Centro de Investigación y Estudios Avanzados
Departamento de Ingeniería Eléctrica
Sección de Computación

Clasificadores Basados en Conjuntos de Representantes

Tesis que para obtener el grado de

Maestro en Ciencias

con especialidad en

Ingeniería Eléctrica

Presenta

Jesús Ariel Carrasco Ochoa

1994

Agradecimientos

A Lucía Angélica De la Vega Doria, por su apoyo durante la realización de este trabajo, durante toda la maestría y durante toda la vida.

A mi madre Guadalupe Ochoa Vda. de Carrasco, por su esfuerzo y apoyo durante toda la vida.

A José Ruiz Shulcloper por haber dirigido el trabajo de tesis y por el apoyo brindado hasta la culminación del mismo.

A Sergio Victor Chapa Vergara y José Oscar Olmedo Aguirre por sus valiosos consejos, los cuales ayudaron a mejorar el resultado de esta tesis.

Al Instituto Nacional de Astrofísica Optica y Electrónica (INAOE), por haberme permitido ingresar en su plan de superación académica y por su apoyo durante los dos años de la maestría.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo económico que me brindó a través de su programa de becas para estudios de posgrado.

Contenido

Introducción

Los especialistas de algunas ciencias poco formalizadas, tales como la Geología, la Medicina, etc, frecuentemente se enfrentan al problema de discriminar objetos de diferentes clases y ocupan para la descripción de los mismos, variables de diferentes tipos, las cuales pueden ser cualitativas o cuantitativas. Las clases entre las que se tiene que discriminar pueden o no ser mutuamente excluyentes, o incluso los objetos pueden pertenecer con diferentes grados a cada una de estas clases, además puede ser que en algunos de los objetos de muestra se desconozcan los valores de uno o más de los atributos que los describen. En adelante, los atributos o características que se utilizan para describir a los objetos serán llamados rasgos.

Por si fuera poco, el proceso que siguen los especialistas, para realizar esta discriminación, normalmente no es muy claro, o peor aún, muchas veces se desconoce la forma de hacerlo. De aquí que con lo único que se cuenta para tratar de encontrar la manera de determinar a qué clase pertenece un cierto objeto, es una muestra de objetos, para los cuales ya se conoce la clase a la que pertenecen.

Partiendo de la suposición de que se tiene una muestra de objetos ya clasificados, es decir, un problema de clasificación supervizada, los cuales en adelante serán llamados problemas de clasificación con aprendizaje, existen varios métodos para realizar la clasificación de nuevos objetos, en particular estaremos interesados en el enfoque lógico combinatorio, dentro del cual existen algunos algoritmos, por ejemplo los algoritmos de votación.

El trabajo de esta tesis se enmarca en la teoría matemática de Reconocimiento de Patrones y sus aplicaciones, dentro del enfoque lógico combinatorio, centrándose en la solución de problemas de clasificación con aprendizaje. Se basa en el método de clasificación propuesto por L. V. Baskakova y Yu. I. Zhuravliov [1].

El método de clasificación usando conjuntos de representantes de Baskakova y Zhuravliov[1], trabaja con una muestra de objetos descritos en términos de rasgos que toman valores en espacios métricos, agrupados en clases no necesariamente disjuntas y con un criterio de semejanza booleano. Este método permite clasificar nuevos objetos en una o más de estas clases, indicando únicamente a cuáles clases pertenece, el objeto en cuestión, y a cuáles no pertenece.

El método propuesto por Baskakova y Zhuravliov[1], se basa en la idea de que algunas combinaciones de valores, para ciertos rasgos, pueden dar información a favor, o en contra, acerca de la pertenencia de un objeto a alguna clase, por lo cual, primero se extraen las combinaciones de valores, para ciertos conjuntos de rasgos, que son capaces de proporcionar información a favor, o en contra, las cuales serán llamadas representantes positivos y negativos respectivamente. Una vez hecho esto, cuando se quiere clasificar un nuevo objeto, se compara contra todos los representantes positivos y negativos, de cada clase, y basándose en la cantidad de información obtenida, a favor y en contra, se decide a cuál o cuáles de las clases pertenece el objeto.

El trabajo de esta tesis consiste en desarrollar el método de clasificación usando conjuntos de representantes, de manera que pueda aplicarse en condiciones menos restrictivas, por ejemplo, permitiendo trabajar con distinta función de semejanza, con clases difusas, etc, lo cual permite realizar una modelación, de problemas de clasificación con aprendizaje, más cercana a la realidad, y de esta manera obtener resultados más confiables para el especialista, puesto que la representación de su problema es más parecida a su modelo de la realidad. Con todo esto se amplía el área de aplicación del método de clasificación usando conjuntos de representantes.

El método de clasificación usando conjuntos de representantes se extiende para que trabaje con rasgos que tomen valores en espacios no restringidos, y empleando distintos tipos de funciones de semejanza, no necesariamente booleanas, especialmente aquellas que toman valores en el intervalo [0,1]. Además se extiende para que trabaje con clases difusas, con esto último se permite que el resultado indique no solamente a cuáles clases pertenece, el objeto a clasificar, sino también el grado con el cual pertenece a cada una de éstas. Con todo esto, el especialista tiene que distorsionar en menor grado, su forma de ver el problema que quiere solucionar, durante el proceso de modelación matemática del mismo, y de esta manera se pueden obtener mejores resultados, con respecto a la calidad de la clasificación, los cuales a la vez son más confiables.

Adicionalmente al desarrollo conceptual del modelo de clasificación usando conjuntos de representantes, se realizó una implementación computacional del método, incluyendo las características nuevas, las cuales permiten tener un sistema automatizado más versátil, para la solución de problemas de clasificación con aprendizaje.

En el capítulo 1 se dan los conceptos básicos para entender la manera de atacar este tipo de problemas de clasificación en el enfoque lógico combinatorio, en el capítulo 2 se decribe el método de clasificación usando conjuntos de representantes, en el capítulo 3 se describe la manera de estimar los parámetros usados por el modelo, en el capítulo 4 se dan algunas extensiones al modelo, en los capítulos 5 y 6 se describe como estimar los parámetros del método extendido, finalmente en el capítulo 7 se describe brevemente la implementación computacional del modelo.

Capítulo 1

Reconocimiento de Patrones

En este capítulo se da una introducción a la teoría matemática de Reconocimiento de Patrones, dentro del enfoque lógico combinatorio, con el fin de que se puedan entender mejor los siguientes capítulos.

Inicialmente, en la sección 1.1. daremos una breve explicación de la modelación matemática de problemas de Reconocimiento de Patrones, posteriormente en la sección 1.2. se formalizan algunos conceptos de Reconocimiento de Patrones, para poder llegar a definir lo que se entiende por un problema de clasificación con aprendizaje, que es el tipo de problemas que se solucionan en esta tesis, en la sección 1.3. se definirán los conceptos de criterio de comparación y de función de semejanza, sobre los cuales están basados los algoritmos de solución de problemas de clasificación. En la sección 1.4. se da una introducción a la teoría de testores, para después, en la sección 1.5. utilizar estos conceptos para definir lo que se entiende por peso informacional de los rasgos y de los objetos. En la sección 1.6. se explica un modelo de algoritmos para la solución de problemas de clasificación con aprendizaje, dentro del cual se usan los conceptos definidos en las secciones anteriores. Finalmente en la sección 1.7. se da una breve introducción a la teoría de subconjuntos difusos.

Todos los conceptos de este capítulo fueron extraídos del trabajo de J. R. Shulcloper[2], excepto la sección 1.7. que se tomó del trabajo de Zadeh[15].

1.1. Modelación Matemática

Para la aplicación de modelos matemáticos de Reconocimiento de Patrones, en ciencias tales como la geología, geofísica, medicina, psicología, criminalística, etc., las cuales en general serán denominadas ciencias poco formalizadas, se necesita de un proceso de modelación, en el cual intervienen especialistas del área en cuestión, a los que llamaremos especialistas no matemáticos, y especialistas matemáticos, los cuales tienen que interactuar para obtener un modelo de la realidad, dentro del campo de estudio de estas ciencias, el que posteriormente será formalizado para crear un modelo matemático, el cual es utilizado para solucionar el problema.

Durante este proceso de modelación resulta importante que los especialistas matemáticos conozcan la esencia del problema que se está tratando de solucionar, así como las limitaciones de las herramientas de que disponen, pero también resulta muy importante que el especialista no matemático entienda la forma en que su problema será solucionado, es decir, el proceso al cual serán sometidos sus datos, así como la manera en que serán interpretados los resultados, de modo que sea capaz de confiar en los sistemas automatizados que se obtengan, y por lo tanto utilizarlos, más aún debe ser capaz de cuestionar su modelo de la realidad, en caso de que los resultados no sean los esperados.

La figura 1.1. muestra un esquema del proceso de modelación matemática, en el cual la retroalimentación permite no sólo modificar el modelo matemático del problema, sino que tambien el modelo que el especialista no matemático tiene de la realidad.

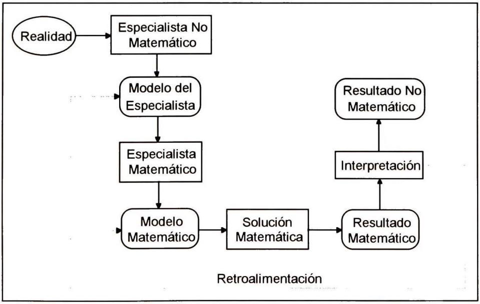


Figura 1.1. Equema General del Proceso de Modelación Matemática

1.2. Planteamiento Formal

En esta sección se dará el planteamiento formal de los problemas de clasificación dentro del enfoque lógico combinatorio.

Se considera un universo M de objetos y un cubrimiento finito $K_1,\ldots,K_\ell$, de M, por subconjuntos propios, es decir, $K_i \subset M$, para $i=1,\ldots,\ell$, cada uno de estos subconjuntos será llamado una clase. Sean $x_1,\ldots,x_n$ los rasgos, en términos de los cuales se describen los objetos de M. Cada rasgo x_i tiene asociado un conjunto M_i que denominaremos conjunto de valores admisibles del rasgo. En dependencia de su conjunto de valores asociado podemos tener rasgos de diferentes tipos, por ejemplo:

Booleanos	si $M_i = \{0,1\}$
K-valentes	si $M_i = \{0,1,\ldots,k\text{-}1\}$; $k > 2$
Reales	si $M_i \subseteq \mathbb{R}$

El conjunto de valores admisibles para un rasgo será complementado con el símbolo "*", el cual denotará ausencia de información, y será utilizado cuando se desconozca el valor que toma un objeto en algún rasgo.

Definición 1.2.1.

Una **descripción estandar de un objeto O**, será un vector n-dimensional $I(O) = (x_1(O),\ldots,x_n(O))$, donde $x_i(O) \in M_i$, $i=1,\ldots,n$, es el valor del rasgo x_i en el objeto O.

Definición 1.2.2.

Si $x_i(O) \neq *$ para toda $i=1,\ldots,n$, diremos que I(O) es una **descripción completa** de O en términos de $x_1,\ldots,x_n$.

Definición 1.2.3.

Llamaremos **vector informacional**, de un objeto O, a un vector ℓ-dimensional $\alpha(O)=(\alpha_1(O),\ldots,\alpha_\ell(O))$ con $\alpha_i(O) \in [0,1] \cup \{*\}$, donde $\alpha_i(O)$ es la pertenencia de O a K_i, y $\alpha_i(O)=*$ significa que se desconoce la pertenencia de O a K_i, $i=1,\ldots,\ell$.

Nota: Si $\alpha_i(O) \in \{0,1\} \cup \{*\}$ diremos que $\alpha(O)$ es un vector informacional booleano, en el cual se admite la ausencia de información.

Definición 1.2.4.

Si $\alpha_i(O) \neq *$ para toda $i=1,\ldots,\ell$, diremos que $\alpha(O)$ es un **vector informacional completo**.

Definición 1.2.5.

Si $\alpha_i(O)$ concuerda con la realidad para toda $i=1,\ldots,\ell$, diremos que $\alpha(O)$ es un **vector informacional correcto**.

Definición 1.2.6.

Por **información estandar de las clases $K_1,\ldots,K_\ell$**, la cual denotaremos por $I(K_1,\ldots,K_\ell)$, entenderemos el conjunto de descripciones estandar de los objetos de estas clases, junto con el vector informacional de cada uno, es decir:

$$I(K_1,\ldots,K_\ell) = \{ I(O_1),\alpha(O_1), \ldots ,I(O_m),\alpha(O_m) \}$$

Definición 1.2.7.

Por **información estandar correcta de las clases $K_1,\ldots,K_\ell$**, entenderemos una información estandar en la cual todos sus vectores informacionales son correctos.

Definición 1.2.8.

Sea M un universo de objetos, descritos en términos de n rasgos $x_1,\ldots,x_n$ y $K_1,\ldots,K_\ell$ un cubrimiento finito de subconjuntos propios de M; sea dada $I(K_1,\ldots,K_\ell)$ una información estandar de las clases, que denominaremos muestra o matriz de aprendizaje; y sea dada $I(O)$ la descripción estandar de un objeto. Un **problema de clasificación** consiste en encontrar un algoritmo A tal que:

$$A(I(K_1,\ldots,K_\ell),I(O)) = \alpha(O)$$

es decir, un algoritmo que tome la información estandar de las clases $K_1,\ldots,K_l$, y la descripción estandar de un objeto O, dando como resultado el vector informacional del objeto O.

En general consideraremos tres tipos de problemas de clasificación, a saber:

Definición 1.2.9.

Un problema en el cual la información estandar de las clases contiene al menos la descripción estandar de un objeto de cada clase, será llamado un problema de **clasificación con aprendizaje**.

Definición 1.2.10.

Un problema en el cual la información estandar de las clases es tal que por lo menos para una clase contiene al menos la descripción estandar de un objeto y por lo menos para una clase no contiene la descripción estandar de ningún objeto, será llamado un problema de **clasificación con aprendizaje parcial**.

Definición 1.2.11.

Un problema en el cual se desconocen las clases en las cuales se agrupan los objetos, y por lo tanto la información estandar de las clases contiene únicamente la descripción estandar de los objetos y el problema en sí consiste en agrupar dichos objetos en clases, será llamado un problema de **clasificación sin aprendizaje**.

En particular estaremos interesados en la solución de problemas de clasificación con aprendizaje, también llamados de clasificación supervisada, que es el tipo de problemas que se resuelve en esta tesis. En las secciones siguientes se darán algunas definiciones adicionales y se explicará un algoritmo para la solución de este tipo de problemas.

1.3. Función de Semejanza y Criterio de Comparación

En esta sección definiremos lo que se entiende por un criterio de comparación, de semejanza y de diferencia, entre valores de rasgos, por una función de semejanza y por una función de semejanza parcial. Además se darán algunos ejemplos.

Definición 1.3.1.

Un **criterio de comparación de semejanza**, entre valores de un rasgo x_i, es una función:

$$CS_i : M_i \times M_i \rightarrow D_i$$

con

M_i el conjunto de valores admisibles para x_i
D_i algún conjunto totalmente ordenado

que cumple:

$$C_i(v_1,v_2) > C_i(v_1,v_3) \Rightarrow v_1 \text{ es más parecido a } v_2 \text{ que a } v_3$$

En general D_i puede ser cualquier conjunto totalmente ordenado, pero estaremos especialmente interesados en criterios de comparación de semejanza en los cuales $D_i \subseteq [0,1]$.

Ejemplo 1.3.1. Criterios de Comparación de Semejanza.

Supongamos que M_i es cualquier conjunto de valores admisibles para x_i, entonces podemos definir:

$$CS_i(v_1,v_2) = \begin{cases} 1 & \text{si } v_1 = v_2 \\ 0 & \text{en otro caso} \end{cases}$$

Supongamos que $M_i = \{ 0, 1, ..., k\text{-}1 \}$, entonces podemos definir:

Sean $a_1,...,a_r \in M_i$, con $a_p \neq 0,k\text{-}1$, para p=1,...,r, entonces podemos partir M_i en r+1 intervalos $[a_p,a_{p+1})$ para p=0,...,r considerando $a_0 = 0$ y $a_{r+1} = k\text{-}1$.

$$CS_i(v_1,v_2) = \begin{cases} 1 & \text{si } v_1, v_2 \in [a_p, a_{p+1}) \text{ para algún } p = 0,\dots,r \\ 0 & \text{en otro caso} \end{cases}$$

Supongamos que $M_i = [a,b]$, un intervalo real, entonces podemos definir:

Sea $\varepsilon_i \in \mathbb{R}$

$$CS_i(v_1,v_2) = \begin{cases} 1 & \text{si } |v_1 - v_2| < \varepsilon_i \\ 0 & \text{en otro caso} \end{cases}$$

Para definir lo que entenderemos por una función de semejanza entre objetos, supondremos que los objetos están descritos en términos de n rasgos, $R=\{x_1,\ldots,x_n\}$.

Definición 1.3.2.

Llamaremos **espacio de representación**, el cual denotaremos por E, al producto cartesiano de los conjuntos de valores admisibles de los rasgos, es decir $E = M_1 \times \ldots \times M_n$.

Definición 1.3.3.

Una **función de semejanza**, entre objetos es una función:

$$\beta : E \times E \to D$$

con

E	el espacio de representación de los objetos
D	algún conjunto totalmente ordenado

que cumple:

$$\beta(O_1,O_2) > \beta(O_1,O_3) \Rightarrow O_1 \text{ es más semejante a } O_2 \text{ que a } O_3$$

En general D puede ser cualquier conjunto totalmente ordenado, pero estaremos especialmente interesados en funciones de semejanza para las cuales $D \subseteq [0,1]$.

Ejemplo 1.3.2. Funciones de Semejanza.

Supongamos que tenemos un criterio de comparación de semejanza booleano CS_i para cada rasgo x_i, i=1,...,n.

$$\beta(O_1,O_2) = \begin{cases} 1 & \text{si } CS_i(x_i(O_1),x_i(O_2)) = 1 \quad \forall\, i = 1,\dots,n \\ 0 & \text{en otro caso} \end{cases}$$

Sea $\varepsilon \in \{0,1,\dots,n\}$

$$\beta(O_1,O_2) = \begin{cases} 1 & \text{si } \left|\{\, i \mid CS_i(x_i(O_1),x_i(O_2)) = 0 \,\}\right| \le \varepsilon \\ 0 & \text{en otro caso} \end{cases}$$

Sea $\lambda \in [0,1]$

$$\beta(O_1,O_2) = \begin{cases} 1 & \text{si } \left(\left|\{\, i \mid CS_i(x_i(O_1),x_i(O_2)) = 1 \,\}\right|/n\right) \ge \lambda \\ 0 & \text{en otro caso} \end{cases}$$

Definición 1.3.4.

Supongamos que $R=\{x_1,\dots,x_n\}$ y sea $\{\Omega\}=\{\Omega_i \mid \Omega_i \subseteq R\}$. Una **función de semejanza parcial**, entre partes de objetos es una función:

$$\beta : E \times E \times \{\Omega\} \to D$$

con

E el espacio de representación de los objetos
D algún conjunto totalmente ordenado

en la cual dos objetos, O_1 y O_2, se comparan atendiendo únicamente a los rasgos especificados por el tercer parámetro, y que cumple:

$\beta(O_1,O_2,\Omega_1) > \beta(O_1,O_3,\Omega_1) \Rightarrow$ O_1 es más semejante a O_2 que a O_3 en los rasgos de Ω_1

1.4. Introducción a la Teoría de Testores

En esta sección definiremos lo que se entiende por un testor y por un testor típico, conceptos que serán, entre otras cosas, utilizados para estimar la relevancia informacional de los rasgos.

Consideremos una matriz MA de descripciones de objetos $O_1,\dots,O_m$, en términos de rasgos $R=\{x_1,\dots,x_n\}$, divididos en ℓ clases disjuntas, por ejemplo:

MA		x_1	...	x_n
K_1	O_1	$x_1(O_1)$	...	$x_n(O_1)$
	:			
	O_{m_1}	$x_1(O_{m_1})$	...	$x_n(O_{m_1})$
:	:	:		:
K_ℓ	$O_{m_{\ell-1}+1}$	$x_1(O_{m_{\ell-1}+1})$	...	$x_n(O_{m_{\ell-1}+1})$
	:			
	O_m	$x_1(O_m)$	...	$x_n(O_m)$

donde

$$K_i \cap K_j = \varnothing \quad \text{si} \quad i \neq j$$

Definición 1.4.1.

El subconjunto de rasgos T, $T \subseteq R$, es un **testor** de MA si y sólo si al eliminar de M todas las columnas excepto las correspondientes a los rasgos de T, ocurre que no hay descripciones iguales en clases distintas, es decir:

$$K_i\big|_T \cap K_j\big|_T = \varnothing \quad \text{si} \quad i \neq j$$

Esta definición nos indica que un testor es un subconjunto de rasgos que continúa diferenciando totalmente a todas las clases.

Definición 1.4.2.

El subconjunto de rasgos T, $T \subseteq R$, es un **testor típico** de MA si y sólo si T es un testor y no existe T' testor, tal que $T' \subset T$.

Esta definición nos indica que un testor típico es un testor en el cual todos los rasgos son indispensables, es decir, si eliminamos cualquiera de ellos el conjunto resultante no es un testor.

1.5. Relevancia Informacional

En esta sección definiremos una manera de calcular la relevancia informacional de los rasgos y de los objetos, la cual estará basada en el concepto de testor típico.

Dentro de un problema de clasificación es posible que no todos los rasgos que se ocupan para describir a los objetos tengan la misma importancia, para la separación de las clases, por lo cual es necesario encontrar una manera de estimar la importancia de cada rasgo. Puesto que los testores típicos son conjuntos totalmente diferenciantes irreducibles, es razonable pensar que un rasgo es más importante si interviene en más testores típicos, utilizando esta idea podemos dar la siguiente definición de peso informacional de un rasgo.

Definición 1.5.1.

El **peso informacional del rasgo x_i**, el cual denotaremos por $P(x_i)$, está definido como:

$$P(x_i) = \frac{|T(x_i)|}{|T|}$$

donde

$T(x_i)$ es el conjunto de todos los testores típicos que contienen a x_i.

T es el conjunto de todos los testores típicos.

En esta definición sólo se considera la cantidad de testores típicos en los que interviene el rasgo, pero resulta razonable pensar que si un rasgo aparece en testores típicos más pequeños significa que necesita menos ayuda, para diferenciar las clases, por lo cual debe ser más importante que uno que aparece en testores típicos con más rasgos. Aplicando esta idea podemos dar una definición en la cual intervengan la cantidad de testores típicos, en los que aparece el rasgo, así como la longitud de los mismos.

Definición 1.5.2.

Sean α y β dos números reales tales que $\alpha,\beta \geq 0$ y $\alpha+\beta=1$, Entonces el **peso informacional del rasgo x_i**, el cual denotaremos por $P(x_i)$, está definido como:

$$P(x_i) = \alpha F(x_i) + \beta H(x_i)$$

donde

$$F(x_i) = \frac{|T(x_i)|}{|T|}$$

$$H(x_i) = \frac{|\,T(x_i)\,|}{\sum\limits_{t\in T(x_i)} |\,t\,|}$$

con

$T(x_i)$ es el conjunto de todos los testores típicos que contienen a x_i.

T es el conjunto de todos los testores típicos.

En esta definición, α y β determinan cuándo la cantidad de testores típicos es más importante que la longitud de los mismos. Claramente si tomamos α=1 y β=0, esta definición coincide con la anterior.

A continuación daremos una definición de peso informacional de un objeto, en la cual intevienen los pesos informacionales de los rasgos que lo describen, así como los valores que toma en cada uno de estos.

Definición 1.5.3.

El **peso informacional del objeto O_i** en la clase K_j, el cual denotaremos por $P_j(O_i)$, está definido como:

$$P_j(O_i) = \frac{1}{\rho}\sum_{k=1}^{n} a_k^j(O_i)P(x_k)$$

donde

$$\rho = \sum_{k=1}^{n} P(x_k)$$

con

$$a_k^j(O_i) = \frac{\#\text{ de objetos en } K_j \text{ que coinciden con } O_i \text{ en } x_k}{|K_j|}$$

$P(x_k)$ = Peso informacional del rasgo x_k.

Estos valores de relevancia informacional, de rasgos y objetos, se utilizan en diversas etapas del proceso de reconocimiento de patrones, en nuestro caso se ocupan para ponderar las comparaciones en los algoritmos de clasificación.

1.6. Algoritmos de Votación

En esta sección se describe la familia de algoritmos de votación, con los cuales se pueden resolver problemas de clasificación con aprendizaje, ver definición 1.2.9. Esta familia de algoritmos está determinada por una serie de parámetros, los cuales al variar forman instancias del modelo.

Inicialmente definiremos algunos conceptos que se ocupan en la descripción del modelo de algoritmos de votación.

Definición 1.6.1.

Un **sistema de conjuntos de apoyo**, el cual denotaremos por $\{\Omega\}$, es un conjunto de subconjuntos de rasgos, el cual servirá para indicar qué partes de los objetos serán comparadas. Cada $\Omega \in \{\Omega\}$ es llamado un conjunto de apoyo.

Definición 1.6.2.

Sea $\Omega \subseteq R$, un subconjunto del conjunto de rasgos, la **ω-parte** de un objeto O, correspondiente a Ω, la cual denotaremos por $\omega I(O)$ o simplemente por ωO, será la subdescripción de O, atendiendo únicamente a los rasgos de Ω.

El modelo de algoritmos de votación se basa en tres conceptos fundamentales que son:

Analogía. Se tiene una función de semejanza que se supone refleja la manera en que se hace la analogía, entre objetos, en el problema real.

Precedencia parcial. Las comparaciones no se hacen entre descripciones completas de objetos, sino entre subdescripciones previamente seleccionadas.

Frecuencia. En los algoritmos de votación se aplica la idea de que un objeto estará en una clase, si se parece a suficientes objetos de la misma.

Cada algoritmo de votación está determinado por 6 parámetros que son:

Ω_A Sistema de Conjuntos de Apoyo

El sistema de conjuntos de apoyo determinará las partes que se compararán durante el desarrollo del algoritmo. Como sistema de conjuntos de apoyo puede utilizarse cualquier subconjunto del conjunto potencia del conjunto de rasgos, por ejemplo, todos los subconjuntos de cardinal fijo k, el conjunto de todos los testores típicos, etc.

β Función de Semejanza

Función de semejanza, la cual determinará la forma en que se compararán los objetos, en sus diferentes partes. Esta función se eligirá de modo que refleje la manera en que se hace la análogía, entre objetos, en el problema real.

f Función de Evaluación por Fila para un Conjunto de Apoyo Fijo

Esta función nos determina qué información aporta el parecido del objeto a clasificar con cada uno de los objetos de la muestra de aprendizaje. Al resultado de esta función se le denomina la votación dada por cada fila, al objeto a clasificar, respecto a un conjunto de apoyo dado.

Dentro de esta función se puede considerar la relevancia informacional de la fila evaluada, así como de los rasgos que intervienen en el conjunto de apoyo considerado. Algunas funciones de evaluación por fila pueden ser:

$$f(O_i,O,\Omega) = \gamma(O_i)\ \beta(\omega O_i,\omega O)$$
$$f(O_i,O,\Omega) = \gamma(O_i)\ P(\Omega)\ \beta(\omega O_i,\omega O)$$

donde

$\gamma(O_i)$ el peso informacional del objeto O_i

$$P(\Omega) = \sum_{x_i \in \Omega} P(x_i)$$

φ Función de Evaluación por Clase para un Conjunto de Apoyo Fijo

Esta función resume la evaluación total por fila para el objeto a clasificar, dentro de cada clase, para un cierto conjunto de apoyo. Al resultado de esta función se le denomina la votación dada por la clase, al objeto a clasificar, respecto a un conjunto de apoyo dado. Algunas funciones de evaluación por clase, para un conjunto de apoyo, pueden ser:

$$\varphi(j,O,\Omega) = \frac{1}{|K_j|} \sum_{O_i \in K_j} f(O_i,O,\Omega)$$

$$\varphi(j,O,\Omega) = \begin{cases} 1 & \text{si } \frac{1}{|K_j|} \sum_{O_i \in K_j} f(O_i,O,\Omega) \geq \lambda\ ;\ \lambda > 0 \\ 0 & \text{en otro caso} \end{cases}$$

ψ Función de Evaluación por Clase para todo el Sistema de Conjuntos de Apoyo

Esta función concentra la evaluación del objeto en cada clase, considerando su evaluación para todos los conjuntos de apoyo. Al resultado de esta función se le denomina la votación dada por la clase, al objeto a clasificar, respecto a todo el sistema de conjuntos de apoyo. Algunas funciones de evaluación por clase, para el sistema de conjuntos de apoyo completo, pueden ser:

$$\psi(j,O) = \frac{1}{|\{\Omega\}|} \sum_{\Omega \in \{\Omega\}} \varphi(j,O,\Omega)$$

$$\psi(j,O) = \begin{cases} 1 & \text{si } \frac{1}{|\{\Omega\}|} \sum_{\Omega \in \{\Omega\}} \varphi(j,O,\Omega) \geq \lambda \ ; \ \lambda > 0 \\ 0 & \text{en otro caso} \end{cases}$$

r_A Regla de Solución

Esta función toma todas las evaluaciones totales por clase y forma el vector informacional del objeto a clasificar, es decir, decide a cuál o cuáles clases pertenece dicho objeto. La regla de solución en general tiene la siguiente forma:

$$r_A(\psi(1,O),\ldots,\psi(\ell,O)) = (\alpha_1(O),\ldots,\alpha_n(O))$$

Algunas maneras de calcular los valores de $\alpha_i(O)$ pueden ser:

$$\alpha_i(O) = \begin{cases} 1 & \text{si } \psi(i,O) > \psi(j,O) \ \ \forall\, j \neq i \\ 0 & \text{en otro caso} \end{cases}$$

$$\alpha_i(O) = \begin{cases} 1 & \text{si } \psi(i,O) - r_1 > \psi(j,O) \ \ \forall\, j \neq i \\ 0 & \text{en otro caso} \end{cases}$$

$$\alpha_i(O) = \begin{cases} 1 & \text{si } \psi(i,O) - r_1 > \psi(j,O) \ \ \forall\, j \neq i \\ & \text{y } \ \psi(i,O) \Big/ \sum_{j=1}^{\ell} \psi(j,O) > r_2 \\ 0 & \text{en otro caso} \end{cases}$$

Con estos parámetros los algoritmos de votación se desarrollan en 5 etapas que son las siguientes:

1) Determinación de los parámetros del modelo.
2) Evaluación por fila, del objeto a clasificar, para cada uno de los conjuntos de apoyo.
3) Evaluación por clase, del objeto a clasificar, para cada uno de los conjuntos de apoyo.
4) Evaluación por clase, del objeto a clasificar, para todo el sistema de conjuntos de apoyo.
5) Aplicación de la regla de solución, para determinar el vector informacional del objeto a clasificar.

Ejemplo 1.6.1.

Supongamos que tenemos la siguiente matriz de aprendizaje:

	MA	x_1	x_2	x_3	x_4	x_5
K_1	O_1	1	1	0	1	1
	O_2	0	1	0	1	1
	O_3	1	0	1	0	1
K_2	O_4	1	1	1	1	1
	O_5	0	0	0	1	1

Esta matriz tiene un sólo testor típico que es $\{x_2,x_3\}$, por lo tanto usando la definición 1.5.1, los pesos informacionales de los rasgos son:

$P(x_1) = 0$
$P(x_2) = 1$
$P(x_3) = 1$
$P(x_4) = 0$
$P(x_5) = 0$

Usando la definición 1.5.3, los pesos informacionales de los objetos son:

$\gamma(O_1) = 2/3$
$\gamma(O_2) = 2/3$
$\gamma(O_3) = 1/3$
$\gamma(O_4) = 1/2$
$\gamma(O_5) = 1/2$

Consideremos

Ω_A Conjunto de todos los testores típicos, es decir, $\{ \Omega_1=\{x_2,x_3\} \}$
β Booleana, coincidencia rasgo a rasgo, es decir, igualdad.

$$f(O_i,O,\Omega) = \gamma(O_i)\,P(\Omega)\,\beta(\omega O_i,\omega O)$$

donde

$\gamma(O_i)$ el peso informacional del objeto O_i

$$P(\Omega) = \sum_{x_i \in \Omega} P(x_i)$$

$$\varphi(j,O,\Omega) = \frac{1}{|K_j|} \sum_{O_i \in K_j} f(O_i,O,\Omega)$$

$$\psi(j,O) = \frac{1}{|\{\Omega\}|} \sum_{\Omega \in \{\Omega\}} \varphi(j,O,\Omega)$$

$$\alpha_i(O) = \begin{cases} 1 & \text{si } \psi(i,O) > \psi(j,O) \;\; \forall\, j \neq i \\ 0 & \text{en otro caso} \end{cases}$$

Si el objeto a clasificar es O=(0 1 1 0 0), entonces la evaluación por fila es:

como $P(\Omega_1) = 1$ entonces:

$$f(O_1,O,\Omega_1) = 0$$
$$f(O_2,O,\Omega_1) = 0$$
$$f(O_3,O,\Omega_1) = 0$$
$$f(O_4,O,\Omega_1) = 1/2$$
$$f(O_5,O,\Omega_1) = 0$$

Por lo tanto la evaluación por clase para el conjunto de apoyo Ω_1 es:

$$\varphi(1,O,\Omega_1) = 0$$
$$\varphi(2,O,\Omega_1) = 1/4$$

Y la evaluación por clase para el sistema de conjuntos de apoyo es:

$$\psi(1,O) = 0$$
$$\psi(2,O) = 1/4$$

y finalmente aplicando la regla de solución tenemos:

$$r_A(0,1/4) = \alpha(O) = (\,0\,,\,1\,)$$

De aquí que el objeto O=(0 1 1 0 0) se clasifica en K_2 y no en K_1.

1.7. Subconjuntos Difusos

En esta sección daremos los conceptos básicos de la teoría de subconjuntos difusos.

Un conjunto difuso A es un conjunto en el cual cada elemento del universo puede tener distinto grado de pertenencia, es decir, cada elemento del universo no esta restringido a pertenecer, o no pertenecer, al conjunto A, sino que puede pertenecer con mayor o menor grado. A continuación daremos una definición formal de lo que entenderemos por un conjunto difuso.

Definición 1.7.1.

Sea U el conjunto univero, un **conjunto difuso** A estará definido por una función $\mu_A: U \rightarrow [0,1]$, que asocia a cada elemento de U un grado de pertenencia al conjunto A.

Para denotar un conjunto difuso utilizaremos pares ordenados de la forma $(x,\mu_A(x))$ donde $x \in U$ y $\mu_A(x)$ es el grado de pertenencia de x a A, escribiendo únicamente aquellos elementos para los cuales $\mu_A(x) \neq 0$. Otra manera comunmente usada para denotar conjuntos difusos es escribiendo $x/\mu_A(x)$ en lugar de la notación de pares ordenados. Por ejemplo:

$$A = \{ (x,0.5), (y,0.3), (z,1) \}$$

o bien:

$$A = \{ x/0.5, y/0.3, z/1 \}$$

Cuando $\mu_A(x)$ es más cercano a 1 decimos que x pertenece más a A, y cuando $\mu_A(x)$ es más cercano a 0 decimos que x pertenece menos a A. Además si tenemos que $\mu_A(x) > \mu_A(y)$ diremos que x pertenece mas a A que y.

La siguiente definición nos dice la manera en que se definen la operaciones de unión, intersección y complemento, para conjuntos difusos.

Definición 1.7.2.

Sean A y B dos conjuntos difusos, entonces las funciones de asociación de pertenencia para las **operaciones de conjuntos difusos** quedan definidas como:

$$\mu_{A \cup B}(x) = \max (\mu_A(x), \mu_B(x))$$
$$\mu_{A \cap B}(x) = \min (\mu_A(x), \mu_B(x))$$
$$\mu_{A^c}(x) = 1 - \mu_A(x)$$

Por ejemplo sean:

$$U = \{ a, b, c, d, e \}$$

y

$$A = \{ a/0.4, b/0.5, d/1 \}$$
$$B = \{ a/0.7, b/0.3, c/1 \}$$

entonces

$$A \cup B = \{ a/0.7, b/0.5, c/1, d/1 \}$$
$$A \cap B = \{ a/0.4, b/0.3 \}$$
$$A^c = \{ a/0.6, b/0.5, c/1, e/1 \}$$
$$B^c = \{ a/0.3, b/0.7, d/1, e/1 \}$$

Como se ha visto, un conjunto difuso está completamente determinado por su función de pertenencia, la cual asigna un valor de pertenencia al conjunto a cada uno de los elementos del universo. Existen varios métodos para estimar las funciones de pertenencia, la mayoría de los cuales están basados en el uso de datos estadísticos, en el capítulo 4 veremos un método alternativo para calcular las pertencias de objetos a clases difusas.

1.8. Conclusión

En este capítulo hemos hecho una revisión de los conceptos de reconocimiento de patrones, dentro del enfoque lógico combinatorio, inicialmente se explicó brevemente la modelación matemática de problemas, después se definieron algunos conceptos de reconocimiento de patrones, los cuales nos permitieron definir lo que se entiende por un problema de clasificación con aprendizaje, que es el tipo de problemas que se resuelve en esta tesis.

Adicionalmente se definieron los conceptos de criterio de comparación, de semejanza y de diferencia, y de función de semejanza, los cuales se utilizan para la definición de los algoritmos de solución de problemas de clasificación.

También se dió una introducción a la teoría de testores, definiendo los conceptos de testor y testor típico. Posteriormente, usando el concepto de testor típico, se definió lo que se entiende por peso informacional de rasgos y objetos, los cuales se usan para ponderar las comparaciones, entre objetos, en los algoritmos de clasificación.

Como ejemplo, de un algoritmo para la solución de problemas de clasificación con aprendizaje, se describió el modelo de algoritmos de votación, usado en el sistema PROGNOSIS[3], junto con un ejemplo, para entender mejor el funcionamiento de este tipo de algoritmos.

Finalmente se dió una breve introducción a la teoría de subconjuntos difusos, con el objeto de explicar los conceptos de conjunto difuso y función de pertenencia.

Capítulo 2

Clasificación Usando Conjuntos de Representantes

En este capítulo se describirá el método de clasificación usando conjuntos de representantes, en su forma original, y se harán notar sus restricciones y limitaciones. Todas las definiciones de este capítulo fueron tomadas del trabajo de Baskakova[1].

2.1. Definición

Se considera un problema de clasificación con aprendizaje, con ℓ clases no necesariamente disjuntas, y una muestra de m objetos descritos en términos de n rasgos, que toman valores en espacios métricos, adicionalmente se tiene el vector informacional de cada objeto, este vector nos indica a cuáles clases pertenece dicho objeto. La información de entrada para el algoritmo estará contenida en una matriz, la cual será llamada, en adelante, matriz de aprendizaje, y tiene la siguiente forma:

MA	x_1	...	x_n	α_1	...	α_ℓ
O_1	$x_1(O_1)$	...	$x_n(O_1)$	$\alpha_1(O_1)$	...	$\alpha_\ell(O_1)$
$\vdots$						
O_m	$x_1(O_m)$	...	$x_n(O_m)$	$\alpha_1(O_m)$	...	$\alpha_\ell(O_m)$

donde

$$x_i(O_j)\text{ en un espacio métrico}$$

$$\alpha_i(O_j) = \begin{cases} 1 & \text{si } O_j \in K_i \\ 0 & \text{si } O_j \notin K_i \end{cases}$$

Definición 2.1.

Un **sistema de conjuntos de apoyo** para una clase K_j, el cual denotaremos por $\{\Omega\}_j$, es un conjunto de subconjuntos de rasgos, el cual servirá para indicar qué partes de los objetos serán comparadas al trabajar con la clase K_j. Cada $\Omega \in \{\Omega\}_j$ es llamado un conjunto de apoyo.

El presente algoritmo pertenece al tipo de algoritmos basados en sistemas de conjuntos de apoyo. Puede utilizarse cualquier sistema de conjuntos de apoyo, no necesariamente el mismo para cada clase. Por ejemplo, todos los conjuntos de cardinal fijo k, o el conjunto de todos los testores típicos. El conjunto de los rasgos que describen a los objetos lo denotaremos por R.

Definición 2.2.

Sea $\Omega \subseteq R$, un subconjunto del conjunto de rasgos, la **ω-parte** de un objeto O correspondiente a Ω, la cual denotaremos por ωO, será la subdescripción de O, atendiendo únicamente a los rasgos de Ω.

Definición 2.3.

El **complemento de una clase** Kj, el cual denotaremos por CKj, está definido como:

$$CK_j = \{ O_i \mid \alpha_j(O_i) = 0 \}$$

2.1.1. Caso Booleano

En esta sección se describirá el método de clasificación usando conjuntos de representantes, en el caso en que todos los rasgos son booleanos.

2.1.1.1. Representantes

En esta sección se definirá lo que se entiende por un representante positivo, un representante negativo y una combinación neutral.

Definición 2.4.

Sea $\Omega \in \{\Omega\}_j$, un conjunto de apoyo para la clase K_j, el **conjunto de representantes positivos** para la clase K_j con respecto a Ω, que lo denotaremos por ΩM_j^1 , se define como el conjunto de todos los valores, para la ω-parte correspondiente, que se presentan η_j veces en las ω-partes de los objetos de K_j y no se presentan ni una vez en las ω-partes de los objetos de CK_j.

Nota.- Cada elemento de este conjunto será llamado un representante positivo.

Definición 2.5.

Sea $\Omega \in \{\Omega\}_j$, un conjunto de apoyo para la clase K_j, el **conjunto de representantes negativos** para la clase K_j con respecto a Ω, que lo denotaremos por ΩM_j^0 , se define como el conjunto de todos los valores, para la ω-parte correspondiente, que se presentan η_j veces en las ω-partes de los objetos de CK_j y no se presentan ni una vez en las ω-partes de los objetos de K_j.

Nota.- Cada elemento de este conjunto será llamado un representante negativo.

Definición 2.6.

Sea $\Omega \in \{\Omega\}_j$, un conjunto de apoyo para la clase K_j, el **conjunto de combinaciones neutrales** para la clase K_j con respecto a Ω, que lo denotaremos por ΩM_j^Δ , se define como el conjunto de todos los valores, para la ω-parte correspondiente, que no sean representantes positivos, ni representantes negativos.

Nota.- Cada elemento de este conjunto será llamado una combinación neutral.

2.1.1.2. Clasificación

El método de clasificación usando conjuntos de representantes utiliza como parámetros los pesos informacionales de los rasgos y de los objetos, los cuales denotaremos por P_i^j i=1,..,n y γ_t^j t=1,..,m respectivamente. Estos pesos informacionales pueden calcularse usando las fórmulas descritas en el capítulo anterior, o bien usando la estimación de parámetros que será descrita más adelante.

Como en este caso se considerarán problemas en los cuales los objetos están descritos en términos de n rasgos booleanos, se puede garantizar que los conjuntos ΩM_j^1, ΩM_j^0 y ΩM_j^Δ son finitos para cada $\Omega \in \{\Omega\}_j$ y para cada j=1,...,ℓ, por lo cual pueden calcularse totalmente. El algoritmo de clasificación es el siguiente:

Algoritmo 2.1.

1) Se definen los sistemas de conjuntos de apoyo $\{\Omega\}_j$ y los parámetros η_j para cada una de las clases K_j, j=1,..,ℓ

2) Se calculan los complementos, CK_j, para cada una de las clases K_j, j=1,..,ℓ

3) Se calculan todos los conjuntos ΩM_j^1, ΩM_j^0 y ΩM_j^Δ para cada uno de los conjuntos de apoyo de cada una de las clases K_j, j=1,..,ℓ

4) Cuando se quiere clasificar un nuevo objeto O se calcula $\Gamma_j(\omega O)$, para cada conjunto de apoyo Ω, de cada clase K_j, con j=1,..,ℓ. $\Gamma_j(\omega O)$ está definida como:

   Sea $\Omega = \{x_{u_1},\ldots,x_{u_k}\}$

   a) $(P_{u_1}^j+\ldots+P_{u_k}^j)(\gamma_{v_1}^j+\ldots+\gamma_{v_t}^j)$ si $\omega O \in \Omega M_j^1$

   donde $\{O_{v_1},\ldots,O_{v_t}\} = \{\, O_v \mid O_v \in K_j \wedge \omega O_v = \omega O \,\}$

   b) $-(P_{u_1}^j+\ldots+P_{u_k}^j)(\gamma_{v_1}^j+\ldots+\gamma_{v_t}^j)$ si $\omega O \in \Omega M_j^0$

   donde $\{O_{v_1},\ldots,O_{v_t}\} = \{\, O_v \mid O_v \in CK_j \wedge \omega O_v = \omega O \,\}$

   c) 0 si $\omega O \in \Omega M_j^\Delta$

5) Se calcula la evaluación total $\Gamma_j(O)$, para cada clase K_j, $j=1,..,\ell$, $\Gamma_j(O)$ está definida como:

$$\Gamma_j(O) = \frac{1}{|\{\Omega\}_j|} \sum_{\Omega \in \{\Omega\}_j} \Gamma_j(\omega O)$$

6) Se calcula el vector informacional, del objeto O, el cual tiene la forma $\alpha(O)=(\alpha_1(O), .. ,\alpha_\ell(O))$, donde:

$$\alpha_j(O) = \begin{cases} 1 & \text{si } \Gamma_j(O) > 0 \\ 0 & \text{si } \Gamma_j(O) < 0 \\ * & \text{si } \Gamma_j(O) = 0 \end{cases}$$

donde $\alpha_j(O)=*$ significa que el algoritmo se abstiene de clasificar al objeto O.

2.1.2. Caso Métrico

En este caso se considerarán problemas en los cuales los objetos están descritos en términos de rasgos x_i que pueden tomar valores en cualquier espacio métrico M_i con métrica ρ_i. Con esto se puede definir una función de semejanza, entre dos ω-partes de objetos O y O', introduciendo nuevos parámetros $\varepsilon_1^j,...,\varepsilon_n^j$ para $j=1,...,n$, de la siguiente manera.

Sea $\Omega = \{x_{u_1},...,x_{u_k}\}$ entonces, cuando se trabaja respecto a la clase K_j la función de semejanza es:

$$\beta(\omega O,\omega O') = \begin{cases} 1 & \text{si} \quad \rho_{u_i}(x_{u_i}(O),x_{u_i}(O')) \leq \varepsilon_{u_i}^j \quad \forall\, i=1,...,k \\ 0 & \text{en otro caso} \end{cases}$$

2.1.2.1. Representantes

La función de semejanza definida puede ser usada para determinar si una ω-parte de un objeto pertenece a ΩM_j^1, ΩM_j^0 o ΩM_j^Δ de la misma forma que en el caso booleano.

Una forma alternativa para decidir si una ω-parte de un objeto pertenece a ΩM_j^1, ΩM_j^0 o ΩM_j^Δ es introduciendo como parámetros los pesos interiores de los rasgos y de los objetos, $w_1,..,w_n$ y $w^1,..,w^m$ respectivamente, además de X_0, X_1 y δ. Con los cuales se definen las evaluaciones $\Gamma_j^+(\omega O)$ y $\Gamma_j^-(\omega O)$, de la siguiente manera:

Sea $\Omega = \{x_{u_1},\ldots,x_{u_k}\}$ entonces

$$\Gamma_j^+(\omega O) = (w_{u_1}+\ldots+w_{u_k})[X_1 \sum_{O_i \in K_j} w^i \beta(\omega O, \omega O_i) + X_0 \sum_{O_i \in CK_j} w^i \overline{\beta}(\omega O, \omega O_i)]$$

$$\Gamma_j^-(\omega O) = (w_{u_1}+\ldots+w_{u_k})[X_1 \sum_{O_i \in CK_j} w^i \beta(\omega O, \omega O_i) + X_0 \sum_{O_i \in K_j} w^i \overline{\beta}(\omega O, \omega O_i)]$$

donde

$$\overline{\beta}(\omega O, \omega O') = \begin{cases} 1 & \text{si} \quad \beta(\omega O, \omega O') = 0 \\ 0 & \text{si} \quad \beta(\omega O, \omega O') = 1 \end{cases}$$

Estas evaluaciones permiten decidir si la ω-parte de un objeto O está en ΩM_j^1, ΩM_j^0 o ΩM_j^Δ de la siguiente manera:

a) $\omega O \in \Omega M_j^1$ si $\Gamma_j^+(\omega O) - \Gamma_j^-(\omega O) > \delta$

b) $\omega O \in \Omega M_j^0$ si $\Gamma_j^-(\omega O) - \Gamma_j^+(\omega O) > \delta$

c) $\omega O \in \Omega M_j^\Delta$ si $|\Gamma_j^+(\omega O) - \Gamma_j^-(\omega O)| \leq \delta$

Nota.- Los pesos interiores de los rasgos y de los objetos pueden ser calculados utilizando las definiciones dadas en el capítulo anterior, para los pesos informacionales, considerando solamente una clase y tomando cada objeto dentro de la misma como si fuera una clase independiente, o bien considerando únicamente dos clases, K_j y CK_j.

2.1.2.2. Clasificación

Dado que en este caso no puede garantizarse que ΩM_j^1, ΩM_j^0 o ΩM_j^Δ sean finitos, entonces estos no pueden ser calculados totalmente, de manera que el algoritmo queda como:

Algoritmo 2.2.

1) Se definen los sistemas de conjuntos de apoyo $\{\Omega\}_j$ y los parámetros η_j para cada una de las clases K_j, $j=1,..,\ell$

2) Se calculan los complementos, CK_j, para cada una de las clases K_j, $j=1,..,\ell$

3) Cuando se quiere clasificar un nuevo objeto O, se decide si ωO pertenece a ΩM_j^1, ΩM_j^0 o ΩM_j^Δ y se calcula $\Gamma_j(\omega O)$, para cada conjunto de apoyo Ω, de cada clase K_j, con $j=1,..,\ell$. $\Gamma_j(\omega O)$ está definida como:

Sea $\Omega = \{x_{u_1},...,x_{u_k}\}$

a) $(P_{u_1}^j+...+P_{u_k}^j)(\gamma_{v_1}^j+...+\gamma_{v_t}^j)$ si $\omega O \in \Omega M_j^1$

donde $\{O_{v_1},...,O_{v_t}\} = \{ O_v \mid O_v \in K_j \wedge \beta(\omega O_v,\omega O)=1 \}$

b) $-(P_{u_1}^j+...+P_{u_k}^j)(\gamma_{v_1}^j+...+\gamma_{v_t}^j)$ si $\omega O \in \Omega M_j^0$

donde $\{O_{v_1},...,O_{v_t}\} = \{ O_v \mid O_v \in CK_j \wedge \beta(\omega O_v,\omega O)=1 \}$

c) 0 si $\omega O \in \Omega M_j^\Delta$

4) Se calcula la evaluación total $\Gamma_j(O)$, para cada clase K_j, $j=1,..,\ell$, $\Gamma_j(O)$ está definida como:

$$\Gamma_j(O) = \frac{1}{|\{\Omega\}_j|} \sum_{\Omega \in \{\Omega\}_j} \Gamma_j(\omega O)$$

5) Se calcula el vector informacional, del objeto O, el cual tiene la forma $\alpha(O)=(\alpha_1(O), .. ,\alpha_\ell(O))$, donde:

$$\alpha_j(O) = \begin{cases} 1 & \text{si } \Gamma_j(O) > 0 \\ 0 & \text{si } \Gamma_j(O) < 0 \\ * & \text{si } \Gamma_j(O) = 0 \end{cases}$$

donde $\alpha_j(O)=*$ significa que el algoritmo se abstiene de clasificar al objeto O.

2.1.3. Ejemplo

Como ejemplo, consideremos la siguiente matriz de aprendizaje:

MA	x_1	x_2	x_3	x_4	x_5	α_1	α_2	α_3
O_1	1	1	0	1	1	1	0	0
O_2	0	1	0	1	1	1	0	0
O_3	1	0	1	0	1	1	0	0
O_4	1	1	1	1	1	0	1	0
O_5	0	0	0	1	1	0	1	0
O_6	0	0	0	0	1	0	1	0
O_7	0	0	0	0	0	0	0	1
O_8	1	1	0	0	0	0	0	1
O_9	1	0	0	0	0	0	0	1

Los pesos informacionales de los rasgos y de los objetos, para cada una de las clases, son los siguientes.

P.I.	K_1	K_2	K_3
x_1	0.500000	0.675000	0.750000
x_2	0.857143	0.937500	1.000000
x_3	0.857143	1.000000	0.750000
x_4	1.000000	0.960000	0.937500
x_5	0.642857	0.650000	0.694444
O_1	0.333333	0.333333	1.000000
O_2	0.666667	0.500000	1.000000
O_3	1.000000	0.666667	0.800000
O_4	0.750000	0.750000	1.000000
O_5	0.500000	1.000000	1.000000
O_6	0.750000	0.500000	0.600000
O_7	1.000000	0.500000	0.600000
O_8	0.750000	1.000000	1.000000
O_9	1.000000	0.666667	0.800000

Para este ejemplo utilizaremos los siguientes sistemas de conjuntos de apoyo:

K_1: $\Omega_1 = \{x_2, x_3, x_4\}$
$\Omega_2 = \{x_1, x_5\}$
$\Omega_3 = \{x_3, x_4, x_5\}$
$\Omega_4 = \{x_1, x_2, x_3\}$

K_2: $\Omega_1 = \{x_1, x_2, x_5\}$
$\Omega_2 = \{x_1, x_2, x_3, x_4\}$
$\Omega_3 = \{x_2, x_4\}$

$$\Omega_4 = \{x_2 , x_3 , x_4 \}$$
$$\Omega_5 = \{x_3 , x_4 , x_5 \}$$
$$\Omega_6 = \{ x_4 , x_5 \}$$

K_3: $\Omega_1 = \{ x_4 , x_5 \}$
$$\Omega_2 = \{x_3 , x_4 , x_5 \}$$
$$\Omega_3 = \{ x_5 \}$$
$$\Omega_4 = \{x_2 , x_4 \}$$
$$\Omega_5 = \{x_1 , x_2 , x_3 , x_4 \}$$

los conjuntos de representantes positivos y negativos, para cada clase, utilizando estos conjuntos de apoyo y considerando $\eta_1=1$, $\eta_2=1$, y $\eta_3=1$, son:

K_1: M_1^1 = { 101-- , 010-- , --101 , -010- , -101- }
M_1^0 = { 100-- , 000-- , 111-- , --000 , --001 , --111 , 1---0 , 0---0 , -100- , -000- , -001- , -111- }

K_2: M_1^1 = { --001 , --111 , -001- , -111- , -0-1- , 0001- , 1111- , 00--1 }
M_1^0 = { ---00 , --000 , --101 , -100- , -010- , -101- , -1-0- , 1000- , 1100- , 1010- , 0101- , 1101- , 10--0 , 11--0 , 00--0 , 10--1 , 01--1 }

K_3: M_1^1 = { 1000- , 1100- , -1-0- , ----0 , --000 , ---00 }
M_1^0 = { 0001- , 1111- , 1010- , 0101- , 1101- , -0-1- , -1-1- , ----1 , --001 , --111 , --101 , --011 , ---01 , ---11 }

Ahora supongamos que se quieren clasificar los siguientes objetos.

O^1 = (0 1 0 1 0)
O^2 = (1 0 1 0 1)
O^3 = (0 0 1 1 1)
O^4 = (0 0 0 1 0)
O^5 = (1 1 1 0 0)
O^6 = (0 1 0 0 0)
O^7 = (1 1 0 1 0)
O^8 = (0 1 1 1 1)
O^9 = (1 0 1 0 0)

por lo tanto las evaluaciones obtenidas para estos objetos, en cada una de las clases, son:

Γ_j	K_1	K_2	K_3
O^1	3.047620	-4.200830	-7.583330
O^2	7.428570	-7.561670	-10.690300
O^3	-1.875000	7.248750	-14.597200
O^4	-7.482140	7.236250	-3.708330
O^5	-3.660710	-7.648330	7.520830
O^6	-8.577380	-13.938300	13.237500
O^7	0.714286	-5.867920	-7.583330
O^8	-3.910710	2.999380	-18.472200
O^9	2.928570	-9.310000	2.833330

finalmente los vectores informacionales de estos objetos son:

$$\alpha(O^1) = (1, 0, 0)$$
$$\alpha(O^2) = (1, 0, 0)$$
$$\alpha(O^3) = (0, 1, 0)$$
$$\alpha(O^4) = (0, 1, 0)$$
$$\alpha(O^5) = (0, 0, 1)$$
$$\alpha(O^6) = (0, 0, 1)$$
$$\alpha(O^7) = (1, 0, 0)$$
$$\alpha(O^8) = (0, 1, 0)$$
$$\alpha(O^9) = (1, 0, 1)$$

Los parámetros de este ejemplo, así como las evaluaciones, fueron calculados usando el sistema C-REP.

2.2. Limitaciones

En esta sección se delimitará el tipo de problemas en los cuales puede aplicarse el método de clasificación usando conjuntos de representantes.

2.2.1. Tipo de Problemas

El modelo de clasificación usando conjuntos de representantes, en su forma original, puede ser aplicado en problemas que tengan las siguientes características:

1) El problema debe ser de clasificación con aprendizaje, es decir, un problema en el cual se tiene un universo dividido en clases, con una muestra de objetos para cada clase, y que consiste en decidir, para un objeto, a cuál o cuáles clases pertenece.

2) Cada objeto de la muestra, debe estar descrito en términos de rasgos que tomen valores en espacios métricos, con métrica conocida.

3) Las clases en que se divide el universo pueden ser no disjuntas, pero deben ser duras, es decir, conjuntos en el sentido clásico.

2.2.2. Restricciones

Aunque las características que debe tener un problema para que pueda aplicarse el modelo de clasificación usando conjuntos de representantes, permiten su aplicación sobre un gran número de problemas, aún existen ciertos tipos de problemas para los cuales el modelo no es aplicable, por ejemplo si la semejanza entre objetos no es booleana, si las clases son difusas, etc.

Por este motivo, el objetivo de esta tesis es extender el modelo, de manera que sea aplicable sobre un mayor universo de problemas. Así como reducir la distorsión de la realidad, y de esta manera incrementar la confiabilidad en los resultados.

2.2.3. Semejanza y Comparación Implícitas

Un punto que no se ha tratado, y que representa una limitación del modelo, es que implícitamente se trabaja con una función de semejanza booleana, entre objetos, definida como la coincidencia rasgo a rasgo, lo cual no siempre es lo deseable en problemas de clasificación, ya que dos objetos, o dos partes de objetos, pueden ser considerados semejantes, por el especialista, si son suficientemente parecidos, aunque no coincidan en todos sus rasgos.

Por este motivo también se extenderá el modelo para que trabaje con diferentes funciones de semejanza entre objetos, especialmente aquellas que tomen valores en el intervalo [0,1].

2.3. Conclusión

En este capítulo se ha dado una descripción del método de clasificación usando conjuntos de representantes, en su forma original, definiendo los conceptos de representante positivo y negativo, sobre los cuales está basado este método de clasificación.

Además se dió un ejemplo de la aplicación del método de clasificación usando conjuntos de representantes, sobre un pequeño conjunto de datos, con el fin de que se pueda entender con claridad el funcionamiento del método.

Una vez explicado el método de clasificación usando conjuntos de representantes, se hicieron notar las restricciones y limitaciones del mismo, las cuales fueron parte de la motivación de las extensiones que se realizan en los capítulos siguientes.

Capítulo 3

Estimación de Parámetros

En este capítulo se describirá un mecanismo para estimar los parámetros del método de clasificación usando conjuntos de representantes, en su forma original, este mecanismo fue tomado del trabajo de Baskakova[1].

3.1. Construcción de los Sistemas de Conjuntos de Apoyo $\{\Omega\}_j$

En esta sección se describirá un método para construir los sistemas de conjuntos de apoyo que se utilizan en el modelo de clasificación usando conjuntos de representantes.

3.1.1. Caso Booleano

Primero describiremos el caso en que los objetos están descritos en términos de n rasgos booleanos. Consideremos que la matriz de aprendizaje MA tiene m objetos, $O_1,\ldots,O_m$, y que se tiene una matriz de control, la cual denotaremos por MC, con q objetos, $O^1,\ldots,O^q$.

Si se quiere determinar el sistema de conjuntos de apoyo $\{\Omega\}_j$, para la clase K_j, primero separemos estas dos matrices en 4 tablas:

T_1	contiene los objetos en MA que están en K_j.
T_2	contiene los objetos en MA que están en CK_j.
T_3	contiene los objetos en MC que están en K_j.
T_4	contiene los objetos en MC que están en CK_j.

para simplificar la notación numeraremos los objetos de la siguiente manera:

$$T_1 = \{ O_1, \dots ,O_r \}$$
$$T_2 = \{ O_{r+1}, \dots ,O_m \}$$
$$T_3 = \{ O^1, \dots ,O^t \}$$
$$T_4 = \{ O^{t+1}, \dots ,O^q \}$$

esquemáticamente queda:

	MA	x_1	...	x_n	
	O_1	$x_1(O_1)$	...	$x_n(O_1)$	
K_j	:				T_1
	O_r	$x_1(O_r)$	...	$x_n(O_r)$	
	O_{r+1}	$x_1(O_{r+1})$	...	$x_n(O_{r+1})$	
CK_j	:				T_2
	O_m	$x_1(O_m)$	...	$x_n(O_m)$	

	MC	x_1	...	x_n	
	O^1	$x_1(O^1)$	...	$x_n(O^1)$	
K_j	:				T_3
	O^t	$x_1(O^t)$	...	$x_n(O^t)$	
	O^{t+1}	$x_1(O^{t+1})$	...	$x_n(O^{t+1})$	
CK_j	:				T_4
	O^q	$x_1(O^q)$	...	$x_n(O^q)$	

Nuestro objetivo consiste en construir una familia de subconjuntos de rasgos, tales que se cumplan las siguientes condiciones:

a) Los objetos de T_1 y T_3, así como los de T_2 y T_4, coincidan tanto como sea posible.

a') Los objetos de T_1 y T_4, así como los de T_2 y T_3, difieran tanto como sea posible.

Primero trataremos de satisfacer la segunda condición, para esto construimos dos tablas (T_1,T_4) y (T_2,T_3), considerando cada una de las subtablas como clases separadas. A continuación calculamos, para cada tabla, el conjunto de todos los testores típicos, $T(T_1,T_4)$ y $T(T_2,T_3)$ respectivamente. A continuación tomemos:

$$M = T(T_1,T_4) \cap T(T_2,T_3)$$

Supongamos que $M \neq \varnothing$, entonces M satisface las segunda condición (a'), puesto que los testores típicos son conjuntos totalmente diferenciantes, es decir no confunden ningún objeto, entre T_1 y T_4, ni entre T_2 y T_3. En este caso, para cada subconjunto de rasgos Ω, perteneciente a M, se checan las siguientes condiciones:

b) Para cada $O^i \in T_3$ existe $O_k \in T_1$ tal que $\omega O^i = \omega O_k$.

b') Para cada $O^i \in T_4$ existe $O_k \in T_2$ tal que $\omega O^i = \omega O_k$.

Si estas condiciones se cumplen, entoces Ω se considera en el sistema de conjuntos de apoyo $\{\Omega\}_j$. Si el proceso obtiene un sistema no vacío $\{\Omega\}_j$, entonces este sistema cumple también la primera condición (a), y por lo tanto la construcción del sistema de conjuntos de apoyo termina.

Si M es vacío, o si el proceso anterior da un sistema vacío, se necesita de un proceso más fino para determinar $\{\Omega\}_j$ por lo cual se aplica el siguiente algoritmo.

Para cada objeto $O^k \in T_3$ y cada objeto $O_i \in T_1$ se contruye el conjunto $M(O^k,O_i)$, el cual contiene a los rasgos en los cuales coinciden O^k y O_i, análogamente se contruyen los conjuntos $M(O^r,O_u)$ con $O^r \in T_4$ y $O_u \in T_2$. En la construcción de $\{\Omega\}_j$ se deben preservar las siguientes condiciones:

c) Para k fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^k,O_i)$, o uno de sus subconjuntos no vacío.

c') Para r fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^r,O_u)$ o uno de sus subconjuntos no vacío.

A cada uno de los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, le asignamos las siguientes características numéricas:

$P^+(M(O^k,O_i))$ = # de pares de objetos (O^s,O_v), con $O^s \in T_3$ y $O_v \in T_1$, tales que $\omega O^s = \omega O_v$.

$P^-(M(O^k,O_i))$ = # de pares de objetos (O^s,O_v), con $O^s \in T_4$ y $O_v \in T_1$, tales que $\omega O^s = \omega O_v$.

De la misma manera, a cada conjunto $M(O^r,O_u)$, $O^r \in T_4$ y $O_u \in T_2$, se le asignan las siguientes características numéricas:

$P^+(M(O^r,O_u))$ = # de pares de objetos (O^s,O_v), con $O^s \in T_4$ y $O_v \in T_2$, tales que $\omega O^s = \omega O_v$.
$P^-(M(O^r,O_u))$ = # de pares de objetos (O^s,O_v), con $O^s \in T_3$ y $O_v \in T_2$, tales que $\omega O^s = \omega O_v$.

Cada uno de estos conjunto se considerará en $\{\Omega\}_j$ mientras mayor sea P^+ y menor sea P^-, la selección se hace de alguna de las siguientes formas:

Modo 1

Se introduce una función de calidad W, para cada conjunto, que crece monótonamente respecto a P^+ y decrece monótonamente respecto a P^-, por ejemplo:

$$W = \frac{P^+}{1+P^-}$$

Con esta función se ordenan los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, en el orden de decrecimiento de W, y se incluyen en $\{\Omega\}_j$ los primeros conjuntos en este orden, teniendo en cuenta que el número de conjuntos incluidos debe preservar la condición (c). Análogamente se ordenan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in T_2$, y se incluyen los primeros en el orden, hasta que se cumpla la condición (c').

Modo 2

Se introduce un orden parcial en los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, de la siguiente manera:

$$M_1 \geq M_2 \quad \text{si} \quad P_1^+ \geq P_2^+ \wedge P_1^- \leq P_2^-$$

La selección de los conjuntos se realiza comenzando con los elementos maximales y se prosigue hasta que se cumpla la condición (c). Análogamente se introduce un orden parcial y se seleccionan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in T_2$, hasta que se cumpla la condición (c').

3.1.2. Caso General

En esta sección se considera el caso en que los objetos están descritos en términos de n rasgos que toman valores en espacios métricos. En este caso se puede seguir un procedimiento similar al del caso booleano, usando la función de semejanza descrita en la sección 2.1.2. Siempre y cuando se seleccionen los parámetros $\varepsilon_1^j,\ldots,\varepsilon_n^j$, para cada una de las clases, los cuales pueden ser estimados de la siguiente manera.

3.2. Determinación de los Umbrales ε_k^j

En esta sección se decribe una manera de estimar los valores de los umbrales ε_k^j. Consideremos el k-ésimo rasgo, $1 \leq k \leq n$, calculemos la distancia media:

$$\begin{aligned}(1)\qquad \rho_k(K_j, CK_j) = {} & \frac{1}{r(q-t)}\sum_{i=1}^{r}\sum_{j=t+1}^{q}\rho_k(x_k(O_i), x_k(O^j)) + \\ & \frac{1}{(m-r)t}\sum_{i=r+1}^{m}\sum_{j=1}^{t}\rho_k(x_k(O_i), x_k(O^j))\end{aligned}$$

Esta magnitud muestra el alejamiento medio, respecto al rasgo k, entre los objetos de aprendizaje y los de control, cuando pertenecen a clases distintas, K_j y CK_j. Ahora calculemos:

$$\begin{aligned}(2)\qquad B_{kj} = {} & \frac{1}{rt}\sum_{i=1}^{r}\sum_{j=1}^{t}\rho_k(x_k(O_i), x_k(O^j)) + \\ & \frac{1}{(m-r)(q-t)}\sum_{i=r+1}^{m}\sum_{j=t+1}^{q}\rho_k(x_k(O_i), x_k(O^j))\end{aligned}$$

La magnitud B_{kj} muestra la distancia media, respecto al rasgo k, entre los objetos de aprendizaje y los de control, cuando pertenecen a la misma clase, K_j o CK_j. Evidentemente, cuando mayor sea (1) y menor sea (2), más adecuado será el rasgo k para el reconocimiento de los objetos de control. Tomemos:

$$F(j,k) = \frac{\rho_k(K_j, CK_j)}{B_{kj}}$$

Claramente F(j,k) crece cuando (1) crece o (2) decrece. Ahora consideremos la siguiente magnitud:

$$\rho'_k(x_k(O_i),x_k(O^j)) = \begin{cases} 0 & \text{si } \rho_k(x_k(O_i),x_k(O^j)) \le \varepsilon_k^j \\ \rho_k(x_k(O_i),x_k(O^j)) & \text{si } \rho_k(x_k(O_i),x_k(O^j)) > \varepsilon_k^j \end{cases}$$

Sustituyendo en (1) y (2), $\rho_k(x_k(O_i),x_k(O^j))$ por $\rho'_k(x_k(O_i),x_k(O^j))$ obtenemos $\rho_k(K_j,CK_j,\varepsilon_k^j)$ y $B_{kj}(\varepsilon_k^j)$. Evidentemente:

$$\rho_k(K_j,CK_j,0) = \rho_k(K_j,CK_j)$$
$$B_{kj}(0) = B_{kj}$$

Análogamente obtenemos $F(j,k,\varepsilon_k^j)$. Ahora, el valor de ε_k^j lo obtenemos como una solución de:

$$\max_{\varepsilon_k^j}(F(j,k,\varepsilon_k^j))$$

donde

$$\varepsilon_k^j \in [0, \max_{i=1,..,m;r=1,..,q}\{\rho_k(x_k(O_i),x_k(O^r))\}].$$

debido a que de nada sirve considerar un ε_k^j mayor que la máxima diferencia que se pueda encontrar.

3.3. Estimación de los Pesos Informacionales

En esta sección se estimarán los parámetros γ_v^j, v=1,..,m, y P_u^j, u=1,..,n, para lo cual supondremos que ya se han determinado los sistemas de conjuntos de apoyo $\{\Omega\}_j$ y los parámetros η_j.

Para calcular los pesos informacionales de los objetos, con respecto a la clase K_j, primero asociemos a cada objeto $O_i \in$ MA, las siguientes magnitudes:

$\Pi_j^+(O_i)$ = # de representantes positivos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

$\Pi_j^-(O_i)$ = # de representantes negativos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

ahora γ_i^j es calculado como:

$$1)\ \gamma_i^j = \frac{\Pi_j^+(O_i)}{1+\Pi_j^-(O_i)} \qquad \text{si } O_i \in K_j$$

$$2)\ \gamma_i^j = \frac{\Pi_j^-(O_i)}{1+\Pi_j^+(O_i)} \qquad \text{si } O_i \in CK_j$$

Para el cálculo de los parámetros P_v^j , v=1,..,n, se pueden usar directamente las magnitudes $F(j,v,\varepsilon_v^j)$, introducidas para la construcción de los umbrales, de la siguiente manera:

$$P_v^j = F(j,v,\varepsilon_v^j)$$

donde ε_v^j es el umbral obtenido al resolver el problema de maximización. Otra manera para calcular los P_v^j, se basa en la misma idea que el método de cálculo de pesos informacionales basado en la teoría de testores. Separemos de $\{\Omega\}_j$ el subsistema $\{\Omega\}_j^i$ de conjuntos de apoyo que contienen al rasgo x_i. Para cada conjunto $\Omega \in \{\Omega\}_j^i$, de entre las ω-partes $\omega O^1,...,\omega O^t$, calculamos las siguientes magnitudes:

$M^+(\omega K_j)$ = # de representantes positivos para K_j
$M^-(\omega K_j)$ = # de representantes negativos para K_j

análogamente de entre las ω-partes $\omega O^{t+1},..,\omega O^q$, calculamos las siguientes magnitudes:

$M^+(\omega CK_j)$ = # de representantes positivos para K_j
$M^-(\omega CK_j)$ = # de representantes negativos para K_j

y entonces calculamos P_i^j como:

$$P_i^j = \frac{1}{|\{\Omega\}_j^i|} \sum_{\Omega \in \{\Omega\}_j^i} \frac{M^+(\omega K_j)+M^-(\omega CK_j)}{1+M^+(\omega CK_j)+M^-(\omega K_j)}$$

3.4. Determinación de los Parámetros η_j

En esta sección se explicará una manera de estimar los valores de los parámetros η_j, los cuales sirven para calcular los representantes de cada clase K_j.

Primero, para cada objeto $O^i \in MC$, separemos los conjuntos $\Omega \in \{\Omega\}_j$ para los cuales ωO^i es un candidato a representante positivo, o negativo, de K_j en MA, es decir se presenta sólo en K_j o en CK_j, pero no en ambos.

Analizaremos el caso en que $O^i \in K_j$, el caso en que $O^i \in CK_j$ se trata de manera análoga, si ωO^i es un candidato a representante positivo, entonces contamos el número de objetos en la tabla T_1 que coinciden con O^i en Ω, a este número lo llamaremos $N(\Omega,O^i)$, tomemos:

$$N(O^i) = \max_{\Omega} N(\Omega, O^i)$$

Claramente se ve que debe cumplirse que:

$$\eta_j \leq N(O^i)$$

puesto que si no se cumpliera, entonces O^i no obtendría una evaluación positiva para la clase K_j, de manera que sería mal clasificado, o al menos, el método se abstendría de clasificarlo; ya que esta desigualdad debe cumplirse para todo $O^i \in K_j$, entonces debe cumplirse:

$$\eta_j \leq \min_{O^i \in K_j} N(O^i)$$

La manera en que se seleccionaron los conjuntos de apoyo, los umbrales y la forma en que se obtienen los representantes positivos, garantiza que:

$$\min_{O^i \in K_j} N(O^i) \geq 1$$

de forma análoga, si $O^i \in CK_j$, y trabajando con los Ω tales que ωO^i es un candidato a representante negativo, se obtiene una desigualdad semejante para los $O^i \in CK_j$

$$\eta_j \leq \min_{O^i \in CK_j} N(O^i)$$

Tomando η_0 igual a la más fuerte de las dos restricciones, tenemos:

$$1 \leq \eta_j \leq \eta_0$$

Ahora bien, si η_0 no es muy grande, entonces se realiza el proceso de reconocimiento, para calcular $\alpha_j(O^i)$, para todo $O^i \in MC$, utilizando $\eta_j=1,\ldots,\eta_0$, y se elige $\eta_j=\eta^*$, donde η^* es el valor para el cual se realiza mejor el reconocimiento. Si η_0 es grande, se puede seguir el siguiente procedimiento.

Sea $O^i \in MC$ y $\Omega \in \{\Omega\}_j$, si $O^i \in K_j$, el caso en que $O^i \in CK_j$ se analiza de forma análoga, si ωO^i es un candidato a representante positivo para K_j en MA, entonces tomemos:

$$t(O^i,\Omega) = \#\text{ de objetos } O_v \in T_1 \text{ tales que } \omega O_v \text{ es semejante a } \omega O^i$$

para que el reconocimiento de O^i sea correcto, debería cumplirse

$$\eta_j \leq t(O^i,\Omega) \qquad (3)$$

si ωO^i es un candidato a representante negativo para K_j en MA, entonces tomemos:

$$w(O^i,\Omega) = \#\text{ de objetos } O_v \in T_2 \text{ tales que } \omega O_v \text{ es semejante a } \omega O^i$$

para que el reconocimiento de O^i sea correcto, debería cumplirse

$$\eta_j > w(O^i,\Omega) \qquad (4)$$

Obteniendo (3) y (4) para cada $O^i \in MC$, obtenemos un sistema de desigualdades impropio, para η_j, para determinar η_j buscamos un subsistema propio maximal, y sus soluciones. De estas soluciones se selecciona aquella para la cual se cumpla el mayor número de desigualdades del tipo (3).

El algoritmo de selección de η_j puede ser modificado, por ejemplo, requiriendo que para cada $O^i \in MC$ se cumpla al menos una de las desigualdades del tipo (3), y se construye un subsistema propio maximal que satisfaga estas condiciones, η_j puede ser cualquiera de las soluciones de este último subsistema.

3.5. Conclusión

En este capítulo hemos explicado la manera de estimar los parámetros que se utilizan en el método de clasificación usando conjuntos de representantes, en su forma original. Este mecanismo de estimación de parámetros permite calcular los sistemas de conjuntos de apoyo y los umbrales η_j, los cuales sirven para determinar si las combinaciones de valores son representantes. Además, suponiendo que se tienen calculados los sistemas de conjuntos de apoyo, y los umbrales η_j, se estiman los valores de los umbrales de cercanía de las métricas de los rasgos y los pesos informacionales de los rasgos y de los objetos.

Capítulo 4

Extensiones al Método de Clasificación

En este capítulo se describirán las extensiones hechas al método de clasificación, de manera que se pueda aplicar sobre un universo más amplio de problemas. Las extensiones que se consideran son:

1) Rasgos no restringidos.
2) Función de semejanza no booleana.
3) Clases difusas.

Todas estas extensiones, las cuales constituyen parte de la aportación de esta tesis, fueron motivadas por situaciones reales que se han ido presentando durante el proceso de modelación matemática de problemas de clasificación en Geofísica y Geología.

4.1. Rasgos en Espacios no Métricos

El modelo de clasificación usando conjuntos de representantes está propuesto para trabajar con rasgos que tomen valores en espacios métricos, y para cada uno de ellos se define un criterio de comparación de semejanza booleano, el cual está basado en la métrica de espacio, y en un umbral que nos indica si dos valores son suficientemente cercanos.

Sin embargo, en la práctica se han encontrado problemas donde no todos los rasgos que describen a los objetos toman valores en un espacio métrico, sino que, toman valores en un cierto conjunto y solamente se cuenta con algún criterio para decidir si dos valores se parecen, o no, lo cual resulta suficiente para la búsqueda de los representantes. Por este motivo, para permitir que el método acepte que los objetos se describan en términos de cualquier tipo de rasgos, sin imponer ninguna restricción a sus espacios de valores, solamente es necesario definir la forma en que se compararán los valores de los rasgos, en base a los criterios que ocupe el especialista, y a partir de estos construir un criterio de comparación de semejanza booleano, para cada uno de los rasgos, y con esto definir la función de semejanza, entre partes de objetos, la cual será ocupada para calcular los representantes.

En la siguiente sección se explican los cambios que se necesitan hacer, al modelo de clasificación usando conjuntos de representantes, para trabajar con objetos descritos en términos de rasgos que toman valores en espacios no restringidos.

4.1.1. Distintos Criterios de Comparación de Semejanza

Primero supongamos que cada rasgo x_i toma valores en el conjunto M_i, y que se tiene definido un criterio de comparación de semejanza booleano, para comparar valores dentro de M_i, de la siguiente manera:

$$CS_i(v_1, v_2) = \begin{cases} 1 & \text{si } v_1 \text{ y } v_2 \text{ coinciden} \\ 0 & \text{en otro caso} \end{cases}$$

donde el hecho de que dos valores $v_1, v_2 \in M_i$ coincidan, estará dado por la forma en que el especialista compare los valores del rasgo x_i.

A partir de estos criterios de comparación, definiremos una función de semejanza booleana, de la siguiente manera:

Sea $\Omega = \{x_{u_1}, \ldots, x_{u_k}\}$ entonces

$$\beta(\omega O, \omega O') = \begin{cases} 1 & \text{si } \; CS_{u_i}(x_{u_i}(O), x_{u_i}(O')) = 1 \;\; , \;\; \forall\, i = 1, \ldots, k \\ 0 & \text{en otro caso} \end{cases}$$

Utilizando esta funcion de semejanza, el resto del algoritmo permanece sin cambio, teniendo en cuenta la cosideración acerca de que no puede garantizarse que ΩM_j^1, ΩM_j^0 o ΩM_j^Δ sean finitos, para lo cual puede seguirse el mecanismo usado en el algoritmo de la sección 2.1.2.2, o restringir los representantes a los valores que aparezcan en la matriz de aprendizaje.

Si se sigue esta segunda opción, las definiciones de representante positivo y de representante negativo tienen que modificarse ligeramente, con lo cual quedan de la siguiente manera:

Definición 4.1.

Sea $\Omega = \{x_{u_1},...,x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes positivos** para la clase Kj con respecto a Ω, que lo denotaremos por ΩM_j^1, se define como el conjunto de todos los valores $(a_1,...,a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in K_j \mid x_{u_1}(O) = a_1 \wedge ... \wedge x_{u_k}(O) = a_k$
2) $(a_1,...,a_k)$ se presente η_j veces en las ω-partes de los objetos de Kj.
3) no se presenta ni una vez en los objetos de CKj.

Nota.- Cada elemento de este conjunto será llamado un representante positivo.

Definición 4.2.

Sea $\Omega = \{x_{u_1},...,x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes negativos** para la clase Kj con respecto a Ω, que lo denotaremos por ΩM_j^0, se define como el conjunto de todos los valores $(a_1,...,a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in CK_j \mid x_{u_1}(O) = a_1 \wedge ... \wedge x_{u_k}(O) = a_k$
2) $(a_1,...,a_k)$ se presente η_j veces en las ω-partes de los objetos de CKj.
3) no se presenta ni una vez en los objetos de Kj.

Nota.- Cada elemento de este conjunto será llamado un representante negativo.

4.2. Distinta Función de Semejanza

Hasta el momento el método de clasificación usando conjuntos de representantes trabaja con una función de semejanza booleana definida como la coincidencia rasgo a rasgo, es decir, dos ω-partes de objetos son semejantes si y sólo si coinciden en todos sus rasgos.

Esto es una manera natural de considerar la semejanza entre partes de objetos, pero para algunos problemas, los especialistas consideran la semejanza, entre partes de objetos, de alguna otra manera, por ejemplo, dos partes de objetos son semejantes, si coinciden en suficientes rasgos, aunque difieran en algunos, o si sus rasgos son suficientemente parecidos, aunque no coincidan del todo. En estos casos, el problema se reduce a definir una función de semejanza booleana, teniendo en cuenta el criterio del especialista, y trabajar con esta función para el cálculo de los representantes positivos y negativos, lo cual permite que el resto del algoritmo funcione sin cambios, puesto que la función de semejanza sólo se utiliza para decidir si un conjunto de valores es un representante positivo, un representante negativo o una combinación neutral.

En otro tipo de problemas, sólo se cuenta con una función de semejanza que no es booleana, es decir no sólo nos indica si dos ω-partes de objetos se parecen o no, sino que nos da una estimación de que tanto se parecen. En este caso, puede definirse una función de semejanza booleana, fijando un umbral, el cual separe los valores de la función para los cuales se considerarán semejantes las ω-partes, de aquellos para los cuales no se considerarán semejantes. Esta solución consiste básicamente en reducir el problema al caso anterior.

Otra manera de resolver el problema, es introduciendo nuevas definiciones para representantes positivos y negativos, las cuales no estén basadas en términos de cuántas veces se presenta en los objetos, sino en términos de qué tanto se parece a estos objetos. Además puesto que es difícil que un conjunto de valores se parezca en 0 a todos los objetos, se hace necesario introducir un nuevo parámetro δ_i, que nos indique cuándo una combinación de valores no se ha presentado mucho en el complemento, y por lo tanto se puede considerar como representante.

Para dar estas nuevas definiciones, de representante positivo y negativo, supondremos que se tiene una función de semejanza β, la cual nos da una estimación acerca de qué tanto se parecen dos ω-parte de objetos, y que toma valores en el intervalo [0,1], esto siempre puede obtenerse, normalizando los valores de esta función. En este caso supondremos que 0 nos indica que no se parecen en nada, y 1 nos indica que pueden considerarse como si fueran iguales, lo cual quiere decir que valores mayores indicarán parecidos mayores.

Definición 4.3

Sea $\Omega = \{x_{u_1},\ldots,x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes positivos** para la clase K_j con respecto a Ω , que lo denotaremos por ΩM_j^1, se define como el conjunto de todos los valores $(a_1,\ldots,a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in K_j \mid x_{u_1}(O) = a_1 \wedge \ldots \wedge x_{u_k}(O) = a_k$

2) $\sum_{O \in K_j} \beta(\omega O, (a_1, \ldots, a_k)) \geq \eta_j$

3) $\sum_{O \in CK_j} \beta(\omega O, (a_1, \ldots, a_k)) < \delta_j$

Nota.- En este caso, los parámetros η_j y δ_j son números reales.

Definición 4.4.

Sea $\Omega = \{x_{u_1}, \ldots, x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes negativos** para la clase K_j con respecto a Ω , que lo denotaremos por ΩM_j^0, se define como el conjunto de todos los valores $(a_1, \ldots, a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in CK_j \mid x_{u_1}(O) = a_1 \wedge \ldots \wedge x_{u_k}(O) = a_k$

2) $\sum_{O \in CK_j} \beta(\omega O, (a_1, \ldots, a_k)) \geq \eta_j$

3) $\sum_{O \in K_j} \beta(\omega O, (a_1, \ldots, a_k)) < \delta_j$

Con estas definiciones, el algoritmo de clasificación trabaja prácticamente igual, excepto que inicialmente se definen los parámetros η_j y δ_j, para cada una de las clases, $j=1,..,\ell$. Además en el momento de la evaluación de la ω-parte debe incluirse, como factor, el parecido que tiene con cada uno de los objetos que lo llevaron a ser representante positivo o negativo.

Claramente estas nuevas definiciones coinciden con las anteriores, si la función de semejanza toma valores en $\{0,1\}$ y se toma $\delta_j=0$.

4.3. Clases Difusas

En esta sección extenderemos el método de clasificación usando conjuntos de representantes para trabajar en situaciones donde las clases en las que se divide el universo no son conjuntos duros, sino que son conjuntos difusos, este caso se presenta cuando la naturaleza del problema hace que la pertenencia de ciertos objetos a algunas clases no sea exactamente 1, sino que puede tomar valores en $[0,1]$, donde 0 indica que el objeto no pertenece a la clase, 1 indica que el objeto pertenece totalmente a la clase, y los valores intermedios representan los distintos grados de pertenencia. En este tipo de problemas la respuesta para cada clase no es si el objeto pertenece o no, sino el grado con el cual pertenece.

En este caso los vectores informacionales, de los objetos, no contienen a que clases pertenecen, sino el grado con que pertenecen a cada clase, es decir, la componente j-ésima del vector informacional, del objeto O, contendrá la pertenencia de este objeto a la clase K_j, es decir:

$\alpha_j(O) = \mu_{K_j}(O)$ donde μ_{K_j} es la función de pertenencia de la clase K_j, $j=1,..,\ell$.

La única restricción que pondremos sobre la muestra es que:

$$\forall(j=1,..,\ell)\ \max_{1\le i\le m}(\alpha_j(O_i)) \ge 0.5$$

Esto quiere decir que para cada clase exista al menos un objeto que esté más cerca de pertenecer totalmente a la clase, que de no pertenecer.

Para resolver este problema se tienen que hacer modificaciones a las definiciones de representante positivo y negativo, de manera que involucre estos grados de pertenencia. Adicionalmente como las clases son difusas, una combinación de valores puede ser representante de varias clases, con diferente grado, por lo cual hay que estimar el grado con el que una combinación de valores es representante de una clase.

Para dar estas nuevas definiciones de representante positivo y negativo, supondremos que se tiene una función de semejanza β, la cual nos da una estimación acerca de que tanto se parecen dos ω-parte de objetos, y que toma valores en el intervalo [0,1]. Además consideraremos que si un objeto O pertenece en $\alpha_j(O)$, a la clase K_j, entonces pertenece en $(1-\alpha_j(O))$ al complemento de esta clase, lo cual coincide con la manera tradicional de considerar los complementos de conjuntos difusos.

Definición 4.5

Sea $\Omega = \{x_{u_1},\ldots,x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes positivos** para la clase Kj con respecto a Ω, que lo denotaremos por ΩM_j^1, se define como el conjunto de todos los valores $(a_1,\ldots,a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in K_j \mid x_{u_1}(O) = a_1 \wedge \ldots \wedge x_{u_k}(O) = a_k$

2) $\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))\alpha_j(O_i) \ge \eta_j$

3) $\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))(1-\alpha_j(O_i)) < \delta_j$

4) $\mu_j(\omega(a_1,\ldots,a_k)) = \dfrac{\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))\alpha_j(O_i)}{\sum_{i=1}^{m} \alpha_j(O_i)}$

Nota.- Puede notarse que $\mu_j(\omega(a_1,\ldots,a_k))$, el grado de difusión del representante positivo, se encuentra en el intervalo:

$$\left[\frac{\eta_j}{\sum_{i=1}^{m} \alpha_j(O_i)}, 1\right]$$

Definición 4.6.

Sea $\Omega = \{x_{u_1},\ldots,x_{u_k}\}$ un conjunto de apoyo para la clase K_j, el **conjunto de representantes negativos** para la clase K_j con respecto a Ω, que lo denotaremos por ΩM_j^0, se define como el conjunto de todos los valores $(a_1,\ldots,a_k)$, para la ω-parte correspondiente, tales que:

1) $\exists O \in CK_j \mid x_{u_1}(O) = a_1 \wedge \ldots \wedge x_{u_k}(O) = a_k$

2) $\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))(1-\alpha_j(O_i)) \geq \eta_j$

3) $\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))\alpha_j(O_i) < \delta_j$

4) $\mu_j(\omega(a_1,\ldots,a_k)) = \dfrac{\sum_{i=1}^{m} \beta(\omega O_i,(a_1,\ldots,a_k))(1-\alpha_j(O_i))}{\sum_{i=1}^{m} (1-\alpha_j(O_i))}$

Nota.- Puede notarse que $\mu_j(\omega(a_1,\ldots,a_k))$, el grado de difusión del representante negativo se encuentra en el intervalo:

$$\left[\frac{\eta_j}{\sum_{i=1}^{m} (1-\alpha_j(O_i))}, 1\right]$$

Si todas las pertenencias de los objetos a las clases fueran 0 o 1, es decir duras, entonces estas definiciones de representante positivo y negativo, coincidirían con las anteriores.

En este caso, el algoritmo de clasificación debe cambiar un poco, puesto que aquí el problema no es decidir, para cada clase, si el nuevo objeto pertenece o no, sino estimar el grado con que pertenece a cada una de las clases, es decir, determinar el vector informacional del nuevo objeto.

Para esto, dado un objeto O, a clasificar, primero determinaremos su evaluación con respecto a un cierto conjunto de apoyo $\Omega=\{x_{u_1},\ldots,x_{u_k}\}$, la cual denotaremos por $\Gamma_j(\omega O)$, y estará dada por:

a) $\frac{1}{|\Omega|}\sum_{i=1}^{k} P^{j}_{u_i} \cdot \frac{1}{\rho}\sum_{i=1}^{m} \gamma^{j}_{v_i}\alpha_j(O_i)\beta(\omega O,\omega O_i)$ si $\omega O \in \Omega M^1_j$

b) $-\frac{1}{|\Omega|}\sum_{i=1}^{k} P^{j}_{u_i} \cdot \frac{1}{\rho}\sum_{i=1}^{m} \gamma^{j}_{v_i}(1-\alpha_j(O_i))\beta(\omega O,\omega O_i)$ si $\omega O \in \Omega M^0_j$

c) 0 si $\omega O \in \Omega M^{\Delta}_j$

donde

$$\rho=\sum_{i=1}^{m}\alpha_j(O_i)\gamma^i_j$$

Las divisiones, entre el cardinal del conjunto de apoyo y ρ, se hacen con objeto de obligar a que $\Gamma_j(\omega O)$ esté en [0,1].

Después se calcula la evidencia positiva total, denotada por $\Gamma^+_j(O)$, y la evidencia negativa total, denotada por $\Gamma^-_j(O)$, para cada clase K_j, $j=1,..,\ell$, las cuales están definidas como:

$$\Gamma^+_j(O) = \sum_{\Gamma_j(\omega O)\geq 0}\Gamma_j(\omega O)\mu_j(\omega O)$$

$$\Gamma^-_j(O) = \sum_{\Gamma_j(\omega O)< 0}\Gamma_j(\omega O)\mu_j(\omega O)$$

Una vez hecho esto, se calcula la evaluación total $\Gamma_j(O)$, para cada clase K_j, $j=1,..,\ell$, $\Gamma_j(O)$ está definida como:

$$\Gamma_j(O) = \begin{cases} 0 & \text{si } \Gamma_j^+(O) = 0 \wedge \Gamma_j^-(O) = 0 \\ \dfrac{\Gamma_j^+(O) + \Gamma_j^-(O)}{2\max\{|\Gamma_j^+(O)|, |\Gamma_j^-(O)|\}} & \text{en otro caso} \end{cases}$$

En este caso, dividir entre el máximo tiene como objetivo obtener un número en [-1,1], que nos indique la proporción de ventaja que tenga la información positiva sobre la negativa, o viceversa. La división entre dos es para obtener un número en [-0.5,0.5].

Con esta evaluación se estimará la función de pertenencia de la clase K_j, mapeando los valores en [-0.5,0.5], de manera que cumpla que:

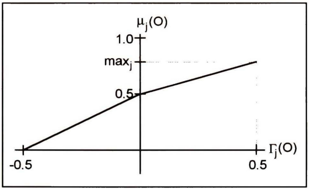
- $\mu_j(O)$ sea monótona creciente
- $\mu_j(O)$ sea continua
- $\mu_j(O) \in [0, \max_j]$
  donde
  $$\max_j = \max_{O_i \in MA} \mu_j(O_i)$$
- $\mu_j(O)=0$ si y solo si $\Gamma_j(O)=-0.5$
- $\mu_j(O)=0.5$ si y solo si $\Gamma_j(O)=0$
- $\mu_j(O)>0.5$ si y solo si $\Gamma_j(O)>0$
- $\mu_j(O)<0.5$ si y solo si $\Gamma_j(O)<0$

Por este motivo, el vector informacional, del objeto O, el cual tiene la forma $\alpha(O)=(\alpha_1(O), \ldots, \alpha_\ell(O))$, donde $\alpha_j(O)=\mu_j(O)$ indica el grado de pertenencia del objeto O a la clase K_j, y está definido como:

$$\alpha_j(O) = \begin{cases} 0.5 + \Gamma_j(O)(2\max_j - 1) & \text{si } \Gamma_j(O) \geq 0 \\ 0.5 + \Gamma_j(O) & \text{si } \Gamma_j(O) < 0 \end{cases}$$

En esta última expresión, si la evaluación final de un objeto es 0, entonces hay la misma evidencia indicando que está en la clase, que indicando que no está en la clase, por lo tanto la pertenencia obtenida es 0.5. Si la evaluación final es mayor que 0, entonces hay más evidencia indicando que está en la clase, y por lo tanto la pertenencia obtenida será mayor que 0.5, análogamente si la evaluación final es menor que 0, la evidencia de no estar en la clase es mayor y por lo tanto la pertenencia obtenida será menor que 0.5. Además si la evaluación final es 0.5, es decir la máxima evaluación posible, entonces se le asigna al objeto la máxima pertenencia existente, y si la evaluación final es -0.5, es decir la mínima evaluación posible, entonces se le asigna al objeto una pertenencia 0.

La gráfica de esta función de pertenencia es:



Como puede verse, la función de pertenencia obtenida es lineal, pero podría buscarse otro tipo de función de pertenencia, siempre y cuando se cumplan las condiciones impuestas.

4.4. Análisis de Complejidad

En esta sección haremos un breve análisis acerca de la complejidad del método extendido de clasificación usando conjuntos de representantes.

Nos concentraremos en el problema del cálculo de los representantes, puesto que la clasificación de un objeto es lineal con respecto al número de representantes obtenidos, y no se necesita más espacio que el requerido para almacenar la descripción del objeto.

Sea n el número de rasgos del problema, m el número de objetos de la matriz de aprendizaje, ℓ el número de clases y k_i el número de conjuntos de apoyo que se utilizarán en la clase K_i, $i=1,\ldots,\ell$.

Cuando se está trabajando con la clase K_i, dado que los representantes están restringidos a valores existentes en la matriz de aprendizaje MA, para cada conjunto de apoyo se tienen que checar, para decidir si son representantes positivos o negativos, las m posibles combinaciones de valores, una por cada objeto de MA, por lo cual el total de candidatos a representante analizados son:

$$k_i m$$

Cada uno de estos candidatos se tiene que comparar contra todos los objetos de MA, para saber si se parece suficiente a los objetos de la clase, y poco a los objetos del complemento, o viceversa, y como esto se hace para cada candidato, en total se hacen:

$k_i m^2$

comparaciones entre ω-partes de objetos. Como este proceso se realiza para cada una de las clases, tenemos que el total de comparaciones entre ω-partes de objetos es:

$$m^2 \sum_{i=1}^{\ell} k_i$$

Por otro lado, como cada combinación puede ser representante positivo o negativo de cada clase, entonces el máximo número de representantes que se pueden obtener es:

$$m \sum_{i=1}^{\ell} k_i$$

Concluyendo, para un problema de n rasgos, m objetos, ℓ clases, k_i conjuntos de apoyo en cada clase K_i, $i=1,\ldots,\ell$, la complejidad en el tiempo es de:

$$m^2 \sum_{i=1}^{\ell} k_i$$

comparaciones entre ω-partes de objetos, y la complejidad en el espacio es de:

$$m \sum_{i=1}^{\ell} k_i \, T(r)$$

donde T(r) es el espacio necesario para almacenar un representante.

4.5. Conclusión

En este capítulo se desarrollaron algunas extensiones al método de clasificación usando conjuntos de representantes, redefiniendo algunos de los conceptos y definiendo algunos nuevos. Todas estas extensiones se hicieron con el fin de reducir las restricciones del algoritmo, y de esta manera ampliar su campo de aplicación, y por la existencia de problemas reales, para los cuales no podía aplicarse el método de clasificación usando conjuntos de representantes, en su forma original.

Las extensiones desarrolladas fueron:

- Permitir trabajar con rasgos en cualquier espacio.
- Permitir trabajar con distintos criterios de comparación
- Permitir trabajar con distinta función de semejanza.
- Permitir trabajar con clases difusas.

Finalmente, también se hizo un análisis de complejidad, en tiempo y espacio, del método de clasificación usando conjuntos de representantes.

Capítulo 5

Estimación de Parámetros con Distinta Función de Semejanza

En este capítulo se describirá una manera de estimar los parámetros para el método extendido de clasificación usando conjuntos de representantes, poniendo especial atención al método resultante de usar una función de semejanza que tome valores en [0,1], y dejando para un posterior análisis el problema de clases difusas. Estas extensiones forman parte de la aportación de esta tesis.

En este capítulo se considera un problema con un universo dividido en ℓ clases, y con objetos descritos en términos de n rasgos de cualquier tipo, pero con criterios de comparación booleanos, además se tiene una función β, de semejanza entre partes de objetos, que toma valores en [0,1]. Consideremos que la matriz de aprendizaje MA tiene m objetos, $O_1,...,O_m$, y que se tiene una matriz de control, la cual denotaremos por MC, con q objetos, $O^1,...,O^q$.

5.1. Construcción de los Sistemas de Conjuntos de Apoyo $\{\Omega\}_j$

En esta sección se describirá un método para construir los sistemas de conjuntos de apoyo que se utilizan en el modelo extendido de clasificación usando conjuntos de representantes.

La manera de construir los conjuntos de apoyo en el método extendido no difiere mucho de la manera en que se construían originalmente, esto debido a que en dicha construcción se consideraba solamente el criterio de existencia de objetos que coincidan en determinado subconjunto de rasgos, lo cual se mantiene cuando se resuelve el problema de infinitud restringiendo los representantes a valores de la matriz de aprendizaje. La selección final de los conjuntos se hace atendiendo al parecido de los objetos en los rasgos de cada conjunto, para lo cual sí interviene el hecho de tener distinta función de semejanza.

Si se quiere determinar el sistema de conjuntos de apoyo $\{\Omega\}_j$, para la clase K_j, primero separemos MA y MC en 4 tablas:

T_1 contiene los objetos en MA que están en K_j.
T_2 contiene los objetos en MA que están en CK_j.
T_3 contiene los objetos en MC que están en K_j.
T_4 contiene los objetos en MC que están en CK_j.

para simplificar la notación numeraremos los objetos de la siguiente manera:

$$T_1 = \{ O_1, \dots ,O_r \}$$
$$T_2 = \{ O_{r+1}, \dots ,O_m \}$$
$$T_3 = \{ O^1, \dots ,O^t \}$$
$$T_4 = \{ O^{t+1}, \dots ,O^q \}$$

esquemáticamente queda:

	MA	x_1	...	x_n	
	O_1	$x_1(O_1)$	...	$x_n(O_1)$	
K_j	:				T_1
	O_r	$x_1(O_r)$	...	$x_n(O_r)$	
	O_{r+1}	$x_1(O_{r+1})$	...	$x_n(O_{r+1})$	
CK_j	:				T_2
	O_m	$x_1(O_m)$	...	$x_n(O_m)$	

	MC	x_1	...	x_n	
	O^1	$x_1(O^1)$	...	$x_n(O^1)$	
K_j	:				T_3
	O^t	$x_1(O^t)$	...	$x_n(O^t)$	
	O^{t+1}	$x_1(O^{t+1})$	...	$x_n(O^{t+1})$	
CK_j	:				T_4
	O^q	$x_1(O^q)$	...	$x_n(O^q)$	

Nuestro objetivo consiste en construir una familia de subconjuntos de rasgos, tales que se cumplan las siguientes condiciones:

a) Los objetos de T_1 y T_3, así como los de T_2 y T_4, coincidan tanto como sea posible.

a') Los objetos de T_1 y T_4, así como los de T_2 y T_3, difieran tanto como sea posible.

Primero para cada objeto $O^k \in T_3$ y cada objeto $O_i \in T_1$ se construye el conjunto $M(O^k,O_i)$, el cual contiene a los rasgos en los cuales coinciden, O^k y O_i, atendiendo a los criterios de comparación de cada rasgo, análogamente se construyen los conjuntos $M(O^r,O_u)$ con $O^r \in T_4$ y $O_u \in T_2$. Estos conjuntos sirven para garantizar que los valores de los representantes que reconozcan a cada objeto se presenten en MA. En la construcción de $\{\Omega\}_j$ se deben preservar las siguientes condiciones:

c) Para k fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^k,O_i)$ o uno de sus subconjuntos no vacío.

c') Para r fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^r,O_u)$ o uno de sus subconjuntos no vacío.

A cada uno de los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, le asignamos las siguientes características numéricas:

$$P^+(M(O^k,O_i)) = \sum_{v=1}^{r}\sum_{s=1}^{t}\beta(\omega O^s,\omega O_v)$$

$$P^-(M(O^k,O_i)) = \sum_{v=1}^{r}\sum_{s=t+1}^{q}\beta(\omega O^s,\omega O_v)$$

De la misma manera, a cada conjunto $M(O^r,O_u)$, $O^r \in T_4$ y $O_u \in T_2$, se le asignan las siguientes características numéricas:

$$P^+(M(O^r,O_u)) = \sum_{v=r+1}^{m}\sum_{s=t+1}^{q}\beta(\omega O^s,\omega O_v)$$

$$P^-(M(O^r,O_u)) = \sum_{v=r+1}^{m}\sum_{s=1}^{t}\beta(\omega O^s,\omega O_v)$$

Estas características nos indican qué tanto se parecen los objetos de MA y MC cuando pertenecen a la misma clase y cuando pertenecen a diferentes clases, considerando como únicas clases K_j y CK_j. Cada uno de estos conjuntos se considerará en $\{\Omega\}_j$ mientras mayor sea P^+ y menor sea P^-, la selección se hace de alguna de las siguientes formas:

Modo 1

Se introduce una función de calidad W, para cada conjunto, que crece monótonamente respecto a P^+ y decrece monótonamente respecto a P^-, por ejemplo:

$$W = \frac{P^+}{1 + P^-}$$

Con esta función se ordenan los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, en el orden de decrecimiento de W, y se incluyen en $\{\Omega\}_j$ los primeros conjuntos en este orden, teniendo en cuenta que el número de conjuntos incluidos debe preservar la condición (c). Análogamente se ordenan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in T_2$, y se incluyen los primeros en el orden, hasta que se cumpla la condición (c').

Modo 2

Se introduce un orden parcial en los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in T_1$, de la siguiente manera:

$$M_1 \geq M_2 \qquad \text{si} \qquad P_1^+ \geq P_2^+ \wedge P_1^- \leq P_2^-$$

La selección de los conjuntos se realiza comenzando con los elementos maximales y se prosigue hasta que se cumpla la condición (c). Análogamente se introduce un orden parcial y se seleccionan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in T_2$, hasta que se cumpla la condición (c').

5.2. Estimación de los Pesos Informacionales

En esta sección se estimarán los parámetros γ_v^j, v=1,..,m, y P_u^j, u=1,..,n, para lo cual supondremos que ya se han determinado los sistemas de conjuntos de apoyo $\{\Omega\}_j$ y los parámetros η_j y δ_j. Esta estimación, igualmente, no difiere gran cosa de la hecha para el método original, sólo que aquí se usan las nuevas definiciones de representante positivo y negativo.

Para calcular los pesos informacionales de los objetos, en la clase K_j, primero asociaremos a cada objeto $O_i \in MA$, las siguientes magnitudes:

$\Pi_j^+(O_i)$ = # de representantes positivos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

$\Pi_j^-(O_i)$ = # de representantes negativos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

ahora γ_i^j es calculado como:

1) $\gamma_i^j = \dfrac{\Pi_j^+(O_i)}{1+\Pi_j^-(O_i)}$ si $O_i \in K_j$

2) $\gamma_i^j = \dfrac{\Pi_j^-(O_i)}{1+\Pi_j^+(O_i)}$ si $O_i \in CK_j$

Para el cálculo de los parámetros P_v^j v=1,..,n, debido a que se eliminó la restricción de espacios métricos, se tiene que utilizar la estimación basada en la cantidad de representantes en los que se presenta cada rasgo.

Primero separemos de $\{\Omega\}_j$ el subsistema $\{\Omega\}_j^i$ de conjuntos de apoyo que contienen al rasgo x_i. Para cada $\Omega \in \{\Omega\}_j^i$, de entre las ω-partes $\omega O^1,..,\omega O^t$, calculamos las siguientes magnitudes:

$M^+(\omega K_j)$ = # de representantes positivos para K_j
$M^-(\omega K_j)$ = # de representantes negativos para K_j

análogamente de entre las ω-partes $\omega O^{t+1},..,\omega O^q$, calculamos las siguientes magnitudes:

$M^+(\omega CK_j)$ = # de representantes positivos para K_j
$M^-(\omega CK_j)$ = # de representantes negativos para K_j

y entonces calculamos P_i^j como:

$$P_i^j = \frac{1}{|\{\Omega\}_j^i|} \sum_{\Omega \in \{\Omega\}_j^i} \frac{M^+(\omega K_j)+M^-(\omega CK_j)}{1+M^+(\omega CK_j)+M^-(\omega K_j)}$$

5.3. Determinación de los Parámetros η_j y δ_j

En esta sección se explicará una manera de estimar los valores de los parámetros η_j y δ_j, que en este caso son números reales, los cuales sirven para calcular los representantes de cada clase K_j.

Como siempre la estimación no será muy diferente, pero en este caso se utilizará la segunda forma, puesto que analizar todas las posibilidades en un intervalo real no tiene sentido.

Sea $O^i \in MC$ y $\Omega \in \{\Omega\}_j$, si $O^i \in K_j$, el caso en que $O^i \in CK_j$ se analiza de forma análoga, primero tomemos:

$$t(O^i,\Omega) = \sum_{v=1}^{r} \beta(\omega O_v, \omega O^i)$$

$$w(O^i,\Omega) = \sum_{v=r+1}^{m} \beta(\omega O_v, \omega O^i)$$

Si ωO^i es un candidato a representante positivo para K_j en MA, es decir se parece más a los objetos de K_j que a los de CK_j, entonces, para que el reconocimiento de O^i sea correcto, deberían cumplirse:

$$\eta_j \leq t(O^i,\Omega) \qquad (3)$$

$$\delta_j \geq w(O^i,\Omega) \qquad (3')$$

Si ωO^i es un candidato a representante negativo para K_j en MA, es decir se parece más a los objetos de CK_j que a los de K_j, entonces, para que el reconocimiento de O^i sea correcto, deberían cumplirse:

$$\eta_j > w(O^i,\Omega) \qquad (4)$$

$$\delta_j < t(O^i,\Omega) \qquad (4')$$

Obteniendo (3) y (4) para cada $O^i \in MC$, obtenemos un sistema de desigualdades impropio, para η_j, para determinar η_j buscamos un subsistema propio maximal y sus soluciones. Análogamente, obteniendo (3') y (4') para cada $O^i \in MC$ obtenemos un sistema de desigualdades impropio, para δ_j, para determinar δ_j buscamos un subsistema propio maximal y sus soluciones. De estas soluciones se seleccionan valores para η_j y δ_j tales que:

1) $\delta_j < \eta_j$
2) Se cumpla el mayor número de desigualdades del tipo (3) y (3').

5.4. Análisis de Complejidad

En esta sección haremos un breve análisis acerca de la complejidad del proceso de estimación de parámetros en el método extendido de clasificación usando conjuntos de representantes. Analizaremos separadamente la estimación de cada uno de los parámetros.

Sea n el número de rasgos del problema, m el número de objetos de la matriz de aprendizaje, q el número de objetos de la matriz de control, ℓ el número de clases, m_i el número de objetos de aprendizaje en la clase K_i y q_i el número de objetos de control en la clase K_i, $i=1,\dots,\ell$.

5.4.1. Estimación de $\{\Omega\}_j$

La estimación del sistema de conjuntos de apoyo, para una clase K_i, involucra 4 pasos que son: obtención de candidatos, evaluación de candidatos, ordenamiento de los candidatos y selección del sistema. En estos dos últimos pasos sólo se hacen comparaciones entre números, por lo cual no impactan significativamente en el proceso, y además no requieren de espacio adicional. Por este motivo el análisis de complejidad lo haremos atendiendo solamente a los dos primeros pasos.

Cuando se está trabajando con la clase K_i, por la manera en que se obtienen los candidatos, se hacen en total

$$m_i q_i+(m-m_i)(q-q_i)$$

comparaciones entre ω-partes de objetos. Para evaluar cada candidato se hacen:

$m_i q$	si se obtuvo de comparar en K_i
$(m-m_i)q$	si se obtuvo de comparar en CK_i

comparaciones entre ω-partes de objetos. Por lo cual en total se hacen:

$$m_i^2 q_i q+(m-m_i)^2(q-q_i)q$$

comparaciones entre ω-partes de objetos. Como esto se repite para cada clase, en total se hacen:

$$q\sum_{i=1}^{\ell}\left[m_i^2 q_i + (m - m_i)^2 (q - q_i)\right]$$

comparaciones entre ω-partes de objetos. Como los candidatos son subconjuntos de rasgos y no se pueden repetir, el máximo número de candidatos es:

$$\mathit{min}(\ m_i q_i + (m - m_i)(q - q_i)\ ,\ 2^n\ )$$

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje en K_i y q_i objetos de control en K_i, $i=1,\ldots,\ell$, la complejidad en el tiempo, de la estimación de los sistemas de conjuntos de apoyo, es de:

$$q\sum_{i=1}^{\ell}\left[m_i^2 q_i + (m - m_i)^2 (q - q_i)\right]$$

comparaciones entre ω-partes de objetos, y la complejidad en el espacio es de:

$$\mathit{min}(\ m_i q_i + (m - m_i)(q - q_i)\ ,\ 2^n\ )T(c)$$

donde T(c) es el espacio necesario para almacenar un candidato a conjunto de apoyo.

5.4.2. Estimación de Pesos Informacionales

Para calcular la complejidad del proceso de estimación de los pesos informacionales supondremos que se tienen k_i conjuntos de apoyo para cada clase K_i, $i=1,\ldots,\ell$, y que se tienen calculados los representantes, para cada clase K_i, y se encontraron r_i representantes en K_i, $i=1,\ldots,\ell$.

Cuando se está trabajando con la clase K_i, para el cálculo de las magnitudes Π^+ y Π^-, cada objeto de aprendizaje se compara contra todos los representantes de K_i, por lo cual se hacen:

$$mr_i$$

comparaciones entre partes de objetos, y como esto se repite para cada clase, entonces se hacen en total:

$$m\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos.

Para almacenar los pesos informacionales de los objetos, sólo se guarda un número de punto flotante por cada peso informacional, de cada objeto en cada clase, por lo tanto se guardan en total

$$m\ell$$

pesos informacionales.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje en K_i, q_i objetos de control en K_i, k_i conjuntos de apoyo para K_i y r_i representantes en K_i, $i=1,...,\ell$, la complejidad en el tiempo, de la estimación de los pesos informacionales de los objetos, es de:

$$m\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$m\ell T(\text{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

Para el cálculo de los pesos informacionales de los rasgos, cuando se está trabajando con la clase K_i, en el peor de los casos se tienen que comparar, contra los objetos de la matriz de control, los representantes formados con todos los conjuntos de $\{\Omega\}_i$, es decir, todos los representantes de K_i, por lo tanto se hacen:

$$nqr_i$$

comparaciones entre partes de objetos y como esto se repite para cada clase, entonces se hacen en total:

$$nq\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos.

Para almacenar los pesos informacionales de los rasgos, sólo se guarda un número de punto flotante por cada peso informacional, de cada rasgo en cada clase, por lo tanto se guardan en total:

$$n\ell$$

pesos informacionales.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje en K_i, q_i objetos de control en K_i, k_i conjuntos de apoyo para K_i y r_i representantes en K_i, $i=1,\ldots,\ell$, la complejidad en el tiempo, de la estimación de los pesos informacionales de los rasgos, es de:

$$nq\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$n\ell T(\text{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

5.4.3. Estimación de los Parámetros η_i y δ_i

Para calcular la complejidad del proceso de estimación de los parámetros η_i y δ_i supondremos que se tienen k_i conjuntos de apoyo para cada clase K_i, $i=1,\ldots,\ell$.

Cuando se está trabajando con la clase K_i, cada objeto de control se compara, con respecto a cada conjunto de apoyo, contra todos los objetos de aprendizaje, por lo cual se hacen:

$$qmk_i$$

comparaciones entre partes de objetos y como esto se repite para cada clase, entonces se hacen en total:

$$qm\sum_{i=1}^{\ell} k_i$$

comparaciones entre partes de objetos.

Para almacenar los parámetros η_i y δ_i, sólo se guarda un número de punto flotante para cada uno, en cada clase, por lo tanto se guardan en total

$$2\ell$$

números de punto flotante.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje en K_i, q_i objetos de control en K_i y k_i conjuntos de apoyo para K_i, $i=1,\ldots,\ell$, la complejidad en el tiempo, de la estimación de los parámetros η_i y δ_i, es de:

$$qm\sum_{i=1}^{\ell} k_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$2\ell T(\text{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

5.5. Conclusión

En este capítulo hemos explicado la manera de estimar los parámetros que se utilizan en el método extendido de clasificación usando conjuntos de representantes, cuando se permite distinta función de semejanza. Este mecanismo de estimación de parámetros permite calcular los sistemas de conjuntos de apoyo y los umbrales η_j y δ_j, los cuales sirven para determinar si las combinaciones de valores son representantes. Además, suponiendo que se tienen calculados los sistemas de conjuntos de apoyo, y los umbrales η_j y δ_j, se estiman los pesos informacionales de los rasgos y de los objetos.

Esta extensión del mecanismo de estimación de parámetros constituye parte de la aportación de esta tesis.

Capítulo 6

Estimación de Parámetros con Clases Difusas

En este capítulo se describirá una manera de estimar los parámetros para el método extendido de clasificación usando conjuntos de representantes, poniendo especial atención al método resultante del problema de clases difusas. Estas extensiones forman parte de la aportación de esta tesis.

Se considera un problema con un universo dividido en ℓ clases difusas, y con objetos descritos en términos de n rasgos de cualquier tipo, pero con criterios de comparación booleanos, además se tiene una función β, de semejanza entre partes de objetos, que toma valores en [0,1]. Consideremos que la matriz de aprendizaje MA tiene m objetos, $O_1,\ldots,O_m$, y que se tiene una matriz de control, la cual denotaremos por MC, con q objetos, $O^1,\ldots,O^q$.

6.1. Construcción de los Sistemas de Conjuntos de Apoyo $\{\Omega\}_j$

En esta sección se describirá un método para construir los sistemas de conjuntos de apoyo que se utilizan en el modelo difuso de clasificación usando conjuntos de representantes.

La manera de construir los conjuntos de apoyo en el modelo difuso no difiere mucho de la manera en que se construían originalmente, esto debido a que en dicha construcción se consideraba solamente el criterio de existencia de objetos que coincidan en determinado subconjunto de rasgos, lo cual se mantiene cuando se resuelve el problema de infinitud restringiendo los representantes a valores de la matriz de aprendizaje. La selección final de los conjuntos se hace atendiendo al parecido de los objetos en los rasgos de cada conjunto, para lo cual sí interviene el hecho de tener distinta función de semejanza, y el hecho de que las clases sean difusas.

Si se quiere determinar el sistema de conjuntos de apoyo $\{\Omega\}_j$, para la clase K_j, primero separemos MC en 2 tablas:

T_3 contiene los objetos O en MC tales que $\alpha_j(O) \geq 0.5$.
T_4 contiene los objetos O en MC tales que $\alpha_j(O) < 0.5$.

para simplificar la notación numeraremos los objetos de la siguiente manera:

$$T_3 = \{ O^1, \dots ,O^t \}$$
$$T_4 = \{ O^{t+1}, \dots ,O^q \}$$

esquemáticamente queda:

	MC	x_1	...	x_n	
	O^1	$x_1(O^1)$	...	$x_n(O^1)$	
$\alpha_j(O_i) \geq 0.5$	:				T_3
	O^t	$x_1(O^t)$	...	$x_n(O^t)$	
	O^{t+1}	$x_1(O^{t+1})$	...	$x_n(O^{t+1})$	
$\alpha_j(O_i) < 0.5$	:				T_4
	O^q	$x_1(O^q)$	...	$x_n(O^q)$	

Nuestro objetivo consiste en construir una familia de subconjuntos de rasgos, tales que se cumplan las siguientes condiciones:

a) Los objetos de T_3 se parezcan a K_j tanto como sea posible, así como los de T_4 se parezcan a CK_j tanto como sea posible.

a') Los objetos de T_3 difieran de CK_j tanto como sea posible, así como los de T_4 difieran de K_j tanto como sea posible.

Primero para cada objeto $O^k \in T_3$ y cada objeto $O_i \in MA$ se construye el conjunto $M(O^k,O_i)$, el cual contiene a los rasgos en los cuales coinciden, O^k y O_i, atendiendo a los criterios de comparación de cada rasgo, análogamente se construyen los conjuntos $M(O^r,O_u)$ con $O^r \in T_4$ y $O_u \in MA$. Estos conjuntos sirven para garantizar que los valores de los representantes que reconozcan a cada objeto se presenten en MA. En la construcción de $\{\Omega\}_j$ se deben preservar las siguientes condiciones:

c) Para k fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^k,O_i)$, o uno de sus subconjuntos no vacío.

c') Para r fijo, $\{\Omega\}_j$ necesariamente debe incluir uno de los $M(O^r,O_u)$ o uno de sus subconjuntos no vacío.

A cada uno de los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in MA$, le asignamos las siguientes características numéricas:

$$P^+(M(O^k,O_i)) = \sum_{v=1}^{m}\sum_{s=1}^{t}\beta(\omega O^s,\omega O_v)\alpha_j(O_v)$$

$$P^-(M(O^k,O_i)) = \sum_{v=1}^{m}\sum_{s=t+1}^{q}\beta(\omega O^s,\omega O_v)\alpha_j(O_v)$$

De la misma manera, a cada conjunto $M(O^r,O_u)$, $O^r \in T_4$ y $O_u \in MA$, se le asignan las siguientes características numéricas:

$$P^+(M(O^r,O_u)) = \sum_{v=1}^{m}\sum_{s=t+1}^{q}\beta(\omega O^s,\omega O_v)(1-\alpha_j(O_v))$$

$$P^-(M(O^r,O_u)) = \sum_{v=1}^{m}\sum_{s=1}^{t}\beta(\omega O^s,\omega O_v)(1-\alpha_j(O_v))$$

Estas características nos indican qué tanto se parecen los objetos de MA y MC considerando pertenencias a la misma clase y considerando pertenencias a diferentes clases, considerando como únicas clases K_j y CK_j. Cada uno de estos conjuntos se considerará en $\{\Omega\}_j$ mientras mayor sea P^+ y menor sea P^-, la selección se hace de alguna de las siguientes formas:

Modo 1

Se introduce una función de calidad W, para cada conjunto, que crece monótonamente respecto a P^+ y decrece monótonamente respecto a P^-, por ejemplo:

$$W = \frac{P^+}{1+P^-}$$

Con esta función se ordenan los conjuntos $M(O^k,O_i)$, con $O^k \in T_3$ y $O_i \in MA$, en el orden de decrecimiento de W, y se incluyen en $\{\Omega\}_j$ los primeros conjuntos en este orden, teniendo en cuenta que el número de conjuntos incluidos debe preservar la condición (c). Análogamente se ordenan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in MA$, y se incluyen los primeros en el orden, hasta que se cumpla la condición (c').

Modo 2

Se introduce un orden parcial para los conjuntos $M(O^k,O_i)$, donde $O^k \in T_3$ y $O_i \in MA$, de la siguiente manera:

$$M_1 \geq M_2 \quad \text{si} \quad P_1^+ \geq P_2^+ \wedge P_1^- \leq P_2^-$$

La selección de los conjuntos se realiza comenzando con los elementos maximales y se prosigue hasta que se cumpla la condición (c).

Análogamente se introduce un orden parcial y se seleccionan los conjuntos $M(O^r,O_u)$, donde $O^r \in T_4$ y $O_u \in MA$, hasta que se cumpla la condición (c').

6.2. Estimación de los Pesos Informacionales

En esta sección se estimarán los parámetros γ_v^j, v=1,..,m, y P_u^j, u=1,..,n, para lo cual supondremos que ya se han determinado los sistemas de conjuntos de apoyo $\{\Omega\}_j$ y los parámetros η_j y δ_j. Esta estimación igualmente no difiere gran cosa de la hecha para el método original, sólo que aquí se usan las nuevas definiciones de representante positivo y negativo.

Para calcular los pesos informacionales de los objetos, en la clase K_j, primero asociaremos a cada objeto $O_i \in MA$, las siguientes magnitudes:

$\Pi_j^+(O_i)$ = # de representantes positivos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

$\Pi_j^-(O_i)$ = # de representantes negativos para K_j que se obtienen al comparar las ω-partes de O_i con los objetos de MA, respecto a $\{\Omega\}_j$.

ahora γ_i^j es calculado como:

$$1)\ \gamma_i^j = \frac{\Pi_j^+(O_i)}{1+\Pi_j^-(O_i)} \qquad \text{si } \alpha_j(O_i) \geq 0.5$$

$$2)\ \gamma_i^j = \frac{\Pi_j^-(O_i)}{1+\Pi_j^+(O_i)} \qquad \text{si } \alpha_j(O_i) < 0.5$$

Para el cálculo de los parámetros P_v^j v=1,..,n, debido a que se eliminó la restricción de espacios métricos, se tiene que utilizar la estimación basada en la cantidad de representantes en los que se presenta cada rasgo.

Primero separemos de $\{\Omega\}_j$ el subsistema $\{\Omega\}_j^i$ de conjuntos de apoyo que contienen al rasgo x_i. Para cada $\Omega \in \{\Omega\}_j^i$, de entre las ω-partes $\omega O^1,..,\omega O^t$, calculamos las siguientes magnitudes:

$M^+(\omega K_j)$ = # de representantes positivos para K_j
$M^-(\omega K_j)$ = # de representantes negativos para K_j

análogamente de entre las ω-partes $\omega O^{t+1},..,\omega O^q$, calculamos las siguientes magnitudes:

$M^+(\omega CK_j)$ = # de representantes positivos para K_j
$M^-(\omega CK_j)$ = # de representantes negativos para K_j

y entonces calculamos P_i^j como:

$$P_i^j = \frac{1}{|\{\Omega\}_j^i|} \sum_{\Omega \in \{\Omega\}_j^i} \frac{M^+(\omega K_j)+M^-(\omega CK_j)}{1+M^+(\omega CK_j)+M^-(\omega K_j)}$$

6.3. Determinación de los Parámetros η_j y δ_j

En esta sección se explicará una manera de estimar los valores de los parámetros η_j y δ_j, que en este caso son números reales, los cuales sirven para calcular los representantes de cada clase K_j.

Como siempre la estimación no será muy diferente, pero en este caso se utilizará la segunda forma, puesto que analizar todas las posibilidades en un intervalo real no tiene sentido.

Sea $O^i \in$ MC y $\Omega \in \{\Omega\}_j$, si $\alpha_j(O^i) \geq 0.5$, el caso en que $\alpha_j(O^i) < 0.5$ se analiza de forma análoga, primero tomemos:

$$t(O^i,\Omega) = \sum_{v=1}^{m} \beta(\omega O_v, \omega O^i)\alpha_j(O_v)$$

$$w(O^i,\Omega) = \sum_{v=1}^{m} \beta(\omega O_v, \omega O^i)(1-\alpha_j(O_v))$$

Si ωO^i es un candidato a representante positivo para K_j en MA, es decir, $t(O^i,\Omega)>w(O^i,\Omega)$, lo cual significa que se parece más a los objetos de K_j que a los de CK_j, entonces para que el reconocimiento de O^i sea correcto, deberían cumplirse:

$$\eta_j \leq t(O^i,\Omega) \qquad (3)$$
$$\delta_j \geq w(O^i,\Omega) \qquad (3')$$

si ωO^i es un candidato a representante negativo para K_j en MA, es decir, $t(O^i,\Omega)<w(O^i,\Omega)$, lo cual significa que se parece más a los objetos de CK_j que a los de K_j, entonces para que el reconocimiento de O^i sea correcto, deberían cumplirse:

$$\eta_j > w(O^i,\Omega) \qquad (4)$$
$$\delta_j < t(O^i,\Omega) \qquad (4')$$

Obteniendo (3) y (4) para cada $O^i \in$ MC, obtenemos un sistema de desigualdades impropio, para η_j, para determinar η_j buscamos un subsistema propio maximal y sus soluciones. Análogamente, obteniendo (3') y (4') para cada $O^i \in$ MC obtenemos un sistema de desigualdades impropio, para δ_j, para determinar δ_j buscamos un subsistema propio maximal y sus soluciones. De estas soluciones se seleccionan valores para η_j y δ_j tales que:

1) $\delta_j < \eta_j$
2) Se cumpla el mayor número de desigualdades del tipo (3) y (3').

6.4. Análisis de Complejidad

En esta sección haremos un breve análisis acerca de la complejidad del proceso de estimación de parámetros en el método extendido de clasificación usando conjuntos de representantes, en el caso en que las clases son difusas.

Como veremos, la complejidad del mecanismo de estimación de parámetros, para un problema de clases difusas, es la misma que la calculada en el capítulo anterior, excepto en la estimación de los sistemas de conjuntos de apoyo.

Sea n el número de rasgos del problema, m el número de objetos de la matriz de aprendizaje, q el número de objetos de la matriz de control, ℓ el número de clases, m_i el número de objetos de aprendizaje con pertenencia ≥ 0.5 a K_i y q_i el número de objetos de control con pertenencia ≥ 0.5 a K_i, $i=1,\ldots,\ell$.

6.4.1. Estimación de $\{\Omega\}_j$

La estimación del sistema de conjuntos de apoyo, para una clase K_i, involucra 4 pasos que son: obtención de candidatos, evaluación de candidatos, ordenamiento de los candidatos y selección del sistema. En estos dos últimos pasos sólo se hacen comparaciones entre números, por lo cual no impactan significativamente en el proceso, y además no requieren de espacio adicional. Por este motivo el análisis de complejidad lo haremos atendiendo solamente a los dos primeros pasos.

Cuando se está trabajando con la clase K_i, por la manera en que se obtienen los candidatos, cada objeto de control se compara con todos los objetos de aprendizaje, por lo tanto se hacen en total

$$mq$$

comparaciones entre ω-partes de objetos. Para evaluar cada candidato se hacen:

$$mq$$

comparaciones entre ω-partes de objetos. Por lo cual en total se hacen:

$$m^2q^2$$

comparaciones entre ω-partes de objetos. Como esto se repite para cada clase, en total se hacen:

$$\ell m^2q^2$$

comparaciones entre ω-partes de objetos. Como los candidatos son subconjuntos de rasgos y no se pueden repetir, el máximo número de candidatos es:

$$min(\ mq\ ,\ 2^n\ )$$

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje con pertenencia ≥ 0.5 a K_i y q_i objetos de control con pertenencia ≥ 0.5 a K_i, $i=1,...,\ell$, la complejidad en el tiempo, de la estimación de los sistemas de conjuntos de apoyo, es de:

$$\ell m^2 q^2$$

comparaciones entre ω-partes de objetos, y la complejidad en el espacio es de:

$$min(mq , 2^n)T(c)$$

donde T(c) es el espacio necesario para almacenar un candidato a conjunto de apoyo.

6.4.2. Estimación de Pesos Informacionales

Para calcular la complejidad del proceso de estimación de los pesos informacionales supondremos que se tienen k_i conjuntos de apoyo para cada clase K_i, $i=1,...,\ell$, y que se tienen calculados los representantes, para cada clase K_i, y se encontraron r_i representantes en K_i, $i=1,...,\ell$.

Cuando se está trabajando con la clase K_i, para el cálculo de las magnitudes Π^+ y Π^-, cada objeto de aprendizaje se compara contra todos los representantes de K_i, por lo cual se hacen:

$$mr_i$$

comparaciones entre partes de objetos y como esto se repite para cada clase, entonces se hacen en total:

$$m\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos.

Para almacenar los pesos informacionales de los objetos, sólo se guarda un número de punto flotante por cada peso informacional, de cada objeto en cada clase, por lo tanto se guardan en total

$$m\ell$$

pesos informacionales.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje con pertenencia ≥ 0.5 a K_i, q_i objetos de control con pertenencia ≥ 0.5 a K_i, k_i conjuntos de apoyo para K_i y r_i representantes en K_i, $i=1,...,\ell$, la complejidad en el tiempo, de la estimación de los pesos informacionales de los objetos, es de:

$$m\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$m\ell T(\text{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

Para el cálculo de los pesos informacionales de los rasgos, cuando se está trabajando con la clase K_i, en el peor de los casos se tienen que comparar, contra los objetos de la matriz de control, los representantes formados con todos los conjuntos de $\{\Omega\}_i$, es decir, todos los representantes de K_i, por lo tanto se hacen:

$$nqr_i$$

comparaciones entre partes de objetos y como esto se repite para cada clase, entonces se hacen en total:

$$nq\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos.

Para almacenar los pesos informacionales de los rasgos, sólo se guarda un número de punto flotante por cada peso informacional, de cada rasgo en cada clase, por lo tanto se guardan en total:

$$n\ell$$

pesos informacionales.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje con pertenencia ≥ 0.5 a K_i, q_i objetos de control con pertenencia ≥ 0.5 a K_i, k_i conjuntos de apoyo para K_i y r_i representantes en K_i, $i=1,...,\ell$, la complejidad en el tiempo, de la estimación de los pesos informacionales de los rasgos, es de:

$$nq\sum_{i=1}^{\ell} r_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$n\ell T(\text{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

6.4.3. Estimación de los parámetros η_i y δ_i

Para calcular la complejidad del proceso de estimación de los parámetros η_i y δ_i supondremos que se tienen k_i conjuntos de apoyo para cada clase K_i, $i=1,\ldots,\ell$.

Cuando se está trabajando con la clase K_i, cada objeto de control se compara, con respecto a cada conjunto de apoyo, contra todos los objetos de aprendizaje, por lo cual se hacen:

$$qmk_i$$

comparaciones entre partes de objetos y como esto se repite para cada clase, entonces se hacen en total:

$$qm\sum_{i=1}^{\ell} k_i$$

comparaciones entre partes de objetos.

Para almacenar los parámetros η_i y δ_i, sólo se guarda un número de punto flotante para cada uno, en cada clase, por lo tanto se guardan en total

$$2\ell$$

números de punto flotante.

Concluyendo, para un problema de n rasgos, m objetos de aprendizaje, q objetos de control, ℓ clases, m_i objetos de aprendizaje con pertenencia ≥ 0.5 a K_i, q_i objetos de control con pertenencia ≥ 0.5 a K_i y k_i conjuntos de apoyo para K_i, $i=1,\ldots,\ell$, la complejidad en el tiempo, de la estimación de los parámetros η_i y δ_i, es de:

$$qm\sum_{i=1}^{\ell} k_i$$

comparaciones entre partes de objetos, y la complejidad en el espacio es de:

$$2\ell T(\mathrm{float})$$

donde T(float) es el espacio necesario para almacenar un número de punto flotante.

6.5. Conclusión

En este capítulo hemos explicado la manera de estimar los parámetros que se utilizan en el método extendido de clasificación usando conjuntos de representantes, cuando las clases son difusas. Este mecanismo de estimación de parámetros permite calcular los sistemas de conjuntos de apoyo y los umbrales η_j y δ_j, los cuales sirven para determinar si las combinaciones de valores son representantes. Además, suponiendo que se tienen calculados los sistemas de conjuntos de apoyo, y los umbrales η_j y δ_j, los pesos informacionales de los rasgos y de los objetos.

Esta extensión del mecanismo de estimación de parámetros constituye parte de la aportación de esta tesis.

Capítulo 7

Sistema C-REP Implementación

En este capítulo se describe la implementación computacional del método extendido de clasificación usando conjuntos de representantes, la cual dió origen al sistema de clasificación con aprendizaje C-REP.

La presente implementación trabaja en una computadora PC basada en un procesador 80386 o mejor, con sistema operativo MS/DOS 5.0 o posterior y Microsoft Windows 3.0 o posterior. Se eligió este tipo de computadoras debido a su gran difusión, eligiendo programar en Windows porque ésta es la interfaz gráfica más popular para computadoras basadas en MS/DOS, la cual permite diseñar interfaces agradables, para el usuario, y además brinda grandes facilidades para el acceso a la memoria extendida.

El sistema está escrito en Borland C/C++ versión 3.1, la elección de C/C++ como lenguaje de programación se basó en el hecho de la gran portabilidad de los programas escritos en este lenguaje, y por la posibilidad de combinar las ventajas de la programación orientada a objetos, con un lenguaje de programación convencional como C. Borland se eligió por ser la versión que se tenía disponible, y porque permite escribir programas para Windows de forma no demasiado complicada.

7.1. Organización del Sistema

En esta sección se describe la organización general del sistema de clasificación con aprendizaje C-REP, el cual cuenta con tres módulos que son:

Módulo de Manejo de Interfaz

Este módulo es el encargado de manejar la interfaz al usuario, controlando el menú principal y los diálogos de captura y notificación, además incluye un submódulo encargado de notificar al usuario acerca del grado de avance de los distintos procesos involucrados en el método de clasificación usando conjuntos de representantes. Este módulo es el que controla la ejecución del sistema, se encarga de tomar las solicitudes, del usuario, las realiza haciendo llamados a las funciones de los otros módulos, y posteriormente toma los resultados y los despliega.

Módulo de Almacenamiento

Este módulo se encarga de crear y mantener las estructuras que sirven para almacenar los datos necesarios para el funcionamiento del método extendido de clasificación usando conjuntos de representantes, además permite almacenar, en el disco de la computadora, y recuperar todas estas estructuras de datos. Para poder recuperar correctamente todos los datos, junto con los mismos se guarda una descripción acerca de los tipos y las cantidades de datos que se están almacenando.

Módulo del Sistema

Este módulo es el más importante del sistema, pues es el encargado de realizar todos los procesos involucrados en el método extendido de clasificación usando conjuntos de representantes. Este módulo se divide en tres partes que son el submódulo de estimación de parámetros, el submódulo de cálculo de representantes y el submódulo de clasificación.

Submódulo de Estimación de Parámetros

Este módulo implementa el mecanismo extendido para la estimación de los parámetros utilizados en el método de clasificación usando conjuntos de representantes, tales como los sistemas de conjuntos de apoyo, los parámetros η_j y los pesos informacionales de los rasgos y de los objetos. Estos parámetros se estiman para cada una de las clases.

Submódulo de Cálculo de Representantes

Es el encargado de calcular, a partir de la matriz de aprendizaje, todos los representantes positivos y negativos, para cada clase, utilizando los parámetros estimados por el submódulo anterior. Además se encarga de calcular la ponderación de cada uno de estos representantes.

Submódulo de Clasificación

Este módulo toma los representantes calculados, junto con sus ponderaciones, y los utiliza para clasificar nuevos objetos.

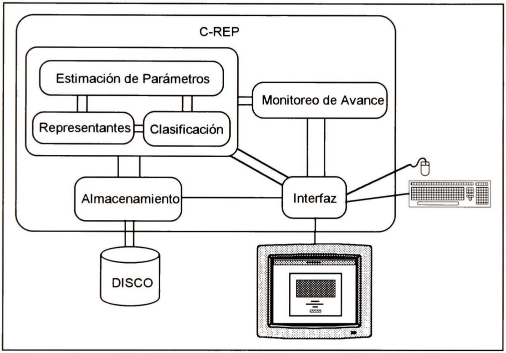


Figura 7.1. Estructura del Sistema C-REP

7.2. Características del Sistema

En esta sección describiremos las características del sistema, es decir, describiremos lo que puede hacer C-REP. C-REP tiene las siguientes características:

- Permite trabajar con rasgos booleanos, k-valentes, reales y nominales.
- Permite trabajar con ausencia de información.
- Permite definir criterios de comparación por igualdad en todos los tipos, por intervalos en el tipo k-valente y por épsilon en el tipo real.
- Permite trabajar con función de semejanza por coincidencia total, con épsilon y con porcentaje mínimo.
- Permite trabajar problemas con clases disjuntas, no disjuntas y difusas.

- Permite leer los datos por teclado o por archivo.
- Permite trabajar con pesos informacionales calculados o dados por el usuario.
- Permite trabajar con cualquier sistema de conjuntos de apoyo.
- Permite estimar los parámetros del modelo.
- Permite almacenar y después recuperar todos los datos calculados, tales como los parámetros y los representantes.
- Permite generar reportes en archivo y en la impresora.
- Permite trabajar con hasta 32 rasgos, hasta 32 clases y con un máximo de 32500 objetos.

7.3. Guía del Usuario

En esta sección daremos una guía rápida, acerca de las funciones accesibles por el usuario, desde los menúes de C-REP.

El menú Principal tiene las siguientes opciones.

Datos Parámetros Representantes Clasificación Salir	**Ayuda**

Cada una de estas opciones nos lleva a un submenú, excepto salir que es la opción que permite terminar la ejecución de C-REP.

7.3.1. Datos

Este submenú contiene las opciones que permiten leer y salvar los datos del sistema, así como las de generación de reportes. Este submenú tiene las siguientes opciones:

| Datos | Parámetros Representantes Clasificación Salir		**Ayuda**
Leer Matriz de Aprendizaje	**>**	
Leer Matriz de Control	**>**	
Salvar Matriz de Aprendizaje		
Salvar Matriz de Control		
Leer Objetos a Clasificar	**>**	
Salvar Objetos a Clasificar		
Reporte de Objetos Clasificados	**>**	

Las opciones **Leer Matriz de Aprendizaje** y **Leer Matriz de Control** permiten leer, por el teclado o de un archivo, la matriz de aprendizaje y la matriz de control, las cuales están formadas por la definición del tipo y características del problema, la descripción de los rasgos con los cuales se definen los objetos, la cantidad de clases y los datos de los objetos de aprendizaje. En caso de que ya exista una matriz de aprendizaje, o de control, esta opción permite crear otra o editar la existente.

Las opciones **Salvar Matriz de Aprendizaje** y **Salvar Matriz de Control** permiten salvar, en un archivo de disco, los datos de la matriz de aprendizaje y la matriz de control, en caso de que se tenga definida alguna.

Las opciones **Leer Objetos a Clasificar** y **Salvar Objetos a Clasificar**, permiten leer y salvar, de archivos en disco, las descripciones de los objetos que serán clasificados.

La opción **Reporte de Objetos Clasificados** permite generar un reporte, en un archivo o en la impresora, de los resultados de la clasificación.

7.3.2. Parámetros

Este submenú contiene las opciones que permiten calcular y salvar los diferentes parámetros que se requieren para el método de clasificación usando conjuntos de representantes. Este submenú tiene las siguientes opciones:

Datos	Parámetros	Representantes	Clasificación	Salir	Ayuda
	Lee C. de Apoyo Salva C. de Apoyo Calcula C. de Apoyo				
	Leer Nj's Salvar Nj's Calcula Nj's				
	Leer P.I. Rasgos Calcula P.I. Rasgos Salva P.I. Rasgos				
	Leer P.I. Objetos Calcula P.I. Objetos Salva P.I. Objetos				
	Semejanza >				

Las opciones **Lee C. de Apoyo**, **Salva C. de Apoyo** y **Calcula C. de Apoyo** permiten leer, salvar y calcular los sistemas de conjuntos de apoyo. Para calcular los sistemas de conjuntos de apoyo se necesita tener definidas la matriz de aprendizaje y la matriz de control.

Las opciones **Leer Nj's**, **Salvar Nj's** y **Calcula Nj's** permiten leer, salvar y calcular los parámetros η_j. Para calcular los parámetros η_j se necesita tener definidos la matriz de aprendizaje, la matriz de control y los sistemas de conjuntos de apoyo.

Las opciones **Leer P.I. Rasgos**, **Calcula P.I. Rasgos** y **Salva P.I. Rasgos** permiten calcular, salvar y recuperar los pesos informacionales de los rasgos. Para calcular los pesos informacionales de los rasgos se necesita tener definidos la matriz de aprendizaje, la matriz de control y los sistemas de conjuntos de apoyo, además ya se deben haber calculado los representantes.

Las opciones **Leer P.I. Objetos**, **Calcula P.I. Objetos** y **Salva P.I. Objetos** permiten calcular, salvar y recuperar los pesos informacionales de los objetos. Para calcular los pesos informacionales de los objetos se necesita tener definidos la matriz de aprendizaje, la matriz de control y los sistemas de conjuntos de apoyo, además ya se deben haber calculado los representantes.

La opción **Semejanza** permite seleccionar la función de semejanza, la cual puede ser coincidencia rasgo a rasgo, denominada igualdad, con épsilon o con porcentaje mínimo.

7.3.3. Representantes

Este submenú contiene las opciones que permiten calcular, evaluar, mostrar, leer y salvar los rasgos complejos. Este submenú tiene las siguientes opciones:

Datos	Parámetros	Representantes	Clasificación	Salir	Ayuda
		Calcular			
		Evaluar			
		Mostrar			
		Salvar			
		Leer			

Para poder calcular los representantes deben haberse definido la matriz de aprendizaje, los sistemas de conjuntos de apoyo y los parámetros η_j, si ya se tienen los pesos informacionales, de rasgos y de objetos, entonces los representantes serán evaluados, en caso contrario, a cada uno se le asignará una ponderación 1.0, y podrán ser evaluados posteriormente.

7.3.4. Clasificación

Este submenú contiene las opciones que permiten clasificar nuevos objetos. Este submenú tiene las siguientes opciones:

Datos	Parámetros	Representantes	Clasificación	Salir	Ayuda
			Clasifica		
			Muestra		

La opción **Clasifica** permite clasificar objetos, previamente leídos, usando los representantes calculados.

La opción **Muestra** permite al usuario ver los resultados de la clasificación.

7.3.5. Ayuda

Este submenú contiene las siguientes opciones:

Datos Parámetros Representantes Clasificación Salir	**Ayuda**
	Acerca de ...

La opción **Acerca de ...** permite ver el diálogo de presentación del sistema de clasificación con aprendizaje C-REP.

7.4. Formato de los Archivos

En esta sección se describen los formatos de los principales archivos de datos utilizados por el sistema C-REP, cada uno de estos archivos está almacenado como texto ASCII, y pueden ser creados con cualquier procesador de texto que guarde sólo los códigos de los caracteres.

Archivos *.MAT

Estos archivos sirven para almacenar los datos de la matriz de aprendizaje, o de control, y tienen el siguiente formato:

Número de rasgos
Rasgos
Tipo de problema
Función de semejanza
Número de clases
Número de objetos
Objetos

Donde cada rasgo incluye los siguientes datos:

Peso informacional del rasgo
Tipo del rasgo
Tipo de comparación
Datos adicionales

Los datos adicionales sólo se incluyen cuando el rasgo es k-valente, real o nominal, y representan el mínimo y máximo valor k-valente, el mínimo y máximo valor real, o los valores nominales. Cada uno de los objetos incluye:

Nombre del objeto
Peso informacional del objeto
Valores para cada rasgo
Pertenencia a cada clase

Los valores de cada rasgo están en correspondencia con el tipo de los mismos, y la pertenencia a cada clase es 0 o 1 para problemas duros disjuntos, o no, o un número real en [0,1] para problemas de clases difusas.

El ejemplo siguiente muestra los datos de un problema con 5 rasgos booleanos y tres clases duras disjuntas, con función de semejanza por igualdad, cada una con tres objetos, siendo en total 9 objetos.

```
5

1.00 0 0
1.00 0 0
1.00 0 0
1.00 0 0
1.00 0 0

201  301

3  9

O1 1.0  1 1 0 1 1   1 0 0
O2 1.0  0 1 0 1 1   1 0 0
O3 1.0  1 0 1 0 1   1 0 0
O4 1.0  1 1 1 1 1   0 1 0
O5 1.0  0 0 0 1 1   0 1 0
O6 1.0  0 0 0 0 1   0 1 0
O7 1.0  0 0 0 0 0   0 0 1
O8 1.0  1 1 0 0 0   0 0 1
O9 1.0  1 0 0 0 0   0 0 1
```

Archivos *.SCA

Estos archivos sirven para almacenar los sistemas de conjuntos de apoyo con el que trabajará el sistema, y tienen el siguiente formato:

Número de clases.
Número de conjuntos de apoyo de la clase 1
Conjuntos de apoyo de la clase 1
:
Número de conjuntos de apoyo de la clase l
Conjuntos de apoyo de la clase l

Donde cada conjunto de apoyo incluye los siguientes datos

Número de rasgos que lo integran
Indice de cada rasgo

El ejemplo siguiente muestra los datos de un archivo con el sistema de conjuntos de apoyo formado por $\{x_1,x_2\}$, $\{x_1,x_2,x_4,x_5\}$ y $\{x_2,x_3,x_5\}$, para cada una de las 2 clases.

2

3
2 1 2
4 1 2 4 5
3 2 3 5

3
2 1 2
4 1 2 4 5
3 2 3 5

Archivos *.PIO y *.PIR

Estos archivos sirven para almacenar los pesos informacionales de los objetos y de los rasgos, respectivamente, y tienen el siguiente formato:

Número de Clases
Número de pesos informacionales
Pesos informacionales de la clase 1
:
Pesos informacionales de la clase ℓ

Donde cada peso informacional es un número real en [0,1].

El ejemplo siguiente muestra un archivo con tres pesos informacionales, para cada una de las 2 clases.

```
2 3
0.85732 0.67384 1.00000
0.45879 0.97658 0.75689
```

Archivos *.OPC

Estos archivos sirven para almacenar los datos de los objetos que van a ser clasificados, y tienen el siguiente formato:

Número de objetos a clasificar
Objetos a clasificar

Donde los objetos a clasificar se describen en el mismo formato que los objetos de la matriz de aprendizaje, solamente que sin las pertenencias a las clases. El valor de los pesos informacionales no será tomado en cuenta.

El ejemplo siguiente muestra un archivo con 6 objetos a clasificar.

```
6

OC1 1.0 0 1 0 1 0
OC2 1.0 1 0 1 0 1
```

OC3 1.0 0 0 1 1 1
OC4 1.0 0 0 0 1 0
OC5 1.0 1 1 0 1 0
OC6 1.0 0 1 1 1 1

Archivos *.PNJ

Estos archivos sirven para almacenar los datos de los parámetros η_j y δ_j, y tienen el siguiente formato:

Número de clases
Valores de los η_j
Valores de los δ_j

Donde los valores de η_j son números reales.

Archivos *.REP

Estos archivos sirven para almacenar los datos de los representantes positivos y negativos, y tienen el siguiente formato:

Número de rasgos
Rasgos
Tipo de problema
Función de semejanza
Número de clases
Número de representantes positivos para la clase 1
Representantes positivos de la clase 1
Número de representantes negativos para la clase 1
Representantes negativos de la clase 1
⋮
Número de representantes positivos para la clase ℓ
Representantes positivos de la clase ℓ
Número de representantes negativos para la clase ℓ
Representantes negativos de la clase ℓ

Donde el número de rasgos y la descripción de los mismos tiene el mismo formato que en los archivos *.MAT. Cada representante, positivo o negativo, tiene el siguiente formato:

Ponderación del representante
Número asociado con el vector característico del conjunto de apoyo
Número del primer rasgo
Valor del primer rasgo
⋮
Número del último rasgo
Valor del último rasgo

El ejemplo siguiente muestra un archivo con 2 representantes positivos y 2 negativos, para cada una de las tres clases, siendo clases duras disjuntas, con función de semejanza por igualdad y con objetos descritos en términos de 5 rasgos booleanos.

```
5

1.00 0 0
1.00 0 0
1.00 0 0
1.00 0 0
1.00 0 0

201  301

3

2
6.00 25 0 0  1 1  4 1
1.00 21 0 0  2 1  4 1
2
6.00 21 0 1  2 0  4 0
3.00 19 0 0  3 0  4 1
2
9.00 13 1 1  2 0  4 1
6.00 11 1 1  3 1  4 1
2
3.00 28 0 1  1 0  2 0
3.00 25 0 1  1 0  4 0
2
6.00 22 0 1  2 0  3 0
6.00 19 0 1  3 0  4 0
2
3.00 13 1 0  2 0  4 0
6.00 7 2 0  3 0  4 0
```

7.5. Estructuras de Datos

En esta sección se describen las principales estructuras de datos utilizadas en la implementación de C-REP. Estas estructuras, se encuentran definidas en el archivo PCR.H, y sirven para almacenar los datos de la matriz de aprendizaje, de los objetos y rasgos, así como los rasgos complejos encontrados y los resultados de la clasificación.

Las estructuras son explicadas en el siguiente formato:

Declaración:

```
struct NOMBRE {
        ...
};
```

Descripción:

Una breve descripción de la información que almacena la estructura.

Miembros:

En esta sección se describen los miembros de la estructura, explicando para que se utilizan.

A continuación describiremos las principales estructuras de C-REP.

```
struct rasgo {
        int     tipo;
        int     comp;
        int     maxk;
        float   min, max;
        float   epsilon;
        lista   *v;
        ilista  *ints;

        rasgo();
        ~rasgo();
};
```

Descripción:

La estructura **rasgo** almacena la descripción de un rasgo, esta estructura es utilizada para guardar la descripción del tipo de datos que contendrá la matriz de aprendizaje.

Miembros :

tipo es un número entero que indica el tipo de valor que puede tomar el rasgo, los valores que actualmente se manejan son:

0 booleano
1 k-valente
2 real
3 nominal

cmp este parámetro indica el criterio de comparación que se utilizará, y depende del tipo del rasgo. Los criterios actualmente permitidos son:

booleano 0 igualdad

k-valente	0 igualdad, 1 por intervalos
real	0 igualdad, 1 con épsilon
nominal	0 igualdad

maxk en este parámetro se almacena el número máximo que puede tomar el rasgo en caso de que sea de tipo k-valente.

min este parámetro almacena el valor mínimo que puede tomar el rasgo en caso de que sea real.

max este parámetro almacena el valor máximo que puede tomar el rasgo en caso de que sea real.

epsilon este parámetro almacena el épsilon para las comparaciones, en caso de que el rasgo sea real y el criterio de comparación sea con épsilon.

****v*** este parámetro apunta a una lista de palabras que contendrán los valores nominales, en caso de que el rasgo sea nominal.

****ints*** este parámetro apunta a una lista de enteros que definirán los intervalos, en caso de que el rasgo sea k-valente y el criterio de comparación sea por intervalos.

rasgo() Construye la estructura y la inicializa.

~rasgo() Destruye la estructura, destruyendo también las listas de *v y *ints.

```
struct valor {
        char    num;
        char    aus;
        int     vi;
        float   vr;
        valor   *sig;

        valor();
};
```

Descripción:

La estructura **valor** almacena el valor de un rasgo, esta estructura es utilizada para guardar los datos de la matriz de aprendizaje.

Miembros:

num es un número entero que indica el número del rasgo del cual se está almacenando el valor.

vi este parámetro almacena un valor entero, se utiliza si el rasgo es booleano, K-valente o nominal, en este último caso el valor es un índice a la lista de valores nominales.

vr este parámetro almacena un valor real, se utiliza si el rasgo es real.

sig este parámetro apunta al siguiente valor, en una lista ligada de valores.

valor() Construye la estructura y la inicializa.

```
struct objetos {
            char      *nom;
            int       ya;
            long      vi;
            valor     *vals;
            objetos   *sig;

            objetos();
            ~objetos();
            void defineNombre(char *n);
};
```

Descripción:

La estructura **objetos** almacena un objeto, esta estructura es utilizada para guardar los datos de la matriz de aprendizaje.

Miembros:

nom apunta a una cadena de caracteres con el nombre del objeto.
ya es un número entero que indica si el objeto ya fue incluido, durante el cálculo de los sistemas de conjuntos de apoyo.
vi este parámetro almacena el vector informacional del objeto.
****vals*** este parámetro apunta a la lista de los valores de los rasgos que describen al objeto.
****sig*** este parámetro apunta al siguiente objeto, en una lista ligada de objetos.

objetos() Construye la estructura y la inicializa.
~objetos() Destruye la estructura, incluyendo la lista de valores de los rasgos.
defineNombre() Define el nombre del objeto, se encarga de buscar espacio en memoria para almacenarlo.

```
struct clases {
            int       nObj;
            objetos   *objs;

            clases();
};
```

Descripción:

La estructura **clases** almacena la descripción de una clase, esta estructura es utilizada para guardar la descripción de la matriz de aprendizaje.

Miembros

nObj este parámetro almacena el número de objetos.
****objs*** este parámetro apunta a los objetos de la clase.

clases() Construye la estructura y la inicializa.

```
struct Crep {
        long     ca;
        float    pi;
        valor    *vals;
        Crep     *sig;

        Crep();
        ~Crep();
};
```

Descripción:

La estructura **Crep** almacena un representante.

Miembros:

ca este parámetro almacena el vector característico del conjunto de apoyo asociado al rasgo complejo.
pi este parámetro almacena la ponderación del representante.
****vals*** este parámetro apunta a los valores de los rasgos que forman el rasgo complejo.
****sig*** este parámetro apunta al siguiente rasgo complejo, en una lista ligada de rasgos complejos.

Crep() Construye la estructura y la inicializa.
~Crep() Destruye la estructura, destruyendo también los valores de los rasgos que la forman.

```
struct cadato {
        long     ca;
        cadato   *sig;

        cadato( long d );
};
```

Descripción:

La estructura **cadato** almacena un conjunto de apoyo.

Miembros:

ca este parámetro almacena el vector característico del conjunto de apoyo.

****sig*** este parámetro apunta al siguiente conjunto de apoyo, en una lista ligada de conjuntos de apoyo.

cadato() Construye la estructura y la inicializa.

```
class calista {
            cadato      *ini
            int         nca;
public:
            calista();
            ~calista();
            void inserta( long d );
            int cuantos();
            int esta( long d );
            cadato *primero();
};
```

Descripción:

La estructura **calista** maneja una lista ligada de conjuntos de apoyo.

Miembros:

****ini*** este parámetro almacena el inicio de la lista ligada de conjuntos de apoyo.

nca este parámetro almacena el número de conjuntos de apoyo contenidos en la lista.

calista() Construye la estructura y la inicializa.

~calista() Destruye la estructura, incluyendo todos los miembros de la lista.

inserta() Inserta un conjunto de apoyo en la lista.

cuantos() Regresa el número de conjuntos de apoyo contenidos en la lista.

esta() Decide si un cierto conjunto de apoyo ya se encuentra en la lista.

primero() Regresa un apuntador al primer elemento de la lista.

7.6. Conclusión

En este capítulo se describió la implementación computacional del método extendido de clasificación usando conjuntos de representantes, el cual es producto de una serie de extensiones hechas al método propuesto por Baskakova y Zhuravliov[1]. El sistema de clasificación con aprendizaje obtenido, C-REP, se describió exponiendo la organización del mismo y explicando sus diferentes módulos, además se mencionaron las características del sistema, las cuales son las siguientes:

- Permite trabajar con rasgos booleanos, k-valentes, reales y nominales.
- Permite trabajar con ausencia de información.
- Permite definir criterios de comparación por igualdad en todos los tipos, por intervalos en el tipo k-valente y por épsilon en el tipo real.
- Permite trabajar con función de semejanza por coincidencia total, con épsilon y con porcentaje mínimo.
- Permite trabajar problemas con clases disjuntas, no disjuntas y difusas.
- Permite leer los datos por teclado o por archivo.
- Permite trabajar con pesos informacionales calculados o dados por el usuario.
- Permite trabajar con cualquier sistema de conjuntos de apoyo.
- Permite estimar los parámetros del modelo.
- Permite almacenar y después recuperar todos los datos calculados, tales como los parámetros y los representantes.
- Permite generar reportes en archivo y en la impresora.
- Permite trabajar con hasta 32 rasgos, hasta 32 clases y con un máximo de 32500 objetos.

También se dió un pequeño manual de usuario del sistema C-REP, se describieron los formatos de los diferentes archivos que utiliza, y se explicaron las principales estructuras de datos del sistema.

Conclusiones

El objetivo de esta tesis era realizar algunas extensiones al método de clasificación usando conjuntos de representantes, y generar a partir de éstas una implementación computacional, del mismo, que permitiera resolver problemas de clasificación con aprendizaje, los cuales no podían ser resueltos con el método de clasificación usando conjuntos de representantes, en su forma original.

Las características originales del método de clasificación usando conjuntos de representantes, así como del método extendido, se engloban en la siguiente tabla:

Características	Método Original	Método Extendido
Más de Dos Clases	Si	Si
Clases Disjuntas	No	Si
Clases No Disjuntas	Si	Si
Clases Difusas	No	Si
Rasgos Métricos	Si	Si
Rasgos No Métricos	No	Si
Criterio de Comparación por Igualdad	Si	Si
Distinto Criterio de Comparación	No	Si
Función de Semejanza por Igualdad	Si	Si
Distinta Función de Semejanza	No	Si
Ausencia de Información	No	Si
Estimación de Parámetros	Si	Si

Adicionalmente el método incluye un mecanismo para estimar los valores de los parámetros utilizados por el modelo, por lo cual también se extendió este mecanismo, de manera que se adecuara a las nuevas características del método de clasificación usando conjuntos de representantes.

Todas estas extensiones fueron motivadas por situaciones reales que se presentaron durante el proceso de modelación matemática de problemas de clasificación en Geofísica y Geología.

La implementación del método extendido de clasificación usando conjuntos de representantes, C-REP, fue realizada en Borland C/C++ 3.1 y diseñada para funcionar en computadoras tipo PC, basadas en un procesador 80386 o mejor, con una interfaz para Microsoft Windows 3.0 o posterior, con lo cual se logró obtener un sistema fácil de usar y que permite la solución de una amplia variedad de problemas de clasificación con aprendizaje.

Para evaluar el funcionamiento del sistema C-REP, éste fue probado con un conjunto de datos proporcionados por el Grupo de Geomatemática del Centro de Investigaciones y Desarrollo del Petróleo de Cuba, los cuales forman una muestra de aprendizaje para la búsqueda de zonas perspectivas de acumulación de hidrocarburos en las secuencias ofiolíticas dentro del territorio insular de la República de Cuba. De estos datos sólo se muestran sus características, y no su significado, ya que forman parte de una investigación, acerca de la cual no se pueden dar todos los detalles.

El problema está constituido por un universo dividido en 13 clases y formado por objetos descritos en términos de 10 rasgos, admitiendo ausencia de información, con la siguiente descripción:

Rasgo	Tipo	Dominio		Comparación
		Mínimo	Máximo	
x_1	k-valente	0	462	Igualdad
x_2	k-valente	0	7	Igualdad
x_3	k-valente	0	10	Igualdad
x_4	k-valente	0	9	Igualdad
x_5	k-valente	0	10	Igualdad
x_6	k-valente	0	5	Igualdad
x_7	k-valente	0	19	Igualdad
x_8	k-valente	0	19	Igualdad
x_9	k-valente	0	17	Igualdad
x_{10}	k-valente	0	3	Igualdad

y con una función de semejanza con un épsilon = 1, es decir:

$$\beta(\omega O, \omega O') = \begin{cases} 1 & \text{si } \omega O \text{ y } \omega O' \text{ difieren en a lo más 1 rasgo} \\ 0 & \text{en otro caso} \end{cases}$$

La matriz de aprendizaje está integrada por 146 objetos, ver apéndice A, agrupados de la siguiente manera:

Clase	# de Objetos
K_1	16
K_2	26
K_3	9
K_4	5
K_5	9
K_6	6
K_7	13
K_8	4
K_9	7
K_{10}	9
K_{11}	6
K_{12}	6
K_{13}	30

Con estos datos se obtuvo la siguiente cantidad de representantes, positivos y negativos, para cada clase.

Clase	# de R. Positivos	# de R. Negativos
K_1	42	341
K_2	152	987
K_3	72	1041
K_4	1	133
K_5	20	315
K_6	14	363
K_7	21	273
K_8	4	225
K_9	0	5
K_{10}	96	1307
K_{11}	7	145
K_{12}	11	533
K_{13}	96	364

Para generar estos representantes, y para el proceso de clasificación, se utilizaron los parámetros obtenidos usando el mecanismo extendido de estimación de parámetros.

La eficiencia de C-REP sobre estos datos fue medida aplicándolo sobre una matriz de control formada por 85 objetos, de los cuales ya se conoce a qué clase pertenecen, la misma que fue utilizada durante el proceso de estimación de parámetros. El resultado fue que de estos 85 objetos se cometieron 2 errores, lo cual nos da una eficiencia del 97.64%, la cual es mayor que la obtenida con el sistema PROGNOSIS[3] después de buscar el algoritmo óptimo.

Naturalmente esto es solamente un ejemplo, con el cual no puede probarse que C-REP obtendrá siempre buenos resultados, pero si demuestra que puede aplicarse para solucionar problemas reales, con una eficiencia suficientemente buena, sin contar que el método de clasificación usando conjuntos de representantes, en su forma original, no podría ser aplicado para resolver este problema, puesto que no cumple con las restricciones del modelo. Otro ejemplo, con el que se probó el sistema C-REP, es el siguiente:

En este ejemplo, los objetos representan puntos en el plano y se describen en términos de los siguientes rasgos:

Rasgo	Tipo	Dominio Mínimo	Dominio Máximo	Comparación
x	real	-2.0	2.0	diferencia < 0.1
y	real	-2.0	2.0	diferencia < 0.1
x+y	real	-4.0	4.0	diferencia < 0.1
x-y	real	-4.0	4.0	diferencia < 0.1

En este caso tenemos dos clases, la primera de las cuales representa una circunferencia de radio 1 centrada en (0,0) y la segunda representa una circunferencia de radio 1 centrada en (1,1). Para cada una de estas clases se tiene una muestra de aprendizaje formada por 30 puntos sobre la circunferencia correspondiente, para ver los objetos de aprendizaje consulte el apéndice A.

En este caso, la función de semejanza que se utilizó fue la coincidencia rasgo a rasgo, con la cual se obtuvieron los siguientes representantes para cada clase:

Clase	# de R. Positivos	# de R. Negativos
K_1	30	30
K_2	30	30

En este caso la eficiencia fue medida utilizando una matriz de control con 10 objetos en cada clase, obteniéndose una eficiencia del 100%. Posteriormente se le proporcionaron 400 objetos que no se encontraban exactamente sobre las circunferencias, de estos objetos se incluyeron 16 en K_1 y 16 en K_2, obteniendo un promedio de distancias entre los objetos clasificados y la circunferencia correspondiente, de 0.028. Las figuras C.1., C.2. y C.3. muestran las gráficas de clasificación de la matriz de control y de los 400 objetos adicionales, para los sistemas C-REP, KORA-{Ω}[7] y PROGNOSIS[3]. En estas gráficas, los objetos incluidos en K_1 se muestran con un +, los objetos incluidos en K_2 con un □ y los objetos que no se incluyeron en ninguna clase se muestran con un ▪.

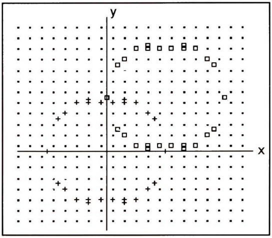


Figura C.1. Resultados de C-REP

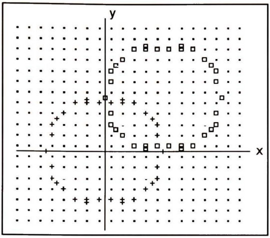


Figura C.2. Resultados de KORA-{Ω}

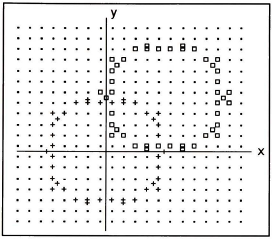


Figura C.3. Resultados de PROGNOSIS

A continuación mostraremos una tabla comparativa entre los sitemas C-REP, KORA-{Ω}[7] y PROGNOSIS[3].

Características	C-REP	KORA-{Ω}	PROGNOSIS
Eficiencia en búsqueda de hidrocarburos	97.64%	96.47%	81.32%
Eficiencia en las circunferencias	100%	100%	100%
Promedio de error de los puntos incluidos	0.028	0.031	0.048
Complejidad en tiempo de aprendizaje	$m^2 \sum_{i=1}^{l} k_i$ (1)	$km \sum_{i=1}^{l} m_i^2$ (2)	No Disponible
Complejidad en el espacio	$m \sum_{i=1}^{l} k_i \, T(r)$ (1)	$kmT(rc)$ (2)	No Disponible

(1) Para más detalles consulte la sección 4.4.

(2) Para más detalles consulte la sección 3.7. de [7].

Por otro lado, vale la pena comentar que el método de clasificación usando conjuntos de representantes no ha sido desarrollado en todas las direcciones posibles, es decir, todavía quedan algunos aspectos que podrían analizarse para extender aún más el algoritmo, por ejemplo:

- Permitir trabajar con rasgos difusos y lingüísticos, es decir, rasgos para los cuales la variable que lo representa es difusa o lingüística.
- Analizar el concepto de representante típico[16][17], así como buscar la manera de calcular todos los representantes típicos de una matriz de aprendizaje. Es decir, tratar de reducir los conjuntos de representantes que se obtienen, sin alterar la calidad de la clasificación.
- Considerar la posibilidad de trabajar con un sistema de conjuntos de apoyo formado por subconjuntos difusos de rasgos.

Finalmente podemos decir que se cumplieron los objetivos fijados al inicio de este trabajo de tesis, puesto que se logró obtener un método extendido de clasificación usando conjuntos de representantes, el cual es capaz de resolver una gama más amplia de problemas.

Apéndice A

Datos de los Ejemplos

En este apéndice se muestran los datos de los ejemplos de prueba mencionados en las conclusiones. En la sección A.1. se muestran los datos del problema de búsqueda de zonas perspectivas de acumulación de hidrocarburos y en la sección A.2. se muestran los datos del ejemplo de las circunferencias.

A.1. Datos para Búsqueda de Hidrocarburos

En este ejemplo, los objetos están descritos en términos de 10 rasgos, con las siguientes características:

Rasgo	Tipo	Dominio		Comparación
		Mínimo	Máximo	
x_1	k-valente	0	462	Igualdad
x_2	k-valente	0	7	Igualdad
x_3	k-valente	0	10	Igualdad
x_4	k-valente	0	9	Igualdad
x_5	k-valente	0	10	Igualdad
x_6	k-valente	0	5	Igualdad
x_7	k-valente	0	19	Igualdad
x_8	k-valente	0	19	Igualdad
x_9	k-valente	0	17	Igualdad
x_{10}	k-valente	0	3	Igualdad

y están agrupados en 13 clases, cada una con las siguientes cantidades de objetos:

Clase	# de Objetos
K_1	16
K_2	26
K_3	9
K_4	5
K_5	9
K_6	6
K_7	13
K_8	4
K_9	7
K_{10}	9
K_{11}	6
K_{12}	6
K_{13}	30
Total	146

A continuación se muestran los datos de los objetos de la matriz de aprendizaje:

Clase	Nombre del Objeto	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
	10TAR28BAN:	271	4	7	1	*	1	10	10	*	*
	10TAR29BAN:	406	4	7	3	*	4	9	10	*	*
	12TAR28BAN:	297	4	7	1	*	4	10	10	*	*
	4TAR28BAN:	404	4	7	8	7	4	9	9	*	*
	4TAR29BAN:	404	4	7	8	7	2	10	10	*	*
	5TAR28BAN:	404	4	7	1	6	5	9	10	*	*
	5TAR30BAN:	301	4	7	6	7	3	10	10	*	*
K_1	6TAR27BAN:	404	2	7	5	6	4	10	10	*	*
	6TAR28BAN:	404	3	7	8	6	1	10	10	*	*
	6TAR29BAN:	271	4	7	3	7	5	9	10	*	*
	7TAR28BAN:	404	2	7	1	6	5	9	10	*	*
	8TAR27BAN:	406	2	7	5	*	2	10	10	*	*
	8TAR28BAN:	404	2	7	4	6	2	9	10	*	*
	8TAR29BAN:	404	3	7	1	6	5	9	10	*	*
	9TAR28BAN:	404	3	7	7	*	4	9	10	*	*
	9TAR29BAN:	409	4	7	8	5	1	9	10	*	*
	2STC25BSN:	404	4	7	3	8	4	10	10	10	3
	2STC29BSN:	404	5	6	1	5	4	10	10	11	3
	3STC25BSN:	119	5	7	7	7	2	10	10	10	3
	3STC26BSN:	404	5	7	8	8	3	10	10	11	3
	3STC28BSN:	404	5	7	8	6	1	10	10	11	3
	3STC29BSN:	404	5	6	7	6	5	10	10	11	3
	2STC26BSN:	404	5	7	5	8	4	10	10	10	3
	1STC27BSN:	404	5	7	5	7	4	10	11	11	3
	1STC28BSN:	404	5	7	1	5	2	10	11	13	3
	4STC25BSN:	30	5	7	4	7	3	10	10	10	3
	4STC26BSN:	404	5	7	7	7	3	10	10	11	3

K_2	4STC27BSN:	404	5	7	1	8	2	10	10	11	3
	4STC28BSN:	404	5	6	1	8	3	10	10	11	3
	4STC29BSN:	404	5	6	1	8	2	10	10	12	3
	5STC26BSN:	119	5	7	1	7	3	10	10	10	3
	5STC27BSN:	404	5	7	1	8	3	10	10	11	3
	5STC28BSN:	404	5	6	1	7	3	10	10	11	3
	1STC29BSN:	404	5	7	1	6	2	10	11	10	3
	2STC28BSN:	408	5	7	1	5	2	10	11	10	3
	3STC27BSN:	404	5	7	7	7	4	10	11	12	3
	45TAR23BSN:	404	4	8	4	6	3	10	11	10	1
	45TAR25BSN:	404	4	7	1	5	2	10	11	10	1
	46TAR24BSN:	404	4	7	1	6	2	10	11	10	1
	43TAR23BSN:	404	4	8	1	5	3	10	11	11	1
	2STC27BSP:	404	5	7	6	5	3	10	11	10	3
	46TAR26BSP:	404	5	7	1	5	2	10	11	14	1
K_3	33VAR22CAN:	108	5	6	8	2	5	10	10	7	3
	34VAR22CAN:	108	5	6	5	1	5	10	10	7	3
	35VAR23CAN:	30	5	6	8	2	1	10	10	7	3
	37VAR25CAN:	30	6	4	8	1	1	10	10	7	3
	39VAR23CAN:	108	6	5	1	1	1	10	10	7	2
	40VAR22CAN:	188	6	5	8	1	1	10	10	7	2
	40VAR23CAN:	30	5	4	5	1	1	10	10	7	3
	40VAR24CAN:	30	5	4	8	1	5	10	10	7	2
	41VAR22CAN:	188	5	4	1	1	5	10	10	7	3
K_4	10TAR30CVN:	262	4	6	9	5	3	11	12	*	*
	10TAR31CVN:	252	4	6	5	6	3	11	12	*	*
	10TAR32CVN:	262	4	6	8	6	5	11	11	*	*
	11TAR30CVN:	409	4	6	4	6	2	11	11	*	*
	6TAR31CVN:	271	4	6	8	6	4	11	12	*	*
K_5	25VAR23KAN:	19	6	7	1	2	5	10	10	8	3
	26VAR21KAN:	188	6	7	1	2	4	10	9	8	2
	28VAR23KAN:	22	6	6	1	4	2	10	10	8	3
	29VAR24KAN:	30	6	6	1	2	2	10	10	7	3
	29VAR25KAN:	30	6	6	5	2	5	10	10	8	3
	30VAR23KAN:	30	6	6	4	3	3	10	9	8	3
	34VAR26KAN:	30	6	5	9	1	5	10	10	8	3
	35VAR27KAN:	30	6	4	4	1	5	10	10	8	3
	41VAR27KAN:	30	5	1	1	1	5	10	10	8	3
K_6	28MAD15MDN:	339	7	10	1	4	5	10	10	14	3
	29MAD12MDN:	462	7	10	7	3	1	10	10	12	2
	29MAD14MDN:	462	7	10	1	3	4	10	10	15	3
	34MAD16MDN:	450	7	10	1	2	3	10	10	15	3
	29MAD13MDN:	462	7	10	1	3	3	10	9	13	1
	34MAD15MDN:	409	7	10	7	2	3	10	9	16	3
K_7	17COR18MTN:	108	4	9	8	1	2	10	10	14	3
	17COR19MTN:	188	4	8	4	1	4	10	10	15	3
	18COR15MTN:	30	4	9	6	1	2	10	10	14	3
	18COR17MTN:	61	4	9	4	1	4	10	10	14	3
	18COR18MTN:	108	4	8	8	1	4	10	10	15	3
	18COR20MTN:	188	4	8	1	1	3	10	10	12	3
	6COR16MTN:	252	5	9	5	1	3	10	10	9	3
	6COR17MTN:	252	5	9	8	1	2	10	10	9	3
	6COR18MTN:	162	5	9	1	1	4	10	10	9	3

	6COR19MTN:	188	5	9	1	1	1	10	10	9	3
	6COR21MTN:	188	5	9	2	1	4	10	9	9	2
	7COR20MTN:	188	5	9	4	1	2	10	10	9	2
	8COR20MTN:	188	4	9	8	1	3	10	11	9	2
K_8	12TAR33BAP:	119	5	6	7	7	4	10	10	*	*
	3TAR32BAP:	404	3	6	1	7	4	10	11	*	*
	6TAR33BAP:	65	4	6	1	7	2	11	11	*	*
	7TAR33BAP:	271	4	6	3	7	2	10	11	*	*
K_9	45TAR26BSP:	408	5	7	6	5	3	10	12	13	1
	43TAR25BSP:	404	4	7	6	6	1	11	12	10	1
	43TAR26BSP:	441	5	7	6	*	3	11	12	12	1
	44TAR25BSP:	404	4	7	9	5	4	11	12	10	1
	44TAR26BSP:	408	5	7	9	6	3	11	12	12	1
	44TAR27BSP:	277	5	7	9	*	2	11	12	13	1
	44TAR23BSP:	387	4	8	8	5	2	10	11	10	1
K_{10}	37VAR19CAP:	188	5	6	2	1	2	10	10	7	3
	37VAR21CAP:	188	5	6	5	1	2	11	11	7	3
	37VAR22CAP:	188	5	6	5	2	1	10	11	7	2
	39VAR20CAP:	188	5	6	6	1	2	11	12	7	3
	39VAR21CAP:	188	5	5	8	2	1	11	11	7	3
	40VAR19CAP:	188	5	6	1	2	2	11	11	7	2
	41VAR19CAP:	188	5	5	1	1	2	10	11	7	3
	41VAR20CAP:	188	6	5	1	2	1	10	11	7	3
	41VAR21CAP:	188	6	5	7	1	1	10	11	7	2
K_{11}	5TAR32CVP:	252	4	6	8	7	2	12	12	*	*
	6TAR32CVP:	297	4	6	1	7	1	12	12	*	*
	7TAR30CVP:	252	4	7	1	6	3	10	11	*	*
	7TAR32CVP:	252	4	6	1	6	4	12	12	*	*
	8TAR31CVP:	252	4	6	5	6	1	12	12	*	*
	9TAR30CVP:	252	4	6	7	6	4	10	11	*	*
K_{12}	30MAD15MDP:	426	7	10	5	3	1	10	10	16	3
	31MAD15MDP:	404	7	10	1	2	3	10	10	17	3
	28MAD14MDN:	334	7	10	1	2	4	10	10	13	3
	9MAD16MDN:	404	7	10	6	4	2	10	11	16	3
	23MAD26MDN:	235	7	10	1	2	4	10	10	9	3
	33MAD16MDN:	409	7	10	5	2	3	10	10	17	3
K_{13}	10COR15MTP:	252	4	9	1	5	3	10	12	9	3
	10COR17MTP:	252	4	9	4	1	4	11	13	9	3
	11COR13MTP:	252	4	9	5	4	4	10	12	9	3
	11COR14MTP:	252	4	9	7	3	5	10	12	9	3
	11COR15MTP:	252	4	9	3	3	1	10	12	9	3
	11COR18MTP:	252	4	9	9	3	5	11	13	9	2
	11COR19MTP:	188	4	9	1	3	4	11	13	10	3
	12COR13MTP:	252	4	9	6	3	1	10	11	9	3
	12COR14MTP:	252	3	9	5	3	1	10	12	9	2
	12COR17MTP:	252	3	9	1	2	2	11	13	9	3
	12COR18MTP:	252	3	9	9	2	1	11	13	10	3
	12COR20MTP:	188	4	9	8	3	1	11	13	11	3
	13COR13MTP:	252	3	9	4	5	2	11	12	9	3
	13COR14MTP:	252	2	9	8	5	3	11	13	9	3
	13COR15MTP:	252	3	9	5	3	2	11	13	9	3
	13COR17MTP:	252	2	9	4	3	2	10	12	10	3
	14COR14MTP:	252	1	9	1	4	4	11	13	10	3

	14COR15MTP:	252	2	9	1	3	5	11	12	10	3
	14COR16MTP:	252	3	9	3	4	1	10	12	10	3
	14COR17MTP:	252	3	9	1	2	2	10	12	10	3
	15COR13MTP:	252	3	9	6	5	3	10	12	10	3
	15COR14MTP:	252	2	9	6	4	2	10	12	10	3
	15COR15MTP:	252	3	9	8	4	2	10	12	10	3
	16COR13MTP:	252	4	9	6	4	5	11	12	10	3
	7COR15MTP:	252	5	9	1	3	4	11	11	9	3
	7COR17MTP:	252	5	9	4	3	2	11	12	9	3
	8COR17MTP:	252	4	9	8	2	1	11	12	9	3
	8COR18MTP:	252	4	9	8	4	5	10	12	9	2
	9COR14MTP:	252	4	9	1	2	4	11	12	9	3
	7COR13MTN:	252	5	9	1	4	2	10	10	9	3

A.2. Datos del Ejemplo de Circunferencias

En este ejemplo, los objetos están descritos en términos de 4 rasgos, con las siguientes características:

Rasgo	Tipo	Dominio		Comparación
		Mínimo	Máximo	
x	real	-2.0	2.0	diferencia < 0.1
y	real	-2.0	2.0	diferencia < 0.1
x+y	real	-4.0	4.0	diferencia < 0.1
x-y	real	-4.0	4.0	diferencia < 0.1

y están agrupados en 2 clases, cada una con las siguientes cantidades de objetos:

Clase	# de Objetos
K_1	30
K_2	30
Total	60

La clase K_1 representa una circunferencia de radio 1 centrada en el origen, y la clase K_2 representa una circunferencia de radio 1 centrada en (1,1). A continuación se muestran los datos de los objetos de la matriz de aprendizaje:

Clase	Nombre del Objeto	x_1	x_2	x_3	x_4
	O1	1.000	0.000	1.000	1.000
	O2	0.978	0.208	1.186	0.770
	O3	0.914	0.407	1.320	0.507
	O4	0.809	0.588	1.397	0.221
	O5	0.669	0.743	1.412	-0.074
	O6	0.500	0.866	1.366	-0.366
	O7	0.309	0.951	1.260	-0.642
	O8	0.105	0.995	1.099	-0.890
	O9	-0.105	0.995	0.890	-1.099

K_1	O10	-0.309	0.951	0.642	-1.260
	O11	-0.500	0.866	0.366	-1.366
	O12	-0.669	0.743	0.074	-1.412
	O13	-0.809	0.588	-0.221	-1.397
	O14	-0.914	0.407	-0.507	-1.320
	O15	-0.978	0.208	-0.770	-1.186
	O16	-1.000	0.000	-1.000	-1.000
	O17	-0.978	-0.208	-1.186	-0.770
	O18	-0.914	-0.407	-1.320	-0.507
	O19	-0.809	-0.588	-1.397	-0.221
	O20	-0.669	-0.743	-1.412	0.074
	O21	-0.500	-0.866	-1.366	0.366
	O22	-0.309	-0.951	-1.260	0.642
	O23	-0.105	-0.995	-1.099	0.890
	O24	0.105	-0.995	-0.890	1.099
	O25	0.309	-0.951	-0.642	1.260
	O26	0.500	-0.866	-0.366	1.366
	O27	0.669	-0.743	-0.074	1.412
	O28	0.809	-0.588	0.221	1.397
	O29	0.914	-0.407	0.507	1.320
	O30	0.978	-0.208	0.770	1.186
K_2	O31	2.000	1.000	3.000	1.000
	O32	1.978	1.208	3.186	0.770
	O33	1.914	1.407	3.320	0.507
	O34	1.809	1.588	3.397	0.221
	O35	1.669	1.743	3.412	-0.074
	O36	1.500	1.866	3.366	-0.366
	O37	1.309	1.951	3.260	-0.642
	O38	1.105	1.995	3.099	-0.890
	O39	0.895	1.995	2.890	-1.099
	O40	0.691	1.951	2.642	-1.260
	O41	0.500	1.866	2.366	-1.366
	O42	0.331	1.743	2.074	-1.412
	O43	0.191	1.588	1.779	-1.397
	O44	0.086	1.407	1.493	-1.320
	O45	0.022	1.208	1.230	-1.186
	O46	0.000	1.000	1.000	-1.000
	O47	0.022	0.792	0.814	-0.770
	O48	0.086	0.593	0.680	-0.507
	O49	0.191	0.412	0.603	-0.221
	O50	0.331	0.257	0.588	0.074
	O51	0.500	0.134	0.634	0.366
	O52	0.691	0.049	0.740	0.642
	O53	0.895	0.005	0.901	0.890
	O54	1.105	0.005	1.110	1.099
	O55	1.309	0.049	1.358	1.260
	O56	1.500	0.134	1.634	1.366
	O57	1.669	0.257	1.926	1.412
	O58	1.809	0.412	2.221	1.397
	O59	1.914	0.593	2.507	1.320
	O60	1.978	0.792	2.770	1.186

Referencias

[1] Modelo de Algoritmos de Reconocimiento con Conjuntos de Representantes y Sistemas de Conjuntos de Apoyo
Baskakova, L. V. y Zhuravliov, Yu. I.
Zhurnal Vichislitielnoi Matemati y Matematicheskoi Fisiki
Tomo 21; No. 5, pp. 1264-75. (1981)

[2] Modelos Matemáticos para el Reconocimiento de Patrones
José Ruiz Shulcloper
Instituto de Cibernética, Matemática y Física
Academia de Ciencias de Cuba (1989)

[3] PROGNOSIS y sus Aplicaciones a las Geociencias
José Ruiz Shulcloper
Memorias del 3er Congreso Iberoamericano de Inteligencia Artificial
IBERAMIA'92. Ed. Limusa, México, pp. 561-586 (1992)

[4] Modelación Matemática de Fenómenos Geocientíficos
José Ruiz Shulcloper
Primer Simposio acerca del Desarrollo de la Matemática
La Habana, Cuba (1990)

[5] Algoritmos de Votación en Medios Difusos
Manuel Lazo Cortez
Universidad Central de las Villas, Cuba (1993)

[6] Reconocimiento de Elementos y Estructuras Espaciales
Instituto de Cibernética, Matemática y Física
Instituto de Geofísica y Astronomía
Editorial Academia (1992)

[7] Extensión al caso Difuso del Algoritmo de Clasificación KORA-3
Lucía Angélica De la Vega Doria
Proyecto de tesis de maestría en el CINVESTAV
Departamento de Ingeniería Eléctrica, Sección de Computación (1994)

[8] Tópicos Acerca de la Teoría de Testores
José Ruiz Shulcloper, Eduardo Alba Cabrera, Nancy López Reyes,
Manuel Lazo Cortés & Eligio E. Barreto Fiu
Serie Amarilla, # 134
CINVESTAV-IPN (1994)

[9] Fuzzy Sets and Systems: Theory and Applications
Didier Dubois & Henri Prade
Academic Press (1980)

[10] Constructing Membership Functions Using Statistical Data
M. Reha Civanlar & H. Joel Trussell
Elsevier Science Publishers B.V. (1986)

[11] On the Derivation of Memberships for Fuzzy Sets in Expert Systems
Lawrence O. Hall, Sue Szabo & Abraham Kandel
Information Sciences (1986)

[12] Fuzzy Sets, Fuzzy Algebra and Fuzzy Statistics
Abraham Kandel & William J. Byatt
Proceedings of the IEEE (1978)

[13] Fuzzy Logic
Lofti A. Zadeh
IEEE Computer, vol 4, pp 83-93 (1988)

[14] Developing Windows™ Applications with Borland® C++ 3
James McCord
SAMS Programming Series (1992)

[15] Fuzzy Sets
Lofti A. Zadeh
Inform. Control, vol. 8, pp. 338-353 (1965)

[16] Acerca de un Modelo Paramétrico de Algoritmos de Reconocimiento tipo KORA.
Elena V. Diukova.
Comunicaciones sobre Matemática Aplicada
Centro de Cálculo, Academia de Ciencias de la URSS (1988)

[17] Método de Síntesis de los Conjuntos Representantes Típicos para tablas K-valentes
R. A. Denisova.
Comunicaciones sobre Matemática Aplicada
Centro de Cálculo, Academia de Ciencias de la URSS (1984)

Los abajo firmantes, integrantes de jurado para el examen de grado que sustentará el **Lic. Jesús Ariel Carrasco Ochoa**, declaramos que hemos revisado la tesis titulada:

Clasificadores Basados en Conjuntos de Representantes

y consideramos que cumple con los requisitos para obtener el grado de Maestro en Ciencias, con especialidad en Ingeniería Eléctrica.

Atentamente

Dr. José Ruíz Shulcloper ____________________

Dr. Sergio Víctor Chapa Vergara ____________________

M. en C. José Oscar Olmedo Aguirre ____________________

AUTOR CARRASCO OCHOA, J.A.		
TITULO CLASIFICADORES BASADOS EN CONJUNTOS DE...		
CLASIF. XM 94.16 **RGTRO** BI 14416		
NOMBRE DEL LECTOR	FECHA PREST.	FECHA DEVOL.
Laura Mendez Segundo	12/07/95	5-VIII-95
Dr. Guillermo Morales	14-12-95	14/XII/95
Manuel Lazo Cortes	14-2-97	10-III-97