left[1 - \frac{1}{8}2\right] \not> 0.75 \rightarrow C_4 = 0$
Para $j = 5$ tenemos que $ECA_5 = 3$, esto es $\left[1 - \frac{1}{8}3\right] \not> 0.75 \rightarrow C_5 = 0$
Para $j = 6$ tenemos que $ECA_6 = 2$, esto es $\left[1 - \frac{1}{8}2\right] \not> 0.75 \rightarrow C_6 = 0$
Para $j = 7$ tenemos que $ECA_7 = 0$, esto es $\left[1 - \frac{1}{8}0\right] > 0.75 \rightarrow C_7 = 1$
Para $j = 8$ tenemos que $ECA_8 = 3$, esto es $\left[1 - \frac{1}{8}3\right] \not> 0.75 \rightarrow C_8 = 0$

Así pues, el vector de restricciones $\mathbf{C}$ de dimensión n obtenido es el siguiente:

$$\mathbf{C} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Una vez que se tiene el vector de restricciones $\mathbf{C}$ de dimensión n, se continúa con el paso 8 del algoritmo propuesto, aplicando la expresión (4.12) sobre cada uno de los elementos que definen el conjunto fundamental de patrones trasladados.

Con la finalidad de ilustrar qué es lo que sucede cuando existen rasgos irrelevantes (para fines de clasificación) en las instancias que conforman el conjunto fundamental de patrones trasladados, se aplica el vector de restricción nula $\mathbf{C}^{null}$ en la expresión (4.12) para cada uno de los elementos que definen el conjunto fundamental de patrones trasladados.

$$\mathbf{y}^{1'} = \mathbf{M} \cdot \left(\mathbf{x}^{1'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 16.81 \\ -16.81 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{1'} = \mathbf{y}^{1} \quad \text{Correcto}$$

$$\mathbf{y}^{2'} = \mathbf{M} \cdot \left(\mathbf{x}^{2'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -2.91 \\ 2.91 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{2'} \neq \mathbf{y}^{2} \quad \textbf{Incorrecto}$$

$$\mathbf{y}^{3'} = \mathbf{M} \cdot \left(\mathbf{x}^{3'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 9.50 \\ -9.50 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{3'} = \mathbf{y}^{3} \quad \text{Correcto}$$

$$\mathbf{y}^{4'} = \mathbf{M} \cdot \left(\mathbf{x}^{4'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 8.89 \\ -8.89 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{4'} = \mathbf{y}^{4} \quad \text{Correcto}$$

$$\mathbf{y}^{5'} = \mathbf{M} \cdot \left(\mathbf{x}^{5'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -12.24 \\ 12.24 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{5'} = \mathbf{y}^{5} \quad \text{Correcto}$$

$$\mathbf{y}^{6'} = \mathbf{M} \cdot \left(\mathbf{x}^{6'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -9.46 \\ 9.46 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{6'} = \mathbf{y}^{6} \quad \text{Correcto}$$

$$\mathbf{y}^{7'} = \mathbf{M} \cdot \left(\mathbf{x}^{7'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -6.78 \\ 6.78 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{7'} = \mathbf{y}^{7} \quad \text{Correcto}$$

$$\mathbf{y}^{8'} = \mathbf{M} \cdot \left(\mathbf{x}^{8'} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -3.81 \\ 3.81 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{8'} = \mathbf{y}^{8} \quad \text{Correcto}$$

Se puede observar que al considerar todas las componentes de cada uno de los patrones de entrada, el patrón de salida $\mathbf{y}^{2'}$ recuperado, asociado con el patrón de entrada $\mathbf{x}^{2'}$, NO es correctamente recuperado.

Para ilustrar qué es lo que sucede cuando se identifican aquellas características irrelevantes (para fines de clasificación) en las instancias que conforman el conjunto fundamental de patrones trasladados, se aplicará el vector de restricción $\mathbf{C}$, obtenido en el paso 7 del algoritmo propuesto, para cada uno de los elementos que definen el conjunto fundamental de patrones trasladados.

$$\mathbf{y}^{1'} = \mathbf{M} \cdot \left(\mathbf{x}^{1'} \cdot \mathbf{C}\right) = \begin{pmatrix} 1.276 \\ -1.276 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{1'} = \mathbf{y}^{1} \quad \text{Correcto}$$

$$\mathbf{y}^{2'} = \mathbf{M} \cdot \left(\mathbf{x}^{2'} \cdot \mathbf{C}\right) = \begin{pmatrix} 1.338 \\ -1.338 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{2'} = \mathbf{y}^{2} \quad \text{Correcto}$$

$$\mathbf{y}^{3'} = \mathbf{M} \cdot \left(\mathbf{x}^{3'} \cdot \mathbf{C}\right) = \begin{pmatrix} 1.279 \\ -1.279 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{3'} = \mathbf{y}^{3} \quad \text{Correcto}$$

$$\mathbf{y}^{4'} = \mathbf{M} \cdot \left(\mathbf{x}^{4'} \cdot \mathbf{C}\right) = \begin{pmatrix} 0.343 \\ -0.343 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{4'} = \mathbf{y}^{4} \quad \text{Correcto}$$

$$\mathbf{y}^{5'} = \mathbf{M} \cdot \left(\mathbf{x}^{5'} \cdot \mathbf{C}\right) = \begin{pmatrix} -1.051 \\ 1.051 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{5'} = \mathbf{y}^{5} \quad \text{Correcto}$$

$$\mathbf{y}^{6'} = \mathbf{M} \cdot \left(\mathbf{x}^{6'} \cdot \mathbf{C}\right) = \begin{pmatrix} -1.153 \\ 1.153 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{6'} = \mathbf{y}^{6} \quad \text{Correcto}$$

$$\mathbf{y}^{7'} = \mathbf{M} \cdot \left(\mathbf{x}^{7'} \cdot \mathbf{C}\right) = \begin{pmatrix} -0.878 \\ 0.878 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{7'} = \mathbf{y}^{7} \quad \text{Correcto}$$

$$\mathbf{y}^{8'} = \mathbf{M} \cdot \left(\mathbf{x}^{8'} \cdot \mathbf{C}\right) = \begin{pmatrix} -1.153 \\ 1.153 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{8'} = \mathbf{y}^{8} \quad \text{Correcto}$$

Como se puede observar, todos los patrones del conjunto fundamental fueron correctamente clasificados.

Analizando el vector de restricción $\mathbf{C}$, obtenido en el paso 7 del algoritmo propuesto, hay que señalar que tanto la tercera como la séptima componentes tienen valor igual con 1, mientras que las demás componentes ($1, 2, 4, 5, 6$ y 8) tienen valor igual con 0; esto implica que tanto la tercera como la séptima componentes de cada una de las instancias que conforman el conjunto fundamental de patrones son rasgos relevantes (para fines de clasificación).

Para mostrar la eficacia del modelo propuesto en la obtención de una representación equivalente (dimensionalmente menor) del conjunto fundamental, a continuación se alterarán cada uno de los patrones de entrada trasladados, aplicando la operación siguiente:

$$x_j^{\mu''} = x_j^{\mu'} \cdot C_j \quad , \forall \mu \in \{1, 2, ..., p\} \tag{4.13}$$

Aplicando la expresión (4.13) para cada uno de los patrones de entrada trasladados, obtenemos un conjunto fundamental trasladado restringido por $\mathbf{C}$, donde los patrones

pertenecientes a la clase 1 son los siguientes:

$$\mathbf{x}^{1''} = \begin{pmatrix} 0 \\ 0 \\ 0.66 \\ 0 \\ 0 \\ 0 \\ -0.13 \\ 0 \end{pmatrix}; \mathbf{x}^{2''} = \begin{pmatrix} 0 \\ 0 \\ 0.63 \\ 0 \\ 0 \\ 0 \\ -0.23 \\ 0 \end{pmatrix}; \mathbf{x}^{3''} = \begin{pmatrix} 0 \\ 0 \\ 0.52 \\ 0 \\ 0 \\ 0 \\ -0.33 \\ 0 \end{pmatrix}; \mathbf{x}^{4''} = \begin{pmatrix} 0 \\ 0 \\ -0.09 \\ 0 \\ 0 \\ 0 \\ -0.43 \\ 0 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{1''} = \mathbf{y}^{2''} = \mathbf{y}^{3''} = \mathbf{y}^{4''} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Del mismo modo que con los patrones de pertenecientes a la clase 1, se aplica la expresión (4.13) sobre los patrones pertenecientes a la clase 2. Esto es:

$$\mathbf{x}^{5''} = \begin{pmatrix} 0 \\ 0 \\ -0.45 \\ 0 \\ 0 \\ 0 \\ 0.24 \\ 0 \end{pmatrix}; \mathbf{x}^{6''} = \begin{pmatrix} 0 \\ 0 \\ -0.48 \\ 0 \\ 0 \\ 0 \\ 0.30 \\ 0 \end{pmatrix}; \mathbf{x}^{7''} = \begin{pmatrix} 0 \\ 0 \\ -0.31 \\ 0 \\ 0 \\ 0 \\ 0.30 \\ 0 \end{pmatrix}; \mathbf{x}^{8''} = \begin{pmatrix} 0 \\ 0 \\ -0.47 \\ 0 \\ 0 \\ 0 \\ 0.30 \\ 0 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{5''} = \mathbf{y}^{6''} = \mathbf{y}^{7''} = \mathbf{y}^{8''} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Al eliminar las componentes con valor igual con cero en cada uno de los patrones de entrada del conjunto fundamental trasladado restringido por $\mathbf{C}$, obtenemos un conjunto

fundamental trasladado dimensionalmente menor. Esto es:

$$\mathbf{x}^{1''} = \begin{pmatrix} 0.66 \\ -0.13 \end{pmatrix}; \mathbf{x}^{2''} = \begin{pmatrix} 0.63 \\ -0.23 \end{pmatrix}; \mathbf{x}^{3''} = \begin{pmatrix} 0.52 \\ -0.33 \end{pmatrix}; \mathbf{x}^{4''} = \begin{pmatrix} -0.09 \\ -0.43 \end{pmatrix}$$

$$\mathbf{x}^{5''} = \begin{pmatrix} -0.45 \\ 0.24 \end{pmatrix}; \mathbf{x}^{6''} = \begin{pmatrix} -0.48 \\ 0.30 \end{pmatrix}; \mathbf{x}^{7''} = \begin{pmatrix} -0.31 \\ 0.30 \end{pmatrix}; \mathbf{x}^{8''} = \begin{pmatrix} -0.47 \\ 0.30 \end{pmatrix}$$

y sus correspondientes etiquetas de clase son: $\mathbf{y}^{1''} = \mathbf{y}^{2''} = \mathbf{y}^{3''} = \mathbf{y}^{4''} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $\mathbf{y}^{5''} = \mathbf{y}^{6''} = \mathbf{y}^{7''} = \mathbf{y}^{8''} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Al aplicar las expresiones (4.4) y (4.6), se obtiene la Memoria Asociativa $\mathbf{M}''$ (dimensionalmente menor). Esto es:

$$\mathbf{M}'' = \begin{pmatrix} 1.718 & -1.129 \\ -1.718 & 1.129 \end{pmatrix}$$

Para comprobar que el conjunto fundamental de patrones trasladados y su representación dimensionalmente menor es equivalente (para fines de clasificación), se aplica la fase de clasificación sin restricciones, es decir, se aplicará el vector de restricción nula $\mathbf{C}^{null}$ en la expresión (4.12) para cada uno de los elementos que definen el conjunto fundamental de patrones trasladados dimensionalmente menores.

$$\mathbf{y}^{1''} = \mathbf{M}''\cdot\left(\mathbf{x}^{1''} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 1.276 \\ -1.276 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{1''} = \mathbf{y}^{1} \quad \text{Correcto}$$

$$\mathbf{y}^{2''} = \mathbf{M}''\cdot\left(\mathbf{x}^{2''} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 1.338 \\ -1.338 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{2''} = \mathbf{y}^{2} \quad \text{Correcto}$$

$$\mathbf{y}^{3''} = \mathbf{M}''\cdot\left(\mathbf{x}^{3''} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 1.279 \\ -1.279 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{3''} = \mathbf{y}^{3} \quad \text{Correcto}$$

$$\mathbf{y}^{4''} = \mathbf{M}''\cdot\left(\mathbf{x}^{4''} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} 0.343 \\ -0.343 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad \mathbf{y}^{4''} = \mathbf{y}^{4} \quad \text{Correcto}$$

$$\mathbf{y}^{5''} = \mathbf{M}''\cdot\left(\mathbf{x}^{5''} \cdot \mathbf{C}^{null}\right) = \begin{pmatrix} -1.051 \\ 1.051 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{5''} = \mathbf{y}^{5} \quad \text{Correcto}$$

$$\mathbf{y}^{6''} = \mathbf{M}''\cdot\left(\mathbf{x}^{6''}\cdot\mathbf{C}^{null}\right) = \begin{pmatrix} -1.153 \\ 1.153 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{6''} = \mathbf{y}^{6} \quad \text{Correcto}$$

$$\mathbf{y}^{7''} = \mathbf{M}''\cdot\left(\mathbf{x}^{7''}\cdot\mathbf{C}^{null}\right) = \begin{pmatrix} -0.878 \\ 0.878 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{7''} = \mathbf{y}^{7} \quad \text{Correcto}$$

$$\mathbf{y}^{8''} = \mathbf{M}''\cdot\left(\mathbf{x}^{8''}\cdot\mathbf{C}^{null}\right) = \begin{pmatrix} -1.153 \\ 1.153 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad \mathbf{y}^{8''} = \mathbf{y}^{8} \quad \text{Correcto}$$

Como se puede observar, todas las instancias del conjunto fundamental de patrones trasladados dimensionalmente menores fueron correctamente clasificadas. Evidentemente, los resultados alcanzados con el conjunto fundamental de patrones trasladados y su representación dimensionalmente menor son equivalentes (para fines de clasificación).

Cabe mencionar que la representación dimensionalmente menor es 75 % más pequeña con respecto a la dimensión original de los patrones que conforman el conjunto fundamental.

Al aplicar el modelo propuesto para la reducción dimensional de los datos, en este caso, es posible reducir la dimensionalidad de los patrones que conforman el conjunto fundamental al 25 %, sin afectar el índice de clasificación.

Capítulo 5

Resultados y Discusión

Este capítulo es de vital importancia en el presente trabajo de tesis, puesto que no solo se ilustran los conceptos descritos en los capítulos anteriores; también se muestra la eficacia del modelo propuesto en tareas de selección de rasgos para la reducción dimensional de los datos, así como el desempeño alcanzado en diferentes bases de datos, tomadas del repositorio de bases de datos de la Universidad de California en Irvine [52].

5.1. Aplicación en Bases de Datos

En cada una de las secciones que conforman este capítulo se presentan de manera sucinta las características de cada uno de los conjuntos de datos utilizados a lo largo de la fase experimental, así como una breve descripción del ámbito en el que han sido aplicados. Cabe mencionar que con la finalidad de que pueda existir una comparación coherente entre los hallazgos experimentales presentados en este capítulo y los resultados experimentales publicados en la literatura científica actual, los índices de clasificación fueron obtenidos aplicando técnicas de validación cruzada; concretamente, K-Fold Cross Validation con K=10 fue aplicada.

Los experimentos se realizaron en una computadora personal (PC), con un procesador Intel Core2 Duo a 2.13 GHz, 2048 MBytes de memoria RAM y 73.2 GBytes de espacio libre en disco duro. Se utilizó el paquete computacional MatLab versión R14 de The MathWorks, Inc., corriendo sobre Windows XP Profesional de Microsoft.

5.1.1. Breast Cancer Database

Esta base de datos fue integrada en el Hospital de la Universidad de Wisconsin, Madison, gracias al Dr. William H. Wolberg y se encuentra disponible en el repositorio de bases de datos de la Universidad de California en Irvine [52]. El conjunto de datos fue integrado a partir de reportes clínicos periódicos de biología celular; por consiguiente, la base de datos refleja un agrupamiento cronológico en ocho diferentes grupos. Cada instancia está conformada por nueve rasgos numéricos y una etiqueta de clase; naturalmente, cada una de las instancias pertenece a una de dos posibles clases: benigno o maligno. El 65.5 % de las instancias contenidas en la base de datos pertenece a la clase 1 (benigno), mientras que el 34.5 % restante pertenece a la clase 2 (maligno). Esta base de datos ha sido ampliamente utilizada para diagnostico médico en citología de cáncer de seno (mama) [68]. Cabe mencionar que la base de datos está conformada por 699 instancias, de las cuales 16 presentan valores faltantes; por ende, estas últimas no fueron consideradas durante la fase experimental del presente trabajo de tesis.

Experimento 5.1 *Dado el conjunto de instancias disponibles en la base de datos* ***Breast Cancer Database****; aplicar los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4 para obtener un vector de restricciones* $\mathbf{C}$ *que permita identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado. Posteriormente, aplicar los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4 para estimar la precisión predictiva del modelo propuesto, aplicando técnicas de validación cruzada; concretamente, aplicar K-Fold Cross Validation con K=10.*

A continuación se muestra el resultado de la ejecución de los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4.

** Breast Cancer Database **
El rendimiento alcanzado usando el rasgo [1] fue: 80.3807 %
El rendimiento alcanzado usando el rasgo [2] fue: 92.9722 %
El rendimiento alcanzado usando el rasgo [3] fue: 92.5329 %
El rendimiento alcanzado usando el rasgo [4] fue: 86.6764 %
El rendimiento alcanzado usando el rasgo [5] fue: 87.8477 %
El rendimiento alcanzado usando el rasgo [6] fue: 91.2152 %
El rendimiento alcanzado usando el rasgo [7] fue: 91.8009 %
El rendimiento alcanzado usando el rasgo [8] fue: 86.8228 %
El rendimiento alcanzado usando el rasgo [9] fue: 78.7701 %

El valor de referencia θ fue: 87.6688 %

En la Figura 5.1 se muestra como se encuentran distribuidos los patrones de entrenamiento, para cada uno de los 9 rasgos, en cada una de las dos posibles clases: benigno o maligno. Es necesario hacer notar que tomando en cuenta el valor de referencia θ, obtenido mediante la expresión (4.10) de la definición 4.7, los rasgos que contribuyen mayormente de manera univariable a la separación entre clases son los siguientes: $2, 3, 5, 6, 7$; tal como se muestra en la Figura 5.2.

Una vez que se tienen identificados los rasgos que contribuyen mayormente de manera univariable a la separación entre clases, se obtiene el vector de restricciones $\mathbf{C}$, aplicando el paso 7 del algoritmo propuesto en el Capítulo 4.

A continuación se muestra el resultado de la ejecución de los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4.

Un total de [24] errores de clasificación en [680] instancias.
El rendimiento promedio fue de [96.4706 %] de precisión predictiva.
El vector de restricciones es: [0 1 1 0 1 1 1 0 0] = 220
El número de rasgos seleccionados fue: 5 de 9
Logrando eliminar [44.4444 %] del espacio original.

Con la finalidad de obtener una estimación confiable del comportamiento del modelo propuesto en presencia de instancias no conocidas, se aplicó la técnica de validación cruzada K-Fold Cross Validation con K=10. Los resultados de las K estimaciones de la

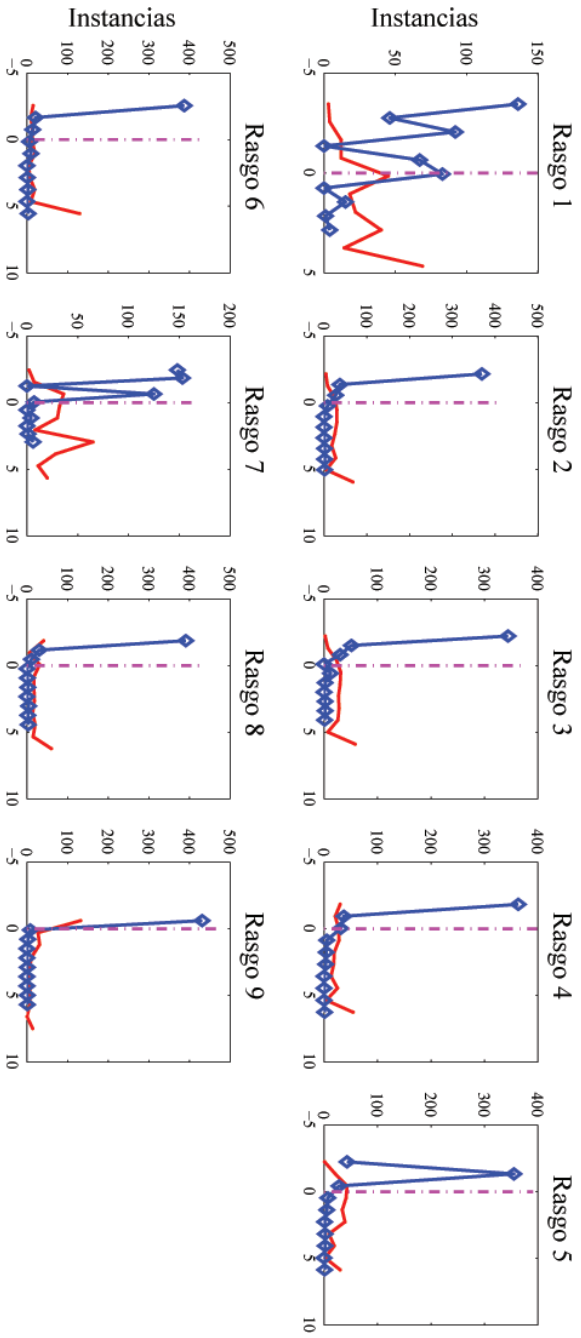


Figura 5.1: Función de Verosimilitud Univariable. Breast Cancer Database.

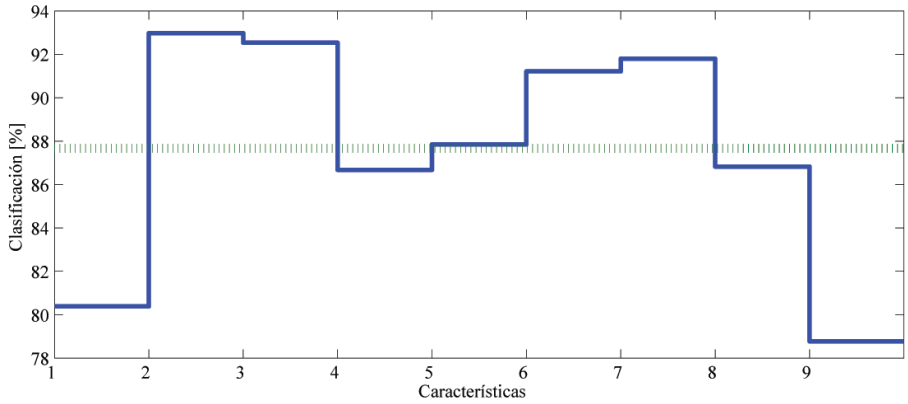


Figura 5.2: Clasificación Univariable. Breast Cancer Database.

Tabla 5.1: Clasificación Multivariable. Breast Cancer Database.

Máscara decimal	Máscara binaria	Desempeño alcanzado
511	1 1 1 1 1 1 1 1 1	97.51 %
220	0 1 1 0 1 1 1 0 0	96.47 %

precisión predictiva del modelo propuesto, para cada uno de los K subconjuntos de prueba mutuamente excluyentes, se muestran en la Figura 5.3. Asimismo, en la Tabla 5.1 se muestran los resultados promediados de la estimación de la precisión predictiva del modelo propuesto, tanto para el conjunto completo de rasgos, así como para el subconjunto de características seleccionadas mediante el vector de restricciones $\mathbf{C}$.

5.1.2. Heart Disease Database

Esta base de datos fue integrada a partir de 270 consultas médicas efectuadas conjuntamente por el V.A. Medical Center en Long Beach, California y por la Cleveland Clinic Foundation. El conjunto de datos fue integrado por el Dr. Robert Detrano y se encuentra disponible en el repositorio de bases de datos de la Universidad de California en Irvine [52]. Cada instancia del conjunto original de datos está conformada por 75 rasgos numéricos y una etiqueta de clase. Esta base de datos ha sido ampliamente utilizada para pronosticar la presencia de enfermedades cardíacas en seres humanos [69]. Cabe mencionar que, aun cuando cada una de las instancias de esta base de datos está conformada por 75 rasgos numéricos

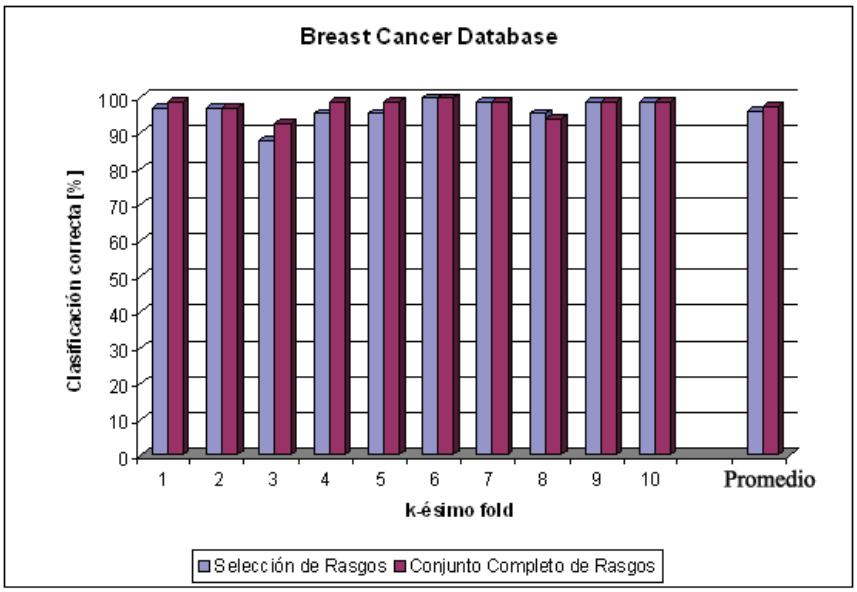


Figura 5.3: Clasificación Multivariable. Breast Cancer Database.

y una etiqueta de clase, en la literatura científica actual, únicamente han sido publicados resultados experimentales usando un subconjunto de 13 características y una etiqueta de clase. Con la finalidad de que pueda existir una comparación coherente entre los hallazgos experimentales presentados en esta sección y los resultados experimentales publicados en la literatura científica actual [70], únicamente fueron consideradas 13 características numéricas y una etiqueta de clase por instancia.

Experimento 5.2 *Dado el conjunto de instancias disponibles en la base de datos* ***Heart Disease Database****; aplicar los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4 para obtener un vector de restricciones* $\mathbf{C}$ *que permita identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado. Posteriormente, aplicar los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4 para estimar la precisión predictiva del modelo propuesto, aplicando técnicas de validación cruzada; concretamente, aplicar K-Fold Cross Validation con K=10.*

A continuación se muestra el resultado de la ejecución de los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4.

** Heart Disease Database **
El rendimiento alcanzado usando el rasgo [1] fue: 64.4444 %
El rendimiento alcanzado usando el rasgo [2] fue: 61.8519 %
El rendimiento alcanzado usando el rasgo [3] fue: 75.1852 %
El rendimiento alcanzado usando el rasgo [4] fue: 50.0000 %
El rendimiento alcanzado usando el rasgo [5] fue: 61.1111 %
El rendimiento alcanzado usando el rasgo [6] fue: 46.6667 %
El rendimiento alcanzado usando el rasgo [7] fue: 58.8889 %
El rendimiento alcanzado usando el rasgo [8] fue: 63.3333 %
El rendimiento alcanzado usando el rasgo [9] fue: 71.4815 %
El rendimiento alcanzado usando el rasgo [10] fue: 67.7778 %
El rendimiento alcanzado usando el rasgo [11] fue: 68.8889 %
El rendimiento alcanzado usando el rasgo [12] fue: 74.0741 %
El rendimiento alcanzado usando el rasgo [13] fue: 76.2963 %

El valor de referencia θ fue: 64.6154 %

En la Figura 5.4 se muestra como se encuentran distribuidos los patrones de entrenamiento, para cada uno de los 13 rasgos, en cada una de las dos posibles clases. Es necesario hacer notar que tomando en cuenta el valor de referencia θ, obtenido mediante la expresión (4.10) de la definición 4.7, los rasgos que contribuyen mayormente de manera univariable a la separación entre clases son los siguientes: $3, 9, 10, 11, 12, 13$; tal como se muestra en la Figura 5.5.

Una vez que se tienen identificados los rasgos que contribuyen mayormente de manera univariable a la separación entre clases, se obtiene el vector de restricciones $\mathbf{C}$, aplicando el paso 7 del algoritmo propuesto en el Capítulo 4.

A continuación se muestra el resultado de la ejecución de los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4.

Un total de [59] errores de clasificación en [270] instancias.
El rendimiento promedio fue de [78.1481 %] de precisión predictiva.
El vector de restricciones es: [0 0 1 0 0 0 0 0 1 1 1 1 1] = 1055
El número de rasgos seleccionados fue: 6 de 13
Logrando eliminar [53.8462 %] del espacio original.

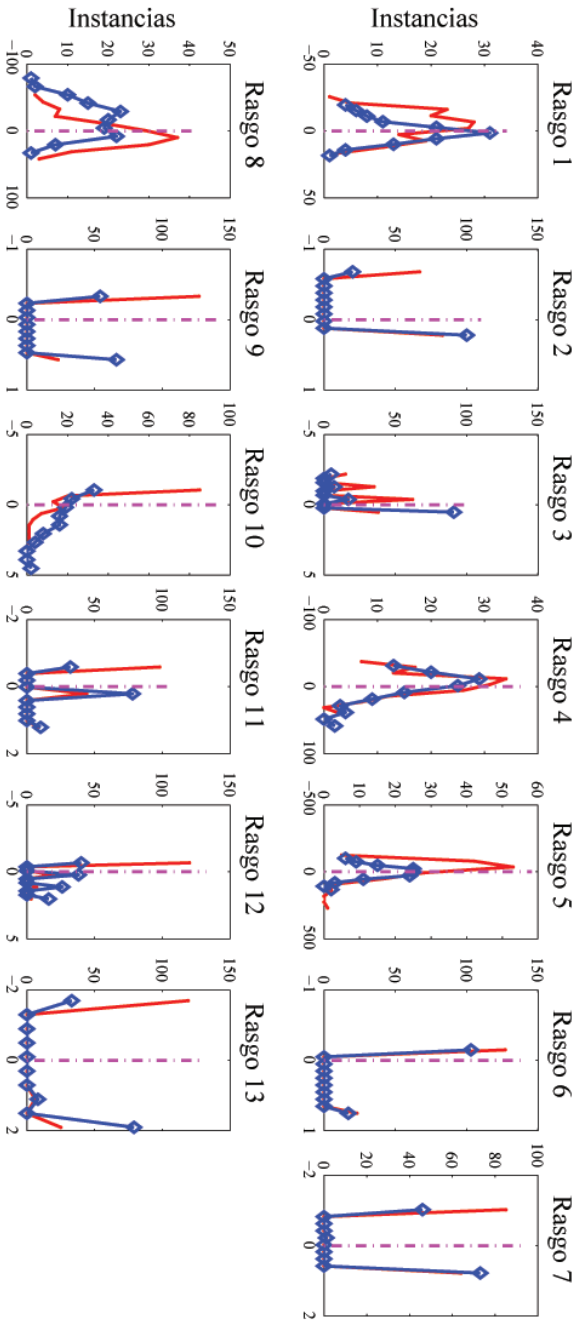


Figura 5.4: Función de Verosimilitud Univariable. Heart Disease Database.

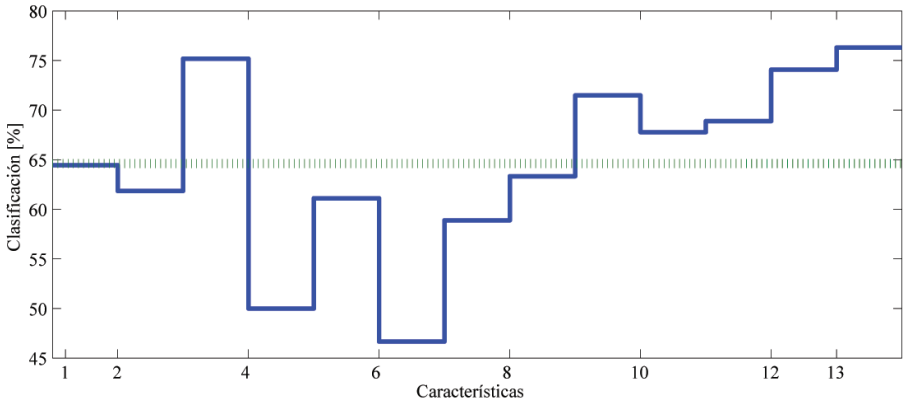


Figura 5.5: Clasificación Univariable. Heart Disease Database.

Tabla 5.2: Clasificación Multivariable. Heart Disease Database.

Máscara decimal	Máscara binaria	Desempeño alcanzado
8191	1 1 1 1 1 1 1 1 1 1 1 1 1	63.70 %
1055	0 0 1 0 0 0 0 0 1 1 1 1 1	78.14 %

Con la finalidad de obtener una estimación confiable del comportamiento del modelo propuesto en presencia de instancias no conocidas, se aplicó la técnica de validación cruzada K-Fold Cross Validation con K=10. Los resultados de las K estimaciones de la precisión predictiva del modelo propuesto, para cada uno de los K subconjuntos de prueba mutuamente excluyentes, se muestran en la Figura 5.6. Asimismo, en la Tabla 5.2 se muestran los resultados promediados de la estimación de la precisión predictiva del modelo propuesto, tanto para el conjunto completo de rasgos, así como para el subconjunto de características seleccionadas mediante el vector de restricciones **C**.

5.1.3. Australian Credit Approval Database

Esta base de datos fue conformada a partir de registros de administración de créditos, con la finalidad de estimar el desempeño de diversos métodos de recuperación de deuda. El conjunto completo de datos se encuentra disponible en el repositorio de bases de datos de la Universidad de California en Irvine [52]. La calificación crediticia (Credit Scoring) es una forma objetiva de evaluar el riesgo asociado con cada sujeto de crédito

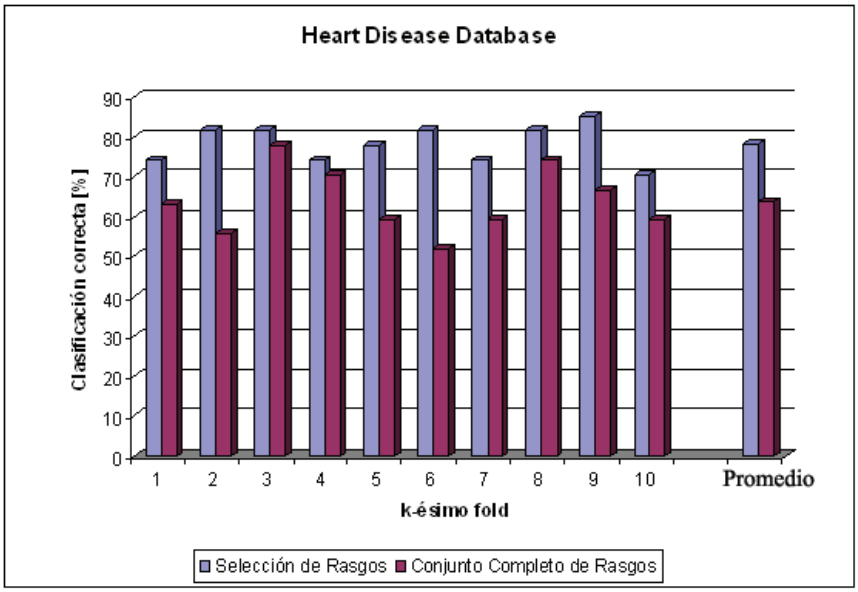


Figura 5.6: Clasificación Multivariable. Heart Disease Database.

mediante la asignación de un valor numérico para cada solicitud de crédito. La estimación adecuada del riesgo crediticio permite a las instituciones de servicios financieros mejorar sus políticas de precios, elevar su competitividad, así como reducir los tiempos de recuperación de deuda. Por el contrario, la estimación inadecuada de este parámetro puede resultar en la aprobación de solicitudes de crédito cuyo periodo de recuperación de deuda puede ser indeterminado [71]. Cada instancia está conformada por 14 rasgos numéricos y una etiqueta de clase; naturalmente, cada una de las instancias pertenece a una de dos posibles clases: bajo riesgo o alto riesgo. Esta base de datos ha sido ampliamente utilizada para clasificar solicitudes de crédito [72].

Experimento 5.3 *Dado el conjunto de instancias disponibles en la base de datos* ***Australian Credit Approval Database****; aplicar los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4 para obtener un vector de restricciones* $\mathbf{C}$ *que permita identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado. Posteriormente, aplicar los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4 para estimar la precisión predictiva del modelo propuesto, aplicando técnicas de*

validación cruzada; concretamente, aplicar K-Fold Cross Validation con K=10.

A continuación se muestra el resultado de la ejecución de los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4.

** Australian Credit Approval **
El rendimiento alcanzado usando el rasgo [1] fue: 52.6087 %
El rendimiento alcanzado usando el rasgo [2] fue: 61.7391 %
El rendimiento alcanzado usando el rasgo [3] fue: 61.0145 %
El rendimiento alcanzado usando el rasgo [4] fue: 55.0725 %
El rendimiento alcanzado usando el rasgo [5] fue: 64.058 %
El rendimiento alcanzado usando el rasgo [6] fue: 61.3043 %
El rendimiento alcanzado usando el rasgo [7] fue: 64.7826 %
El rendimiento alcanzado usando el rasgo [8] fue: 85.5072 %
El rendimiento alcanzado usando el rasgo [9] fue: 73.3333 %
El rendimiento alcanzado usando el rasgo [10] fue: 66.2319 %
El rendimiento alcanzado usando el rasgo [11] fue: 52.029 %
El rendimiento alcanzado usando el rasgo [12] fue: 55.7971 %
El rendimiento alcanzado usando el rasgo [13] fue: 51.8841 %
El rendimiento alcanzado usando el rasgo [14] fue: 55.0725 %

El valor de referencia θ fue: 61.4596 %

En la Figura 5.7 se muestra como se encuentran distribuidos los patrones de entrenamiento, para cada uno de los 14 rasgos, en cada una de las dos posibles clases. Es necesario hacer notar que tomando en cuenta el valor de referencia θ, obtenido mediante la expresión (4.10) de la definición 4.7, los rasgos que contribuyen mayormente de manera univariable a la separación entre clases son los siguientes: $2, 5, 7, 8, 9, 10$; tal como se muestra en la Figura 5.8. Una vez que se tienen identificados los rasgos que contribuyen mayormente de manera univariable a la separación entre clases, se obtiene el vector de restricciones $\mathbf{C}$, aplicando el paso 7 del algoritmo propuesto en el Capítulo 4.

A continuación se muestra el resultado de la ejecución de los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4.

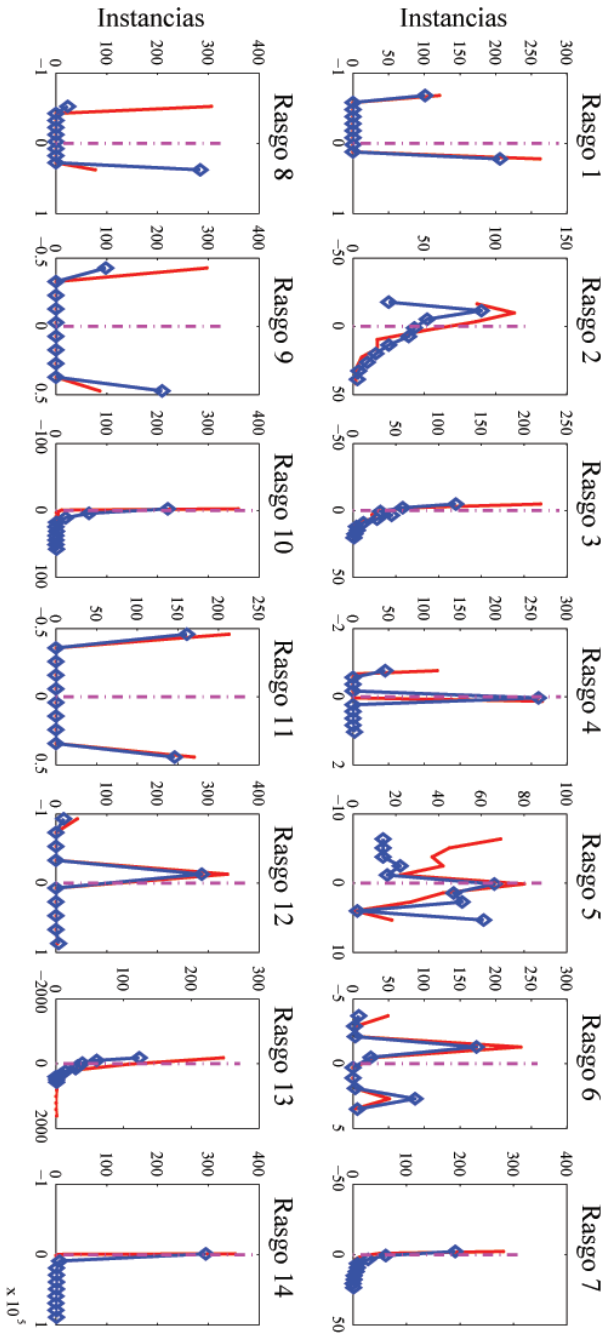


Figura 5.7: Función de Verosimilitud Univariable. Australian Credit Approval Database.

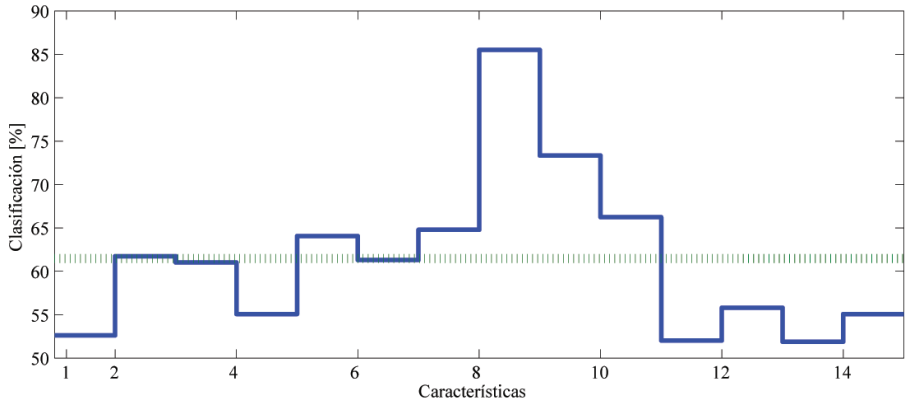


Figura 5.8: Clasificación Univariable. Australian Credit Approval Database.

Tabla 5.3: Clasificación Multivariable. Australian Credit Approval Database.

Máscara decimal	Máscara binaria	Desempeño alcanzado
16383	1 1 1 1 1 1 1 1 1 1 1 1 1 1	65.07 %
4848	0 1 0 0 1 0 1 1 1 1 0 0 0 0	66.37 %

Un total de [232] errores de clasificación en [690] instancias.
El rendimiento promedio fue de [66.3768 %] de precisión predictiva.
El vector de restricciones es: [0 1 0 0 1 0 1 1 1 1 0 0 0 0] = 4848
El número de rasgos seleccionados fue: 6 de 14
Logrando eliminar [57.1429 %] del espacio original.

Con la finalidad de obtener una estimación confiable del comportamiento del modelo propuesto en presencia de instancias no conocidas, se aplicó la técnica de validación cruzada K-Fold Cross Validation con K=10. Los resultados de las K estimaciones de la precisión predictiva del modelo propuesto, para cada uno de los K subconjuntos de prueba mutuamente excluyentes, se muestran en la Figura 5.9. Asimismo, en la Tabla 5.3 se muestran los resultados promediados de la estimación de la precisión predictiva del modelo propuesto, tanto para el conjunto completo de rasgos, así como para el subconjunto de características seleccionadas mediante el vector de restricciones **C**.

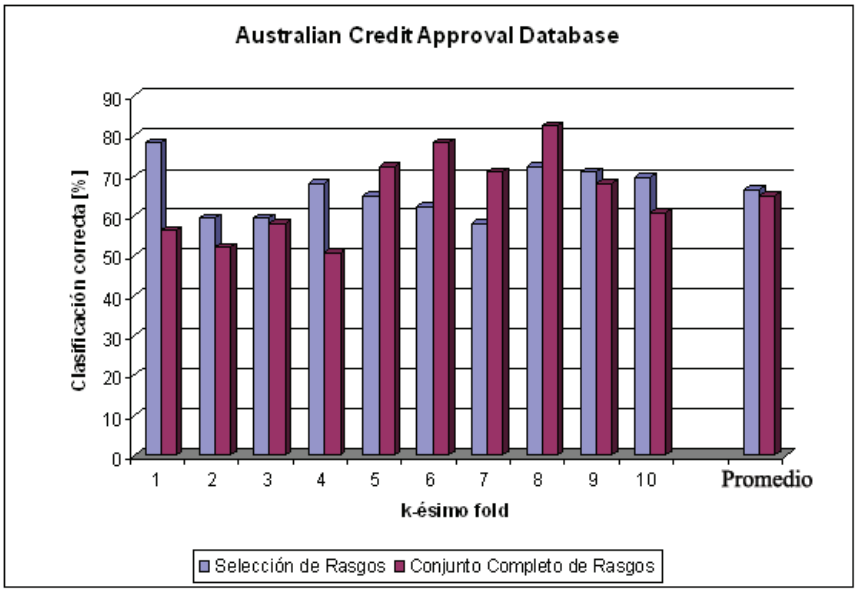


Figura 5.9: Clasificación Multivariable. Australian Credit Approval Database.

5.1.4. Hepatitis Database

Este conjunto de datos fue donado por el Jozef Stefan Institute de la antigua Yugoslavia, hoy Eslovenia. La base de datos se encuentra disponible en el repositorio de bases de datos de la Universidad de California en Irvine [52]. Este conjunto de datos ha sido ampliamente utilizado para predecir la presencia de hepatitis [68]. Cada instancia está conformada por diecinueve rasgos numéricos y una etiqueta de clase; naturalmente, cada una de las instancias pertenece a una de dos posibles clases: ausencia o presencia del padecimiento. Cabe mencionar que la base de datos está conformada por 155 instancias.

Experimento 5.4 *Dado el conjunto de instancias disponibles en la base de datos* ***Hepatitis Database****; aplicar los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4 para obtener un vector de restricciones* $\mathbf{C}$ *que permita identificar aquellas características que preserven o maximicen la separación entre clases en un conjunto de patrones de aprendizaje dado. Posteriormente, aplicar los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4 para estimar la precisión predictiva del modelo propuesto, aplicando técnicas de validación cruzada; concretamente, aplicar K-Fold Cross Validation con K=10.*

A continuación se muestra el resultado de la ejecución de los pasos 1 al 6 del algoritmo propuesto en el Capítulo 4.

** Hepatitis Database **
El rendimiento alcanzado usando el rasgo [1] fue: 63.8710 %
El rendimiento alcanzado usando el rasgo [2] fue: 30.9677 %
El rendimiento alcanzado usando el rasgo [3] fue: 55.4839 %
El rendimiento alcanzado usando el rasgo [4] fue: 33.5484 %
El rendimiento alcanzado usando el rasgo [5] fue: 53.5484 %
El rendimiento alcanzado usando el rasgo [6] fue: 69.0323 %
El rendimiento alcanzado usando el rasgo [7] fue: 70.9677 %
El rendimiento alcanzado usando el rasgo [8] fue: 32.9032 %
El rendimiento alcanzado usando el rasgo [9] fue: 56.7742 %
El rendimiento alcanzado usando el rasgo [10] fue: 73.5484 %
El rendimiento alcanzado usando el rasgo [11] fue: 72.9032 %
El rendimiento alcanzado usando el rasgo [12] fue: 82.5806 %
El rendimiento alcanzado usando el rasgo [13] fue: 80.0000 %
El rendimiento alcanzado usando el rasgo [14] fue: 79.3548 %
El rendimiento alcanzado usando el rasgo [15] fue: 67.7419 %
El rendimiento alcanzado usando el rasgo [16] fue: 71.6129 %
El rendimiento alcanzado usando el rasgo [17] fue: 68.3871 %
El rendimiento alcanzado usando el rasgo [18] fue: 45.1613 %
El rendimiento alcanzado usando el rasgo [19] fue: 66.4516 %

El valor de referencia θ fue: 61.8336 %

Una vez que se han aplicado los pasos 1 a 5 del algoritmo propuesto en el Capítulo 4, se calcula el error de clasificación acumulado de acuerdo con la expresión (4.9) de la definición 4.6. Es necesario hacer notar que tomando en cuenta el valor de referencia θ, obtenido mediante la expresión (4.10) de la definición 4.7, los rasgos que contribuyen mayormente de manera univariable a la separación entre clases son los siguientes: $1, 6, 7, 10-17$ y 19; tal como se muestra en la Figura 5.10.

Una vez que se tienen identificados los rasgos que contribuyen mayormente de manera univariable a la separación entre clases, se obtiene el vector de restricciones $\mathbf{C}$, aplicando el paso 7 del algoritmo propuesto en el Capítulo 4.

A continuación se muestra el resultado de la ejecución de los pasos 8 y 9 del algoritmo propuesto en el Capítulo 4.

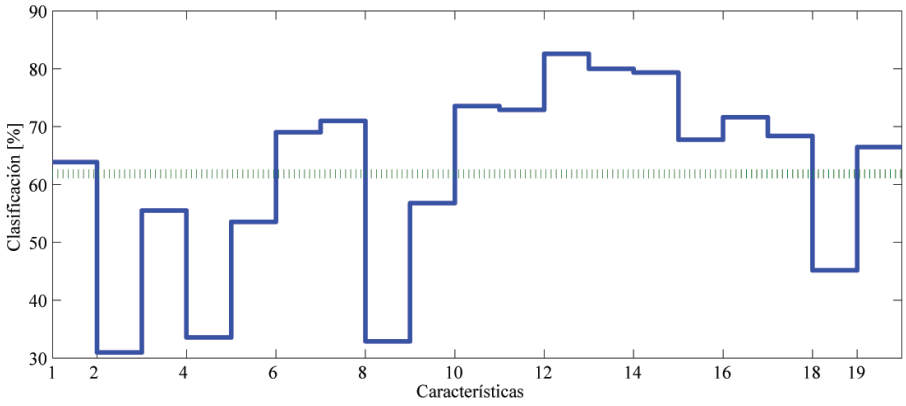


Figura 5.10: Clasificación Univariable. Hepatitis Database.

Tabla 5.4: Clasificación Multivariable. Hepatitis Database.

Máscara binaria	Desempeño alcanzado
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	65.33 %
1 0 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 1	64.17 %

Un total de [54] errores de clasificación en [155] instancias.
El rendimiento promedio fue de [64.1706 %] de precisión predictiva.
El número de rasgos seleccionados fue: 12 de 19
Logrando eliminar [36.8421 %] del espacio original.

Con la finalidad de obtener una estimación confiable del comportamiento del modelo propuesto en presencia de instancias no conocidas, se aplicó la técnica de validación cruzada K-Fold Cross Validation con K=10. Los resultados de las K estimaciones de la precisión predictiva del modelo propuesto, para cada uno de los K subconjuntos de prueba mutuamente excluyentes, se muestran en la Figura 5.11. Asimismo, en la Tabla 5.4 se muestran los resultados promediados de la estimación de la precisión predictiva del modelo propuesto, tanto para el conjunto completo de rasgos, así como para el subconjunto de características seleccionadas mediante el vector de restricciones **C**.

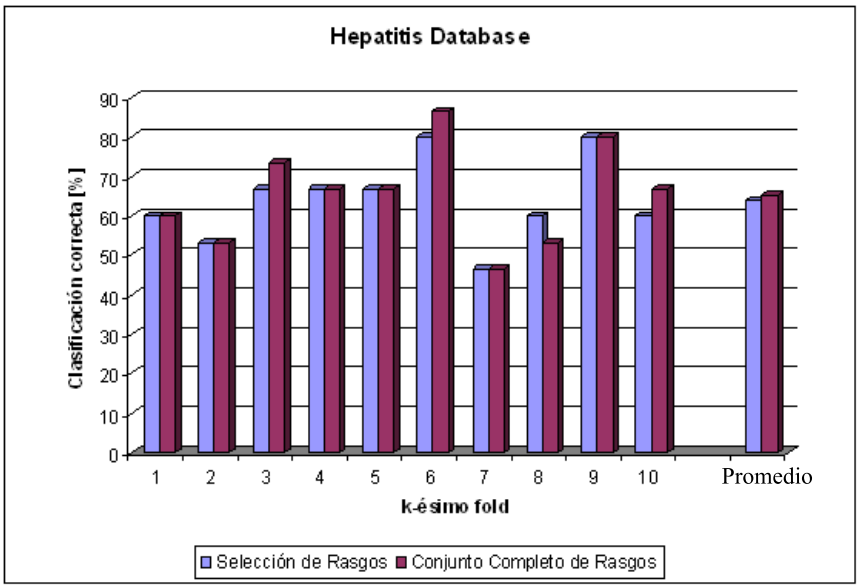


Figura 5.11: Clasificación Multivariable. Hepatitis Database.

Tabla 5.5: Resultados de la Selección de Rasgos.

	Breast	Heart	Credit	Hepatitis
Tamaño original del conjunto de datos	9	13	14	19
Rasgos seleccionados	5	6	6	12
Reducción dimensional de los datos	55.55 %	46.15 %	42.85 %	63.15 %

5.2. Análisis de Resultados

Los resultados expuestos en la Tabla 5.5, muestran el índice de reducción dimensional de los datos para los conjuntos de prueba utilizados a lo largo de la fase experimental del presente trabajo de tesis; cabe mencionar que el número de rasgos seleccionados para dos de las bases de datos (Heart Disease Database y Australian Credit Approval Database) fue menor al 50 %. El índice de reducción dimensional para la base de datos con menor número de rasgos (Breast Cancer Database) fue cercano al 60 %, mientras que para la base de datos con mayor número de rasgos (Hepatitis Database) fue superior al 60 %. La representación gráfica de los resultados expuestos en la Tabla 5.5, se muestran en la Figura 5.12. Estos resultados ponen de manifiesto la eficacia del algoritmo propuesto en el Capítulo 4

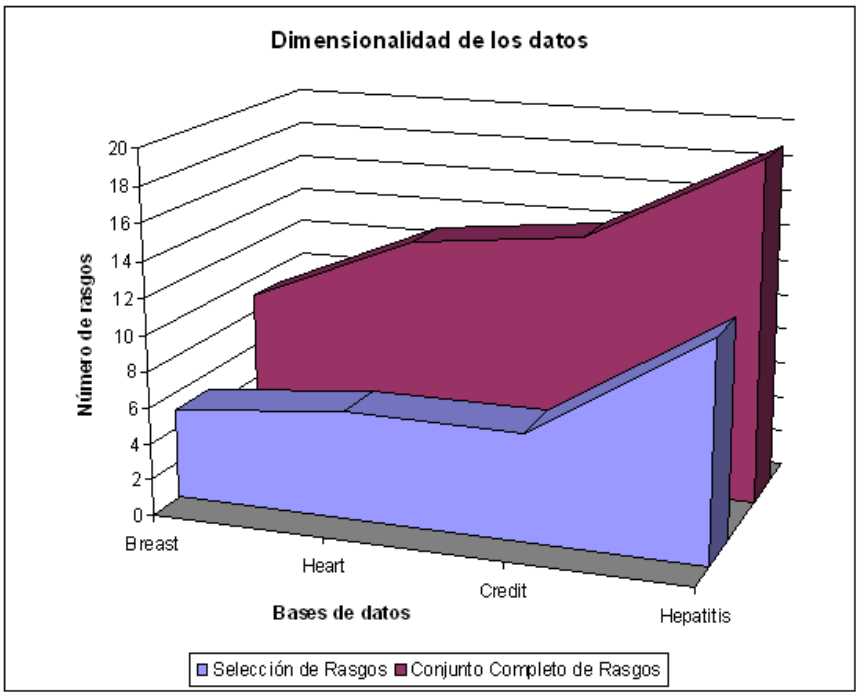


Figura 5.12: Comparación del número de rasgos utilizados para cada base de datos.

para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental, por medio del vector de restricciones **C**.

Es necesario señalar que cuando las condiciones de las instancias que conforman el conjunto fundamental de patrones permiten la eliminación de información irrelevante (para fines de clasificación), el desempeño alcanzado por el subconjunto de rasgos (dimensionalmente menor), obtenido mediante el vector de restricciones **C**, es claramente superior. Dicha situación puede observarse en la base de datos Heart Disease Database, donde al eliminar la información irrelevante (para fines de clasificación), se alcanzan incrementos del 15 % en el índice de precisión predictiva; tal como se muestra en la Figura 5.13.

En la Figura 5.14 se muestra mediante un gráfico tiempo vs. número de rasgos, el tiempo requerido por el algoritmo *HCM* (presentado en [66] y analizado en el Capítulo 3 del presente trabajo de tesis) para encontrar el subconjunto óptimo de características. Cabe mencionar que el algoritmo *HCM* es un método que reduce la dimensionalidad de los

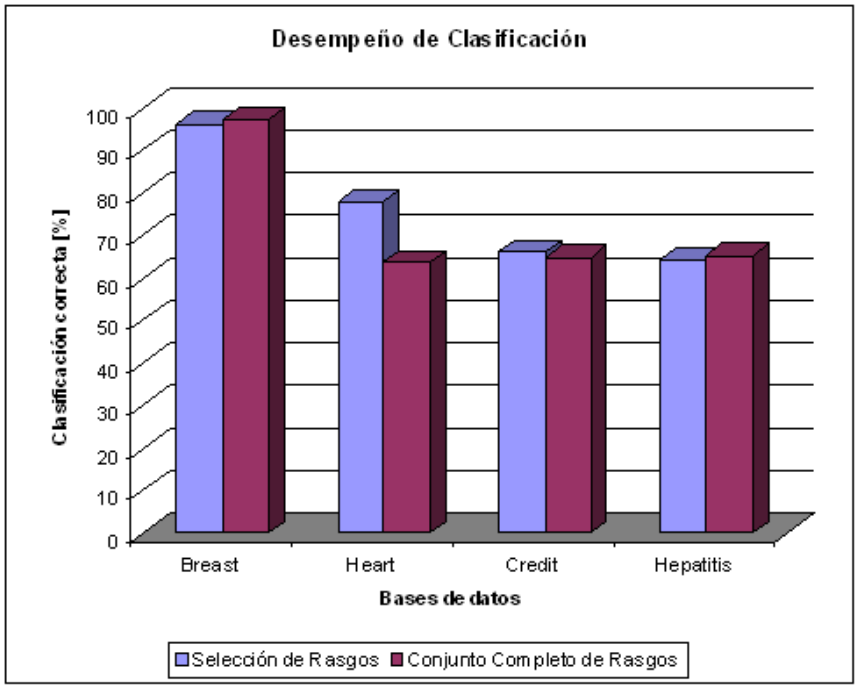


Figura 5.13: Comparación del índice de clasificación para cada base de datos.

patrones del conjunto fundamental, aplicando el enfoque *Wrapper* (exploración exhaustiva del espacio de características).

Con la finalidad de encontrar un límite razonable (en el número de características que describen un problema) para el cual todavía es factible aplicar el enfoque *Wrapper,* se presenta en la Tabla 5.6 el tiempo requerido por el algoritmo *HCM* para encontrar el subconjunto óptimo de características.

Tomando como base los cálculos del tiempo requerido por el algoritmo *HCM* para encontrar el subconjunto óptimo de características, se puede establecer que cuando el número de características que describen un problema supera los 200 rasgos, ya no es recomendable la aplicación del enfoque *Wrapper* para llevar a cabo procesos de selección de rasgos.

Cabe mencionar que el tiempo requerido por el algoritmo HCM para encontrar el subconjunto óptimo de características presenta un crecimiento en tiempo polinomial (en función del número de características que describen el problema a resolver). Asimismo, es

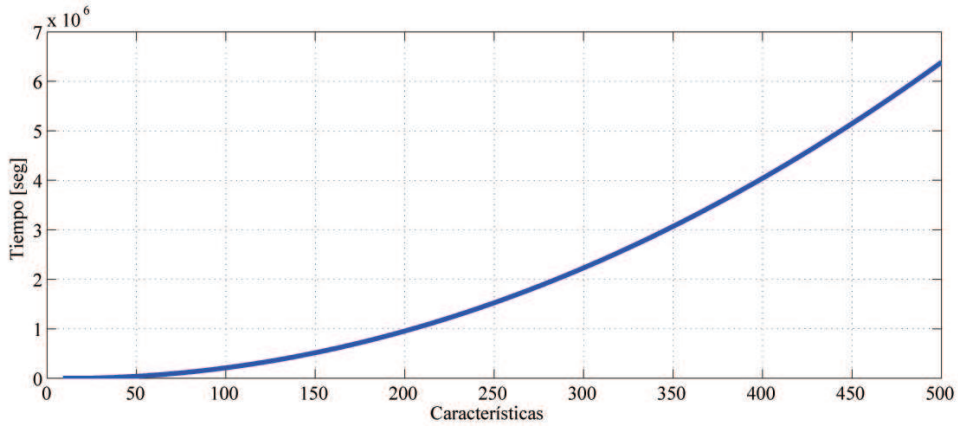


Figura 5.14: Tiempo requerido por el algoritmo *HCM* para encontrar el subconjunto óptimo.

Tabla 5.6: Tiempo requerido por el algoritmo *HCM* para encontrar el subconjunto óptimo.

Número de rasgos	Tiempo	
9	4.6	seg
13	39.2	seg
14	3.8	min
20	36.1	min
30	2.7	hrs
50	11.1	hrs
100	2.4	días
200	11	días
500	73	días
1000	302	días
2000	1221	días
5000	7687	días

necesario señalar que el algoritmo propuesto en el presente trabajo de tesis presenta un crecimiento en tiempo lineal (en función del número de características que describen el problema a resolver).

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

1. En este trabajo de tesis se introduce un nuevo modelo para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental: el Enfoque Asociativo para la Selección de Rasgos.

2. Se define un vector de restricciones: el vector de restricciones C que permite identificar, en los patrones que conforman el conjunto fundamental, aquellos rasgos que contribuyen mayormente de manera univariable a la separación entre clases; asimismo, este vector de restricciones permite eliminar, en los patrones que conforman el conjunto fundamental, aquellas características irrelevantes (para fines de clasificación).

3. El modelo propuesto para reducir la dimensionalidad de los patrones que conforman el conjunto fundamental, exhibe un desempeño experimental competitivo, al ser comparado con otros importantes métodos de Selección de Rasgos descritos en la literatura actual.

6.2. Trabajo Futuro

1. Tomar como base el trabajo de investigación de George Forman [73] sobre métricas para selección de características en aprendizaje supervisado de textos, para desarrollar el enfoque asociativo para la clasificación de textos.

2. Tomar como base el trabajo de investigación de Vojtěch Franc y Bogdan Savchynskyy [74] sobre clasificadores de patrones basados en máximos de sumas y estructuras de vecindarios arbitrarios, para incrementar la capacidad discriminativa del Enfoque Asociativo para la Selección de Rasgos.

3. Tomar como base el trabajo de investigación de Jean-Philippe Pellet y André Elisseeff [75] sobre modelos de Markov en aprendizaje causal, para identificar variables fuertemente relevantes mediante el Enfoque Asociativo para la Selección de Rasgos.

Capítulo 7

Publicaciones

A continuación se muestran algunas publicaciones relacionadas con el presente trabajo de tesis.

1. **M. Aldape-Pérez**, I. Román-Godínez, O. Camacho-Nieto, Thresholded learning matrix for efficient pattern recalling, in: CIARP '08: Proceedings of the 13th Iberoamerican congress on Pattern Recognition, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 445–452.

2. **M. Aldape-Pérez, C**. Yáñez-Márquez, A. J. Argüelles-Cruz, Optimized associative memories for feature selection, in: IbPRIA '07: Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 435–442.

Referencias

[1] A. A. Shklyaev, M. Ichikawa, Fabrication of germanium and silicon nanostructures using a scanning tunneling microscope, Physics-Uspekhi 49 (9) (2006) 887.

[2] Y. Massoud, A. Nieuwoudt, Modeling and design challenges and solutions for carbon nanotube-based interconnect in future high performance integrated circuits, J. Emerg. Technol. Comput. Syst. 2 (3) (2006) 155–196.

[3] R. Jensen, Q. Shen, Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches, Wiley-IEEE Press, 2008.

[4] S.-W. Hla, Scanning tunneling microscope atom and molecule manipulations: Realizing molecular switches and devices, Japanese Journal of Applied Physics 47 (7) (2008) 6063–6069.

[5] J. Deng, A. Lin, G. C. Wan, H.-S. P. Wong, Carbon nanotube transistor compact model for circuit design and performance optimization, J. Emerg. Technol. Comput. Syst. 4 (2) (2008) 1–20.

[6] G. E. Begtrup, W. Gannett, T. D. Yuzvinsky, V. H. Crespi, A. Zettl, Nanoscale reversible mass transport for archival memory, Nano Letters 9 (5) (2009) 1835–1838.

[7] J. A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, 1st Edition, Springer, 2007.

[8] D. W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics), Wiley-Interscience, 1992.

[9] I. Eccles, M. Su, Illustrating the curse of dimensionality numerically through different data distribution models, in: ISICT '04: Proceedings of the 2004 international symposium on Information and communication technologies, Trinity College Dublin, 2004, pp. 232–237.

[10] R. E. Bellman, Adaptive control processes - A guided tour, Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[11] G. H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, Morgan Kaufmann, 1994, pp. 121–129.

[12] J. H. Friedman, U. Fayyad, On bias, variance, 0/1-loss, and the curse-of-dimensionality, Data Mining and Knowledge Discovery 1 (1997) 55–77.

[13] A. L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.

[14] D. Franois, High-dimensional Data Analysis: From Optimal Metrics to Feature Selection, VDM Verlag, Saarbrücken, Germany, Germany, 2008.

[15] K. S. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is "nearest neighbor" meaningful?, in: ICDT '99: Proceeding of the 7th International Conference on Database Theory, Springer-Verlag, London, UK, 1999, pp. 217–235.

[16] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering, ACM Trans. Knowl. Discov. Data 3 (1) (2009) 1–58.

[17] J. E. Malek, A. M. Alimi, R. Tourki, Problems in pattern classification in high domensional spaces: behavior of a class of combined neuro-fuzzy classifiers, Fuzzy Sets Syst. 128 (1) (2002) 15–33.

[18] D. Angluin, J. Westbrook, W. Zhu, Robot navigation with distance queries, SIAM Journal on Computing 30 (2000) 2000.

[19] M. Pinzolas, J. J. Astrain, J. R. G. de Mendívil, J. Villadangos, Isolated hand-written digit recognition using a neurofuzzy scheme and multiple classification, J. Intell. Fuzzy Syst. 12 (2) (2002) 97–105.

[20] M. Pardo, G. Faglia, G. Sberveglieri, M. Corteb, F. Masulli, M. Riani, V. V. Brescia-italy, Monitoring reliability of sensors in an array by neural networks, Sensors and Actuators, B 67 (2000) 2000.

[21] T.-C. Cheng, A. Biswas, Maximum trimmed likelihood estimator for multivariate mixed continuous and categorical data, Comput. Stat. Data Anal. 52 (4) (2008) 2042–2065.

[22] Y. Liu, H. T. Loh, A. Sun, Imbalanced text classification: A term weighting approach, Expert Syst. Appl. 36 (1) (2009) 690–701.

[23] C. Erdman, J. W. Emerson, A fast bayesian change point analysis for the segmentation of microarray data, Bioinformatics 24 (19) (2008) 2143–2148.

[24] H.-Q. Wang, H.-S. Wong, H. Zhu, T. T. C. Yip, A neural network-based biomarker association information extraction approach for cancer classification, J. of Biomedical Informatics 42 (4) (2009) 654–666.

[25] R. J. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches, 1st Edition, Wiley, 1991.

[26] D. Koller, M. Sahami, Toward optimal feature selection, Morgan Kaufmann, 1996, pp. 284–292.

[27] G. C. Cawley, N. L. C. Talbot, I. Guyon, A. Saffari, Preventing over-fitting during model selection using bayesian regularisation, JMLR 8.

[28] L. Yu, H. Liu, I. Guyon, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (2004) 1205–1224.

[29] K. Torkkola, I. Guyon, A. Elisseeff, Feature extraction by non-parametric mutual information maximization, Journal of Machine Learning Research 3 (2003) 1415–1438.

[30] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. 5 (2004) 1531–1555.

[31] S. Gadat, L. Younes, A stochastic algorithm for feature selection in pattern recognition, J. Mach. Learn. Res. 8 (2007) 509–547.

[32] M. H. Hassoun, Associative Neural Memories: Theory and Implementation, illustrated edition Edition, Oxford University Press, USA, 1993.

[33] T. Kohonen, Self-Organization and Associative Memory, 3rd Edition, Springer, 1989.

[34] C. Yáñez-Márquez, Memorias asociativas basadas en relaciones de orden y operadores binarios, Ph.D. thesis, Centro de Investigación en Computación, México. (2002).

[35] M. E. Acevedo-Mosqueda, Memorias asociativas bidireccionales alfa-beta, Ph.D. thesis, Centro de Investigación en Computación, México (2006).

[36] P. K. Simpson, Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations, Mcgraw-Hill (Tx), 1990.

[37] K. Steinbuch, H. Frank, Nichtdigitale lernmatrizen als perzeptoren, Biological Cybernetics 1 (1961) 117–124.

[38] S. Mitra, L. J. Avra, E. J. McCluskey, An output encoding problem and a solution technique, IEEE Trans. on CAD of Integrated Circuits and Systems 18 (6) (1999) 761–768.

[39] D. J. Willshaw, O. P. Buneman, H. C. Longuet-Higgins, Non-Holographic associative memory, Nature 222 (5197) (1969) 960–962.

[40] T. Kohonen, Correlation matrix memories, IEEE Transactions on Computers C-21 (1972) 353–359.

[41] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79 (1982) 2554–2558.

[42] Y. Abu-Mostafa, J. St. Jacques, Information capacity of the hopfield model, Information Theory, IEEE Transactions on 31 (4) (1985) 461–464.

[43] G. Ritter, P. Sussner, J. Diaz-de Leon, Morphological associative memories, Neural Networks, IEEE Transactions on 9 (2) (1998) 281–293.

[44] P. Sussner, M. Valle, Gray-scale morphological associative memories, Neural Networks, IEEE Transactions on 17 (3) (2006) 559–570.

[45] J. H. Sossa Azuela, R. Barrón, R. A. Vázquez, New associative memories to recall real-valued patterns, in: CIARP, 2004, pp. 195–202.

[46] C. K. Loo, M. Rao, Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy artmap, IEEE Transactions on Knowledge and Data Engineering 17 (11) (2005) 1589–1593.

[47] N. A. Chuzhanova, A. J. Jones, A. J. Jones, S. Margetts, Feature selection for genetic sequence classification, Bioinformatics 14 (1998) 139–143.

[48] S. Baek, C.-A. Tsai, J. J. Chen, Development of biomarker classifiers from high-dimensional data, Brief Bioinform 10 (5) (2009) 537–546.

[49] H. Almuallim, T. G. Dietterich, Learning with many irrelevant features, in: AAAI, 1991, pp. 547–552.

[50] C.-N. Hsu, H.-J. Huang, T.-T. Wong, Why discretization works for naive bayesian classifiers, in: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 399–406.

[51] R. Kohavi, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.
URL `http://dx.doi.org/10.1016/S0004-3702(97)00043-X`

[52] A. Asuncion, D. Newman, UCI machine learning repository (2007).
URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`

[53] I. Guyon, A. Saffari, G. Dror, G. Cawley, 2008 special issue: Analysis of the ijcnn 2007 agnostic learning vs. prior knowledge challenge, Neural Netw. 21 (2-3) (2008) 544–550.

[54] S. Klement, A. M. Mamlouk, T. Martinetz, Reliability of cross-validation for svms in high-dimensional, low sample size scenarios., in: V. Kurková, R. Neruda, J. Koutník (Eds.), ICANN (1), Vol. 5163 of Lecture Notes in Computer Science, Springer, 2008, pp. 41–50.

[55] S. Dudoit, M. J. van der Laan, S. Keleş, A. M. Molinaro, S. E. Sinisi, S. L. Teng, Loss-based estimation with cross-validation: applications to microarray data analysis, SIGKDD Explor. Newsl. 5 (2) (2003) 56–68.

[56] I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider, M. Uhr, Competitive baseline methods set new standards for the nips 2003 feature selection benchmark, Pattern Recogn. Lett. 28 (12) (2007) 1438–1444.

[57] I. Guyon, S. Gunn, M. Nikravesh, L. A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[58] A. Y. Ng, Preventing .ºverfitting.ºf cross-validation data, in: ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 245–253.

[59] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, Morgan Kaufmann, 1995, pp. 1137–1143.

[60] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, Wiley-Interscience, 2000.

[61] W. Freeman, D. Brainard, Bayesian decision theory, the maximum local mass estimate, and color constancy, Computer Vision, IEEE International Conference on 0 (1995) 210.

[62] C. P. Robert, The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation, 2nd Edition, Springer Verlag, New York, 2007.

[63] J. O. Berger, Statistical Decision Theory and Bayesian Analysis, 2nd Edition, Springer, 1993.

[64] R. Santiago-Montero, Clasificador híbrido de patrones basado en la lernmatrix de steinbuch y el linear associator de anderson-kohonen, Master's thesis, Centro de Investigación en Computación, México. (2003).

[65] R. Santiago-Montero, C. Yáñez-Márquez, J. L. Díaz de León, Clasificador híbrido de patrones, in: CIARP, 2002.

[66] M. Aldape-Perez, C. Yanez-Marquez, L. O. L. Leyva, Feature selection using a hybrid associative classifier with masking techniques, in: MICAI '06: Proceedings of the Fifth Mexican International Conference on Artificial Intelligence, IEEE Computer Society, Washington, DC, USA, 2006, pp. 151–160.

[67] A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2) (1997) 153–158.

[68] F. Kharbat, L. Bull, M. Odeh, Mining breast cancer data with xcs, in: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, 2007, pp. 2066–2073.

[69] G. A. Carpenter, N. Markuzon, Artmap-ic and medical diagnosis: instance counting and inconsistent cases, Neural Netw. 11 (2) (1998) 323–336.

[70] R. Das, I. Turkoglu, A. Sengur, Effective diagnosis of heart disease through neural networks ensembles, Expert Syst. Appl. 36 (4) (2009) 7675–7680.

[71] G. Bouchard, G. Celeux, Selection of generative models in classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 544–554.

[72] K.-J. Kim, S.-B. Cho, Evolutionary ensemble of diverse artificial neural networks using speciation, Neurocomput. 71 (7-9) (2008) 1604–1618.

[73] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (2003) 1289–1305.

[74] V. Franc, B. Savchynskyy, Discriminative learning of max-sum classifiers, J. Mach. Learn. Res. 9 (2008) 67–104.

[75] J.-P. Pellet, A. Elisseeff, Using markov blankets for causal structure learning, J. Mach. Learn. Res. 9 (2008) 1295–1342.