Procesamiento difuso de características para la clasificación de clases traslapadas en el caso de leucemia aguda

Por

Alejandro Rosales Pérez

Tesis sometida como requisito parcial para obtener
el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Tonantzintla, Puebla

Supervisada por:

Dr. Carlos Alberto Reyes García
Investigador del INAOE
Dra. María del Pilar Gómez Gil
Investigadora del INAOE

©INAOE 2011
Derechos reservados
El autor otorga al INAOE el permiso de
reproducir y distribuir copias de esta tesis
en su totalidad o en partes

Resumen

En el campo de la medicina es deseable contar con métodos cada vez más precisos que permitan identificar enfermedades de manera adecuada. Esta identificación, desde el punto de vista computacional, involucra varias etapas, entre las que podemos destacar la clasificación. La clasificación ha sido usada en el reconocimiento de diferentes enfermedades, siendo la leucemia aguda una de éstas. Por otro lado, en varios problemas de clasificación existen tipos y sub-tipos, donde estos últimos tienen características morfológicas similares entre ellos, lo que dificulta su reconocimiento. En el presente trabajo se propone la determinación automática de un ensamble de modelos de clasificación difusos adecuados al problema representado en los datos de entrenamiento. El modelo está basado en un pre-procesamiento difuso de las características para reducir el grado de traslape en el conjunto de datos. Esto permite proveer a las características una descripción lingüística, las cuales son usadas en la clasificación. El sistema automáticamente determina el conjunto de modelos de clasificación difusos "óptimos" para los diferentes conjuntos de datos, que presentan traslape entre sus clases. Para conjuntos de datos con tipos y subtipos, como en la base de datos de leucemia aguda, un segundo paso es ejecutado, el cual consiste en la construcción de un camino jerárquico para la clasificación. De esta manera, la información de las clases de tipos de leucemia aguda es usada para clasificar los subtipos de ésta.

Para obtener un modelo "óptimo", usamos algoritmos genéticos para encontrar la combinación correcta de método de selección de características, tipo de procesamiento difuso y clasificador difuso. Este algoritmo genético permite minimizar el error a través del esquema de validación cruzada, y determinar el conjunto de clasificadores difusos más diversos, dado el conjunto de entrenamiento. Estos clasificadores son parte de un ensamble.

El método propuesto es probado con varios conjuntos de datos de leucemia aguda y conjuntos de datos disponibles en el repositorio UCI. Entre los resultados más importantes destacan 95.58 % en clasificación correcta de tipos de leucemia aguda; 90.72 % en clasificación correcta de subtipos de leucemia linfoide aguda, hasta un 94.94 % en subtipos de leucemia mieolide aguda. En la clasificación usando la jerarquía se reporta un 93.02 % en sub-tipos de leucemia aguda. Con respecto a la clasificación de cinco conjuntos de datos del repositorio UCI, nuestro método propuesto alcanzó un 98.19 % en el mejor caso.

Abstract

In the field of medicine it is desirable to have accurate methods to identify diseases properly. From a computational point of view, this identification involves several steps, among which we highlight classification. Classification has been used in the recognition of different diseases, acute leukemia being one of these. In the other hand, several classification problems contains types and subtypes, where the subtypes are morphologically similar among them, making difficult their differentiation. This thesis proposes an automatic selection of adequate ensemble of fuzzy classification models, adjusted to a specific problem represented by a training data set. Our model is based on a fuzzy processing of the features to reduce the degree of overlapping in the data set. This also provides a linguistic description of features, which are used in the building of the classifier. The system automatically determines the set of fuzzy "optimal" classification models for different data sets, showing overlap among classes. For data sets with types and subtypes, as in a leukemia data set, a second step is executed, which consists of building a hierarchical classification path. In this way, information of class types of acute leukemia is used to classify subtypes of it.

To get an optimal classification model, we use genetic algorithms to find a right combination of feature selection methods, type of fuzzy processing and fuzzy classifiers. This genetic algorithm allows to minimize the error over a cross validation

scheme, and it determines the set of the most diverse fuzzy classifiers for the training data. These classifiers are part of an ensemble.

The proposed method is tested in several data sets of acute leukemia and data sets from the UCI repository. Among the most important results obtained with our method are a 95.58 % in correct classification of lymphoblastic and mieolyd types of acute leukemia; a 90.72 % in correct classification of subtypes of acute lymphoblastic leukemia, up to 94.94 % in mieolyd acute leukemia subtypes. Classification using a hierarchical path reported a 93.02 % in subtypes of acute leukemia. With respect to classification of five data bases taken from UCI repository our proposed method got a 98.19 % in the best case.

Agradecimientos

Primeramente, quiero agradecer a Dios por haberme permitido culminar un proyecto más en mi vida.

A mis padres, quienes siempre confiaron en mí, me han dado todo su amor, y apoyo.

A mis asesores de tesis, Dr. Carlos Alberto y Dra. María del Pilar, por todos sus consejos, apoyo y guía durante el desarrollo de este trabajo de investigación

A mis sinodales los Dres. Jesús Antonio, Leopoldo Altamirano y Eduardo Morales, por el tiempo dedicado en la revisión de este documento de tesis y que, gracias a sus comentarios, me han permitido enriquecerlo.

A mis amigos del INAOE Adrián, Alejandro, Armando, Jaime, Miriam, Ricardo, Rigoberto, con quienes he compartido momentos agradables en esta etapa de mi vida; y quienes siempre me apoyaron en momentos difíciles.

A los investigadores de ciencias computacionales por todas las enseñanzas y conocimientos transmitidos, y a la coordinación por todas las facilidades brindadas para el desarrollo de esta tesis.

A CONACyT por el apoyo económico otorgado al becario con número de registro 234672, para los estudios de posgrado.

Dedicatorias

A mi madre que siempre me ha dado
su amor incondicional y sincero.

A mi padre que creyó en mí.

A mi hermana y sobrinitos, a
quienes amo y adoro.

A Yesica, musa de mi inspiración y
el amor de mi vida, cuyo
sentimiento ha llenado de luz mi
camino

Índice general

Índice de figuras **XIII**

Índice de tablas **XV**

1. Introducción **1**

1.1. Descripción de problema 1

1.1.1. Objetivos 4

1.2. Metodología 5

1.3. Solución Propuesta 5

1.4. Contribución 7

1.5. Organización del documento 7

2. Fundamento teórico **9**

2.1. Introducción 10

2.2. Selección de atributos 10

2.2.1. Métodos de filtrado 11

2.2.2. Método de envoltura 16

2.3. Conjuntos difusos 16

2.3.1. Funciones de pertenencia 17

2.4. Clasificadores difusos 20
2.4.1. Red neuronal relacional difusa 20
2.4.2. K vecinos más cercanos difusos 26
2.4.3. Árbol de decisión difuso 29
2.4.4. Bosque de decisión difuso 36
2.5. Ensamble de clasificadores 37
2.5.1. Nivel de combinación 38
2.5.2. Diversidad en clasificadores 40
2.6. Algoritmo evolutivo 41
2.7. Métodos de evaluación 45
2.7.1. Validación cruzada con k particiones 45
2.7.2. Matriz de confusión 46
2.7.3. Curvas ROC 47
2.8. Curvas de costo 49
2.9. Resumen 50

3. Trabajo relacionado 53
3.1. Clases traslapadas 53
3.2. Selección de modelo 56
3.3. Clasificación de leucemia aguda 58
3.4. Resumen 60
3.5. Conclusiones 60

4. Método propuesto 63
4.1. Arquitectura general 63
4.1.1. Codificación del cromosoma 64
4.1.2. Operaciones genéticas 73

4.1.3. Prevención de la extinción 76
4.1.4. Función de aptitud 76
4.2. Modelo jerárquico 78
4.3. Discusión 80

5. Experimentos y Resultados 83
5.1. Descripción de los Experimentos 83
5.2. Clasificación de células de leucemia 84
5.2.1. Experimentos realizados 84
5.2.2. Resultados 86
5.3. Comparación con otros trabajos 97
5.4. Bases de datos del repositorio UCI 99
5.5. Discusión 101

6. Conclusiones y Trabajo Futuro 103
6.1. Contribución 104
6.2. Trabajo futuro 105

Apéndices 106

A. Descripción de la base de datos de leucemia aguda 109

B. Comparación de resultados y pruebas de significancia estadística 113
B.1. Comparación de resultados 113
B.2. Prueba de significancia estadística 116

Bibliografía 125

Índice de figuras

1.1. Representación de instancias en base de datos de leucemia 3
1.2. Modelo para clasificación de datos traslapados 6

2.1. Ejemplos de funciones de pertenencia 18
2.2. Ejemplos de árboles de decisión 31
2.3. Árbol de decisión difuso 36
2.4. Matriz de confusión binaria 46
2.5. Ejemplos de curvas ROC 48
2.6. Curva de costo 51

4.1. Arquitectura del método propuesto 64
4.2. Proceso de cruza y mutación 74
4.3. Jerarquía propuesta para la clasificación de leucemia aguda 78
4.4. Diagrama de la jerarquía 80

5.1. Ejemplos de funciones de pertenencia resultantes 96

Índice de tablas

2.1. Conjunto de datos para el ejemplo de construcción de un árbol difuso 33

3.1. Tabla comparativa del trabajo relacionado 61

4.1. Codificación de bits para selección de atributos 65
4.2. Codificación de bits para parámetros de selección de atributos . . 66
4.3. Codificación de bits para parámetros de selección de atributos . . 67
4.4. Codificación de bits para el tipo de función de pertenencia 68
4.5. Codificación de bits para el número de propiedades lingüísticas . . 68
4.6. Codificación de bits para parámetros de las funciones de pertenencia 69
4.7. Codificación de bits para parámetros de clasificadores 71
4.8. Codificación de bits para parámetros de clasificadores 72

5.1. Resultados obtenidos con modelo individual en clasificación binaria 87
5.2. Resultados obtenidos con modelo individual en clasificación multiclase 88
5.3. Resultados obtenidos con modelos con ensamble en clasificación binaria 90
5.4. Resultados obtenidos con modelos con ensamble en clasificación multicase 91

5.5. Modelos seleccionados con el método propuesto 93
5.6. Resultados de clasificación jerárquica 97
5.7. Comparación de resultados 98
5.8. Prueba de significación estadística 99
5.9. Resultados sobre bases de datos del UCI 100

A.1. Descripción de base de datos de leucemia 110
A.2. Características morfológicas 111
A.3. Características estadísticas, textura y componentes principales . . 112
A.4. Características de proporción 112

B.1. Comparación de resultados obtenidos en base de datos de leucemia aguda 114
B.2. Umbrales para distintos valores de α 118
B.3. Prueba de Wilcoxon 118
B.4. Valores críticos de la distribución de Wilcoxon para $N = 34$ . . . 121
B.5. Diferencias en los resultados de clasificación y orden. El orden es mostrado entre paréntesis 121

Capítulo 1

Introducción

1.1. Descripción de problema

La tarea de clasificación se refiere a mapear un ejemplo, descrito por sus características, a una etiqueta de clase [39]. La clasificación ha sido ampliamente usada en diversas aplicaciones tales como el reconocimiento de caracteres manuscritos [32, 35], diagnóstico de enfermedades [2], reconocimiento de objetos en imágenes [5], entre otros. Sin embargo, muchas de estas aplicaciones tienden a compartir características entre sus diferentes clases, lo que hace difícil la asignación a una clase específica. Por ejemplo, la leucemia aguda es una enfermedad difícil de diagnosticar ya que sus subtipos conocidos como "L1" y "L2" tienden a compartir características que los hacen pertenecer a un grupo más general conocido como "LLA" (Leucemia Linfoide Aguda), pero tienen también características propias que los describen como pertenecientes a alguno de los subtipos. A ejemplos de este tipo, donde diferentes clases comparten características comunes, se les conoce como *clases traslapadas*.

El problema de clases traslapadas se refiere a que instancias de diferentes clases

tienen características similares. Si dichas instancias son representadas como puntos en el espacio de características, estos puntos estarían cercanos, dificultando así su separación de forma lineal.

Actualmente, para la clasificación de problemas con clases traslapadas, existen dos enfoques principales [35]: el primero consiste en unir las clases que se traslapen en una meta-clase; el segundo consiste en la separación de las clases mediante estrategias como la extracción de características más descriptivas o en el refinamiento de la clasificación. Cada una de estas estrategias tiene ventajas y desventajas. Por ejemplo, con la combinación de clases en una meta-clase se reduce el número de clases y, en la clasificación, se asume que la sustitución de las clases (dentro de la meta-clase) es correcta. Sin embargo, los límites entre las regiones de decisión son ignorados [35]. Por otro lado, el refinamiento de la clasificación puede ser tratado mediante la construcción de clasificadores no lineales, los cuales tienden a producir límites complejos entre las regiones de decisión de las clases, lo que puede afectar la generalidad del clasificador (sobreajustar el clasificador al conjunto de entrenamiento). Otra opción es la combinación de clasificadores; esto último pretende conjuntar la decisión individual de diferentes clasificadores y con ello hacer mínimo el error de clasificación, mas surge otro problema: la selección de los clasificadores y la manera de combinarlos, conocido como problema de selección [22].

Un caso particularmente interesante se presenta en la base de datos de leucemia aguda construida por el IMSS en cooperación con el INAOE [37], la cual presenta traslape entre sus clases. La base de datos de leucemia aguda se caracteriza por contar con una jerarquía de clases; en el nivel superior de la jerarquía se encuentran las clases “LLA” (Leucemia Linfoide Aguda) y “LMA” (Leucemia Mieloide Aguda). La clase “LLA” se divide en las clases “L1” y “L2”, mientras que la clase “LMA” se

divide en las clases “M2”, “M3” y “M5”. La base de datos cuenta con 633 ejemplos –descritos por características extraídas del núcleo, citoplasma y la célula– y cinco clases. Debido a las características que comparten algunos subtipos de leucemia, una instancia puede parecer un ejemplo válido para más de una clase; siendo necesaria su asignación a una de las clases, se requiere clasificarla con el menor error posible. En la figura 1.1 se muestra la representación de instancias para dos características tomadas de la base de datos de leucemia aguda del INAOE-IMSS [37] que presenta traslape entre las clases. De la figura 1.1 es visible que las cinco clases no son linealmente separables si sólo se consideran estas dos características.

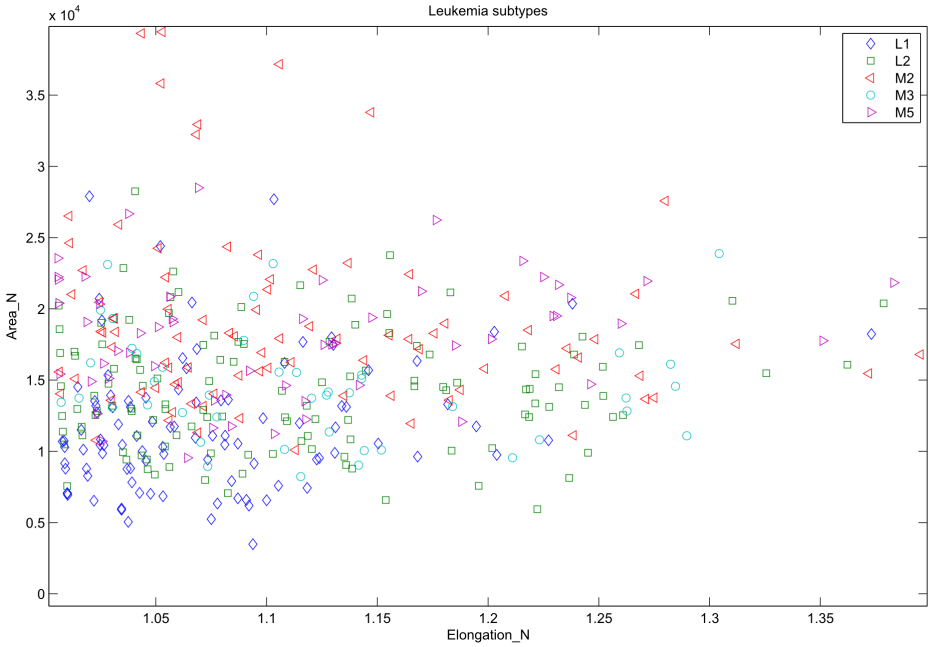


Figura 1.1: Representación de instancias usando dos características y cinco clases en el conjunto de datos de leucemia aguda INAOE-IMSS

Algunos trabajos que han realizado clasificación de leucemia aguda proponen el uso de algoritmos de aprendizaje, como redes neuronales, para la clasificación

de los tipos de leucemia [37, 2]. Sin embargo, el problema central se encuentra en la clasificación satisfactoria de los subtipos. En el presente trabajo se aborda el problema de clases traslapadas a través de ensambles, lo cual ha probado tener éxito en otros trabajos como en [19, 45], y añadiendo un componente de pre-procesamiento difuso de características, lo cual no ha sido abordado con anterioridad en los trabajos de clasificación de leucemia aguda basados en la morfología de la célula. El reto principal en el uso de ensambles está en la definición de qué clasificadores combinar y cómo hacerlo, lo cual se ataca en esta investigación con el uso de algoritmos genéticos.

1.1.1. Objetivos

Objetivo general

Desarrollar un modelo para la clasificación de clases traslapadas basado en procesamiento difuso de características y sustentado en ensambles, apropiado para el caso de leucemia aguda.

Objetivos específicos

- Definir un método automático de selección de características para una base de datos determinada.

- Definir un método automático para determinar el tratamiento difuso, basado en grados de traslape, aplicable a un conjunto de características.

- Definir un método para la construcción automática de ensambles basado en un repositorio de clasificadores.

- Definir un algoritmo genético para la selección automática de un modelo de clasificación.

1.2. Metodología

En esta sección se describe la metodología que se siguió para el desarrollo exitoso del presente trabajo de investigación.

1. Analizar los métodos existentes para detección de traslapes y para manejo de subclases; evaluar sus ventajas y desventajas.
2. Analizar los métodos existentes para la construcción de ensambles, evaluar sus ventajas y desventajas.
3. Diseñar e implementar un modelo automático de difusión de características.
4. Diseñar e implementar un modelo de selección de clasificadores y ensambles.
5. Probar el modelo con el caso de leucemia aguda y otras bases de datos.

1.3. Solución Propuesta

En la figura 1.2 se muestra la arquitectura del sistema propuesto. Cada bloque se describe a continuación:

1. Selección automática de atributos.

   La selección de atributos permite seleccionar un conjunto de atributos relevantes para la base de datos; las estrategias utilizadas son filtrado y envoltura.

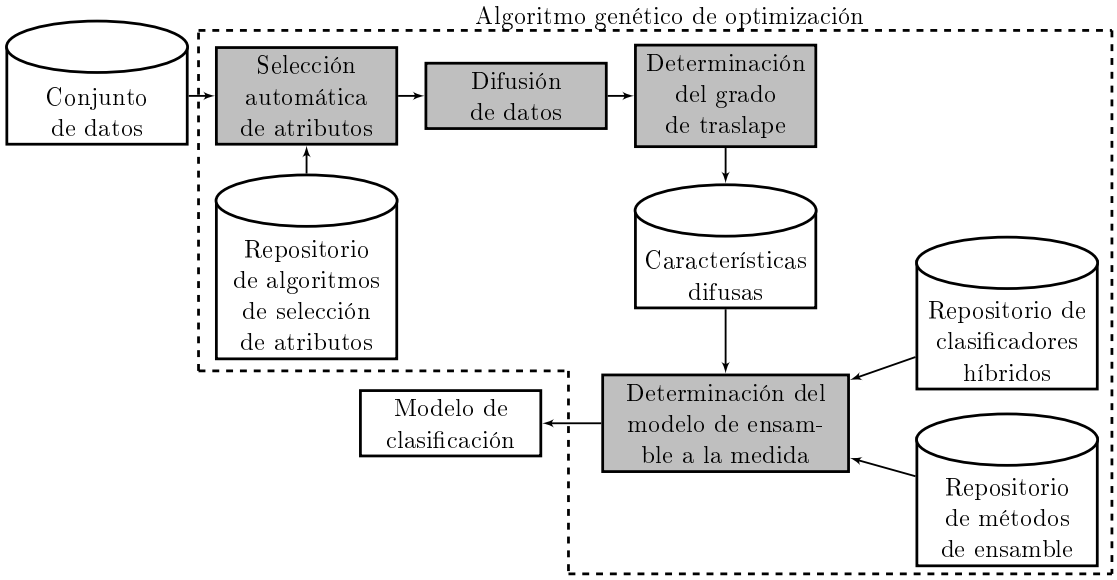


Figura 1.2: Modelo para clasificación de datos traslapados

2. Difusión de los datos.

   Para este trabajo se usa lógica difusa para asignar a una instancia, grados de pertenencia a las diferentes clases. Para lograr lo anterior es necesario dar un tratamiento difuso a las características del conjunto de datos. Los conjuntos difusos son, inicialmente, uniformemente distribuidos y se determina el tipo de función de pertenencia adecuada para el problema.

3. Determinación del grado de traslape.

   Este módulo determina el grado de traslape del conjunto de datos difusos y los parámetros de las funciones de pertenencia adecuados para el conjunto de datos.

4. Determinación del modelo de ensamble a la medida.

A partir de un repositorio de clasificadores se escogerán aquellos que sean más adecuados al problema, así como la manera en como combinar los mismos.

Algoritmo de optimización

Para determinar la combinación adecuada de algoritmo de selección de atributos, tipo de función de pertenencia, clasificador difuso y método de combinación se propone usar algoritmos evolutivos, en particular algoritmos genéticos.

El sistema se evalúa con la base de datos de leucemia aguda descrita en el apéndice A y con bases de datos del repositorio UCI [51]. Sin embargo, la solución propuesta permite su aplicación en bases de datos de dominios diferentes.

1.4. Contribución

La principal contribución de esta tesis es un método novedoso que permite seleccionar, de forma automática, un modelo de clasificación para un problema dado. Además, el sistema permite determinar el tratamiento difuso adecuado al problema para la asignación de grados de pertenencia de cada ejemplo a cada una de las clases. También, el método propuesto permite determinar el conjunto de clasificadores difusos que se conjuntan en un ensamble y la forma en la que sus decisiones individuales son combinadas.

1.5. Organización del documento

En el capítulo 2 se describe el fundamento teórico relacionado con la presente tesis, en particular sobre la selección de atributos, conjuntos difusos, clasificadores, métodos de ensamble y algoritmos de optimización. Seguidamente, en el capítulo

3 los trabajos más relevantes relacionados al tema de investigación son descritos. En el capítulo 4 se presenta el método propuesto; posteriormente, en el capítulo 5 los experimentos y resultados obtenidos son descritos a detalle. En el capítulo 6 se presentan las conclusiones de la tesis y posibles líneas a seguir en futuros trabajos; en el apéndice A se describe la base de datos usada en los experimentos y, finalmente, en el apéndice B los resultados obtenidos son comparados con otros trabajos y se presentan las pruebas de significación estadística relacionadas a dichos resultados.

Capítulo 2

Fundamento teórico

En este capítulo se describen los aspectos teóricos relacionados a este trabajo de investigación. Primeramente, en la sección 2.2 se aborda el tema de selección de atributos, describiendo los enfoques existentes y algunos algoritmos. En la sección 2.3 se describen los conceptos relacionados a la teoría de conjuntos difusos y los tipos de funciones de pertenencia, específicamente las funciones usadas en el desarrollo del trabajo. Posteriormente, en la sección 2.4 se presenta la teoría relacionada a clasificadores híbridos, los cuales hacen uso de la teoría de los conjuntos difusos. En la sección 2.5 se aborda el tema de ensamble de clasificadores. En la sección 2.6 se detallan aspectos relacionados a la teoría de los algoritmos evolutivos utilizados en el trabajo de investigación. Finalmente, en la sección 2.7 se describen las métricas de evaluación utilizadas para determinar el desempeño de los modelos de clasificación generados.

2.1. Introducción

En general, dentro del reconocimiento de patrones existen dos enfoques de aprendizaje: supervisado y no supervisado. En el supervisado, el aprendizaje es realizado a través de un conjunto de entrenamiento etiquetado, es decir un conjunto de ejemplos cuya salida es conocida. En el aprendizaje no supervisado, las salidas de los ejemplos no son conocidas [24].

La clasificación con aprendizaje supervisado asigna una etiqueta de clase a un ejemplo de entrada. La clasificación involucra dos pasos: el aprendizaje y la deducción. En el paso de aprendizaje, se presenta un conjunto de entrenamiento con sus correspondientes salidas y un algoritmo de aprendizaje; el objetivo del algoritmo de aprendizaje es producir una función hipótesis h, también conocido como clasificador, que mapee cada entrada a su salida correspondiente. En el paso de deducción, un ejemplo es presentado al clasificador y una etiqueta de clase es obtenida como salida [55]. Además del clasificador, otros aspectos involucrados en la construcción de un modelo de aprendizaje son la selección de atributos, así como algún procesamiento de los mismos. En la siguiente sección se describen algunas de éstas técnicas

2.2. Selección de atributos

La clasificación consiste en predecir a que clase pertenece un ejemplo que es descrito por sus características, también llamadas atributos. El conjunto de atributos usados para describir al ejemplo tiene un efecto sobre el desempeño del algoritmo de clasificación. Algunos de los atributos pueden ser irrelevantes, es decir, que no afectan de ninguna manera al atributo clase, o redundantes, aquellos

que no añaden algo nuevo para diferenciar las clases. Además, estos atributos pueden tener un efecto negativo sobre la precisión del clasificador, velocidad de aprendizaje o en la complejidad del modelo generado. Reduciendo el número de atributos irrelevantes o redundantes es posible mejorar el tiempo de aprendizaje, así como mejorar el entendimiento de los modelos generados y la precisión [8, 36].

La selección de atributos pretende determinar un subconjunto de M atributos de los N atributos originales, para $M < N$, tal que los atributos seleccionados sean relevantes al concepto meta, esto es, la etiqueta de clase. Existen dos enfoques básicos para la selección de atributos: los métodos de filtrado y los de envoltura. Los métodos de filtrado son independientes del algoritmo de aprendizaje y se basan en las características generales de los atributos, tales como la separación de las clases o atributos que tengan una alta correlación con la clase; este enfoque tiende a ser rápido. Por otro lado, los métodos de envoltura son dependientes del algoritmo de aprendizaje y, en general, tienden a dar buenos resultados; sin embargo, son más lentos que los de filtrado [53]. Enseguida se describen algunos métodos de filtrado para selección de atributos, los cuales están contenidos dentro del repositorio de algoritmos de selección de atributos construido en el modelo propuesto (véase figura 1.2), y el método de envoltura considerados en este trabajo de investigación.

2.2.1. Métodos de filtrado

Los métodos de filtrado usan heurísticas que se basan en las características generales de los datos, independientemente del algoritmo de aprendizaje, para evaluar un subconjunto de atributos. Estos métodos son más aptos para datos con alta dimensión que los métodos de envoltura.

Selección de atributos basada en teoría de información

La selección de atributos basada en teoría de la información evalúa la ganancia de información de cada atributo individualmente, ordenándolos en forma descendente de acuerdo con el mérito obtenido [24]. La información proporcionada por el atributo puede ser calculada como:

$$I(A_i) = -\sum_{j=1}^{k} P(v_j) \lg_2 P(v_j) \tag{2.1}$$

donde A_i es el i-ésimo atributo del conjunto de datos, k es el número de valores del atributo, $P(v_j)$ es la probabilidad del j-ésimo valor del atributo.

La entropía de un atributo mide el grado de desorden o dispersión de los datos con respecto a la clase, y se expresa como:

$$E(A_i) = \sum_{j=1}^{k} P(v_j) I(C|A_i = v_j) \tag{2.2}$$

donde A_i es el i-ésimo atributo del conjunto de datos, k es el número de valores del atributo, $P(v_j)$ es la probabilidad del j-ésimo valor del atributo y $I(C|A_i = v_j)$ es la información proporcionada por el atributo clase dado el j-ésimo valor del i-ésimo atributo.

La ganancia de información se define como la cantidad de información ganada al usar un determinado atributo, es decir, que tan bien se pueden discriminar las clases a partir de un atributo dado. La ganancia de información se calcula como:

$$G(A_i) = I(C) - E(A_i) \tag{2.3}$$

donde $I(C)$ representa la información obtenida por el atributo clase y $E(A_i)$ la entropía del i-ésimo atributo.

Los atributos son ordenados en forma descendente y se seleccionan los M mejores atributos (lo cuáles, en esta tesis, son determinados por un algoritmo genético). Cabe hacer notar que cuando los atributos son valores continuos, éstos primero deben de ser discretizados.

Selección de atributos por χ^2

Agunas medidas estadísticas también son usadas para evaluar individualmente los atributos. La selección de atributos por χ^2 es una medida usada para determinar la dependencia de un atributo con la clase. Cada atributo es evaluado con la expresión de la ecuación siguiente:

$$\chi^2(A_i) = \sum_{p=1}^{k} \sum_{q=1}^{r} \frac{(o_{pq} - e_{pq})^2}{e_{pq}} \tag{2.4}$$

donde k es el número de valores únicos para el atributo A_i, r es el número de clases, o_{pq} es la frecuencia conjunta observada entre el atributo y la clase y e_{pq} es el valor esperado de frecuencia, que puede ser calculado como:

$$e_{pq} = \frac{|A_i = v_p||C = c_q|}{N} \tag{2.5}$$

donde N es el número total de ejemplos, $|A_i = v_p|$ es el número de ejemplos con el valor v_p en el atributo A_i y $|C = c_q|$ es el número de ejemplos de la clase c_q.

Los atributos del conjunto de datos son evaluados y ordenados en forma descendente, se seleccionan los M mejores atributos (lo cuáles, en esta tesis, son determinados por el algoritmo genético). Los atributos numéricos primeramente son discretizados.

Selección de atributos basado en correlación

Hall [23] propone seleccionar atributos basados en correlación [1]. La idea principal de este enfoque es seleccionar un subconjunto de atributos que tengan una alta correlación con la clase y baja correlación entre ellos. Bajo este enfoque, se selecciona un subconjunto de atributos y se evalúa usando la expresión siguiente:

$$Ms = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2.6}$$

donde Ms es el mérito del subconjunto de atributos, k es el número de atributos seleccionados, $\overline{r_{ff}}$ es el promedio del grado de correlación entre los atributos y $\overline{r_{cf}}$ es el promedio del grado de correlación de los atributos con la clase.

Para la selección del subconjunto se requiere de una estrategia de búsqueda y se selecciona el subconjunto de atributos con el mérito más alto.

Relief y *ReliefF*

Relief es un algoritmo basado en el peso de los atributos y está inspirado en algoritmos de aprendizaje basados en instancias [8]. Se genera un vector de pesos para cada atributo, el cual es inicializado en cero. Del conjunto de entrenamiento, *Relief* selecciona un subconjunto del mismo. Posteriormente, una instancia es seleccionada aleatoriamente y se busca su vecino más cercano de la misma clase (conocido como *nearHit*) y el vecino más cercano de la otra clase (conocido como *nearMiss*); esto es realizado con una medida de distancia. El vector de pesos es actualizado de la siguiente manera:

$$W_j = W_j - diff(x_j, nearHit)^2 + diff(x_j, nearMiss)^2 \tag{2.7}$$

[1] La correlación es una medida del grado de dependencia de las variables [23]

donde W_j y x_j son, respectivamente, el peso y valor del j-ésimo atributo; $diff$ representa la diferencia existente entre los dos valores.

El proceso es repetido tantas veces como instancias se tengan en el subconjunto. Intuitivamente, *Relief* se basa en la idea de que los atributos más relevantes son aquellos que pueden ser distinguidos mejor y, por tanto, pueden ser distinguibles entre elementos de la otra clase. El subconjunto de atributos seleccionados es aquel cuyo peso sea mayor que un umbral, el cual es definido por el usuario y, en esta tesis, se seleccionan los atributos que tengan un peso mayor que cero.

Uno de los principales problemas de *Relief* es que sólo trabaja en problemas con dos clases y es sensible a datos con ruido. Con la finalidad de evitar estos problemas se ha propuesto *ReliefF* [48], el cual sigue la idea de *Relief*. *ReliefF* selecciona aleatoriamente una instancia y se buscan los k vecinos más cercanos de la misma clase, llamados *nearest hit* (nH_h), y los k vecinos más cercanos para cada una de las otras clases, llamados *nearest miss* (nM_h). Posteriormente, se actualiza el vector de pesos usando la ecuación (2.8) y el proceso es repetido.

$$W_j = W_j - \frac{1}{m \times k}\left(\sum_{h=1}^{k} diff(x_j, nH_h) + \frac{P(c_l)}{1 - P(c_{R_i})}\sum_{h=1}^{k} diff(x_j, nM_h)\right) \quad (2.8)$$

donde W_j y x_j son, respectivamente, el peso y valor del j-ésimo atributo, $P(C_l)$ es la probabilidad a *priori* de la clase l y $P(c_{R_i})$ es la probabilidad a *priori* de la clase a la que pertenece la instancia R_i; $diff$ representa la diferencia existente entre los dos valores. El algoritmo 2.1 describe el proceso de selección de atributos por *ReliefF*

Algoritmo 2.1 *ReliefF* [48]

Entrada: Conjunto de datos de entrenamiento etiquetado
Salida: Vector de pesos W
  Establecer el vector de pesos W a ceros
  Para $i = 1$ **hasta** m **hacer**
    Seleccionar aleatoriamente una instancia R_i
    Encontrar los k vecinos más cercanos de la misma clase de R_i
    Para cada clase $C \neq clase\,(R_i)$ **hacer**
      Encontrar los k vecinos más cercanos de R_i de la clase c_l
    Fin Para
    Para cada atributo en el conjunto de datos **hacer**
      Actualizar vector de pesos usando la ecuación (2.8)
    Fin Para
  Fin Para

2.2.2. Método de envoltura

La selección de atributos por estrategia de envoltura usa un algoritmo de clasificación para evaluar al subconjunto de atributos. El desempeño de un algoritmo de clasificación puede ser afectado por las características del conjunto de datos. A diferencia del método de filtrado, este método intenta considerar el clasificador final para evaluar el subconjunto de atributos, y seleccionar el subconjunto de atributos que obtenga la mayor precisión, dado el algoritmo de clasificación. En general, los atributos seleccionados por este método tienden a brindar mejores resultados que los métodos de filtrado; sin embargo, su costo computacional es mayor.

2.3. Conjuntos difusos

La teoría de los conjuntos difusos fue propuesta por Zadeh en 1965 [56]. Los conjuntos difusos son una extensión de los conjuntos duros para tratar con el concepto de verdad parcial, lo cual permite modelar la incertidumbre del lenguaje.

En la teoría de los conjuntos difusos, a diferencia de los conjuntos clásicos, un elemento pertenece a un conjunto en un determinado grado, indicando el grado de certeza con el que pertenece a ese conjunto. Formalmente un conjunto difuso se define de la siguiente manera:

Sea $\mathbf{X}$ el universo de discurso, y $x \in \mathbf{X}$ un elemento del dominio $\mathbf{X}$. El conjunto difuso $\mathbf{A}$ es caracterizado por una función que mapea el grado de pertenencia del elemento x al conjunto $\mathbf{A}$:

$$\mu_A : \mathbf{X} \to [0, 1] \tag{2.9}$$

La función μ_A se encarga de mapear el grado de pertenencia de un elemento al conjunto difuso, y comúnmente se le conoce como función de pertenencia o función característica del conjunto difuso. La función de pertenencia puede tener varias formas, entre las que destacan la tipo triangular, trapezoidal, campana y gaussiana, entre otras. Enseguida se describen las funciones de pertenencia más comunes, usadas en este trabajo para la difusión de características:

2.3.1. Funciones de pertenencia

Función de pertenencia triangular

La función triangular se define como:

$$f(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \tag{2.10}$$

Los parámetros a, b y c representan, respectivamente, los puntos izquierdo, centro y derecho del triángulo. Un ejemplo se muestra en la figura 2.1a.

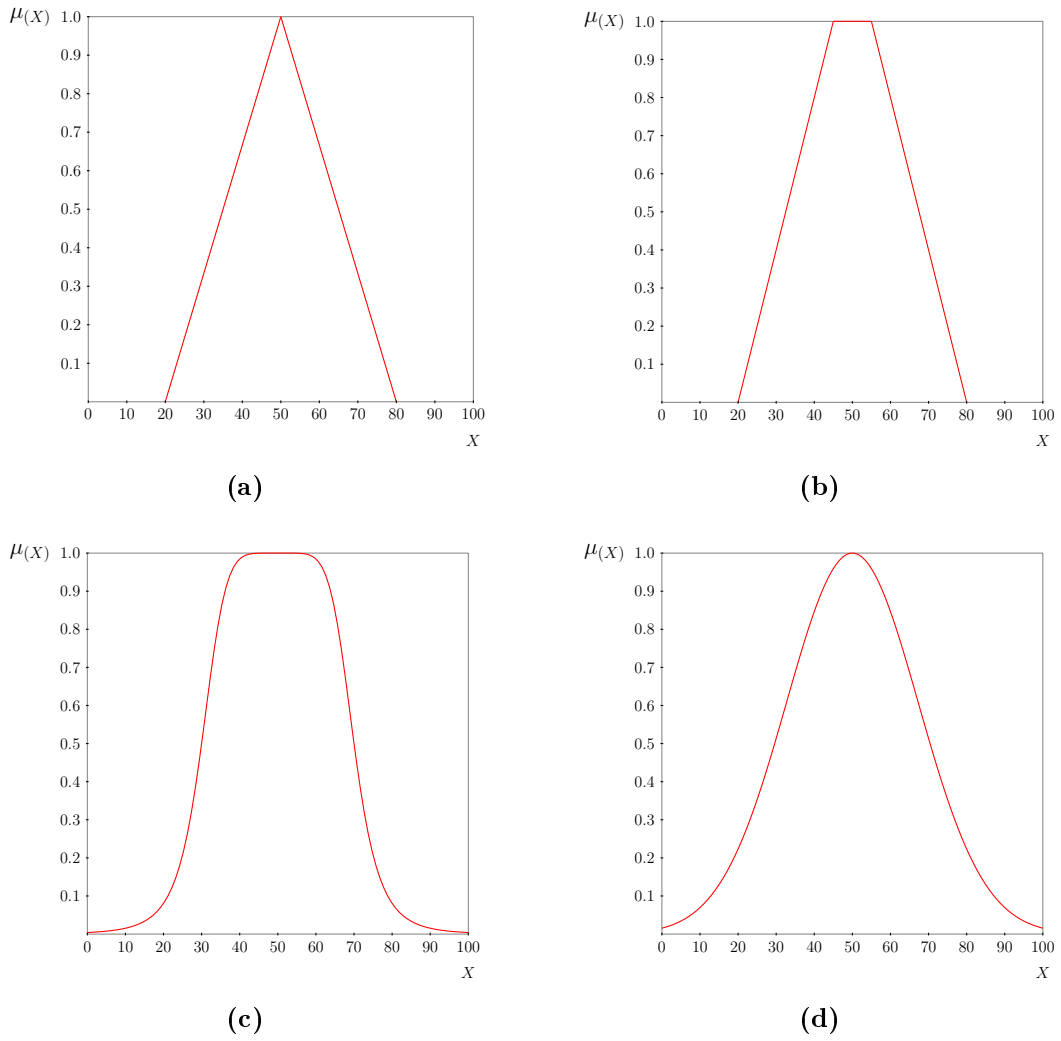


Figura 2.1: Ejemplos de funciones de pertenencia. (a) función tipo triangular con $a = 20$, $b = 50$ y $c = 80$; (b) función tipo trapezoidal con $a = 20$, $b = 45$, $c = 55$ y $d = 80$; (c) función tipo campana con $a = 20$, $b = 30$ y $c = 50$; (d) función tipo gaussiana con $a = 50$, $\sigma = 17$

Función de pertenencia trapezoidal

La función trapezoidal está definida por los puntos a, b, c y d que representan, respectivamente, el punto donde la función comienza a crecer, el punto donde la función comienza a valer uno, el punto donde la función empieza a decrecer y el punto en donde la función regresa a cero. La ecuación (2.11) describe este tipo de función y en la figura 2.1b se muestra un ejemplo de esta función.

$$f(x; a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{cases} \tag{2.11}$$

Función de pertenencia campana

La ecuación (2.12) define la función campana mostrada en la figura 2.1c. En la ecuación (2.12) a, b, c representan, respectivamente, el ancho, la pendiente y el centro.

$$f(x; a, b, c) = \frac{1}{1 + (\frac{x-c}{a})^{2b}} \tag{2.12}$$

Función de pertenencia gaussiana

La función gaussiana está definida por la ecuación (2.13) y su forma se presenta en la figura 2.1d.

$$f(x;\sigma,c) = \exp^{\frac{-(x-c)^2}{2\sigma^2}} \tag{2.13}$$

donde c es la media y σ la desviación estándar de la función.

2.4. Clasificadores difusos

De acuerdo con Kuncheva [33] un clasificador difuso es aquel que usa la lógica difusa en cualquier parte de su proceso, ya sea que trabaje con datos difusos, durante el proceso de entrenamiento o durante su operación.

Los clasificadores descritos a continuación forman parte del repositorio de clasificadores mostrado en la figura 1.2.

2.4.1. Red neuronal relacional difusa

La red neuronal relacional difusa (FRNN, por sus siglas en inglés) [47], basada en la estructura propuesta por Pedrycz [40], está formada por dos capas: la capa de entrada y la capa de salida. En la capa de entrada se tienen $N \times n$ neuronas, donde n es el número de características que describen al patrón de entrada y N es el número de propiedades lingüísticas asociadas a cada característica. La capa de salida está formada por k neuronas, cada una de ellas está asociada con cada una de las k clases. Además, al incluir la teoría de los conjuntos difusos, la FRNN permite tener regiones de decisión no lineales y traslapadas.

En el modelo de la FRNN cada neurona de la capa de entrada está conectada con cada neurona de la capa de salida. Las conexiones son descritas por medio de pesos, o relaciones difusas, $\mathbf{R} : \mathbf{X} \times \mathbf{Y} \rightarrow [0,1]$ de las neuronas de la capa de entrada a las neuronas de la capa de salida.

El proceso completo del modelo de la FRNN está dividido en dos partes, una para el aprendizaje y la otra para el procesamiento. La fase de aprendizaje se encarga de colectar los ejemplos de entrada y determinar las relaciones difusas con las clases. La fase de procesamiento utiliza las relaciones difusas, obtenidas en la fase de aprendizaje, para clasificar ejemplos desconocidos. Enseguida se describe a cada una de ellas.

Fase de aprendizaje

La fase de aprendizaje está compuesta por tres módulos: el extractor de características lingüísticas, el estimador de salida deseada y el entrenamiento de la red neuronal.

El proceso de aprendizaje comienza con la entrada de los ejemplos de entrenamiento, los cuales están descritos por n características. El extractor de características lingüísticas transforma las n características en $N \times n$ características, tal que cada una de las n características está asociada con N propiedades lingüísticas, es decir, cada característica es representada por su valor de pertenencia a cada una de las propiedades lingüísticas. Los valores de pertenencia son obtenidos de las funciones de pertenencia descritas en la sección 2.3.1.

El segundo módulo, el estimador de salida deseada, se encarga de calcular el grado de pertenencia de cada ejemplo –del conjunto de entrenamiento– para cada una de las k clases. Para definir el valor de pertenencia, primeramente, se calcula la distancia pesada del j-ésimo ejemplo a la l-ésima clase, de las k clases, como se muestra en la siguiente ecuación:

$$z_{jl} = \sqrt{\sum_{i=1}^{n} \left(\frac{\mathbf{F_{ji}} - \mu_{li}}{\sigma_{li}} \right)^2} \;\; | \;\; l = 1 \ldots k \tag{2.14}$$

donde $\mathbf{F_{ji}}$ es la i-ésima característica del j-ésimo ejemplo, μ_{li} y σ_{li} representan, respectivamente, la media y desviación estándar de la i-ésima característica de la l-ésima clase.

Una vez que la distancia pesada es calculada, el valor de pertenencia del j-ésimo ejemplo a la l-ésima clase es definido por:

$$\mu_l(\mathbf{F_j}) = \frac{1}{1 + \left(\frac{z_{jl}}{fd}\right)^{fe}} \tag{2.15}$$

donde fe es el generador exponencial difuso y fd es el generador denominacional difuso que controlan la cantidad de difusión, en este caso el valor de pertenencia.

De la ecuación (2.15) se observa que una mayor distancia de una instancia a la clase es representada por un valor de pertenencia menor a esa clase. Lo que significa que como los límites de las regiones de decisión son difusos, una instancia puede pertenecer con diferentes grados a más de una clase.

El entrenamiento de la FRNN tiene como objetivo modificar los pesos (relaciones difusas) de las conexiones de la capa de entrada con la capa de salida. Para ello, la salida de la red es obtenida y se calcula el error en la capa de salida. El error es representado como la diferencia entre la salida real y la salida deseada. Finalmente, la salida de la red neuronal es una matriz que contiene las relaciones difusas para poder mapear futuros ejemplos a las correspondientes clases.

La modificación de los pesos es realizada iterativamente, dado un número de épocas de entrenamiento, como sigue [47]:

1. Se calcula la salida de la FRNN usando la composición *máx-mín* (producto relacional *circlet*), como se muestra en la ecuación siguiente:

$$\mathbf{Y_l} = \left[\bigvee_{i=1}^{n} (\mathbf{X}(x_{ji}) \wedge \mathbf{R}(a_{il}))\right] \bigvee \mathbf{V}(v_l) \tag{2.16}$$

donde $\mathbf{X_j} = \{x_1, x_2, \ldots, x_n\}$ y $\mathbf{Y} = \{y_1, y_2, \ldots, y_k\}$ son, respectivamente, el conjunto de neuronas en la capa de entrada y en la capa de salida; $\mathbf{R}$ representa la matriz de pesos o relaciones difusas de $\mathbf{X}$ a $\mathbf{Y}$ y $\mathbf{V} = \{v_1, v_2, \ldots, v_k\}$ representa el *bias*[2] para cada clase.

2. Se calcula el incremento de los valores de las relaciones difusas:

$$\Delta a_l = \frac{\partial Q}{\partial a_l} = -\sum_{j:y_j>t_j} \frac{\partial f(x_j; a_l v_l)}{\partial a_l} + \sum_{j:y_j<t_j} \frac{\partial f(x_j; a_l v_l)}{\partial a_l} \tag{2.17}$$

tal que:

$$\frac{\partial f(x_j; a_l v_l)}{\partial a_l} = \begin{cases} 1 & \text{si (a)} \bigvee_{i=1}^{n} (a_{il} \wedge x_{ji}) \geq v_l \\ & \quad \text{(b)} \bigvee_{i=1, j\neq s}^{n} (a_{il} \wedge x_{ji}) \leq (a_{sl} \wedge x_{js}) \\ & \quad \text{(c)} a_{sl} \leq x_{js} \\ 0 & \text{en otro caso} \end{cases}$$

donde $\vee$ representa la operación difusa de unión (máximo) y $\wedge$ la operación difusa de intersección (mínimo). De esta forma la diferencial vale 1 si se cumplen las tres reglas, de lo contrario, vale 0.

3. Se calcula el incremento para el vector de *bias* de acuerdo con la ecuación (2.18).

$$\Delta v_l = \frac{\partial Q}{\partial v_l} = -\sum_{j:y_j>t_j} \frac{\partial f(x_j; a_l v_l)}{\partial v_l} + \sum_{j:y_j<t_j} \frac{\partial f(x_j; a_l v_l)}{\partial v_l} \tag{2.18}$$

tal que:

$$\frac{\partial f(x_j; a_l v_l)}{\partial v_l} = \begin{cases} 1 & \text{si (a)} \bigvee_{i=1}^{n} (a_{il} \wedge x_{ji}) \leq v_l \\ 0 & \text{en otro caso} \end{cases}$$

[2]Es el valor mínimo con el cual una instancia pertenence a una clase

En las ecuaciones (2.17) y (2.18) t_j y y_j representan, respectivamente, el valor de la salida real y la salida deseada.

4. Los valores de las relaciones difusas y de *bias* son actualizados conforme a las ecuaciones (2.19) y (2.20).

$$a\left(m+1\right)=a\left(m\right)+\Psi_1\left(m\right)\left[\frac{\Delta a\left(m+1\right)+\eta\Delta a\left(m\right)}{Nn}\right] \tag{2.19}$$

$$v\left(m+1\right)=v\left(m\right)+\Psi_2\left(m\right)\left[\frac{\Delta v\left(m+1\right)+\eta\Delta v\left(m\right)}{Nn}\right] \tag{2.20}$$

donde Nn es el número de caraterísticas difusas, m es la época actual, Ψ_1 y Ψ_2 (tal que $\Psi_1(m)=\Psi_2(m)=1/\left(\sqrt{m}+1\right)$) son funciones no crecientes que controlan la influencia en el incremento de Δa_l y Δ_{vl}, η es la tasa de aprendizaje que especifica el nivel de modificación de los parámetros a aprender.

Una vez que el entrenamiento ha terminado, la salida es una matriz $\mathbf{R}$ que contiene los valores de las relaciones difusas que servirán para mapear nuevos ejemplos a las correspondientes clases.

Fase de procesamiento

Una vez que la fase de aprendizaje se ha completado, la información que se obtuvo es utilizada para clasificar patrones desconocidos; la fase de procesamiento es la que se encarga de esta tarea. Los módulos que forman la fase de procesamiento son: extractor de propiedades lingüísticas, clasificador difuso y módulo de toma de decisiones.

El primer módulo, el extractor de propiedades lingüísticas, es similar al descrito en la fase de aprendizaje; sin embargo, en la fase de procesamiento no se calculan

los parámetros de las funciones de pertenencia. Los valores de pertenencia son calculados usando las funciones de pertenencia correspondientes en la fase de entrenamiento.

El clasificador difuso usa los valores de pertenencia calculados por el extractor de propiedades lingüísticas, de la fase de procesamiento, y la matriz de relaciones $\mathbf{R}$ para clasificar nuevas instancias a sus correspondientes clases. La clasificación puede ser llevada en tres formas diferentes [47]:

- Mediante la composición *máx-mín*, como se expresa en la siguiente ecuación:

$$Y\left(y_j\right) = \bigvee \left[\mathbf{X}\left(x_{ji}\right) \wedge \mathbf{R}\left(a_{il}\right)\right] = \text{máx}\left[\text{mín}\left(\mathbf{X}\left(x_{ji}\right), \mathbf{R}\left(a_{il}\right)\right)\right] \tag{2.21}$$

- Al reemplazar el operador mín por la media geométrica, se obtiene la salida *máx-media geométrica*, que es definida por la ecuación siguiente:

$$Y\left(y_j\right) = \text{máx}\left[\left(\mathbf{X}\left(x_{ji}\right) * \mathbf{R}\left(a_{il}\right)\right)^{1/2}\right] \tag{2.22}$$

- A través del *producto relacional cuadrado*, definido por:

$$Y\left(y_j\right) = \text{mín}\left(\mathbf{X}\left(x_{ji}\right) \leftrightarrow \mathbf{R}\left(a_{il}\right)\right) \tag{2.23}$$

  donde $\leftrightarrow$ es el operador de bi-implicación y, por la bi-implicación de Lukasiewicz [4], se tiene que:

$$Y\left(y_j\right) = \text{mín}\left(1 - \left|\mathbf{X}\left(x_{ji}\right) - \mathbf{R}\left(a_{il}\right)\right|\right)$$

En los tres casos la salida es un vector que contiene los grados de pertenencia de la instancia a las diferentes clases. El tipo de salida es determinado por el algoritmo genético. Finalmente, el módulo de toma de decisiones asigna la instancia a la clase con el valor de pertenencia más alto y, si la clase asignada es igual a la clase deseada, se cuenta como una clasificación correcta.

2.4.2. K vecinos más cercanos difusos

El clasificador de los K vecinos más cercanos (*KNN*, por sus singlas en inglés) [7, 13] es un método de clasificación supervisada. Consiste en almacenar, únicamente, los datos de entrenamiento sin crear algún tipo de abstracción en forma de reglas u otro tipo. Para clasificar una nueva instancia se calcula su distancia con los ejemplos previamente almacenados, esto es realizado usando una medida de similitud. Seguidamente, la instancia es clasificada por la mayoría de votos de sus k vecinos más cercanos. El algoritmo 2.2 describe el método de los vecinos más cercanos.

Uno de los problemas encontrados en el uso del clasificador *KNN* es que, normalmente, cada vector de ejemplo es considerado con igual importancia en la asignación de la clase, causando dificultad en conjuntos de datos donde los ejemplos se traslapen, es decir ejemplos de diferentes clases con características similares. Además, una vez que una instancia es asignada a una clase, no es posible saber el grado de pertenencia a dicha clase. Estos problemas llevan a la inclusión de la teoría de los conjuntos difusos al clasificador *KNN*, obteniendo el clasificador de los K-vecinos más cercanos difuso [31].

El clasificador difuso de los K vecinos más cercanos es similar a la versión dura en el sentido de que el conjunto de entrenamiento debe ser etiquetado, el cual sólo es almacenado para posteriormente encontrar los K vecinos más cercanos. Sin embargo, una vez encontrados los vecinos, el proceso de clasificación difiere.

El objetivo de la versión difusa es asignar a la instancia a ser clasificada grados de pertenencia a las diferentes clases. El valor de pertenencia asignado está en función de la distancia de los K vecinos más cercanos y el grado de pertenencia de esos vecinos a las posibles clases. El algoritmo 2.3 describe la versión difusa.

Algoritmo 2.2 K vecinos más cercanos [31]

Sea $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ un conjunto de ejemplos etiquetados.
Entrada: y, instancia a ser clasificada
Salida: Una etiqueta de clase para la instancia.
Establecer K, tal que $1 \leq K \leq m$.
Para $i = 1$ **hasta** m **hacer**
  Calcular la distancia de y a x_i
  Si $i \leq K$ **entonces**
    Incluir x_i al conjunto de los K vecinos más cercanos.
  de lo contrario Si x_i es más cercano a y que los vecinos almacenados **entonces**
    Borrar el más lejano en el conjunto de los k más cercanos.
    Incluir x_i al conjunto de los K vecinos más cercanos.
  Fin Si
Fin Para
Determinar la clase representada por la mayoría en el conjunto de los K vecinos más cercanos.
Si existen un empate **entonces**
  Calcular la suma de la distancia de los vecinos de las clases que empaten.
  Si no existe empate **entonces**
    Clasificar y con la clase que tenga la menor suma
  de lo contrario
    Clasificar y en alguna de las clases con la menor suma
  Fin Si
de lo contrario
  Clasificar y en la clase representada por la mayoría
Fin Si

Algoritmo 2.3 K vecinos más cercanos difusos [31]

Sea $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ un conjunto de ejemplos etiquetados de un problema de l clases.
Sea $u_i(x)$ la pertenencia del vector x y u_{ij} la pertenencia en la i-ésima clase del j-ésimo ejemplo etiquetado (véase ecuación (2.24)).

Entrada: y, instancia a ser clasificada
Salida: Vector con los grados de pertenencia a las diferentes clases.

```
Establecer K, tal que 1 ≤ K ≤ m.
Para i = 1 hasta m hacer
  Calcular la distancia de y a x_i
  Si i ≤ K entonces
    Incluir x_i al conjunto de los K vecinos más cercanos.
  de lo contrario Si x_i es más cercano a y que los vecinos almacenados entonces
    Borrar el más lejano en el conjunto de los k más cercanos.
    Incluir x_i al conjunto de los K vecinos más cercanos.
  Fin Si
Fin Para
Para i = 1 hasta l hacer
  Calcula u_i(x) usando 2.24
Fin Para
```

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij} \left(1/\left|x - x_j\right|^{2/(p-1)}\right)}{\sum_{j=1}^{k} \left(1/\left|x - x_j\right|^{2/(p-1)}\right)} \tag{2.24}$$

De la ecuación (2.24) se observa que el valor de pertenencia de x está influenciado por el inverso de la distancia de sus vecinos más cercanos y sus valores de pertenencia a las clases. La distancia permite pesar la asignación del valor de pertenencia, es decir, el valor de pertenencia es mayor para los ejemplos más cercanos. La variable p permite controlar el peso que tendrá la distancia (de acuerdo con Keller [31], valores de p diferentes de dos no varían significativamente la tasa de error y por tanto, en esta tesis, se usa un valor de dos) . La asignación del valor de pertenencia de los ejemplos a las etiquetas de clases puede ser realizado de diferentes maneras. Primero, se puede asignar la pertenencia total a su etiqueta

de clase conocida. Otra alternativa es asignar el grado de pertenencia con base en la distancia de la media geométrica a cada clase.

De los algoritmos se observa que se debe calcular la distancia del ejemplo a ser clasificado con la de los ejemplos del conjunto de entrenamiento, existen diversas medidas de distancia, entre éstas, y consideradas en la presente tesis, se encuentran:

- Euclideana,

$$distancia = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.25}$$

- Manhatan

$$distancia = \sum_{i=1}^{n} |x_i - y_i| \tag{2.26}$$

- Chord

$$distancia = \sum_{i=1}^{n} |\sqrt{x_i} - \sqrt{y_i}| \tag{2.27}$$

- Correlación

$$distancia = \frac{1}{1 + correlación} \tag{2.28}$$

donde x es la instancia a ser clasificada, y la instancia en el conjunto de entrenamiento y n es el número de atributos.

2.4.3. Árbol de decisión difuso

Los árboles de decisión difusos son una generalización de los árboles de decisión [28, 42, 44]. En esencia, un árbol de decisión es un grafo acíclico dirigido cuyos nodos están representados por los atributos (variables de entrada) y los enlaces están asociados con valores discretos de los atributos asociados con los

correspondientes nodos. El nodo hoja, o nodo terminal, corresponde a los valores de salida. En los árboles de decisión sólo es tomado un camino del nodo raíz hasta alcanzar un nodo hoja (véase figura 2.2a) [41].

El árbol de decisión difuso incorpora la noción de lógica difusa que le permite tratar con la incertidumbre e imprecisión, así como la expresión en lenguaje natural [52]. El árbol de decisión difuso, a diferencia del árbol de decisión, permite tomar más de un camino y alcanzar más de un nodo hoja (véase figura 2.2b) [41].

El árbol de decisión difuso permite una representación de forma más entendible usando variables lingüísticas. El conocimiento puede ser representado como un conjunto de reglas "*si* - *entonces*". La construcción del árbol comienza con la selección del atributo que genere la menor ambigüedad posible, el cual se convierte en el nodo raíz, y crece en forma descendente.

La medida de ambigüedad $E_g(Y)$ es definida por:

$$E_g(Y) = g(\pi) = \sum_{i=1}^{n} \left(\pi_i^* - \pi_{i+1}^*\right) \log i \tag{2.29}$$

donde $\pi^* = \{\pi_1^*, \pi_2^*, \ldots, \pi_n^*\}$ es la distribución de posibilidad[3] normalizada, tal que $\pi_i^* \geq \pi_{i+1}^*$ y $\pi_{n+1}^* = 0$, y π_i^* es el grado de pertenencia normalizado en que el atributo es subconjunto de la clase.

La ambigüedad de un atributo está expresada por:

$$E_g\left(A\right) = \frac{1}{m} \sum_{i=1}^{m} E_g\left(A\left(u_i\right)\right) \tag{2.30}$$

donde $E\left(A\left(u_i\right)\right) = g\left(\mu_{T_S}\left(u_i\right) / \operatorname{máx}_{1 \leq j \leq s}\left(\mu_{T_j}\left(u_i\right)\right)\right)$. $T_1, \ldots, T_s$ son los términos lingüísticos de un atributo con m objetos y μ_{T_j} es el grado de pertenencia del atri-

[3]La distribución de posibilidad se define como el grado de facilidad con el cual un evento puede ocurrir

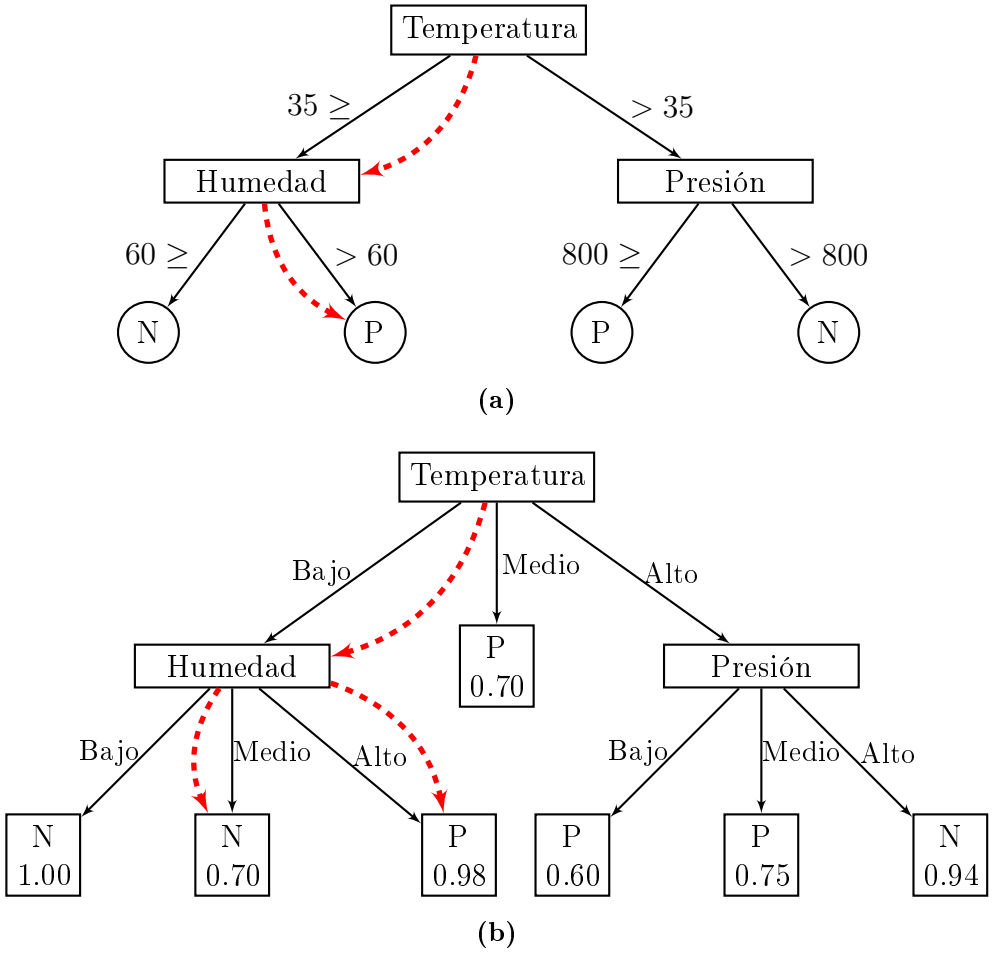


Figura 2.2: Ejemplo de árbol de decisión, donde N y P representan las clases del problema. (a) representa un árbol de decisión clásico, donde el árbol toma un sólo camino. (b) muestra un árbol de decisión difuso tomando más de un único camino. Las líneas punteadas representan el camino que el árbol toma para un caso.

buto al T_j término lingüístico. Cuando existe traslape entre los términos lingüísticos de un atributo, entonces existe ambigüedad. El subconjunto difuso $S(A, B)$ mide el grado al cual A es un subconjunto de B, y está dado por:

$$S(A, B) = \frac{\sum_{u \in U} \text{mín}\left(\mu_A(u), \mu_B(u)\right)}{\sum_{u \in U} \mu_A(u)} \tag{2.31}$$

Dada una evidencia difusa E, la posibilidad de clasificar un objeto en la clase C_j es definida como: $\pi(C_j|E) = S(E, C_j) / \text{máx}(S(E, C_j))$, tal que $S(E, C_j)$ representa el grado de verdad para la regla de clasificación ("*Si E entonces* C_j"). Con una sola evidencia la ambigüedad en la clasificación es definida por la ecuación (2.32), por el contrario, cuando existe más de una evidencia la ambigüedad en la clasificación con una partición $P\{E_1, \ldots, E_p\}$ dada la evidencia F es expresada por la ecuación (2.33).

$$G(E) = g(\pi(C|E)) \tag{2.32}$$

donde $\pi(C|E) = \{\pi(C_1|E), \ldots, \pi(C_k|E)\}$.

$$G(P|F) = \sum_{i=1}^{p} w(E_i|F)\, G(E_i \wedge F) \tag{2.33}$$

donde $G(E_i \wedge F)$ es la ambigüedad con la evidencia difusa $E_i \wedge F = \text{mín}(E_i, F)$ y $w(E_i|F) = \sum_{u \in U} \text{mín}\left(\mu_{E_i}(u), \mu_F(u)\right) / \sum_{j=1}^{p} \left(\sum_{u \in U} \text{mín}\left(\mu_{E_i}(u), \mu_F(u)\right)\right)$

Los atributos son asignados a los nodos con base en el menor grado de ambigüedad en la clasificación. Un atributo se convierte en nodo hoja si el nivel del subconjunto es mayor que un valor de verdad β preestablecido. El nivel de verdad β es un umbral que controla el crecimiento del árbol; un valor de β bajo puede llevar a un árbol pequeño – el cual puede tener menor precisión en la clasificación, ya que se produce un menor número de reglas y éstas pueden ser muy generales–, un valor de β alto produce un árbol más grande [52].

Ejemplo de construcción de un árbol difuso

En esta subsección se describe un ejemplo de construcción de un árbol difuso, considerando un ejemplo pequeño, el cual es tomado de [52]. La base de datos consta de dos atributos, $A1$ y $A2$, descritos por dos propiedades lingüísticas (bajo y alto) y dos clases, positiva (P) y negativa (N). La tabla 2.1 muestra 2 patrones, A y B, donde se asume que los atributos han sido previamente transformados a valores difusos y se conoce el grado de pertenencia a cada clase.

Tabla 2.1: Conjunto de datos para el ejemplo de construcción de un árbol difuso

$A1_B$	$A1_A$	$A2_B$	$A2_A$	P	N
0.65	0.35	0.40	0.60	0.90	0.10
0.55	0.45	0.30	0.70	0.15	0.85

El proceso de construcción del árbol comienza estableciendo el valor del umbral β, que será el valor a partir del cual una regla se puede considerar como verdadera, y el cual en este ejemplo es fijado a 0.65. Posteriormente, es necesario determinar la ambigüedad en la clasificación de cada atributo en el conjunto de datos, lo cual es realizado de la siguiente manera:

Primeramente, se calcula la ambigüedad del atributo $A1_B$; usando la ecuación (2.31), se tiene que:

$$S\left(A1_B, P\right) = \frac{\text{mín}\left(0.65, 0.90\right) + \text{mín}\left(0.55, 0.15\right)}{0.65 + 0.15} = \frac{0.65 + 0.15}{1.2} = \frac{0.80}{1.2} = 0.667$$

y

$$S\left(A1_B, N\right) = \frac{\text{mín}\left(0.65, 0.10\right) + \text{mín}\left(0.55, 0.85\right)}{0.10 + 0.55} = \frac{0.10 + 0.55}{1.2} = \frac{0.65}{1.2} = 0.542$$

La distribución de posibilidad es $\pi = \{0.667, 0.542\}$ y en su forma normalizada es $\pi^* = \{1.000, 0.813\}$.

Por la ecuación (2.29) y (2.32), se tiene que la ambigüedad es:

$$G\left(A1_B\right) = g\left(\pi\left(C|A1_B\right)\right) = \sum_{i=1}^{2}\left(\pi_i^* - \pi_{i+1}^*\right)\log i = (1.000 - 0.813)\log 1 + (0.813 - 0.000)\log 2 = 0.563$$

Siguiendo un proceso similar al anterior, se calcula la ambigüedad del atributo $A1_A$, teniendo que $G\left(A1_A\right) = 0.630$. Por tanto, la ambigüedad total del atributo $A1$ es determinada por la ecuación (2.30), como sigue:

$$E_g\left(A1\right) = \frac{1}{m}\sum_{i=1}^{m} G\left(A1\left(u_i\right)\right) = \frac{0.563 + 0.630}{2} = 0.597$$

De forma similar se calcula la ambigüedad para el atributo $A2$, por tanto $E_g\left(A2\right) = 0.577$. Comparando los dos atributos, se tiene que $A2$ tiene menos ambigüedad que $A1$, por tanto $A2$ es elegido como la raíz del árbol. Ahora se calcula el subconjunto de valores para el atributo $A2$ dada la clase, usando la ecuación (2.31).

$$S\left(A2_B, P\right) = \frac{\text{mín}\left(0.40, 0.90\right) + \text{mín}\left(0.30, 0.15\right)}{0.40 + 0.30} = \frac{0.40 + 0.15}{0.70} = \frac{0.55}{0.70} = 0.786$$

$$S\left(A2_B, N\right) = \frac{\text{mín}\left(0.40, 0.10\right) + \text{mín}\left(0.30, 0.85\right)}{0.40 + 0.30} = \frac{0.10 + 0.30}{0.70} = \frac{0.40}{0.70} = 0.571$$

$$S\left(A2_A, P\right) = \frac{\text{mín}\left(0.60, 0.90\right) + \text{mín}\left(0.70, 0.15\right)}{0.60 + 0.70} = \frac{0.60 + 0.15}{1.30} = \frac{0.75}{1.30} = 0.577$$

$$S\left(A2_A, N\right) = \frac{\text{mín}\left(0.60, 0.10\right) + \text{mín}\left(0.70, 0.85\right)}{0.60 + 0.70} = \frac{0.10 + 0.70}{1.30} = \frac{0.80}{1.30} = 0.615$$

Como el valor de $S\left(A2_B, P\right)$ es mayor que el valor de verdad ($\beta = 0.65$), por tanto $A2_B$ se convierte en un nodo hoja. Ahora, se prosigue el crecimiento del árbol usando $A2_A$, y se calcula la ambigüedad conjunta con $A1$ usando la ecuación (2.33), como se muestra a continuación:

$$G\left(A2_A \ y \ A1|C\right) = \sum_{i=1}^{2} w\left(A1_i|A2_A\right) G\left(A2_A \wedge A1\right)$$

donde:

$$\begin{aligned} w\left(A1_B|A2\right) &= \frac{\sum_{u\in U} \text{mín}\left(\mu_{A1_B}\left(u\right), \mu_{A2_A}\left(u\right)\right)}{\sum_{j=1}^{p}\left(\sum_{u\in U} \text{mín}\left(\mu_{A1_B}\left(u\right), \mu_{A2_A}\left(u\right)\right)\right)} \\ &= \frac{\text{mín}\left(0.65, 0.60\right) + \text{mín}\left(0.55, 0.70\right)}{\left(\text{mín}\left(0.65, 0.60\right) + \text{mín}\left(0.55, 0.70\right)\right) + \left(\text{mín}\left(0.35, 0.60\right) + \text{mín}\left(0.45, 0.70\right)\right)} \\ &= \frac{0.60 + 0.55}{0.60 + 0.55 + 0.35 + 0.45} = \frac{1.15}{1.95} = 0.590 \end{aligned}$$

De manera similar se calcula el peso para $A1_A$, $w\left(A1_A|A2_A\right) = 0.410$, entonces:

$$G\left(A2_A \ y \ A1|C\right) = 0.590 \times 0.601 + 0.410 \times 0.630 = 0.354 + 0.259 = 0.613$$

Al ser este último valor obtenido menor que $G\left(A2_A\right) = 0.650$, entonces añadir la partición $A1$ disminuye la ambigüedad. Usando la ecuación (2.31) se calcula los valores de los subconjuntos, los cuales son los siguientes: $S\left(A2_A \wedge A1_B, P\right) = 0.652$, $S\left(A2_A \wedge A1_B, N\right) = 0.565$, $S\left(A2_A \wedge A1_A, P\right) = 0.625$ y $S\left(A2_A \wedge A1_A, N\right) = 0.686$. Con base en los resultados de los subconjuntos se seleccionan aquellos que sean mayores al valor de verdad pre-establecido (0.65). El árbol construido se muestra en la figura 2.3 y las reglas obtenidas del árbol son las siguientes:

Si $A2$ es bajo entonces Clase es P con un valor de verdad de 0.786.

Si $A2$ es alto y $A1$ es bajo entonces Clase es P con un valor de verdad de 0.652.

Si $A2$ es alto y $A1$ es alto entonces Clase es N con un valor de verdad de 0.686.

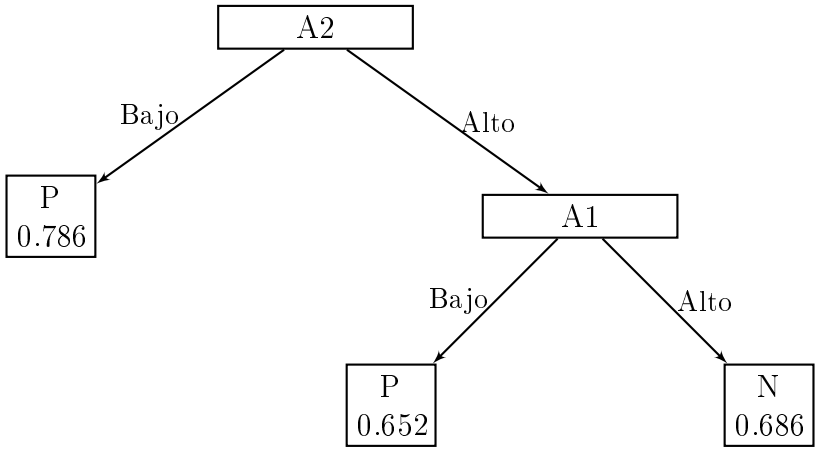


Figura 2.3: Estructura final del árbol de decisión difuso del ejemplo

2.4.4. Bosque de decisión difuso

El bosque de decisión difuso está inspirado en el algoritmo clásico del bosque aleatorio de Breiman [6]. Breiman considera el bosque como un conjunto de árboles de decisión, cuyas decisiones individuales son combinadas. La metodología del bosque de decisión difuso sigue el procedimiento mostrado en el algoritmo 2.4.

Algoritmo 2.4 Bosque de decisión difuso

Escoger aleatoriamente N ejemplos con reemplazo del conjunto de datos completo
Para cada sub-conjunto de datos construir un árbol de decisión difuso
Repetir los pasos anteriores hasta haber construido todos los árboles
Promediar las salidas de todos los árboles

El algoritmo 2.4 requiere que se especifique el número de árboles. Se crean diferentes árboles difusos usando diferentes sub-conjuntos de entrenamiento y la salida del bosque es obtenida mediante la combinación de las salidas de cada árbol.

2.5. Ensamble de clasificadores

El ensamble de clasificadores consiste en conjuntar un grupo de clasificadores tal que sus predicciones individuales sean combinadas de alguna manera (voto mayoritario, pesado, etc.) para clasificar nuevos ejemplos. En general, con el ensamble se logra una mejor predicción final y más robusta.

De acuerdo con Dietterich [10], para que un ensamble pueda funcionar se debe cumplir que los clasificadores sean precisos y diversos. Un clasificador preciso es aquel que tiene una tasa de error menor que una selección al azar de una etiqueta de clase. Por clasificadores diversos se entiende que los errores cometidos por los diferentes clasificadores sean distintos.

De acuerdo con Kuncheva [34] existen cuatro maneras de construir ensambles; éstas son:

- Nivel de combinación. Define la manera en la que se combinarán las decisiones individuales de los diferentes clasificadores.
- Nivel del clasificador. El ensamble está formado por diferentes clasificadores bases.
- Nivel de características. Diferentes subconjuntos de características son usados para entrenar los clasificadores.
- Nivel de datos. Diferentes subconjuntos de datos son usados para entrenar los clasificadores.

Aunque en estos cuatro niveles están agrupados muchos de los métodos de ensamble, existen otros temas de interés en el campo, como es la diversidad en clasificadores. A continuación se describen el nivel de combinación y la diversidad de clasificadores.

2.5.1. Nivel de combinación

El nivel de combinación especifica la manera en que las decisiones individuales de los múltiples clasificadores son combinadas. Las salidas pueden ser combinadas de diversas maneras, tales como la votación o el apilado, entre otros.

Uno de los métodos más populares para combinar las salidas es el concepto de votación [39]. La votación es la técnica de ensamble más sencilla y puede ser aplicada a cualquier tipo de clasificador que produzca una única etiqueta de clase. En el voto mayoritario, cada clasificador asigna una etiqueta de clase al ejemplo de entrada, y la clase que obtenga la mayor cantidad de votos entre los miembros del ensamble es elegida [34, 39, 49, 53].

Además de la votación, otras estrategias para combinar la salida de múltiples clasificadores son el promedio simple, promedio pesado y apilado, los cuales son usados en esta tesis y descritos a continuación.

Promedio simple

El promedio simple es una de las formas más populares de combinar las salidas de múltiples clasificadores. La combinación por promedio simple es calculada de la siguiente forma:

$$h_i^{prom}(x) = \frac{1}{M} \sum_{m=1}^{M} h_i^m(x) \tag{2.34}$$

donde M es el número de clasificadores a conjuntar en el ensamble y $h_i^m(x)$ es el valor de salida del m-ésimo clasificador para la i-ésima clase.

El promedio simple reduce la varianza de los valores de salida para las clases. Debido a su simplicidad es uno de los más usados y se ha demostrado que es un método de ensamble efectivo, particularmente en conjuntos de datos complejos

[39].

Promedio pesado

El promedio pesado también ha sido propuesto como método de ensamble. A diferencia del promedio simple, el cual considera a todos los clasificadores de igual importancia; el promedio pesado asigna a cada clasificador un peso, el cual es determinado con base en la precisión de éste. El promedio pesado es calculado como sigue:

$$h_i^{prom}(x) = \frac{1}{\sum_{i=1}^{m} w_m} \sum_{m=1}^{M} w_m h_i^m(x) \tag{2.35}$$

donde M es el número de clasificadores a conjuntar en el ensamble, w_m es el peso asignado al m-ésimo clasificador y $h_i^m(x)$ es el valor de salida del m-ésimo clasificador para la i-ésima clase.

El promedio pesado es matemáticamente similar al promedio simple, pero añadiéndole grados de libertad, lo cual, en ocasiones, puede proveer mejores resultados que el promedio simple.

Este tipo de método de combinación requiere determinar el peso de cada clasificador. Diferentes estrategias para determinar dicho peso han sido propuestas [34]. Una forma sencilla de determinar el peso al clasificador es con base en su precisión. El peso es calculado de la manera siguiente:

$$w_i = \log_{10}\left(\frac{P_i}{1 - P_i}\right) \tag{2.36}$$

donde w_i y P_i representan, respectivamente, el peso asignado al i-ésimo clasificador y la precisión del mismo.

Apilado

Otra forma de conjuntar las decisiones de los clasificadores individuales es por apilado. La idea del apilado es construir un clasificador, denominado meta-clasificador, que corrija los errores de los clasificadores individuales.

En resumen, en el apilado, se crean l clasificadores base con un conjunto de datos reservado para el entrenamiento y, posteriormente, un conjunto de datos de prueba es usado sobre los l clasificadores. Las salidas de los l clasificadores, obtenidas con el conjunto de prueba, son usadas como atributos del meta-clasificador, el cual es entrenado con las salidas de los l clasificadores, esto es las clasificaciones del conjunto de prueba.

2.5.2. Diversidad en clasificadores

La diversidad de los clasificadores es un requerimiento vital para un ensamble exitoso. Se desea que los clasificadores, miembros del ensamble, sean los más precisos posibles, y en caso de cometer un error, que estos errores sean diferentes entre los diversos miembros.

En la práctica, se han definido medidas de diversidad en clasificadores. La entropía se ha usado como medida en diversidad de clasificadores, y la cual es usada en este trabajo. La medida de diversidad por entropía para L clasificadores sobre un conjunto de datos con N muestras está definida por la expresión siguiente:

$$Ent = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{L - \lfloor L/2 \rfloor - 1} \operatorname{mín} \left\{ \sum_{i=1}^{L} y_{j,i}, L - \sum_{i=1}^{L} y_{j,i} \right\} \tag{2.37}$$

donde $y_{j,i}$ vale 1 si el i-ésimo clasificador predice correctamente el j-ésimo ejemplo, de lo contrario vale 0.

2.6. Algoritmo evolutivo

La evolución es un proceso de optimización donde el objetivo es mejorar la habilidad de un organismo (sistema) para sobrevivir en ambientes competitivos y cambiantes. La computación evolutiva se refiere a los sistemas para resolver problemas basados en cómputo que usen modelos computacionales de proceso evolutivo, tales como la selección natural, supervivencia del más apto y reproducción, como componentes fundamentales [14].

La evolución a través de la selección natural de una población de cromosomas elegidos aleatoriamente puede ser vista como una búsqueda a través del espacio de los posibles valores del cromosoma. Debido a lo anterior, los algoritmos evolutivos son un tipo de búsqueda estocástica para tratar de hallar una solución óptima a un problema en particular. En general, el proceso evolutivo tiene los siguientes componentes:

- un cromosoma que codifica las posibles soluciones al problema;
- una función que evalúa la aptitud, o fuerza, de un individuo;
- una población inicial;
- operaciones para la selección de individuos, y
- operaciones para la reproducción de los individuos.

En el algoritmo 2.5 se muestra el algoritmo evolutivo genérico con los componentes previamente mencionados.

Algoritmo 2.5 Algoritmo evolutivo genérico [14]

Sea $t = 0$ el contador de generaciones.
Crear una población inicial, $P(0)$, de n individuos.
Mientras condición de paro no se cumpla **hacer**
  Evaluar la aptitud, $f(x_i(t))$, de cada individuo $x_i(t)$
  Realizar la reproducción para crear a los hijos.
  Seleccionar la nueva población, $P(t+1)$
  hacer $t = t + 1$.
Fin Mientras

Cromosoma

Un individuo representa una solución candidata al problema de optimización. Las características del individuo son representadas por un cromosoma; dichas características se refieren a las variables de optimización. Cada variable a ser optimizada se le conoce como gen y el valor asociado a dicha variable es conocido como alelo.

Los cromosomas requieren de un tipo de representación, siendo la representación binaria el esquema clásico. Holland [26] da una justificación para la representación binaria argumentando que dicha representación, binaria, permite tener más esquemas que una representación decimal. Un esquema es una plantilla que describe a un subgrupo de cadenas que comparten ciertas similitudes en algunas posiciones a lo largo de su longitud.

Población inicial

Los algoritmos evolutivos son estrategias de búsqueda estocástica; su búsqueda la basan en una población, entonces, los algoritmos evolutivos mantienen una población de posibles soluciones candidatas.

El primer paso para resolver un problema de optimización por algoritmos

evolutivos es crear una población inicial de posibles soluciones. Generalmente, la población inicial es creada asignando un valor aleatorio, dentro del conjunto de valores permitidos, a cada gen del cromosoma. El objetivo de asignar valores aleatorios es garantizar que la población inicial es una representación uniforme de todo el espacio de búsqueda. Si las regiones no son cubiertas por la población inicial, es posible que dichas regiones no sean visitadas en el proceso de búsqueda.

El tamaño de la población inicial tiene consecuencias en términos de complejidad computacional y habilidad de exploración. Una población muy grande generará una diversidad de soluciones al problema –mejorando la capacidad de exploración–, sin embargo, requerirá más tiempo por generación para evaluar la población, incrementando la complejidad computacional. Por otro lado, una población pequeña sólo representará una pequeña parte del espacio de búsqueda –afectando la capacidad de exploración–, pero el tiempo requerido por generación es menor, mejorando la complejidad computacional.

Función de aptitud

Para poder determinar la habilidad de supervivencia de un individuo en el algoritmo evolutivo, se usa una función matemática para saber que tan bien el cromosoma representa la solución. Esta función, denominada función de aptitud, mapea la representación de un cromosoma a un valor escalar:

$$f : \Gamma \rightarrow \mathbb{R} \tag{2.38}$$

donde Γ representa el cromosoma.

La función de aptitud representa la función objetivo, la cual describe el problema de optimización. Usualmente, la función de aptitud proporciona una medida absoluta de la aptitud del cromosoma, es decir, evalúa que tan bien el cromosoma

representa la solución.

Selección

El principal objetivo de los operadores de selección es hacer resaltar las mejores soluciones. Esto es alcanzado en dos principales pasos:

- **Selección de la nueva población.** Se selecciona una nueva población de soluciones candidatas al finalizar cada generación y que servirá como la población para la nueva generación. La nueva población puede ser seleccionada únicamente de los descendientes o de ambos, padres e hijos. El operador de selección debe garantizar que los mejores individuos sobrevivan en las siguientes generaciones.

- **Reproducción.** Los hijos son creados a través de la aplicación de operaciones de cruza, mutación y elitismo. Para la cruza, los individuos con mejor aptitud deben tener más oportunidad de reproducirse para garantizar que los hijos contienen el material genético de los mejores individuos. Para la mutación, los mecanismos de selección deben enfocarse en los individuos más débiles, o menos aptos, con la finalidad de incrementar su oportunidad de supervivencia. Finalmente, con el elitismo se asegura que los mejores individuos sobrevivan en las siguientes generaciones; esto se logra copiando los mejores individuos en la generación siguiente sin sufrir el proceso de mutación.

Criterios de paro

Los operadores de evolución se aplican mientras no se cumpla una condición de paro. La condición de paro más simple es limitar el algoritmo a un número de

generaciones o hasta que un valor de aptitud haya sido alcanzado (que en nuestro caso se buscó que fuese igual a un error de cero por ciento). Alternativamente, otros criterios de paro incluyen que no exista mejora en el valor de aptitud durante un determinado número de generaciones o que no existan cambios en la población.

2.7. Métodos de evaluación

Es importante determinar el desempeño de un clasificador, ya que esto nos permite comparar entre dos modelos distintos. Generalmente, los algoritmos de clasificación son evaluados mediante la comparación de la clasificación predicha por el clasificador y la clasificación real, la cual debe ser conocida. Existen diversas maneras de evaluar el clasificador, la más común es la validación cruzada. De igual manera, la matriz de confusión resulta útil para saber cuáles clases son las que presentan mayor confusión; asimismo, algunas métricas pueden ser definidas a partir de la matriz de confusión. Finalmente, otra métrica usada es el área bajo la curva ROC, por las siglas en inglés de *Reciver Operating Characteristic*. A continuación se detallan algunas técnicas de evaluación que permiten determinar el desempeño de un clasificador, utilizadas en esta tesis.

2.7.1. Validación cruzada con k particiones

El método de validación cruzada con k particiones consiste en dividir el conjunto de datos en k subconjuntos, aproximadamente iguales, tal que los conjuntos sean mutuamente excluyentes. Las fases de entrenamiento y pruebas son realizadas k veces. En la iteración i se selecciona el subconjunto S_i como un conjunto de prueba y los subconjuntos restantes como conjunto de entrenamiento. Se entrena el clasificador usando el conjunto de entrenamiento y, con el conjunto de prueba,

se evalúa el desempeño del clasificador. Al finalizar las k iteraciones, se obtiene el promedio de todas las evaluaciones.

2.7.2. Matriz de confusión

La matriz de confusión contiene información acerca de las clasificaciones reales y las predichas por un clasificador. El desempeño del clasificador es comúnmente evaluado con los datos contenidos en la matriz. En la figura 2.4 se muestra una matriz de confusión binaria.

		Predicha	
		P	N
Real	P	VP	FN
	N	FP	VN

Figura 2.4: Matriz de confusión binaria

En la matriz de confusión, tanto los verdaderos positivos (VP) como los verdaderos negativos (VN) representan las instancias correctamente clasificadas. Por el contrario, los falsos positivos (FP) y los falsos negativos (FN) representan las instancias incorrectamente clasificadas.

Algunas métricas que pueden ser definidas en términos de una matriz binaria son:

- la **exactitud**, que es la porción de instancias correctamente clasificadas:

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.39}$$

- la **tasa de verdaderos positivos**, también llamada sensibilidad, que es la porción de ejemplos positivos que fueron clasificados correctamente:

$$TVP = \frac{VP}{VP + FN} \tag{2.40}$$

- la **tasa de verdaderos negativos**, o especificidad, que es la porción de ejemplos negativos clasificados correctamente:

$$TVN = \frac{VN}{VN + FP} \tag{2.41}$$

2.7.3. Curvas ROC

El análisis de curvas ROC (*Receiver Operating Characteristic*) permite evaluar el desempeño de los clasificadores binarios [24]. Este análisis considera que un clasificador es mejor que otro si tiene una mayor área bajo la curva ROC que lo describe.

La curva ROC es una gráfica de la tasa de falsos positivos (1-especificidad), sobre el eje **X**, contra la tasa de verdaderos positivos (sensibilidad), sobre el eje **Y**. La gráfica es construida conectando el punto $(0, 0)$ con el punto (TFP, TVP) del clasificador y, este último con el punto $(1, 1)$, formándose así la curva ROC.

El punto $(0, 1)$ representa el clasificador perfecto, es decir, clasifica correctamente todos los casos positivos y negativos, representado por la curva en color verde de la figura 2.5a. Por el contrario, los puntos $(0, 0)$ y $(1, 1)$ representan a malos clasificadores; el punto $(0, 0)$ representa un clasificador que predice que todos los casos son negativos; el punto $(1, 1)$ representa un clasificador que predice todos los casos como positivos, ambos cosas son representados por la curva naranja de la figura 2.5a. Finalmente, el punto $(1, 0)$ representa al peor clasificador que tiene todas sus clasificaciones incorrectas, representado por la curva roja de la figura 2.5a.

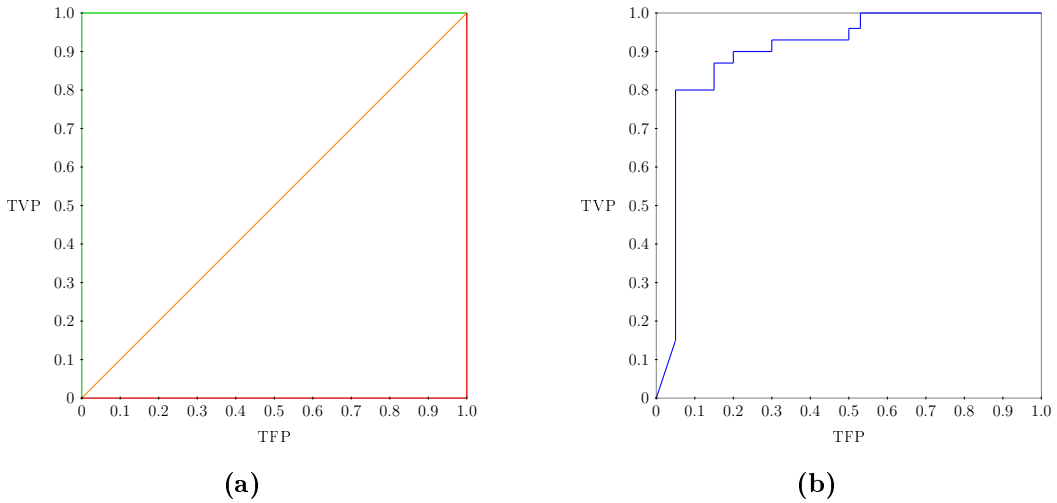


Figura 2.5: Dos ejemplos de curvas ROC para clasificadores binarios. En (a) se muestran los casos de cruva ROC para el mejor (gráfica verde), malo (gráfica naranja) y peor clasificador (gráfica roja); (b) es un ejemplo de la cruva ROC empírica para un clasificador con salida continua

Existen clasificadores cuya salida no es una etiqueta de clase sino un valor continuo; dichas salidas pueden ser consideradas una estimación de la pertenencia de un ejemplo a la clase. Si para ese tipo de clasificadores se varía el umbral en el cual un ejemplo pertenece a una clase, se producen varios pares de puntos (TFP,TVP), los cuales, al ser conectados en conjunto con los puntos $(0,0)$ y $(1,1)$, producen la curva ROC empírica para el clasificador, como se muestra en la figura 2.5b.

Para problemas con más de dos clases, Dreiseitl [11] ha propuesto determinar el área bajo la curva ROC como el promedio de las áreas bajo la curva ROC de los diferentes clasificadores binarios, siguiendo un enfoque una contra todos.

2.8. Curvas de costo

Las curvas de costo fueron introducidas por Drummond y Holte [12]. El costo esperado normalizado (entre cero y uno, tal que el mejor desempeño es cero y el peor es uno) es graficado sobre el eje y y los puntos de operación[4] sobre el eje x, normalizados entre 0 y 1. El punto de operación es la función de probabilidad de costos (PCF, por sus siglas en inglés), y es calculada como sigue:

$$PCF\,(+) = \frac{p\,(+)\,C\,(-|+)}{p\,(+)\,C\,(-|+) + p\,(-)\,C\,(+|-)} \tag{2.42}$$

donde $C\,(-|+)$ es el costo de clasificar erróneamente un ejemplo positivo como negativo, $C\,(+|-)$ es el costo de clasificar erróneamente un ejemplo negativo como positivo, $p\,(+)$ es la probabilidad de un ejemplo positivo y $p\,(-) = 1 - p\,(+)$.

Existen dos simplificaciones para el $PCF(+)$. La primera es cuando los costos

[4] El término punto de operación se refiere a la combinación especifica de costo de clasificación incorrecta y las distribuciones de clase.

de las clasificaciones erróneas son iguales; en este caso $PCF(+) = p(+)$. La segunda, cuando $p(+) = p(-)$, tal que $PCF(+) = C(-|+) / (C(-|+) + p(-) C(+|-))$. Por tanto, los costo de clasificación errónea son normalizado tal que su suma sea uno.

Existe una dualidad entre las curvas ROC y las curvas de costo. Esto implica que un punto en el espacio ROC equivale a una línea en el espacio de costos y un línea en el espacio ROC es representado por un punto en el espacio de costos. El conjunto de puntos que definen la curva ROC, son un conjunto de líneas en el espacio de costo.

En la figura 2.6 se muestra un ejemplo de la curva de costo. La línea punteada horizontal representa el peor clasificador, con todas sus clasificaciones incorrectas. El eje x representa al mejor clasificador con todas las predicciones correctas, y las líneas punteadas diagonales representan a clasificadores que todas sus predicciones son de una clase.

2.9. Resumen

El diseño de un modelo de clasificación puede involucrar diferentes etapas, tales como la selección de atributos, pre-procesamiento de los mismos y la construcción del clasificador, procesos que son abordados en el presente trabajo.

Generalmente, un clasificador asigna una etiqueta de clase a un ejemplo de entrada, el cual es descrito por sus características o atributos. Sin embargo, no siempre todos los atributos son relevantes para predecir la clase. La selección de atributos pretende determinar aquellos atributos relevantes a la clase. Dos enfoques para la selección de atributos, y usados en esta tesis, son de filtrado y de envoltura. Por lo general, los métodos de filtrado son más rápidos que los de

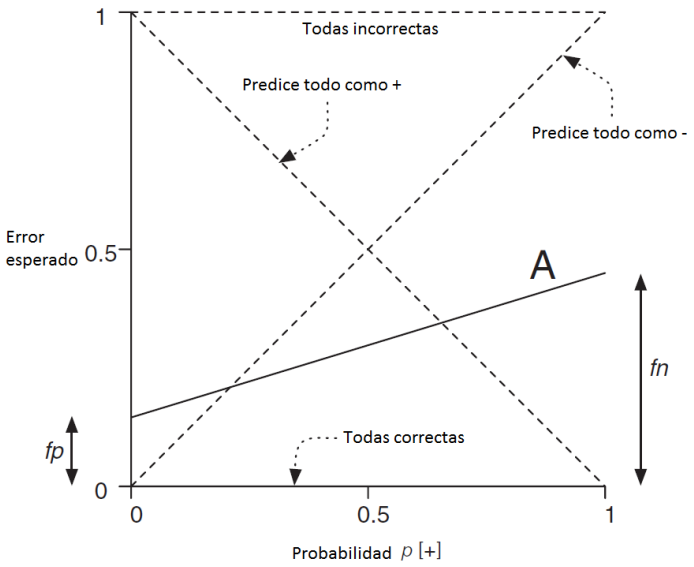


Figura 2.6: Curvas de costo para el mejor, malo y peor clasificador. Imagen tomada de [24]

envoltura, pero éstos últimos, usualmente, brindan mejores resultados.

El procesamiento de los atributos puede ser realizado con lógica difusa. La lógica difusa permite tratar con pertenencias o verdades parciales, esto implica que, a diferencia de la lógica clásica, un elemento puede pertenecer al mismo tiempo a más de un conjunto. Para procesar los atributos es necesario definir una función de pertenencia difusa, la cual se encarga de mapear el valor del atributo a su correspondiente valor difuso. Las funciones de pertenencia más comunes, mas no las únicas, son: triangular, trapezoidal, gaussiana y campana.

Los clasificadores difusos son aquellos que incluyen la idea de la lógica difusa, ya sea durante su proceso de aprendizaje o inducción. Existen diferentes clasificadores difusos, entre los cuales destacan la red neuronal relacional difusa, K-vecinos más cercanos difusos, árbol de decisión difuso y el bosque de decisión difuso. Además, es posible combinar la predicción individual de cada clasificador en

un ensamble; dichas combinaciones puede ser por un promedio simple, promedio pesado o apilado.

Conocer el desempeño de un clasificador es importante, ya que permite saber qué modelo es mejor para un conjunto de datos dado, lo cual es importante en el trabajo desarrollado para seleccionar el modelo de clasificación más adecuado al conjunto de datos. Existen diversas maneras de evaluar el clasificador, las más comunes son: validación cruzada y el área bajo la curva ROC. La validación cruzada parte el conjunto de entrenamiento en k partes, tal que $k-1$ partes son usadas para entrenar el clasificador y 1 es usada para probar el mismo; este proceso es realizado iterativamente hasta que todas las partes han sido usadas como prueba. El área bajo la curva ROC es una gráfica de la tasa de falsos positivos contra la tasa de verdaderos positivos y es aplicada a clasificadores binarios; extensiones del área bajo la curva ROC para problemas multiclase proponen promediar el área bajo la curva ROC para los diferentes clasificadores binarios, siguiendo un enfoque uno contra todos.

Capítulo 3

Trabajo relacionado

En este capítulo se describen algunos trabajos relevantes relacionados al trabajo de tesis, tales como la clasificación de clases traslapadas, así como algunos algoritmos de selección de modelo y de clasificación de leucemia aguda realizados recientemente.

3.1. Clases traslapadas

De acuerdo con Kuncheva [34] una clase traslapada se define de la siguiente manera:

> "Si en una región de decisión R_i, con clase w_i como verdadera, se encuentra un ejemplo del conjunto de etiquetas Z con clase w_j como verdadera, tal que $j \neq i$, entonces las clases w_i y w_j se llaman clases traslapadas."

Existen dos enfoques para la clasificación de problemas con clases traslapadas: el primero consiste en la unión de las clases traslapadas en una nueva clase denomi-

nada meta-clase; un segundo enfoque propone la separación de las clases mediante la extracción de características con mayor poder discriminatorio –obtener nuevas características que permitan diferenciar mejor la clase o transformación de las características existentes en otras– o en la combinación de múltiples clasificadores [35].

A continuación se describen algunos trabajos relacionados a la clasificación de problemas con clases traslapadas.

Koerich [32] propone una metodología para la clasificación de problemas con clases traslapadas de N clases. Además, la metodología propuesta asume que la información contextual del problema está disponible. Esta metodología está compuesta por cuatro pasos: el primero consiste en la obtención del conocimiento, donde un clasificador es diseñado usando las N clases y su desempeño es evaluado obteniéndose su matriz de confusión, a partir de la cual se determina cuáles son las clases que más se confunden. En el segundo paso, la información contextual en conjunto con los datos obtenidos de la matriz de confusión son usados para determinar qué clases se unen en una nueva clase denominada meta-clase. Seguidamente, en el tercer paso, el clasificador es diseñado nuevamente con las meta-clases y, finalmente, en el cuarto paso, el clasificador es validado. Los experimentos fueron realizados sobre una base de datos de caracteres manuscritos, NIST, alcanzando un 87.79 % sobre el conjunto de pruebas.

Liu [35] propone un esquema de entrenamiento parcial discriminatorio para la clasificación de clases traslapadas en el análisis de documentos. El esquema de entrenamiento parcial discriminatorio es aplicado a problemas con N clases. En este trabajo, primeramente, tratan de distinguir las clases usando información contextual y, si esto no es posible, aplicar el entrenamiento parcial. La idea de este entrenamiento consiste en crear N clasificadores binarios, un clasificador para ca-

da posible clase, donde cada clasificador es entrenado con un conjunto de datos, tal que los ejemplos de la clase que se desea identificar son considerados como positivos. Los ejemplos de las clases con las que se presente traslape no son considerados como positivos ni como negativos, y los ejemplos del resto de las clases son considerados como ejemplos negativos; de esta manera cada clasificador es entrenado con los ejemplos de la clase a reconocer y los ejemplos de las clases con las que no existe traslape. La decisión final es realizada mediante la combinación de los N clasificadores binarios.

Podolak [43], por su parte, propone una arquitectura tipo jerárquica para la clasificación de clases traslapadas. Esta arquitectura se basa en una jerarquía tipo árbol, donde cada nodo del árbol corresponde a una red neuronal, la cual es entrenada como un clasificador débil[1]. En la raíz del árbol se tiene una red neuronal para las posibles clases del problema. Una vez que la red neuronal es entrenada, los resultados de sus clasificaciones incorrectas son analizados a través de una matriz de confusión y se determina cuáles son las clases que más se confunden. Se asume que las clases que presentan la mayor confusión tienen información similar y, por tanto, están traslapadas. Estas clases constituyen un subproblema para el cual una red neuronal es entrenada, creándose un subnivel en el árbol. De esa forma, cada nodo del árbol es una red neuronal entrenada para las clases que más se confunden. Durante la clasificación, un ejemplo puede tomar más de un camino del árbol; la decisión final es tomada mediante la conjunción de las probabilidades a *posteriori*.

[1]Un clasificador débil es aquel cuya precisión es ligeramente mejor que una selección al azar [43].

Szmidt y Kukier[50] abarcan el problema de clasificación con clases desbalanceadas[2] y traslapadas. Se realiza un tratamiento de las características usando los conjuntos difusos intuicionistas de Atanassov [3], asignándole un grado de pertenencia, un grado de no pertenencia y un margen de indecisión a cada clase. Los conjuntos difusos son determinados por una distribución de frecuencia relativa de los datos de las clases. La decisión final es realizada por la clase a la que tenga una mayor pertenencia; se obtiene un porcentaje promedio de 98.8 % sobre la base de datos de vinos del UCI [51].

3.2. Selección de modelo

La selección de modelo es la tarea de escoger el mejor modelo que describa a un conjunto de datos; esto implica la selección de parámetros con base en un conjunto de entrenamiento [25, 18]. De acuerdo con Guyon [21] la selección de modelo involucra numerosos problemas, entre los que destacan: selección del mejor subconjunto de características o atributos, selección del método de pre-procesamiento, selección del mejor algoritmo de aprendizaje de máquina y la selección de los hiperparámetros. Algunos trabajos que han realizado selección de modelo son descritos enseguida.

Acevedo [1] usa optimización por colonia de hormigas para ajustar los parámetros de una máquina de vectores de soporte. La máquina de vectores de soporte emplea un kernel de función de base radial, siendo sus parámetros ajustados. Se define cada hormiga como un vector solución de los parámetros para el kernel. Además, como función de aptitud se usa el error usando validación cruzada con

[2]Las clases desbalanceadas se refiere a que una clase es representada con una gran cantidad de ejemplos mientras que la otra con una cantidad pequeña de ejemplos [29].

k particiones; el valor de k es establecido igual al número de ejemplos de entrenamiento. Este algoritmo se probó sobre algunas bases de datos para nariz electrónica, reportándose, en promedio, 80.33 % de clasificación correcta.

Gorissen *et al.* [20] usa algoritmos genéticos para la selección de un metamodelo. El sistema es aplicado a tareas de regresión. Mediante el uso de algoritmos genéticos se seleccionan tanto el modelo como los parámetros asociados a los mismos, conocidos como hiper-parámetros. Se consideran siete modelos; incluyendo máquinas de vectores de soporte, redes neuronales artificiales, funciones de base radial, entre otros métodos de *kernel*. Además, se definen operadores para creación, selección, cruza y mutación. También, se consideró el uso de múltiples subpoblaciones y el ensamble de diferentes modelos mediante un promedio simple. Finalmente, la función de aptitud se define como la raíz cuadrada del error relativo.

Escalante *et al.* [16] utilizan optimización por enjambre de partículas para la selección de un modelo de clasificación (PSMS, por sus siglas en inglés), llamado selección de modelo completo. Dado un conjunto de algoritmos de selección de atributos, métodos de pre-procesamiento y algoritmos de aprendizaje de máquina, la selección completa se define como la tarea de escoger la mejor combinación de éstos que minimice el error de clasificación. El proceso de búsqueda es realizado por enjambre de partículas, donde cada partícula representa un modelo completo y es codificado en un vector. La función de aptitud es la tasa de error balanceado, es decir, el error medio en las clases positivas y negativas. Una extensión a este trabajo es propuesta por Escalante *et al.* [17], quienes siguen la idea de usar optimización por enjambre de partículas para la selección de un modelo completo. Sin embargo, a diferencia de [16], un ensamble de clasificadores es considerado. Este método usa PSMS para buscar modelos de clasificación adecuados a un con-

junto de datos que luego son ensamblados. Se consideran tres maneras distintas de realizar el ensamble: en el primer caso, las mejores soluciones obtenidas cada k iteraciones son consideradas en el ensamble; en el segundo, los modelos con mejor aptitud en cada iteración son considerados; en el tercer caso, las mejores soluciones, una vez finalizado el proceso de búsqueda, son consideradas en el ensamble. En todos los casos, las salidas de los diferentes modelos son combinadas por promedio simple.

3.3. Clasificación de leucemia aguda

Morales [37] realiza la clasificación de leucemia aguda a nivel de tipos, esto es, se identifican entre los tipos linfoide y mieloide. Los datos de las células de leucemia aguda son obtenidos a partir de imágenes de frotis de médula ósea. Las imágenes son pre-procesadas y sus características son extraídas. Por otro lado, la clasificación es realizada usando los algoritmos de *logistic* y una red neuronal artificial multicapa, con dos capas ocultas y seis neuronas por capa. Los resultados de la clasificación son de: 89.90 % para *logistic* y 91.79 % para la red neuronal artificial.

Jarratwisaru [30] propone un método de clasificación de leucemia aguda, diferenciando entre los tipos linfoide y mieloide. Asimismo, los subtipos asociados a cada tipo también son considerados. Primeramente, imágenes de frotis de médula ósea son segmentadas usando un modelo de contorno y color. Posteriormente, la parte de clasificación es realizada con características extraídas del núcleo. Se reporta un 90 % de clasificación correcta.

El trabajo de Galindo [19] presenta un método de obtención de características para la clasificación de subtipos de leucemia aguda. Las características obtenidas

a partir de las imágenes son de textura, morfológicas, estadísticas y geométricas y de componentes principales. Además, en la clasificación de subtipos se consideran los subtipos L1 y L2 para la tipo linfoide, de la cual se obtiene un 85 % de clasificación correcta; M2, M3 y M5 para la mieloide, se reporta un 91 % de clasificación correcta.

Reta *et al.* [46] proponen un algoritmo de segmentación de células con leucemia a partir de información contextual en imágenes. El núcleo y citoplasma, así como la célula completa, son segmentados. De éstos se extraen características morfológicas, estadísticas y de textura; asimismo, características de componente principales también son extraídas. En la clasificación, se reportan resultados de las diferentes clasificaciones binarias para tipos linfoide y mieloide y los subtipos L1, L2, M2, M3 y M5. Por su parte, Reta [45] propone un algoritmo de diagnóstico, el cual combina la información de la clasificación de tipos y subtipos para determinar el tipo de leucemia. Posteriormente, esa información es usada para la determinación del subtipo. Con el algoritmo de diagnóstico se reporta un 95 % de clasificación en tipos y un 90 %, en promedio, en la clasificación de subtipos.

Nasir *et al.* [38] desarrollan la clasificación de leucemia, diferenciando entre células normales y anormales. La base de datos que ellos usan cuenta con 800 muestras, 300 de las cuales pertenecen a células normales y 500 a anormales. Primeramente, proponen la extracción de características de forma, tamaño y color. Los autores usan un perceptrón multicapa para la clasificación de las muestras. Reportan un 94.51 % de clasificación correcta con las características usadas.

Adjouadi *et al.* [2] proponen un método de clasificación basado en redes neuronales para la clasificación de leucemia aguda. Los datos son obtenidos a partir de parámetros extraídos con un citómetro[3]. En la clasificación, una red neuronal

[3]El citómetro es un analizador de las características de la células.

artificial es usada para diferenciar entre casos normales y con leucemia aguda; de igual manera, en el caso de leucemia aguda, se usa una red neuronal diferenciar entre los tipos linfoide y mieloide. La red neuronal diseñada para la clasificación consta de dos capas ocultas con 13 neuronas en la primera capa oculta y 6 en la segunda. La red es entrenada con una conjunto de 220 ejemplos de las cuales 60 corresponden a ejemplos de células con leucemia y 160 a células normales. Con este método se alcanza un resultado del 93.33 % de precisión en la clasificación.

3.4. Resumen

En la tabla 3.1 se describen los trabajos relacionados presentados en este capítulo. En esta tabla puede observarse que la mayoría de los trabajos no se han enfocado a todas las tareas planteadas en esta investigación y que, por tanto, se aborda en este trabajo.

3.5. Conclusiones

De la sección 3.1 se observa que la mayoría de los trabajos asumen que se cuenta con información contextual para distinguir entre las diferentes clases; no obstante, la información contextual no siempre está disponible. Por otro lado, estos trabajos asumen que el traslape sólo existe entre algunas clases y no consideran el caso cuando todas las clases están traslapadas, lo cual dificulta aún más la tarea de clasificación.

Por otra parte, algunos trabajos que se han enfocado a la selección de modelo, descritos en la sección 3.2, han realizado, únicamente, la selección de los parámetros para un modelo en particular. Por otro lado, otros trabajos intentan

Tabla 3.1: Tabla comparativa del trabajo relacionado

Ref	Clases traslapadas	Lógica difusa	Selección de modelo	Clasificación de leucemia	Observaciones
[32]	X				Se asume que la información contextual esta disponible
[35]	X				Se asume que el traslape no es entre todas las clases
[43]	X				El método sólo ha sido definido para un modelo de clasificación
[50]	X	X			Únicamente ha sido probado con bases de datos de bajo traslape
[1]			X		Se considera un modelo
[20]			X		Ha sido usado en tareas de regresión
[15]			X	X	El promedio simple es el único método de ensamble considerado
[37]				X	Sólo se clasifican a nivel de tipos de leucemia
[30]				X	Se enfocan en la tarea de segmentación y su base de datos cuenta con pocas muestras
[19]				X	Se enfoca en la tarea de extracción de características
[45]				X	Se enfoca en la tarea de segmentación
[2]	X			X	Los datos son obtenidos a partir de un citómetro
Propuesta	X	X	X	X	

seleccionar el modelo con sus parámetros relacionados y, en algunos casos, usando otros métodos, tales como selección de atributos o de pre-procesamiento. Aunque algunos trabajos han considerado la opción de realizar ensambles, éstos son realizados de una sola manera, por promedio simple. Los trabajos que consideran el ensamble aún no consideran la búsqueda de un método de ensamble adecuado.

Finalmente, con respecto a los trabajos que realizan clasificación de leucemia aguda, en algunos casos ésta es realizada a nivel de tipos: linfoide y mieloide. En otros casos, la clasificación es realizada a nivel de subtipos, mas no se explora la alternativa de seleccionar un modelo de clasificación adecuado para cada caso. Asimismo, el traslape no es considerado. En el trabajo que se propone aquí, se realiza la selección del modelo de clasificación, describiendo en qué grado pertenece una célula de leucemia aguda a cada uno de los tipos y subtipos, mediante el uso de lógica difusa.

Capítulo 4

Método propuesto

En este capítulo se describe el trabajo desarrollado para abordar el problema planteado en la presente tesis. Se incluye una descripción detallada de la arquitectura del método, así como el algoritmo genético usado para la selección del modelo. Asimismo, se describe la jerarquía propuesta para la clasificación de leucemia aguda.

4.1. Arquitectura general

Este método escoge, a partir de un repositorio, una combinación de componentes que minimicen el error en clasificación. Estos componentes incluyen métodos de selección de atributos, clasificadores híbridos, métodos de ensamble y tipo de difusión de los datos. Esto último permite reducir el grado de traslape en las clases. Este proceso es realizado a través de un algoritmo genético. En la figura 4.1 se muestra la arquitectura de la propuesta.

El diseño del algoritmo genético y de cada módulo es explicado en detalle a continuación.

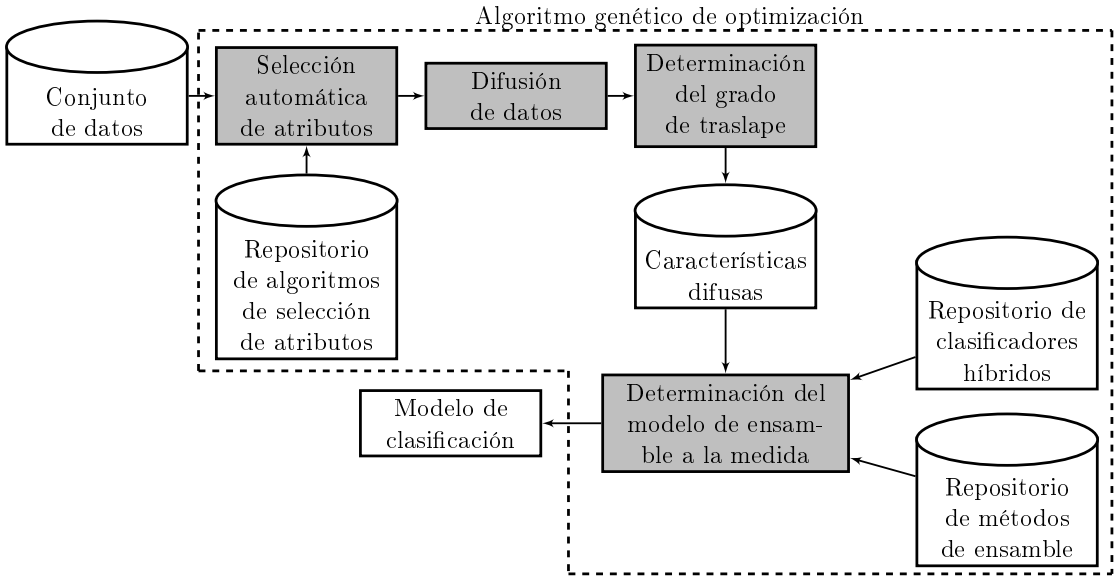


Figura 4.1: Arquitectura del método propuesto

4.1.1. Codificación del cromosoma

El cromosoma representa una posible solución al problema, que se desea optimizar a través de un algoritmo genético, y es codificado en una cadena binaria. El método propuesto construye un modelo que minimiza el error de validación cruzada en clasificación, sobre un conjunto de datos. La figura 4.1 muestra un diagrama general del método. El modelo generado está compuesto por una etapa de selección de atributos, seguido de un pre-procesamiento difuso de los datos y la construcción de un ensamble de clasificadores. En este proceso se determinan los algoritmos de clasificación difusos, así como sus parámetros relacionados y la manera en la que éstos serán combinados. A continuación se explica a detalle la codificación del cromosoma en cada módulo del sistema propuesto.

Selección automática de atributos

Al conjunto de datos de entrada se le aplica una selección de atributos, esto con el fin de reducir la dimensión del conjunto de datos y mejorar la precisión en la clasificación, al trabajar con los atributos relevantes al problema.

Los atributos relevantes son obtenidos a partir de la aplicación de un método de filtrado y de envoltura. En el enfoque de filtrado, un método de selección de atributos disponible en el "repositorio de algoritmos de selección de atributos" es seleccionado y aplicado sobre el conjunto de datos; posteriormente, el método de envoltura es aplicado para refinar la tarea de selección de atributos. De esta forma, primeramente se reduce la dimensión del conjunto de datos con el enfoque de filtrado y, por consecuencia, se reduce la complejidad en el proceso de búsqueda del método de envoltura.

En el enfoque de filtrado, los métodos de selección de atributos disponibles en el repositorio, descritos en la sección 2.2, son escogidos por el algoritmo genético; se utilizan dos bits con la codificación, mostrada en la tabla 4.1.

Tabla 4.1: Codificación de bits para determinar el método de selección de atributos

Codificación	Método de selección
$[0, 0]$	ganancia de información
$[0, 1]$	χ^2
$[1, 0]$	*reliefF*
$[1, 1]$	basado en correlación

Los métodos de selección disponibles requieren de la definición de parámetros. Por ejemplo, ganancia de información, χ^2 y *reliefF* son métodos que asignan a cada atributo una calificación con base en las propiedades de éstos, Esto hace

necesario especificar con cuantos atributos se trabajará. Por otro lado, el método basado en correlación requiere de una estrategia de búsqueda, que en nuestro caso es abordada con un algoritmo genético. Con la finalidad de determinar los parámetros para cada método de selección de atributos, cuatro bits en la cadena del cromosoma son usados. La tabla 4.2 muestra la codificación de los bits para los métodos de *reliefF* y el basado en correlación. Para el método de *reliefF* se especifica el número de vecinos cercanos y el porcentaje del conjunto de datos a usar, se seleccionan los atributos cuyo peso es mayor a cero; en el método de correlación se especifican los parámetros del algoritmo genético para la búsqueda. La tabla 4.3 muestra la codificación para ganancia de información y χ^2; en la tabla se muestra los posibles valores de la cadena binaria codifican valores porcentuales, donde cada valor porcentual representa el porcentaje de atributos mejor calificados, ya sea por ganancia de información o χ^2, que son escogidos del conjunto de atributos originales.

Tabla 4.2: Codificación de bits de parámetros para *relief* y correlación

		Bits del gen (Alelos)			
	Parámetro	00	01	10	11
ReliefF	No. muestras	80 %	85 %	90 %	95 %
	Vecinos	1	3	5	7
Correlación	Generaciones	20	25	50	100
	Población	5	10	15	20

Posteriormente, con los atributos seleccionados por filtrado se aplica envoltura. El enfoque de envoltura (sección 2.2.2) usa un clasificador para evaluar el subconjunto de atributos. Para asegurar que el subconjunto de atributos sea adecuado

Tabla 4.3: Codificación de bits para determinar método de selección de atributos

Codificación	Porcentaje de atributos seleccionados	Codificación	Porcentajes de atributos seleccionados
$[0,0,0,0]$	50 %	$[1,0,0,0]$	77 %
$[0,0,0,1]$	53 %	$[1,0,0,1]$	80 %
$[0,0,1,0]$	57 %	$[1,0,1,0]$	83 %
$[0,0,1,1]$	60 %	$[1,0,1,1]$	87 %
$[0,1,0,0]$	63 %	$[1,1,0,0]$	90 %
$[0,1,0,1]$	67 %	$[1,1,0,1]$	93 %
$[0,1,1,0]$	70 %	$[1,1,1,0]$	97 %
$[0,1,1,1]$	73 %	$[1,1,1,1]$	100 %

al modelo, se usa el clasificador determinado por el cromosoma del genético, explicado en la página 70.

Difusión de datos

El objetivo de aplicar una difusión a los datos es dar a los mismos una descripción lingüística, mediante la transformación de los valores de los datos de entrada a valores difusos. Esta transformación es realizada con la ayuda de una función de pertenencia difusa.

En la codificación del cromosoma propuesto se asigna un gen de dos bits para determinar el tipo de función de pertenencia (véase sección 2.3 para más detalle de las funciones de pertenencia) de acuerdo con la codificación mostrada en la tabla 4.4. Por otro lado, el grado de detalle en la descripción lingüística está

determinado por el número de propiedades lingüísticas. Se ha considerado el uso de tres propiedades lingüísticas (bajo, medio y alto), de cinco (muy bajo, bajo, medio, alto y muy alto) y de siete (muy bajo, bajo, más o menos bajo, medio, más o menos alto, alto y muy alto). De forma análoga a la determinación del tipo de función de pertenencia, un gen de dos bits es concatenado al cromosoma para determinar el número de propiedades lingüísticas usadas para describir el conjunto de datos; la codificación de los bits es resumida en la tabla 4.5.

Tabla 4.4: Codificación de bits para determinar el tipo de función de pertenencia

Codificación	Función de pertenencia
$[0, 0]$	Triangular
$[0, 1]$	Trapezoidal
$[1, 0]$	Gaussiana
$[1, 1]$	Campana

Tabla 4.5: Codificación de bits para determinar el número de propiedades lingüísticas

Codificación	Número de propiedades lingüísticas
$[0, 0]$	3
$[0, 1]$	5
$[1, 0]$	7

Determinación del grado de traslape

Las funciones de pertenencia son representadas por cuatro parámetros (a, b, c, d) y, en este paso, esos parámetros son ajustados. Este módulo se encarga de deter-

minar en qué grado se traslapan los conjuntos difusos tal que el grado de traslape del conjunto de datos de valores difusos sea mínimo.

Los parámetros de las funciones son ajustados y se evalúa su utilidad de la siguiente manera:

$$traslape = \frac{\sum_{i=1}^{n} \delta\left(x_i, N_1\left(x_i\right)\right)}{\sum_{i=1}^{n} \delta\left(x_i, N_2\left(x_i\right)\right)} \tag{4.1}$$

donde $N_1(x_i)$ y $N_2(x_i)$ son, respectivamente, los vecinos más cercanos intra-clase e inter-clase de x_i; δ es una función de distancia que en nuestro caso es la Euclideana.

El ajuste de los parámetros requiere de una estrategia de búsqueda, la cual es realizada a través de un algoritmo genético. Se codifica un cromosoma con $16 \times n \times m$ bits, donde n es el número de atributos relevantes seleccionados y m es el número de propiedades lingüísticas; además, cada parámetro es representado por un gen de cuatro bits. Cada posible valor de los cuatro bits representa un valor de escalamiento de los parámetros. En la tabla 4.6 cada valor del gen está asociado a un número entre 0.50 y 1.50, los cuales representan el valor de escalamiento (VE) para incrementar o decrementar apropiadamente los parámetros de las funciones de pertenencia.

Tabla 4.6: Codificación binaria para ajustar los parámetros de las funciones de pertenencia

Gen	VE	Gen	VE	Gen	VE	Gen	VE
$[0,0,0,0]$	0.50	$[0,1,0,0]$	0.75	$[1,0,0,0]$	1.20	$[1,1,0,0]$	1.40
$[0,0,0,1]$	0.55	$[0,1,0,1]$	0.80	$[1,0,0,1]$	1.25	$[1,1,0,1]$	1.45
$[0,0,1,0]$	0.60	$[0,1,1,0]$	0.90	$[1,0,1,0]$	1.30	$[1,1,1,0]$	1.50
$[0,0,1,1]$	0.70	$[0,1,1,1]$	1.10	$[1,0,1,1]$	1.35	$[1,1,1,1]$	1.00

La función de aptitud para este algoritmo genético es la expresión mostrada en la ecuación (4.1).

Determinación del modelo de ensamble a la medida

El objetivo es construir un modelo de clasificación basado en un ensamble de clasificadores adecuado al conjunto de datos de entrada. Con este método proponemos determinar los parámetros de los modelos individuales que reduzcan el error en validación cruzada y saber cuáles de estos modelos se habrán de conjuntar en un meta-modelo, es decir, en un ensamble.

La configuración del clasificador híbrido, así como sus parámetros está configurada en un cromosoma. Los clasificadores a usar son: la red neuronal relacional difusa (FRNN, por sus siglas en inglés), vecinos más cercanos difusos (FKNN, por sus siglas en inglés), árbol de decisión difuso (FDT, por sus siglas en inglés) y bosque de decisión difuso (FDF, por sus siglas en inglés); descritos en la sección 2.4. La motivación principal de escoger estos algoritmos de clasificación es que éstos permiten obtener una salida difusa y así poder conocer el grado con el que una instancia pertenece a la clase. Es importante hacer notar que cada uno de estos clasificadores requiere parámetros propios para su configuración, los cuales son distintos en cada caso. Por ejemplo, el FDT requiere que se especifique el valor de verdad como parámetro, mientras que el FDF, además del valor de verdad requiere que se especifique el tamaño del bosque o número de árboles. Con la intención de tener una mejor representación del cromosoma para cada posible clasificador se propone trabajar con un cromosoma de longitud variable, donde el tamaño del cromosoma está en función de la cantidad de parámetros para cada clasificador, así como de la cantidad de bits requeridos por cada parámetro. De esta forma, la FRNN tiene un total de 14 bits, FKNN es codificado con 5 bits,

FDF con 6 bits y FDT con 3 bits.

Debido a que la longitud en bits de algunos individuos es diferente, los individuos cuya longitud en bits sean iguales son agrupados en una sola población. Existen entonces diferentes poblaciones en el algoritmo genético, tal que cada población representa un modelo individual caracterizado por un tipo de algoritmo de clasificación. Todas las poblaciones son agrupadas como sub-poblaciones dentro del algoritmo genético. En las tablas 4.7 y 4.8 se muestra, respectivamente, la codificación con dos y tres bits para cada parámetro de los clasificadores híbridos. El algoritmo genético es el encargado de determinar los parámetros adecuados a cada tipo de clasificador.

Tabla 4.7: Codificación de dos bits de los parámetros de los clasificadores

	Codificación de bits			
Parámetro	$[0,0]$	$[0,1]$	$[1,0]$	$[1,1]$
FRNN				
Producto relacional	Max-min	Cuadrado	Sub-triángulo	Super-triángulo
FKNN				
Medida de similitud	*Manhatan*	Euclidiana	Chord	Correlación

Con el objetivo de determinar cuáles son los modelos individuales que presentan mayor diversidad y por lo tanto son los mejores candidatos a ser conjuntados en un ensamble, un segundo algoritmo genético es usado. El segundo algoritmo genético es ejecutado al finalizar cada generación del primero y usa los modelos individuales obtenidos por éste en esa generación. No obstante, no todos los modelos individuales son considerados por el segundo algoritmo genético, sino sólo lo k mejores modelos, de acuerdo con su valor de aptitud. El cromosoma del segundo algoritmo genético tiene una longitud de k bits, un valor de 1 en el bit i

Tabla 4.8: Codificación de tres bits de los parámetros de los clasificadores

	Codificación de bits							
Parámetro	$[0,0,0]$	$[0,0,1]$	$[0,1,0]$	$[0,1,1]$	$[1,0,0]$	$[1,0,1]$	$[1,1,0]$	$[1,1,1]$
FDT								
Valor de verdad	1.00	0.95	0.90	0.85	0.80	0.78	0.75	0.70
FDF								
Valor de verdad	1.00	0.95	0.90	0.85	0.80	0.78	0.75	0.70
Árboles	3	5	7	8	9	10	12	15
FRNN								
Épocas	2	5	10	15	20	25	30	50
Tasa de aprendizaje	0.10	0.15	0.20	0.25	0.30	0.33	0.40	0.50
Bias	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50
Pesos	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
FKNN								
Vecinos	1	2	3	4	5	7	8	10

representa la consideración del i-ésimo modelo en el ensamble; por el contrario, un valor de 0 significa que dicho modelo no será considerado. Al terminar el segundo algoritmo genético se obtiene un conjunto de modelos individuales diversos. Éstos son combinados por promedio simple, promedio pesado y apilado y un conjunto de pruebas es usado para medir el desempeño del ensamble. Al final se escoge el ensamble con el menor error sobre el conjunto de pruebas.

Es importante hacer notar que una vez seleccionado un ensamble por el segundo algoritmo genético, el primer algoritmo genético avanza a la siguiente generación. Esto última provoca que los modelos individuales sean evolucionados, causando que el desempeño del ensamble sea afectado, pero el segundo algoritmo genético no afecta al primero. Por esta razón, al finalizar cada generación del primer algoritmo genético, un segundo algoritmo genético es ejecutado para determinar los modelos a conjuntarse en el ensamble.

4.1.2. Operaciones genéticas

Los cromosomas que representan la solución al problema evolucionan en cada generación, con el propósito de mejorar las soluciones. El proceso de evolución asemeja la teoría de la evolución de Darwin, y es realizado por medio de un conjunto de operaciones genéticas. La sección 2.6 explica a detalle este proceso.

Las operaciones genéticas usadas en el método propuesto son la selección, cruza, mutación y elitismo, las cuales son explicadas en detalle a continuación.

Selección

La selección se realiza por torneo y pretende escoger dos individuos de la población para su reproducción. En la selección por torneo se escogen aleatoriamente n individuos, donde n es el tamaño del torneo, los cuales son sometidos a un torneo. Para determinar los individuos que se habrán de reproducir, el operador de selección por torneo es aplicado dos veces, una vez para cada padre. El individuo con mejor valor de aptitud, en cada caso, es elegido. Una vez seleccionados los individuos se calcula la distancia de Hamming entre éstos y, si la distancia de Hamming es mayor o igual que un umbral β entonces ambos padres son seleccionados; de lo contrario, uno de los padres es conservado, mientras que el operador de selección por torneo es aplicado nuevamente para el segundo padre. Lo anterior es con la finalidad de mantener la diversidad en la población y evitar una convergencia prematura.

Reproducción

La reproducción es realizada a través de las operaciones de cruza y mutación. Además, durante el proceso de reproducción cada individuo generado es compara-

do con los existentes en la población y, si los valores binarios de dicho individuo corresponden a los valores binarios de otro individuo existente en la población, éste es eliminado y un nuevo proceso de reproducción es realizado.

En la cruza, la información genética de los padres previamente seleccionados es combinada para crear nuevos individuos hijos que contendrán la información de los padres. La cruza es realizada usando el esquema de recombinación uniforme media, en el cual los bits diferentes de cada padre son seleccionados y la mitad de éstos son intercambiados. En la figura 4.2 se muestra un ejemplo de este proceso, los bits distintos en cada padre son representados por celdas sombreadas, debajo de cada celda se indican los bits que se intercambiarán por medio de un círculo; las celdas sombreadas en los hijos representan los bits intercambiados por ambos padres.

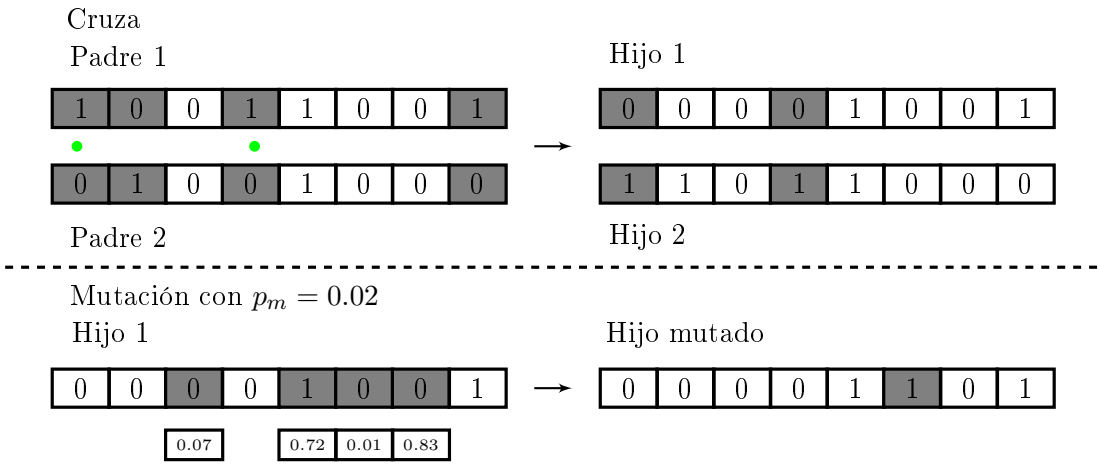


Figura 4.2: Proceso de cruza y mutación.

La cruza puede ser intrapoblación o interpoblación, dependiendo de los padres seleccionados. En la cruza interpoblación el material genético de padres que representan diferentes clasificadores es intercambiado. En este tipo de cruza, se deter-

minan los bits que son diferentes de los primeros l bits, donde l es el tamaño del individuo de menor longitud, y luego se cruzan siguiendo el esquema de recombinación uniforme. Además de los hijos creados, un tercer individuo es creado, el cual es la combinación, por promedio simple, de los modelos de clasificación representados por los padres en un ensamble. De los tres individuos creados se seleccionan los dos con mejor valor de aptitud.

Por otro lado, la mutación es uniforme y pretende añadir material genético nuevo a los individuos. La mutación sigue la idea del esquema de recombinación uniforme; con la diferencia que en este paso los bits que son idénticos son seleccionados. Posteriormente, un vector con números aleatorios entre 0 y 1 es generado. Se intercambia el valor de los bits en las posiciones cuyo valor aleatorio sea menor o igual que la probabilidad de mutación p_m, la cual es proporcionada por el usuario. La figura 4.2 muestra un ejemplo de mutación con $p_m = 0.02$; las celdas sombreadas representan los bits idénticos en cada padre y debajo de cada una de éstas se muestra su probabilidad de mutación asociada; la celda sombreada del hijo mutado muestra el valor del bit mutado.

Elitismo

Para preservar las mejores soluciones encontradas, se usa el operador de elitismo. El elitismo pretende conservar las mejores soluciones en cada generación, este operador selecciona los n mejores individuos de cada generación y los copia a la siguiente generación sin someterlos a operaciones de cruza y mutación.

4.1.3. Prevención de la extinción

El algoritmo genético propuesto para la selección de modelo difuso trabaja con diferentes sub-poblaciones, cada una de diferente longitud, la longitud de cada sub-población depende de la cantidad de parámetros requeridos por cada modelo individual. Sin embargo, durante el proceso de reproducción y evolución de los individuos es posible que una determinada sub-población sea favorecida con respecto a las demás, es decir, los individuos de esa sub-población pueden ir creciendo durante las generaciones en tanto los individuos de otras sub-poblaciones pueden ir decreciendo.

Para evitar que una sub-población se extinga durante el transcurso de las generaciones, el tamaño de cada sub-población es determinado previamente. Si el tamaño de una sub-población decrece por debajo de un umbral α, los individuos sobrevivientes de dicha sub-población son copiados, mutados y posteriormente reinsertados en la sub-población correspondiente. Además, como es deseable mantener el tamaño de la población total constante, los individuos menos aptos de las sub-poblaciones más grandes son eliminados.

Aunque la estrategia seguida para evitar la extinción de una sub-población contradice la idea del elitismo y la supervivencia del más apto, ésta es adoptada con el propósito de mantener la diversidad en los modelos individuales para su consideración en un ensamble.

4.1.4. Función de aptitud

La función de aptitud es un componente importante del algoritmo genético porque permite saber que tan buenas son las soluciones representadas por el cromosoma.

En este trabajo se ha planteado el uso de dos funciones de aptitud: la primera evalúa el desempeño de cada modelo individual; la segunda, evalúa la diversidad de los modelos individuales seleccionados para el ensamble. Es importante mencionar que aún cuando se utilizan dos funciones de aptitud, esta estrategia no está planteada como un problema de optimización multi-objetivo, sino que se aborda como dos algoritmos genéticos mono-objetivos.

El desempeño de los modelos individuales es evaluado usando la tasa de error medio de clasificación por clase (TEM), expresada por:

$$TEM = \frac{1}{l}\sum_{i=1}^{l} e_i \tag{4.2}$$

donde e_i es la tasa de error cometido por la i-ésima clase y l es el número de clases disponibles en el conjunto de datos.

El objetivo es minimizar el error medio de clasificación. La función de aptitud es definida como el porcentaje de TEM calculado sobre la validación cruzada con dos particiones[1]. Esta función de aptitud resulta útil para estimar el error de un modelo, y al considerar el error por clase permite hacer una buena estimación del error en conjuntos de datos no balanceados.

Para el segundo caso, la evaluación de la diversidad de los modelos individuales se realiza usando una medida de entropía. Esta métrica mide el grado de diversidad de L clasificadores sobre un conjunto de datos con N ejemplos y se expresa como:

$$Ent = \frac{1}{N}\sum_{j=1}^{N} \frac{1}{L - \lfloor L/2 \rfloor - 1} \operatorname{mín}\left\{\sum_{i=1}^{L} y_{j,i}, L - \sum_{i=1}^{L} y_{j,i}\right\} \tag{4.3}$$

donde $y_{j,i}$ vale 1 si el i-ésimo clasificador predice correctamente el j-ésimo ejemplo,

[1]Se usa dos particiones con la finalidad de disminuir el costo computacional en la evaluación de los modelos

de lo contrario vale 0.

En este segundo caso se desea maximizar la diversidad entre los diferentes clasificadores; una mayor diversidad de los clasificadores implica que éstos cometen errores diferentes. La función de aptitud del segundo genético se define como la media del grado de diversidad de los diferentes pares de clasificadores a conjuntarse.

4.2. Modelo jerárquico

En este trabajo de tesis también se propone la construcción de una jerarquía para la clasificación de tipos y sub-tipos de leucemia aguda. La figura 4.3 muestra la jerarquía propuesta.

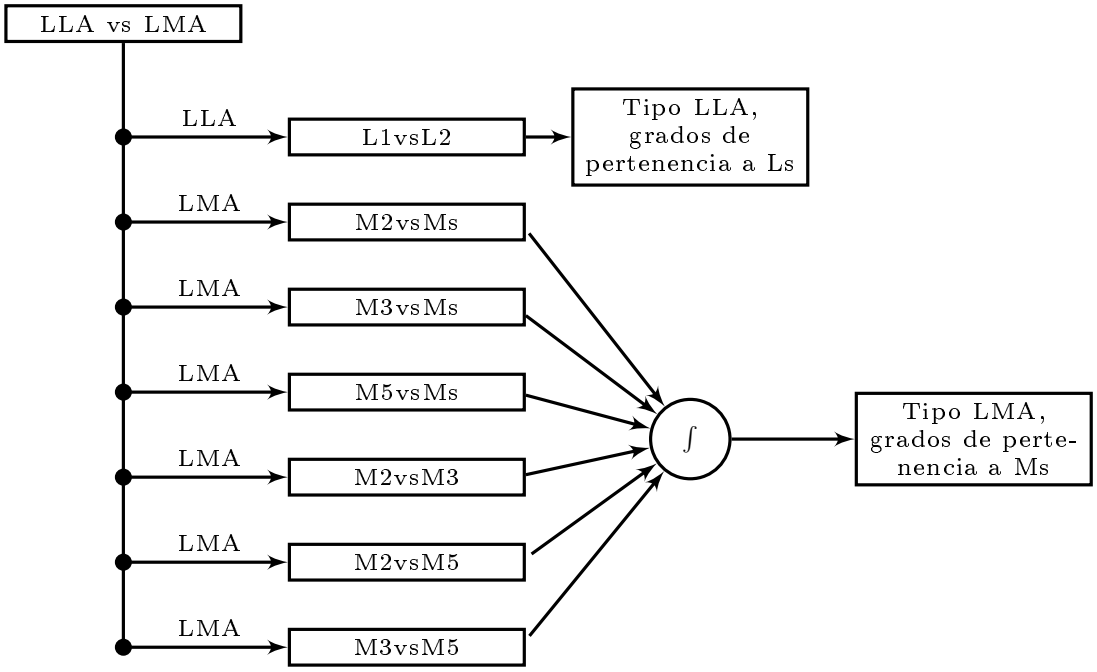


Figura 4.3: Jerarquía propuesta para la clasificación de leucemia aguda

Debido a que las clasificaciones son difusas, es posible conocer en qué grado pertenece cada ejemplo de leucemia a alguno de los dos tipos. La información de los grados de pertenencia es usada por la jerarquía para determinar el camino a seguir.

Para el caso de LLA, existe un clasificador binario para diferenciar L1 de L2; al final se muestran los grados de pertenencia a cada uno de los subtipos. En el caso de LMA, existen clasificadores binarios uno contra todos, cada uno realiza la clasificación y sus predicciones son combinadas, el resultado es el grado de pertenencia a cada uno de los subtipos de LMA.

Sea P la instancia a ser clasificada y $gp_x(P)$ el grado de pertenencia de la instancia P a la clase x, el proceso de clasificación de P es como sigue:

- Si $gp_{LLA}(P) < gp_{LMA}(P)$ entonces $tipo = LMA$

  De lo contrario, si $gp_{LLA}(P) > gp_{LMA}(P)$ entonces $tipo = LLA$

  De lo contrario, $tipo = indeterminado$

- Si $tipo = LLA$ entonces clasificar P usando L1vsL2 y regresar los grados de pertenencia de P a cada subtipo de LLA.

- Si $tipo = LMA$ entonces clasificar P usando M2vsMs, M3vsMs y M5vsMs

  Si $gp_{M2}(P) \approx gp_{M3}(P)$ entonces clasificar P usando M2vsM3 y regresar los grados de pertenencia de P a cada subtipo de LMA.

  De lo contrario, si $gp_{M2}(P) \approx gp_{M5}(P)$ entonces clasificar P usando M2vsM5 y regresar los grados de pertenencia de P a cada subtipo de LMA.

  De lo contrario, si $gp_{M3}(P) \approx gp_{M5}(P)$ entonces clasificar P usando M3vsM5 y regresar los grados de pertenencia de P a cada subtipo de LMA.

De lo contrario, regresar los grados de pertenencia de P a cada subtipo a cada subtipo de LMA.

Este proceso es mostrado en la figura 4.4

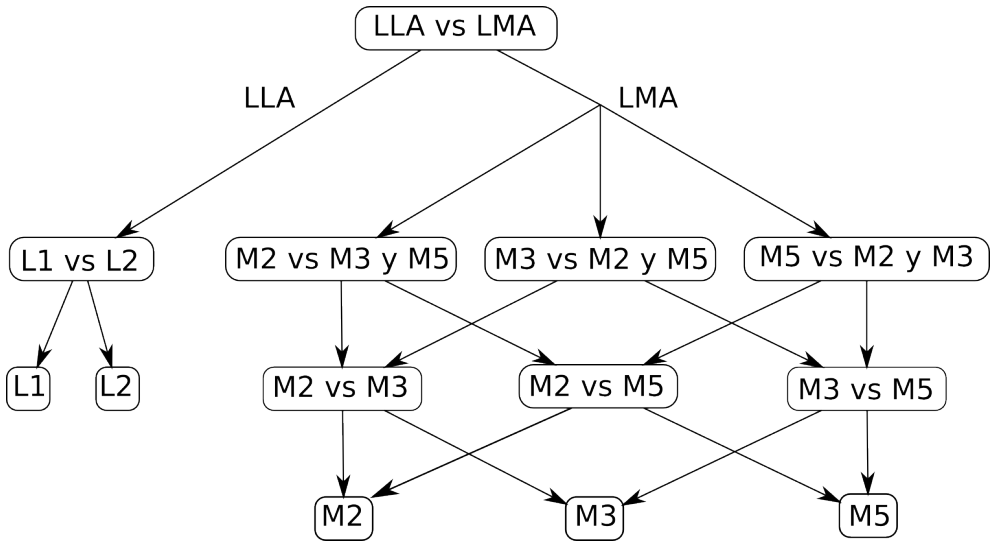


Figura 4.4: Diagrama de la jerarquía para el ejemplo de clasificación de leucemia aguda

4.3. Discusión

En este capítulo se describió la arquitectura del modelo desarrollado en la presente tesis. El método propuesto utiliza los algoritmos genéticos para la tarea de seleccionar una combinación de métodos que reduzcan el error en validación cruzada en una base de datos dada. Debido al proceso de búsqueda realizada por el algoritmo de optimización, el método es útil en bases de datos de diferentes dominios; sin embargo, la carga computacional requerida por los dos algoritmos genéticos es alta.

Por otro lado, se propone el uso de lógica difusa, lo cual nos permite conocer el grado con el cual una instancia pertenece a una clase. De esta forma, en problemas con clases traslapadas, es posible conocer qué tanta información tiene dicha instancia con respecto a cada una de las clases.

También, se considera el uso de un ensamble de clasificadores, determinando el conjunto de clasificadores que van a ser combinados. Esto último es realizado mediante una medida de diversidad entre los diferentes modelos.

Capítulo 5

Experimentos y Resultados

En este capítulo se presentan los experimentos y resultados obtenidos con el método propuesto. En la sección 5.1 se describen los experimentos; en la sección 5.2 se explican a detalle los experimentos realizados sobre la base de datos de leucemia aguda y los resultados alcanzados; en la sección 5.3 se comparan los resultados obtenidos en la clasificación de leucemia aguda con los reportados en otros trabajos. Posteriormente, en la sección 5.4 se presentan los resultados obtenidos sobre algunas bases de datos del repositorio UCI [51]. Finalmente, en la sección 5.5 se hace una discusión de los resultados obtenidos.

5.1. Descripción de los Experimentos

Se realizaron diversos experimentos para probar el método propuesto en el presente trabajo de investigación. Se experimentó usando la base de datos de leucemia construida por el Instituto Mexicano del Seguro Social (IMSS) en cooperación con INAOE[1], así como con algunas bases de datos disponibles en el repositorio del UCI

[1] Ver descripción detallada de la base de datos en el apéndice A

[51]. La evaluación de los experimentos se realizó usando validación cruzada con 10 particiones. Se consideraron las siguientes métricas: exactitud, tasa de verdaderos positivos (TVP), tasa de verdaderos negativos (TVN) y el área bajo la curva ROC, explicadas en la sección 2.7.

5.2. Clasificación de células de leucemia

En esta sección se presentan los resultados de los diversos experimentos realizados con la base de datos de leucemia aguda. La base de datos de leucemia aguda usada se describe en el apéndice A.

5.2.1. Experimentos realizados

Los experimentos ejecutados sobre la base de datos de leucemia aguda fueron tanto de clasificación binaria como de problema multiclase. En el caso de clasificación binaria se agruparon distintos conjuntos de datos por tipo y subtipo; en la clasificación multiclase, se agruparon por subtipos. A continuación se muestran los siete diferentes grupos de datos usados.

- Grupos de datos para clasificación binaria:
  - LLA contra LMA
  - L1 contra L2
  - M2 contra M3 y M5
  - M3 contra M2 y M5
  - M5 contra M2 y M3
- Grupos de datos para clasificación multiclase:
  - M2 contra M3 contra M5

- L1 contra L2 contra M2 contra M3 contra M5

En total se tienen siete diferentes grupos de datos; para cada uno de éstos se tienen seis conjuntos de datos con características diferentes (formándose así un total de cuarenta y dos conjuntos de datos). El proceso de extracción de características fue realizado por Reta [45]. Los seis conjuntos se encuentran formados de la siguiente manera:

- Características extraídas de la región de la célula completa (C):
  - Morfológicas, estadísticas y de textura (A).
  - Componentes principales (B).
  - Morfológicas, estadísticas, de textura y componentes principales (C).
- Características extraídas de las regiones de núcleo y citoplasma (N&Ci):
  - Morfológicas, estadísticas y de textura (A).
  - Componentes principales (B).
  - Morfológicas, estadísticas, de textura y componentes principales (C).

Con base en lo anterior, se cuenta con un total de 42 conjuntos de datos de leucemia aguda, los cuales son probados con el método propuesto. Para cada uno de los 42 conjuntos de datos se encuentra un modelo individual y un conjunto de modelos con ensamble que se adecuan a cada conjunto de datos.

Todos los experimentos fueron realizados usando los siguientes parámetros: 24 individuos por generación (inicialmente 6 individuos por sub-población), el valor β es fijado en 4, α en 3, las poblaciones evolucionaron durante 50 generaciones, la probabilidad de cruza es de 0.8 y la de mutación de 0.05.

Los modelos obtenidos con el método propuesto son usados en la jerarquía propuesta, lo que permite aprovechar la relación que existen entre los tipos y subtipos de leucemia aguda.

5.2.2. Resultados

A continuación se presentan los resultados obtenidos para cada experimento. El método de evaluación usado es validación cruzada con 10 particiones; las métricas usadas son la exactitud y el área bajo la curva ROC. Además, en los problemas de clasificación binaria se considera la tasa de verdaderos positivos (TVP) y la tasa de verdaderos negativos (TVN). Estos resultados son reportados para cada región (célula o núcleo y citoplasma), para cada uno de los diferentes grupos de características extraídas (A, B o C).

Selección de modelo individual

En la tabla 5.1 se muestran los resultados de las clasificaciones binarias, con modelo individual, para las bases de datos de leucemia aguda; la tabla 5.2 muestra los resultados obtenidos para las clasificaciones multiclase con modelo individual.

A partir de los datos presentados en las tablas 5.1 y 5.2 se observa que, en la clasificación de leucemia aguda, los mejores resultados se obtienen al considerar las características morfológicas, estadísticas y de textura de las regiones del núcleo y citoplasma. Por el contrario, los peores resultados son obtenidos con características de componentes principales.

Tabla 5.1: Resultados de clasificación binaria con modelo individual para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica.

	Conjunto de datos					
No.	Región	Características	Exactitud	TVP	TVN	ROC
	ALL vs AML					
1	Célula	A	88.78 ± 8.26	89.88 ± 4.95	86.75 ± 12.57	93.90 ± 3.96
2		B	76.62 ± 19.33	78.31 ± 13.12	74.15 ± 26.88	83.06 ± 13.09
3		C	88.78 ± 12.11	84.75 ± 17.39	92.31 ± 14.37	91.68 ± 9.43
4	N&Ci	A	90.52 ± 13.20	92.31 ± 20.78	88.47 ± 12.29	95.38 ± 10.69
5		B	78.99 ± 9.93	77.97 ± 12.95	79.89 ± 16.14	83.27 ± 6.52
6		C	93.52 ± 4.59	91.52 ± 6.86	95.27 ± 6.53	96.13 ± 3.04
	L1vsL2					
7	Célula	A	87.36 ± 3.87	82.45 ± 8.79	91.26 ± 8.86	91.89 ± 2.58
8		B	75.20 ± 5.65	65.73 ± 9.21	82.36 ± 5.71	76.69 ± 3.99
9		C	84.84 ± 9.63	84.27 ± 15.05	85.27 ± 13.05	88.34 ± 6.84
10	N&Ci	A	86.89 ± 6.27	85.27 ± 11.76	87.97 ± 10.08	91.92 ± 4.02
11		B	79.30 ± 3.23	80.36 ± 7.96	78.52 ± 5.45	81.51 ± 2.17
12		C	87.00 ± 7.55	85.27 ± 8.46	88.35 ± 9.22	91.03 ± 5.06
	M2vsM3M5					
13	Célula	A	92.41 ± 7.33	92.67 ± 6.93	92.18 ± 10.11	96.22 ± 4.60
14		B	73.80 ± 5.08	71.22 ± 10.98	76.00 ± 17.29	78.89 ± 4.04
15		C	90.43 ± 9.52	90.44 ± 9.42	90.36 ± 13.73	92.46 ± 7.93
16	N&Ci	A	89.90 ± 4.73	88.22 ± 10.39	91.36 ± 7.07	92.50 ± 2.65
17		B	74.50 ± 11.87	69.00 ± 16.61	79.82 ± 18.61	79.37 ± 8.71
18		C	91.56 ± 7.65	88.67 ± 9.92	94.27 ± 6.54	94.83 ± 4.81
	M3vsM2M5					
19	Célula	A	93.47 ± 2.29	85.00 ± 14.14	96.00 ± 3.44	96.66 ± 3.23
20		B	81.38 ± 4.39	51.00 ± 19.83	90.71 ± 4.69	70.79 ± 8.93

Tabla 5.1: Resultados de clasificación binaria con modelo individual para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica (continuación).

	Conjunto de datos					
No.	Región	Características	Exactitud	TVP	TVN	ROC
21		C	90.02 ± 8.89	78.00 ± 20.30	94.00 ± 10.33	89.03 ± 8.77
22	N&Ci	A	94.42 ± 3.03	85.50 ± 15.54	97.33 ± 3.44	91.42 ± 7.15
23		B	80.83 ± 8.40	46.50 ± 5.30	92.00 ± 9.85	74.46 ± 15.22
24		C	87.63 ± 12.16	68.50 ± 6.26	93.33 ± 6.29	91.90 ± 6.35
	M5vsM2M3					
25	Célula	A	92.41 ± 6.89	91.00 ± 13.43	93.00 ± 7.98	95.40 ± 5.41
26		B	71.18 ± 5.87	44.33 ± 11.34	81.67 ± 6.71	66.02 ± 4.73
27		C	93.47 ± 5.90	93.33 ± 12.77	93.67 ± 6.13	93.50 ± 7.38
28	N&Ci	A	94.37 ± 4.43	89.00 ± 13.43	96.43 ± 5.05	96.71 ± 3.86
29		B	72.84 ± 9.63	54.33 ± 7.98	80.29 ± 21.08	70.60 ± 10.26
30		C	92.41 ± 2.68	90.67 ± 13.68	92.95 ± 3.37	95.52 ± 3.26

Tabla 5.2: Resultados de clasificación multiclase con modelo individual para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica.

	Conjunto de datos			
No.	Región	Características	Exactitud	ROC
	M2vsM3vsM5			
31	Célula	A	81.82 ± 12.85	96.09 ± 8.32

Tabla 5.2: Resultados de clasificación multiclase con modelo individual para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica (continuación).

	Conjunto de datos			
No.	Región	Características	Exactitud	ROC
32		B	67.68 ± 18.49	71.90 ± 17.22
33		C	82.32 ± 4.16	90.92 ± 13.38
34	N&Ci	A	81.31 ± 10.02	93.54 ± 17.34
35		B	69.19 ± 7.59	74.81 ± 7.51
36		C	86.87 ± 8.51	93.23 ± 8.38
	L1vsL2vsM2vsM3vsM5			
37	Célula	A	75.40 ± 8.50	85.04 ± 5.76
38		B	60.92 ± 12.28	73.10 ± 8.65
39		C	69.20 ± 15.18	85.05 ± 8.99
40	N&Ci	A	72.87 ± 10.64	80.47 ± 7.34
41		B	61.38 ± 14.98	76.48 ± 8.34
42		C	68.51 ± 10.98	83.50 ± 10.10

Selección de modelo con ensamble

En la tabla 5.3 se muestran los resultados obtenidos en clasificaciones binarias y en la 5.4 para las clasificaciones multiclase con ensamble. De estas tablas es importante observar que los resultados obtenidos con el modelo con ensamble muestran una reducción en la desviación estándar en comparación con los obtenidos por los modelos individuales. En promedio, la desviación estándar es

reducida en un 6.12 % considerando la exactitud y 2.98 % en el área bajo la curva ROC.

Tabla 5.3: Resultados de clasificación binaria con ensamble para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica.

	Conjunto de datos					
No.	Región	Características	Exactitud	TVP	TVN	ROC
	ALL vs AML					
1	Célula	A	89.10 ± 0.94	82.72 ± 1.31	94.67 ± 1.20	95.12 ± 5.35
2		B	76.30 ± 1.18	78.31 ± 1.48	74.55 ± 1.36	83.66 ± 4.47
3		C	86.41 ± 1.16	81.36 ± 1.42	90.83 ± 1.63	92.44 ± 4.74
4	N&Ci	A	93.83 ± 0.59	92.88 ± 0.18	94.67 ± 1.20	95.23 ± 5.13
5		B	78.99 ± 1.19	77.97 ± 1.20	79.88 ± 1.87	81.20 ± 4.71
6		C	95.58 ± 0.85	93.22 ± 1.03	97.63 ± 1.24	95.01 ± 4.78
	L1vsL2					
7	Célula	A	84.39 ± 3.02	81.37 ± 3.13	86.67 ± 6.30	92.46 ± 4.18
8		B	81.01 ± 2.08	72.55 ± 4.08	87.41 ± 3.37	82.22 ± 5.03
9		C	90.72 ± 3.63	85.29 ± 4.93	94.82 ± 4.85	93.55 ± 6.20
10	N&Ci	A	89.45 ± 2.02	84.31 ± 4.90	93.33 ± 5.29	94.46 ± 4.15
11		B	82.70 ± 2.12	80.39 ± 0.77	84.44 ± 3.88	87.43 ± 5.44
12		C	90.30 ± 2.60	88.24 ± 3.45	91.85 ± 4.02	92.99 ± 5.07
	M2vsM3M5					
13	Célula	A	89.89 ± 4.49	90.53 ± 3.36	89.32 ± 6.66	96.32 ± 2.33
14		B	80.30 ± 3.55	72.63 ± 4.42	87.38 ± 3.95	82.51 ± 5.56
15		C	90.40 ± 2.78	88.42 ± 3.02	92.23 ± 4.09	95.73 ± 2.95
16	N&Ci	A	87.87 ± 5.41	84.21 ± 4.68	91.26 ± 6.78	95.70 ± 2.54
17		B	78.95 ± 2.64	69.00 ± 1.17	70.87 ± 5.56	82.89 ± 5.21
18		C	93.93 ± 2.93	90.53 ± 3.36	97.09 ± 4.39	94.28 ± 4.43
	M3vsM2M5					

Tabla 5.3: Resultados de clasificación binaria con ensamble para cada experimento. Los resultados son el promedio y la desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica (continuación).

	Conjunto de datos					
No.	Región	Características	Exactitud	TVP	TVN	ROC
19	Célula	A	93.94 ± 2.40	85.11 ± 2.42	96.69 ± 3.16	96.93 ± 4.70
20		B	80.81 ± 3.99	51.06 ± 6.26	90.06 ± 3.43	76.02 ± 4.87
21		C	85.35 ± 2.23	63.83 ± 7.53	87.42 ± 2.10	95.61 ± 5.19
22	N&Ci	A	90.40 ± 3.47	85.50 ± 7.25	98.67 ± 2.72	96.66 ± 4.38
23		B	85.86 ± 2.25	51.06 ± 7.98	96.68 ± 3.47	79.37 ± 4.20
24		C	93.43 ± 2.40	78.72 ± 2.42	98.01 ± 3.16	94.32 ± 4.70
	M5vsM2M3					
25	Célula	A	92.42 ± 4.09	87.50 ± 8.34	94.37 ± 4.31	97.66 ± 3.80
26		B	71.21 ± 1.89	44.64 ± 6.30	81.69 ± 3.49	76.01 ± 5.02
27		C	87.37 ± 3.10	80.35 ± 5.08	90.14 ± 4.55	95.03 ± 3.67
28	N&Ci	A	94.94 ± 3.96	89.29 ± 8.61	97.18 ± 3.57	97.55 ± 3.43
29		B	78.28 ± 1.83	62.50 ± 5.45	84.50 ± 2.41	83.22 ± 3.97
30		C	91.91 ± 2.55	82.43 ± 1.72	95.77 ± 3.61	93.45 ± 5.84

Tabla 5.4: Resultados de clasificación multiclase con ensamble para cada experimento. Los resultados son el promedio y desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica.

	Conjunto de datos			
Número	Región	Características	Exactitud	ROC
	M2vsM3vsM5			
31	Célula	A	82.83 ± 2.30	96.97 ± 2.64

Tabla 5.4: Resultados de clasificación multiclase con ensamble para cada experimento. Los resultados son el promedio y desviación estándar de los resultados obtenidos usando validación cruzada con 10 particiones para cada métrica (continuación).

	Conjunto de datos			
Número	Región	Características	Exactitud	ROC
32		B	67.68 ± 2.72	78.18 ± 4.45
33		C	85.35 ± 2.66	93.98 ± 2.51
34	N&Ci	A	86.87 ± 2.49	96.64 ± 2.67
35		B	72.76 ± 2.70	81.83 ± 3.56
36		C	86.87 ± 4.35	95.43 ± 3.41
	L1vsL2vsM2vsM3vsM5			
37	Célula	A	74.25 ± 0.99	95.84 ± 2.61
38		B	65.98 ± 1.29	79.19 ± 6.67
39		C	79.31 ± 0.98	94.21 ± 4.17
40	N&Ci	A	76.32 ± 1.04	96.09 ± 4.56
41		B	68.97 ± 1.09	83.22 ± 7.79
42		C	77.01 ± 1.52	93.61 ± 3.70

Modelos seleccionados

En la tabla 5.5 se describen los mejores modelos individuales –conformados por un método de selección de atributos (MSA), un tipo de función de pertenencia difusa (TFP), número de propiedades lingüísticas (NPL) y un método de clasificación (MC)– para cada uno de los conjuntos de datos. Asimismo, el método de combinación (ME) de los múltiples modelos individuales es especificado para

cada uno de los conjuntos de datos.

Tabla 5.5: Modelo de clasificación y método de combinación resultantes del proceso evolutivo para cada base de datos

Número	MSA	TFP	NPL	MC	ME
1	Ganancia de Información	Gaussiana	7	FDT	Voto pesado
2	Basado en Correlación	Campana	5	FRNN	Apilado
3	*ReliefF*	Triangular	5	FKNN	Voto simple
4	*ReliefF*	Trapezoidal	3	FKNN	Voto pesado
5	χ^2	Gaussiana	5	FDT	Voto simple
6	*ReliefF*	Triangular	3	FKNN	Voto pesado
7	Basado en Correlación	Campana	5	FDF	Voto simple
8	*ReliefF*	Gaussiana	7	FDT	Voto pesado
9	Basado en Correlación	Trapezoide	5	FKNN	Voto pesado
10	*ReliefF*	Campana	7	FDF	Voto simple
11	Basado en Correlación	Gaussiana	5	FRNN	Apilado
12	χ^2	Gaussiana	5	FDF	Voto pesado
13	Basado en Correlación	Trapezoide	3	FKNN	Voto simple
14	Ganancia de Información	Gaussiana	7	FDT	Apilado
15	Basado en Correlación	Trapezoide	5	FKNN	Voto pesado
16	*RelieF*	Campana	7	FDF	Voto simple
17	Ganancia de Información	Gaussiana	3	FDT	Voto simple
18	χ^2	Gaussiana	5	FDF	Voto pesado
19	Ganancia de Información	Gaussiana	3	FDT	Voto simple
20	Ganancia de Información	Gaussiana	7	FRNN	Voto pesado
21	*ReliefF*	Triangular	5	FKNN	Voto pesado
22	Basado en Correlación	Campana	3	FDF	Voto pesado

Tabla 5.5: Modelo de clasificación y método de combinación resultantes del proceso evolutivo para cada base de datos (continuación).

Número	MSA	TFP	NPL	MC	ME
23	Ganancia de Información	Campana	7	FDT	Voto simple
24	Ganancia de Información	Gaussiana	5	FDF	Voto pesado
25	*ReliefF*	Triangular	3	FKNN	Voto pesado
26	Ganancia de Información	Gaussiana	7	FDT	Apilado
27	*ReliefF*	Campana	5	FDF	Voto pesado
28	χ^2	Gaussiana	5	FDF	Voto pesado
29	Ganancia de Información	Gaussiana	7	FDF	Voto pesado
30	Basado en Correlación	Campana	3	FDF	Voto pesado
31	*ReliefF*	Triangular	5	FDF	Voto simple
32	χ^2	Campana	5	FDF	Voto pesado
33	Basado en Correlación	Trapezoide	5	FKNN	Voto pesado
34	χ^2	Gaussiana	5	FDF	Voto simple
35	*ReliefF*	Campana	7	FDF	Voto simple
36	Basado en Correlación	Gaussiana	7	FDF	Voto pesado
37	*ReliefF*	Gaussiana	7	FDF	Voto simple
38	Ganancia de Información	Campana	3	FDF	Voto simple
39	*ReliefF*	Triangular	3	FKNN	Voto pesado
40	Ganancia de Información	Gaussiana	7	FDF	Voto simple
41	Basado en Correlación	Campana	3	FDF	Apilado
42	Basado en Correlación	Gaussiana	5	FDT	Voto pesado

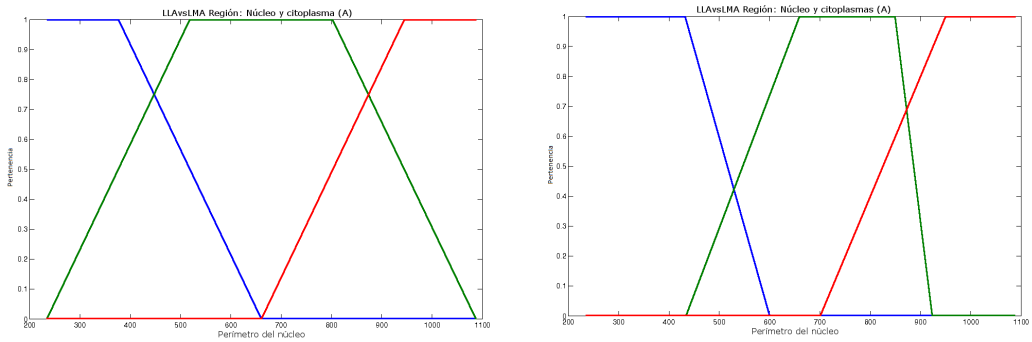
Funciones de pertenencia

Las funciones de pertenencia de cada característica también son ajustadas al problema. En la figura 5.1 se muestran algunos ejemplos de funciones de pertenencia resultantes (derecha) para algunas características de la base de datos de leucemia aguda, después de haber iniciado con funciones uniformemente distribuidas (izquierda). Debido a la cantidad de características no se incluyen todas.

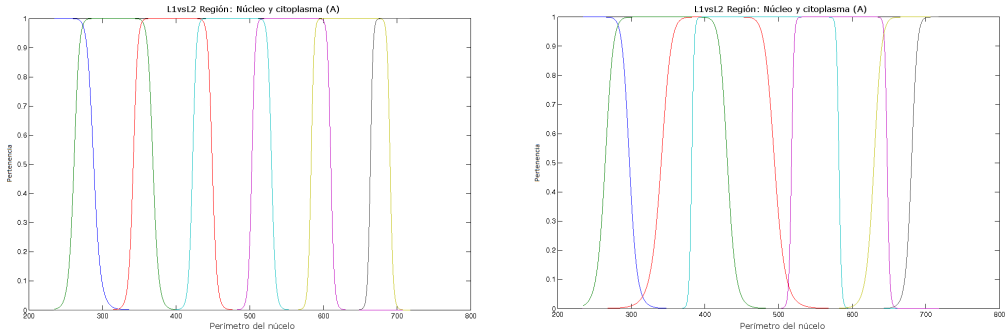
Clasificación jerárquica

La tabla 5.6 muestra los resultados obtenidos aplicando la jerarquía para la clasificación, descrita en la sección 4.2. Es importante hacer notar que, debido a la complejidad del método, en la clasificación jerárquica no se usó validación cruzada con 10 particiones como método de evaluación. En este caso se seleccionaron aleatoriamente 63 ejemplos del conjunto de datos para prueba, y el resto fueron usados como ejemplos de entrenamiento. En la jerarquía sólo se consideraron los modelos obtenidos utilizando las características morfológicas, estadísticas y de textura tanto del núcleo como del citoplasma ya que, en promedio, son con las que mejor resultados se obtuvieron.

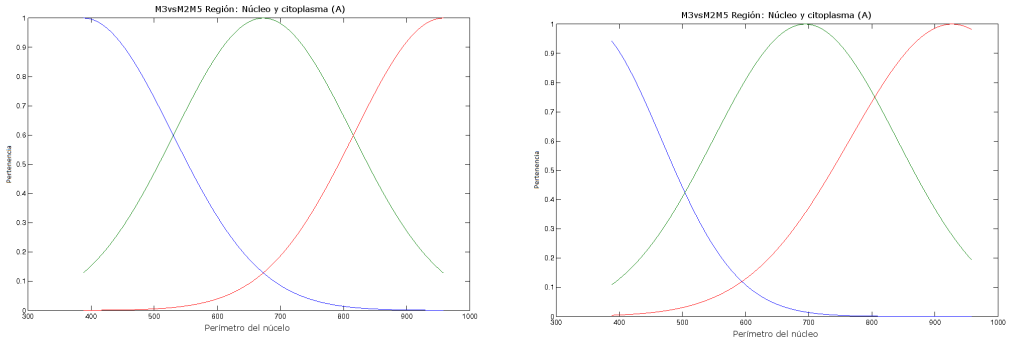
De la tabla 5.6 se puede observar que al clasificar por tipos, donde el error es menor, se reduce el grado de confusión en los subtipos. Con el uso de la jerarquía se alcanza, en promedio, 93.02 % de clasificación correcta.



(a)



(b)



(c)

Figura 5.1: Funciones de pertenencia para la característica perímetro del núcleo. La primera columna representa las funciones de pertenencia iniciales, las cuales son uniformemente distribuidas; en la segunda columna, las funciones de pertenencia resultantes. En (a) se muestran las funciones de pertenencia para la base de datos LLA contra LMA. En (b), de la base de datos L1 contra L2. En (c) para la base de datos de M3 contra M2 y M5

Tabla 5.6: Resultados de las clasificaciones del conjunto de pruebas usando la jerarquía

Clase	No. Ejemplos	% Correctos
Tipos	63	96.93 %
LLA	31	93.55 %
LMA	32	100 %
Subtipos	43	93.02 %
L1	12	91.67 %
L2	13	92.31 %
M2	9	88.89 %
M3	4	100 %
M5	5	100 %

5.3. Comparación con otros trabajos

El desempeño del trabajo propuesto se comparó con los reportados en otros trabajos para el caso de leucemia aguda. Las mismas bases de datos son usadas y la métrica utilizada para la comparación es el área bajo la curva ROC, la cual es usada en los trabajos previos. En la tabla 5.7 se muestran los mejores resultados reportados en los trabajos de Reta [45] y Escalante [15] y los obtenidos con el método propuesto, usando ensambles, en los problemas de clasificación binaria y multiclase.

A fin de probar la validez estadística de estos experimentos, en la tabla 5.8 se presentan los resultados de la pruebas de signos de Wilcoxon [54]. Esta tabla compara los resultados obtenidos con el método propuesto usando ensambles contra los resultados reportandos por Reta [45] y Escalante [15] y los obtenidos con el

Tabla 5.7: Tabla comparativa de los mejores resultados reportados en trabajos previos y los obtenidos con la propuesta

Conjunto de datos				
Región	Características	Reta [45]	Escalante [15]	Método propuesto
LLAvsLMA				
N&Ci	C	92.17	**96.66**	95.01
L1vsL2				
N&Ci	A	90.69	94.10	**94.46**
M2vsM3M5				
N&Ci	A	**95.90**	95.70	95.70
M3vsM2M5				
N&Ci	C	89.85	93.24	**94.32**
M5vsM2M3				
N&Ci	A	95.52	**97.68**	97.55
M2vsM3vsM5				
N&Ci	C	92.80	93.94	**95.43**
L1vsL2vsM2vsM3vsM5				
N&Ci	C	92.33	**94.21**	93.61

método sin usar ensambles. T^+ representan la suma de los órdenes cuya diferencia es positiva, T^- es la suma de los órdenes cuya diferencia es negativa y T es el mínimo de T^+ y T^-. La hipótesis nula, H_0, es rechazada si T es menor que 111, el cual es un valor determinado por el nivel de significancia, α, que en este caso es de 0.001. Si la hipótesis nula es rechazada, implica que la diferencia entre los dos métodos es significativa. La prueba de signos de Wilcoxon es explicada a detalle en el apéndice B.2. La tabla completa de comparación para cada base de datos,

así como las pruebas de significación estadística se presentan en el apéndice B.

Tabla 5.8: Prueba de signos de Wilcoxon para comparar los resultados obtenidos con los ensambles del método propuesto contra los reportados en trabajos previos y los obtenidos con el modelo individual

H_0	T^+	T^-	T	Hipótesis ($\alpha = 0.001$)
Reta[45]	576	19	19	Se rechaza la hipótesis nula
PSMS Escalante [15]	542	53	53	Se rechaza la hipótesis nula
EPSMS Escalante [15]	343.3	250.3	250.3	No se rechaza la hipóstesis nula
Propuesta Individual	544	51	51	Se rechaza la hipótesis nula

5.4. Bases de datos del repositorio UCI

Con el objetivo de sensar experimentalmente la generalidad del método propuesto, éste se probó sobre algunos conjuntos de datos disponibles en el repositorio UCI [51]. Las bases de datos utilizadas fueron:

- Cáncer de mama de Wisconsin. Esta base de datos cuenta con 569 ejemplos y dos clases, (212 ejemplos para la clase positiva y 357 ejemplos para la clase negativa). Cada ejemplo es decrito por un vector con 31 atributos más un atributo de clase.

- Transfusión de sangre. La base de datos está formada por 748 ejemplos descrito por 4 atributos y dos clases, de los cuales 178 son de la clase positiva y 570 de la negativa.

- Ionósfera. Es un conjunto de datos compuesto por 351 ejemplos y descrito

por 34 atributos, más un atributo de clase. Este es un problema de clasificación binaria.

- Iris. Esta es una base de datos compuesta por 150 ejemplos y tres clases, 50 ejemplos por clase. Los ejemplos son descritos por 4 atributos más un atributo de clase para identificar entre los tres tipos de iris: setosa, versicolor, virginica.

- Vinos. Es una base de datos compuesta por 178 ejemplos y tres clases. Cada ejemplo es descrito por 13 atributos, obtenidos de un análisis químico.

En la tabla 5.9 se muestran los resultados en clasificación obtenidos con el método propuesto y con otros métodos como son el $C4.5$, *Naive Bayes* (NB) y El de k vecinos más cercanos (k-NN). El método de evaluación es validación cruzada con 10 particiones; las métricas utilizadas son la exactitud y el área bajo la curva ROC.

Tabla 5.9: Resultados de clasificaciones sobre las bases de datos del UCI. Se reporta el promedio sobre validación cruzada con 10 particiones

Conjunto de datos	Exactitud				ROC			
	Propuesta	C4.5	NB	k-NN	Propuesta	C4.5	NB	k-NN
Cáncer de mama	**96.86**	92.97	96.13	92.68	**98.89**	92.30	95.60	97.60
Transfusión de sangre	76.74	**77.81**	69.11	75.40	68.71	**70.00**	58.60	**70.00**
Ionosfera	**93.71**	91.45	86.32	82.62	**97.72**	89.20	82.50	93.50
Iris	**97.33**	96.00	95.33	96.00	**100**	96.80	96.60	99.40
Vinos	**98.19**	93.26	93.26	94.99	**100**	95.20	95.10	99.60

5.5. Discusión

El método propuesto permite seleccionar de forma automática un modelo de clasificación adecuado a una base de datos. Con base en los resultados obtenidos es posible observar que, en la mayoría de los casos, éstos son mejorados, con respecto a lo reportado en otros trabajos. Esto es importante ya que permite obtener clasificaciones más precisas del tipo y subtipo de leucemia aguda y en otras bases de datos.

En Reta [45] se afirmó que el considerar características combinadas del núcleo y citoplasma brinda mejores resultados en clasificación que considerar únicamente características obtenidas de la célula. Sin embargo, de acuerdo con los resultados presentados en las tablas 5.1 a la 5.4 se observa que existen casos para los cuales al considerar características de la célula se obtuvieron mejores resultados. Puede observarse que los resultados obtenidos usando las características de la célula como del núcleo y citoplasma son consistentes.

Asimismo, del método propuesto es importante destacar dos aspectos: el primero, la relevancia de la determinación de los parámetros adecuados a un modelo de clasificación, ya que estos parámetros determinan la calidad del desempeño del modelo de clasificación. Como segundo aspecto importante, nótese que, gracias al enfoque difuso presentado y al uso de modelos de clasificación afines al enfoque, es posible conocer la fuerza o grado en que cada ejemplo pertenece a los diferentes tipos y subtipos. Esto último es una estrategia que no se había abordado en trabajos anteriores de leucemia aguda y que es útil, especialmente en aquellos casos en que debido al traslape de clases el reconocimiento es difícil. De igual manera, es importante resaltar que el tiempo requerido en la determinación de los modelos de clasificación para los 42 conjuntos de datos de leucemia aguda fue de 4 días.

Capítulo 6

Conclusiones y Trabajo Futuro

En este trabajo de tesis se propone la selección automática de un modelo de clasificación, basado en un pre-procesamiento difuso de las características, adecuados al problema de leucemia aguda. La determinación automática del modelo de clasificación involucra la elección del método de selección de atributos, el número de propiedades lingüísticas, el tipo de función de pertenencia, así como los parámetros de éstos que minimicen la imprecisión provocada por el grado de traslape. Asimismo, se selecciona el algoritmo de clasificación difuso y los parámetros asociados al mismo y la selección de los múltiples modelos de clasificación que son combinados en un ensamble. Además, la ventaja principal de la selección automática del modelo de clasificación es que no requiere conocimiento del dominio de los datos involucrados.

De los resultados obtenidos con el método propuesto en la presente tesis se puede observar que éstos fueron mejorados en comparación con trabajos previos que no realizan selección de modelo. Con esto se demuestra experimentalmente la importancia de elegir adecuadamente el modelo de clasificación para un problema determinado. También puede notarse que el desempeño del método propuesto

es comparable con resultados obtenidos por métodos que realizan selección de modelo reportados en la literatura, añadiendo la ventaja que otorga conocer el grado de pertenencia de una instancia a cada clase como información adicional. Esta información es especialmente útil en aquellos casos donde el traslape entre clases es alto, por lo que su diferenciación se dificulta.

Es importante resaltar que los resultados obtenidos en clasificación de leucemia aguda fueron consistentes tanto en la región de célula completa como en las regiones de núcleo y citoplasma. También, es destacable que, a diferencia de otros enfoques que combinan múltiples modelos de clasificación por promedio simple, el método propuesto determina la forma de combinarlos, siendo en la mayoría de los casos por promedio pesado.

6.1. Contribución

En el presente trabajo de tesis se resaltan las siguientes contribuciones:

- un método para la clasificación de clases traslapadas mediante la asignación de grados de pertenencia a los diferentes ejemplos de leucemia aguda a tipos y sub-tipos,
- determinación de un modelo de clasificación adecuado a cada conjunto de datos,
- determinación de un ensamble de clasificadores y la forma en la que se combinan las predicciones de múltiples clasificadores individuales.

Como resultado directo de esta investigación, a la fecha se han realizado las siguientes publicaciones:

- Rosales-Pérez A, Reyes-García CA, and Gómez-Gil P. "Genetic Fuzzy Relational Neural Network for Infant Cry Classification" LNCS 6718, pp. 288–296, 2011. Martínez-Trinidad et al. (Eds.): Springer-Verlag Berlin Heidelberg. Proceedings of the Third Mexican Congress on Pattern Recognition (MCPR 2011), July 2011.

- Rosales-Pérez A, Reyes-García CA, Gómez-Gil P, Gonzalez JA, Altamirano L. "Genetic Selection of Fuzzy Model for Acute Leukemia Classification." To be published in the Proceedings of the 10th. Mexican International Conference on Artificial Intelligence, MICAI 2011 in LNAI. November 2011.

6.2. Trabajo futuro

Como trabajo futuro se proponen las siguiente líneas de investigación con respecto a la selección del modelo:

- extender el método propuesto para otros métodos de selección de atributos, así como algoritmos de clasificación;

- incluir otros métodos de pre-procesamiento de los datos, tales como balanceo de clases, estandarización y normalización, entre otros;

- explorar el uso de otros métodos de optimización como estrategias evolutivas, colonia de hormigas o enjambre de partículas;

- estudiar la factibilidad de usar optimización multiobjetivo.

Por el lado de clasificación de leucemia se propone como posibles trabajos futuros la recolección de muestras, principalmente de sub-tipos de los que no se

cuenta con ejemplos, así como la clasificación de células bifenotípicas y extracción de características más descriptivas.

Apéndices

Apéndice A

Descripción de la base de datos de leucemia aguda

Para los experimentos usamos la base de datos de leucemia aguda construida por el IMSS en cooperación con INAOE. Esta base de datos cuenta con 633 ejemplos de frotis de médula ósea, los cuales fueron digitalizados por Morales [37] usando una cámara con lente Carl Zeiss con un objetivo de 100x conectada a un microscopio. Las imágenes digitalizadas cuentan con una resolución de 800×600 píxeles. Las células fueron seleccionadas y clasificadas manualmente en diferentes tipos de leucemia aguda por un médico experto en el dominio. La tabla A.1 muestra la cantidad de ejemplos por tipo y subtipo de leucemia aguda disponibles en esta base de datos.

Las imágenes de cada célula fueron segmentadas en tres regiones: célula completa, núcleo y citoplasma. Para cada región se extrajeron características, las cuales fueron extraídas por Reta [45]. Las características extraídas son de tipo morfológicas, estadísticas, de textura, de proporción de tamaño y de componentes principales. Las características morfológicas fueron extraídas de cada región (véase

Tabla A.1: Cantidad de ejemplos por tipo y subtipo de leucemia aguda en la base de datos del IMSS-INAOE

LLA		LMA	
Subtipo	Cantidad	Subtipo	Cantidad
L1	102	M2	95
L2	135	M3	47
		M5	56
Sin clasificar	58	Sin clasificar	140
Total	295	Total	338

tabla A.2). Por otro lado, las de textura, estadísticas y componentes principales se obtuvieron por cada canal del RGB y de la escala de grises para cada región (véase tabla A.3). Finalmente, la tabla A.4 describe las características de proporción de tamaño usadas.

Tabla A.2: Características extraídas para las regiones de la célula completa, núcleo y citoplasma. En la columna correspondiente a cada región se indica si dicha característica fue extraída o no.

Características	Núcleo	Citoplasma	Célula
Morfológicas			
Área	Sí	Sí	Sí
Perímetro	Sí	No	Sí
Circularidad	Sí	No	Sí
Anchura	Sí	No	Sí
Altura	Sí	No	Sí
Elongación	Sí	No	Sí
Eje mayor	Sí	No	Sí
Eje menor	Sí	No	Sí
Excentricidad	Sí	No	Sí
Extensión	Sí	No	Sí
Diámetro equivalente	Sí	No	Sí
Número de Euler	Sí	No	Sí
Área convexa	Sí	No	Sí
Solidez	Sí	No	Sí

Tabla A.3: Características extraídas para las regiones de la célula completa, núcleo y citoplasma. En la columna correspondiente a cada región se indica si dicha característica fue extraída o no

Características	Núcleo	Citoplasma	Célula
Estadísticas			
Moda	Sí	Sí	Sí
Media	Sí	Sí	Sí
Desviación Estándar	Sí	Sí	Sí
Varianza	Sí	Sí	Sí
Densidad óptica integrada	Sí	No	Sí
Promedio de densidad óptica	Sí	No	Sí
Textura			
Entropía	Sí	No	Sí
Contraste	Sí	No	Sí
Correlación	Sí	No	Sí
Energía	Sí	No	Sí
Homogeneidad	Sí	No	Sí
Componentes principales			
PCAs	Sí	No	Sí

Tabla A.4: Características de proporción de tamaño extraídas

Característica	Núcleo/Célula	Núcleo/Citoplasma
Área	Sí	Sí
Perímetro	Sí	No

Apéndice B

Comparación de resultados y pruebas de significancia estadística

En este apéndice se presenta una tabla completa de comparación con los resultados obtenidos en el presente trabajo de tesis y los resultados en trabajos previos en la clasificación de leucemia aguda. Asimismo, pruebas de significancia estadística son realizadas para determinar si las diferencias obtenidas son estadísticamente significativas.

B.1. Comparación de resultados

Los resultados obtenidos con el método propuesto son mostrados en las tablas 5.1 a la 5.4; éstos son comparados con los reportados por Reta [45] y Escalante [15]. La comparación es realizada sobre trabajos que realizaron clasificación de leucemia aguda basándose en la apariencia morfológica de la célula; además, los trabajos de Reta [45] y Escalante [15], quienes usaron los mismos conjuntos de datos que los usados en los experimentos en este trabajo de tesis. Para la comparación de

los resultados se usa la métrica del área bajo la curva ROC, la cual es usada como métrica de evaluación en trabajos previos. En la tabla B.1 se muestran los resultados reportados en los trabajos previos y los reportados en la presente tesis; el mejor resultado para cada caso está en negrita.

Tabla B.1: Comparación de resultados de clasificación para cada experimento. Los resultados son el promedio de los resultados obtenidos usando validación cruzada con 10 particiones. En cada caso el mejor resultado es presentado en negrita

Conjunto de datos		Reta [45]	Escalante [15]		Método	
Región	Características		PSMS	EPSMS	Individual	Ensamble
ALL vs AML						
Célula	A	89.26 %	93.87 %	**95.14** %	93.90 %	95.12 %
	B	79.06 %	83.41 %	83.43 %	83.06 %	**83.66** %
	C	89.92 %	93.15 %	**95.04** %	93.68 %	92.44 %
N&Ci	A	90.65 %	95.16 %	**96.16** %	95.38 %	95.23 %
	B	81.27 %	79.74 %	80.32 %	**83.27** %	81.20 %
	C	92.17 %	95.66 %	**96.66** %	96.13 %	95.01 %
L1 vs L2						
Célula	A	81.40 %	87.68 %	92.24 %	91.89 %	**92.46** %
	B	76.08 %	**88.54** %	84.11 %	76.69 %	82.22 %
	C	82.25 %	79.07 %	92.38 %	88.34 %	**93.55** %
N&Ci	A	90.69 %	74.79 %	94.10 %	91.92 %	**94.45** %
	B	83.67 %	**87.45** %	87.41 %	81.51 %	87.43 %
	C	88.61 %	88.06 %	**93.09** %	91.03 %	93.00 %
M2 vs M3 y M5						
Célula	A	80.45 %	91.93 %	95.78 %	96.22 %	**96.32** %

Tabla B.1: Comparación de resultados de clasificación para cada experimento. Los resultados son el promedio de los resultados obtenidos usando validación cruzada con 10 particiones. En cada caso el mejor resultado es presentado en negrita (continuación).

Conjunto de datos		Reta [45]	Escalante [15]		Método	
Región	Características		PSMS	EPSMS	Individual	Ensamble
	B	71.06 %	72.99 %	80.33 %	78.89 %	**82.51** %
	C	84.12 %	**95.93** %	95.70 %	92.46 %	95.72 %
N&Ci	A	95.90 %	**96.71** %	95.70 %	92.50 %	95.70 %
	B	78.93 %	79.06 %	**83.48** %	79.37 %	82.89 %
	C	94.68 %	93.65 %	94.36 %	**94.83** %	94.28 %
M3 vs M2 y M5						
Célula	A	78.82 %	88.25 %	92.57 %	96.66 %	**96.93** %
	B	78.67 %	72.93 %	**79.86** %	70.79 %	76.02 %
	C	71.01 %	92.74 %	94.15 %	89.03 %	**95.61** %
N&Ci	A	87.97 %	91.51 %	96.62 %	91.42 %	**96.66** %
	B	73.91 %	77.26 %	76.89 %	74.46 %	**79.37** %
	C	89.85 %	93.06 %	93.24 %	91.90 %	**94.33** %
M5 vs M2 y M3						
Célula	A	86.64 %	87.32 %	95.28 %	95.40 %	**97.66** %
	B	73.14 %	61.77 %	75.78 %	66.02 %	**76.01** %
	C	84.98 %	92.29 %	**95.85** %	93.50 %	95.03 %
N&Ci	A	95.52 %	92.94 %	**97.68** %	96.71 %	97.55 %
	B	73.32 %	79.24 %	**85.91** %	70.60 %	83.22 %
	C	93.54 %	88.97 %	93.07 %	**95.52** %	93.45 %
M2 vs M3 vs M5						
Célula	C	78.66 %	92.37 %	93.94 %	90.92 %	**93.98** %

Tabla B.1: Comparación de resultados de clasificación para cada experimento. Los resultados son el promedio de los resultados obtenidos usando validación cruzada con 10 particiones. En cada caso el mejor resultado es presentado en negrita (continuación).

Conjunto de datos		Reta [45]	Escalante [15]		Método	
Región	Características		PSMS	EPSMS	Individual	Ensamble
N&Ci	C	92.80 %	92.36 %	93.94 %	93.23 %	**95.43** %
L1 vs L2 vs M2 vs M3 vs M5						
Célula	C	84.03 %	91.13 %	93.78 %	85.05 %	**94.21** %
N&Ci	C	92.33 %	90.62 %	**94.21** %	83.50 %	93.61 %

B.2. Prueba de significancia estadística

En esta sección se describen las pruebas de significancia estadística de los resultados obtenidos con el método propuesto, con los reportados en otros trabajos. Esta prueba determina si una diferencia es estadísticamente significativa o no. Una diferencia estadísticamente significativa indica que existen evidencias estadísticas de que hay una diferencia; esto no implica que la diferencia sea grande o importante [27].

La prueba de los signos de Wilcoxon (*Wilcoxon signed rank test*) [54] se usó para este tipo de prueba. Esta prueba ordena por rango los valores de las diferencias en desempeño de dos clasificadores para cada conjunto de datos, ignorando los

signos y compara la suma de los órdenes para las diferencias positivas y negativas [9]. Esta prueba es realizada de la siguiente forma:

Sea d_i la diferencia en desempeño de dos clasificadores sobre el i-ésimo conjunto de datos de N conjuntos de datos. A las diferencias se les asigna un orden de acuerdo con su valor absoluto y en caso de empate se asigna el promedio de éstos. Sea T^+ la suma de todos los órdenes cuya diferencia es positiva y T^- la suma de los órdenes cuya diferencia es negativa. Los órdenes de $d_i = 0$ son divididos a la mitad entre las dos sumas, T^+ y T^-; en caso de existir un número impar de diferencias igual a cero, uno es ignorado.

$$T^+ = \sum_{d_i>0} orden(d_i) + \frac{1}{2}\sum_{d_i=0} orden(d_i) \tag{B.1}$$

$$T^- = \sum_{d_i<0} orden(d_i) + \frac{1}{2}\sum_{d_i=0} orden(d_i) \tag{B.2}$$

Sea T el mínimo de T^+ y T^-, entonces la hipótesis nula es rechazada si el valor de T es menor que un valor crítico, el cual es dependiente de N y el nivel de significancia, α. Los valores críticos, generalmente, son encontrados en una tabla; sin embargo, una aproximación para determinar la significancia de T puede ser calculada como:

$$Z = \frac{|T - \frac{1}{4}N\,(N+1)\,|}{\sqrt{\frac{1}{24}N\,(N+1)\,(2N+1)}} \tag{B.3}$$

Si Z es mayor que un umbral dependiente del nivel de significancia α, entonces, se rechaza la hipótesis nula [57]. En la tabla B.2 se muestran los umbrales para distintos valores de α.

La prueba de Wilcoxon es realizada cuatro veces: la primera para comparar el método propuesto con ensambles contra el de Reta [45]; la segunda y tercera, com-

Tabla B.2: Umbrales para distintos valores de α

α	0.500	0.400	0.300	0.200	0.100	0.050	0.020	0.010	0.002	0.001
Umbral	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291

para, respectivamente, contra PSMS y EPSMS de Escalante [15]; la cuarta, contra el mejor modelo individual obtenido por el método propuesto, esto para comprobar la utilidad del uso de ensambles. A continuación se muestra detalladamente la realización de la prueba de Wilcoxon para el primer caso.

La tabla B.3 muestra los resultados en clasificación, usando el área bajo la curva ROC como métrica, obtenidos por Reta [45] y el método con ensambles; también, se muestra la diferencia en desempeño en cada conjunto de datos y el orden asignado con base en el valor absoluto de la diferencia.

Tabla B.3: Resultados en clasificación por leucemia aguda, diferencia en el desempeño en cada conjunto de datos y el orden asignado

Conjunto de datos		Reta [45]	Propuesta Ensamble	Diferencia	Orden
Región	Características				
ALL vs AML					
Célula	A	89.26	95.12	5.86	20
	B	79.06	83.66	4.60	18
	C	89.92	92.44	2.52	07
N&Ci	A	90.65	95.23	4.58	17
	B	81.27	81.20	−0.07	01
	C	92.17	95.01	2.84	10
L1 vs L2					
Célula	A	81.40	92.46	11.06	27

Tabla B.3: Resultados en clasificación por leucemia aguda, diferencia en el desempeño en cada conjunto de datos y el orden asignado (continuación).

Conjunto de datos		Reta [45]	Propuesta		
Región	Características		Ensamble	Diferencia	Orden
	B	76.08	82.22	6.14	21
	C	82.25	93.55	11.30	28
N&Ci	A	90.69	94.45	3.76	12.5
	B	83.67	87.43	3.76	12.5
	C	88.61	93.00	4.39	15
M2 vs M3 y M5					
Célula	A	80.45	96.32	15.87	32
	B	71.06	82.51	11.45	29
	C	84.12	95.72	11.60	30
N&Ci	A	95.90	95.70	−0.20	03
	B	78.93	82.89	3.96	14
	C	94.68	94.28	−0.40	04
M3 vs M2 y M5					
Célula	A	78.82	96.93	18.11	33
	B	78.67	76.02	−2.65	09
	C	71.01	95.61	24.60	34
N&Ci	A	87.97	96.66	8.69	22
	B	73.91	79.37	5.46	19
	C	89.85	94.33	4.48	16
M5 vs M2 y M3					
Célula	A	86.64	97.66	11.02	26
	B	73.14	76.01	2.87	11

Tabla B.3: Resultados en clasificación por leucemia aguda, diferencia en el desempeño en cada conjunto de datos y el orden asignado (continuación).

Conjunto de datos		Reta [45]	Propuesta		
Región	Características		Ensamble	Diferencia	Orden
	C	84.98	95.03	10.05	24
N&Ci	A	95.52	97.55	2.03	06
	B	73.32	83.22	9.90	23
	C	93.54	93.45	−0.09	02
M2 vs M3 vs M5					
Célula	C	78.66	93.98	15.32	31
N&Ci	C	92.80	95.43	2.63	08
L1 vs L2 vs M2 vs M3 vs M5					
Célula	C	84.03	94.21	10.18	25
N&Ci	C	92.33	93.61	1.28	05

De las ecuaciones (B.1) y (B.2) tenemos que $T^+ = 576$ y $T^- = 19$, haciendo $T = \text{mín}(T^+, T^-)$, entonces $T = 19$. La tabla B.4 muestra los valores críticos para diferentes niveles de significancia α y $N = 34$. Para un nivel de significancia $\alpha = 0.001$ y $N = 34$, T debe ser menor que 111, por tanto la hipótesis nula es rechazada, lo que implica que la diferencia entre los dos métodos es significativa.

La tabla B.5 muestra las diferencias en clasificación del método propuesto con ensambles contra PSMS y EPSM de Escalante y el mejor modelo individual del método propuesto; el orden para cada diferencia es mostrado entre paréntesis.

Tabla B.4: Valores críticos de la distribución de Wilcoxon para $N = 34$

	α							
N	0.500	0.200	0.100	0.050	0.020	0.010	0.005	0.001
34	257	221	200	182	162	148	136	111

Tabla B.5: Diferencias en los resultados de clasificación y orden. El orden es mostrado entre paréntesis

Conjunto de datos		Escalante [15]		Método
Región	Características	PSMS	EPSMS	Caso individual
ALL vs AML				
Célula	A	1.25(09)	−0.02(02.3)	1.22(09)
	B	0.25(04)	0.23(11)	0.60(06)
	C	−0.71(07)	−2.60(31)	−1.24(10)
N&Ci	A	0.07(02)	−0.93(21)	−0.15(02)
	B	1.46(11)	0.88(20)	−2.07(13.5)
	C	−0.65(06)	−1.65(26)	−1.12(08)
L1 vs L2				
Célula	A	4.78(25)	0.22(10)	0.57(05)
	B	−6.32(28)	−1.89(27)	5.53(28)
	C	14.48(33)	1.17(23)	5.21(25)
N&Ci	A	19.66(34)	0.35(13)	2.53(18)
	B	−0.02(01)	0.02(02.3)	5.92(29)
	C	4.94(26)	−0.09(08)	1.97(12)
M2 vs M3 y M5				
Célula	A	4.39(22)	0.54(16)	0.10(01)
	B	9.52(30)	2.18(28)	3.62(23)

Tabla B.5: Diferencias en los resultados de clasificación y orden (continuación)

Conjunto de datos		Escalante [15]		Método
Región	Características	PSMS	EPSMS	Caso individual
	C	−0.21(03)	0.02(02.3)	3.26(21)
N&Ci	A	−1.01(08)	0(01)	3.20(20)
	B	3.83(20)	−0.59(17)	3.52(22)
	C	0.63(05)	−0.08(07)	−0.55(04)
M3 vs M2 y M5				
Célula	A	8.68(29)	4.36(34)	0.27(03)
	B	3.09(19)	−3.84(33)	5.23(26)
	C	2.87(15)	1.46(24)	6.58(30)
N&Ci	A	5.15(27)	0.04(05)	5.24(27)
	B	2.11(13)	2.48(30)	4.91(24)
	C	1.27(10)	1.09(22)	2.43(17)
M5 vs M2 y M3				
Célula	A	10.03(31)	2.38(29)	2.26(16)
	B	14.24(32)	0.23(12)	9.99(32)
	C	2.74(14)	−0.82(19)	1.53(11)
N&Ci	A	4.61(24)	−0.13(09)	0.84(07)
	B	3.98(21)	−2.69(32)	12.62(34)
	C	4.48(23)	0.38(14)	−2.07(13.5)
M2 vs M3 vs M5				
Célula	C	1.61(12)	0.04(06)	3.06(19)
N&Ci	C	3.07(17)	1.49(25)	2.20(15)
L1 vs L2 vs M2 vs M3 vs M5				
Célula	C	3.08(18)	0.43(15)	9.16(31)

Tabla B.5: Diferencias en los resultados de clasificación y orden (continuación)

Conjunto de datos		Escalante [15]		Método
Región	Características	PSMS	EPSMS	Caso individual
N&Ci	C	2.99(16)	−0.60(18)	10.11(33)

Para un valor de $\alpha = 0.001$ y $N = 34$, $T < 111$, de la tabla B.5 se obtienen los siguientes datos:

1. Al comparar con PSMS se obtiene que la suma de los ordenes de las diferencias positivas, T^+, es 542, la suma de los órdenes de las diferencias negativas, T^-, es de 53, por tanto T es 53 y al ser menor que 111 la hipótesis nula se rechaza, indicando que la diferencia entre los dos métodos es significativa.

2. Al comparar los resultados obtenidos con EPSMS se obtiene que T^+ es igual a 343.3, T^- es 250.3 y T es 250.3. Al no ser T menor que 111, entonces la hipótesis nula no es rechazada y, por tanto, la diferencia entre ambos métodos no es significativa.

3. Finalmente, de la comparación contra el mejor individual se obtiene que T^+ es 544, T^- es 51 y T es 51. La hipótesis nula es rechazada, lo que implica que la diferencia entre ambos métodos es significativa.

Bibliografía

[1] Acevedo, J., Maldonado, S., Lafuente, S., Gomez, H., and Gil, P. Model selection for support vector machines using ant colony optimization in an electronic nose application. In *Ant Colony Optimization and Swarm Intelligence*, M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, pp. 468–475. 10.1007/11839088_47.

[2] Adjouadi, M., Ayala, M., Cabrerizo, M., Zong, N., Lizarraga, G., and Possman, M. Classification of leukemia blood samples using neural networks. *Annals of biomedical Engineering 38*, 4 (2009), 1473–1482.

[3] Atanassov, K., and Gargov, G. Interval valued intuitionistic fuzzy sets. *Fuzzy Sets and Systems 31*, 3 (1989), 343–349.

[4] Bandler, W., and Kohout, L. Fuzzy relational products as a tool for analysis and synthesis of the behaviour of complex natural and artificial systems. *Fuzzy sets: theory and applications to policy analysis and information systems* (1980), 341–367.

[5] Boutell, M., Luo, J., Shen, X., and Brown, C. Learning multi-label scene classification* 1. *Pattern Recognition 37*, 9 (2004), 1757–1771.

[6] Breiman, L. Random forest. *Machine learning 45*, 1 (1996), 5–32.

[7] Cover, T., and Hart, P. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on 13*, 1 (1967), 21–27.

[8] Dash, M., and Liu, H. Feature selection for classification. *Intelligent data analysis 1*, 3 (1997), 131–156.

[9] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research 7* (2006), 1–30.

[10] Dietterich, T. Ensemble methods in machine learning. *Multiple classifier systems* (2000), 1–15.

[11] Dreiseitl, S. Training multiclass classifiers by maximizing the volume under the roc surface. In *Computer Aided Systems Theory – EUROCAST 2007*, R. Moreno Díaz, F. Pichler, and A. Quesada Arencibia, Eds., vol. 4739 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, pp. 878–885. 10.1007/978-3-540-75867-9_110.

[12] Drummond, C., and Holte, R. Explicitly representing expected cost: An alternative to roc representation. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (2000), ACM, pp. 198–207.

[13] Duda, R., and Hart, P. Pattern recognition and scene analysis, 1973.

[14] Engelbrecht, A. P. *Computational intelligence: an introduction*, 2 ed. Wiley, 2007.

[15] Escalante, H. J. Acute leukemia classification with ensemble particle swarm model selection. *Comunicación personal.*

[16] Escalante, H. J., Montes, M., and Sucar, L. E. Particle swarm model selection. *The Journal of Machine Learning Research 10* (2009), 405–440.

[17] Escalante, H. J., Montes, M., and Sucar, L. E. Ensemble particle swarm model selection. In *Neural Networks (IJCNN), The 2010 International Joint Conference on* (Julio 2010), IEEE, pp. 1–8. Advances in Neural Networks Research: IJCNN '10, 2010 International Joint Conference on Neural Networks IJCNN '10.

[18] Forster, M. Key Concepts in Model Selection: Performance and Generalizability* 1. *Journal of Mathematical Psychology 44*, 1 (2000), 205–231.

[19] Galindo, M. C. Obtención de características de subtipos de leucemia en imágenes digitales de células sanguíneas para su clasificación. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2008.

[20] Gorissen, D., Dhaene, T., and Turck, F. D. Evolutionary model type selection for global surrogate modeling. *J. Mach. Learn. Res. 10* (2009), 2039–2078.

[21] Guyon, I. A practical guide to model selection.

[22] Guyon, I., Saffari, A., Dror, G., and Cawley, G. Analysis of the ijcnn 2007 agnostic learning vs. prior knowledge challenge. *Neural Networks 21*, 2-3 (2008), 544–550. Advances in Neural Networks Research: IJCNN '07, 2007 International Joint Conference on Neural Networks IJCNN '07.

[23] Hall, M. A. *Correlation-based feature selection for machine learning.* PhD thesis, The University of Waikato, 1999.

[24] Han, J., and Kamber, M. *Data mining: concepts and techniques.* Morgan Kaufmann, 2006.

[25] Hastie, T., Tibshinari, R., and Firedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2001.

[26] Holland, J. *Adaptation in natural and artificial systems.* MIT Press Cambridge, MA, USA, 1992.

[27] Hubbard, R., Armstrong, J., Berger, J., Bradlow, E., Goodman, S., et al. Why we don't really know what "statistical significance" means: a major educational failure. *Journal of Marketing Education 28*, 2 (2006), 114–120.

[28] Janikow, C. Fuzzy decision trees: issues and methods. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 28*, 1 (2002), 1–14.

[29] Japkowicz, N. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI'2000)* (2000), vol. 1, Citeseer, pp. 111–117.

[30] Jarratwisaru, Y., Sukarayodhin, S., Thakaew, S., Kiattisin, S., and Chamnongthai, K. Detection of an acute leukemia using an active contour model. *Proceedings of the 4th International Joint Conference on Computer Science and Software Engineering* (2007), 1–4.

[31] Keller, J. M., Gray, M. R., and Givens, J. A. A fuzzy-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics 15*, 4 (1985), 581.

[32] Koerich, A. L. Improving classification performance using metaclasses. In *IEEE International Conference on Systems, Man and Cybernetics* (Noviembre 2003), vol. 1, pp. 717–722.

[33] Kuncheva, L. *Fuzzy classifier design.* Physica Verlag, 2000.

[34] Kuncheva, L. *Combining pattern classifiers: methods and algorithms.* Wiley-Interscience, 2004.

[35] Liu, C.-L. Partial discriminative training for classification of overlapping classes in document analysis. *International Journal on Document Analysis and Recognition 11* (2008), 53–65. 10.1007/s10032-008-0069-1.

[36] Molina, L. C., Belanche, L., and Nebot., A. Feature selection algorithms: A survey and experimental evaluation. *In other words 4*, R4SE1DTH.

[37] Morales, B. A. Extracción de características en imágenes de células de médula ósea para la clasificación de leucemias agudas. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2007.

[38] Nasir, A. S. A., Mashor, M. Y., and Rosline, H. Detection of acute leukaemia cells using variety of features and neural networks. In *5th Kuala Lumpur International Conference on Biomedical Engineering 2011*, N. A. A. Osman, W. A. B. W. Abas, A. K. A. Wahab, H.-N. Ting, and R. Magjarevic, Eds., vol. 35 of *IFMBE Proceedings*. Springer Berlin Heidelberg, 2011, pp. 40–46.

[39] Oza, N., and Tumer, K. Classifier ensembles: Select real-world applications. *Information Fusion 9*, 1 (2008), 4–20.

[40] Pedrycz, W. Neurocomputations in relational systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 13*, 3 (1991), 289–297.

[41] Pedrycz, W., and Gomide, F. *Fuzzy systems engineering: toward human-centric computing.* Wiley-IEEE Press, 2007.

[42] Pedrycz, W., and Sosnowski, Z. C-fuzzy decision trees. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 35*, 4 (2005), 498–511.

[43] Podolak, I. T. Hierarchical classifier with overlapping class groups. *Expert Systems with Applications 34*, 1 (2008), 673 – 682.

[44] Qin, Z., and Lawry, J. Decision tree learning with fuzzy labels. *Information Sciences 172*, 1-2 (2005), 91–129.

[45] Reta, C. Segmentación y clasificación de células con leucemia a partir de información contextual en imágenes digitales. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2009.

[46] Reta, C., Altamirano, L., Gonzalez, J., Diaz, R., and Guichard, J. Segmentation of Bone Marrow Cell Images for Morphological Classification of Acute Leukemia. In *Florida Artificial Intelligence Research Society Conference* (2010), pp. 86–91.

[47] Reyes, C. *On the design of a fuzzy relational neural network for automatic speech recognition.* PhD thesis, Doctoral Dissertation, The Florida State University, Tallahassee, Fl, 1994.

[48] Robnik-Šikonja, M., and Kononenko, I. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning 53*, 1 (2003), 23–69.

[49] Rokach, L. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis 53*, 12 (2009), 4046–4072.

[50] Szmidt, E., and Kukier, M. Atanassov's intuitionistic fuzzy sets in classification of imbalanced and overlapping classes. In *Intelligent Techniques and Tools for Novel System Architectures*, P. Chountas, I. Petrounias, and J. Kacprzyk, Eds., vol. 109 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, 2008, pp. 455–471. 10.1007/978-3-540-77623-9_26.

[51] University of California, I. Uci machine learning repository, 2011.

[52] Wang, H. *Intelligent Data Analysis: Developing New Methodologies Through Pattern Discovery and Recovery*. IGI Global, 2008.

[53] Webb, A. *Statistical pattern recognition*. A Hodder Arnold Publication, 1999.

[54] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin 1*, 6 (1945), 80–83.

[55] Witten, I., and Frank, E. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005.

[56] Zadeh, L. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353.

[57] Zar, J. *Biostatistical analysis*, vol. 564. Prentice hall Upper Saddle River, NJ, 1999.