





CINVESTAV-IPN  
Biblioteca de Ingeniería Eléctrica



FB000009824

**CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELECTRICA**



CENTRO DE INVESTIGACION Y ESTUDIOS AVANZADOS  
DEL  
INSTITUTO POLITECNICO NACIONAL

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA  
SECCION COMPUTACION



“EXTENSIÓN DEL ALGORITMO REC  
PARA EL CÁLCULO DE TESTORES TÍPICOS”

tesis que para obtener el grado de  
Maestra en Ingeniería Eléctrica  
presenta

Verónica Jiménez Jacinto

Director: Dr. José Ruiz Shulcloper  
Coodirector: Dr. Manuel S. Lazo Cortés

Noviembre, 1995

CENTRO DE INVESTIGACION Y DE  
ESTUDIOS AVANZADOS DEL  
I. P. N.  
BIBLIOTECA  
INGENIERIA ELÉCTRICA



# ÍNDICE

	Pag.
Introducción . . . . .	. 1
Capítulo 1. Selección de Variables . . . . .	. 3
1. Métodos de selección de variables . . . . .	. 4
1.1 Transformaciones de Agrupamientos y ordenamiento de características . . . . .	. 5
1.2 Agrupamiento en la selección de Variables . . . . .	. 8
1.3 Selección de Características a través de la <b>minimización</b> de la entropía . . . . .	.11
1.4 Selección de características a través de expansiones ortogonales . . . . .	.12
1.5 Selección de variables a través de la aproximación funcional . . . . .	.18
1.6. Selección de características a través de la <b>maximización</b> de la divergencia . . . . .	.21
1.7 Enfoque básico de la separabilidad de las clases . . . . .	.25
1.8 Selección de Características binarias . . . . .	.27
1.9 Selección gráfica de variables . . . . .	.28
2. Resumen de requerimientos . . . . .	.28
Capítulo 2. El algoritmo RECPLUS . . . . .	.30
Capítulo 3. Subconjuntos y Testores Difusos . . . . .	.37
3.1 Conjuntos Difusos . . . . .	. 37
3.2 Operaciones y Relaciones con Conjuntos Difusos . . . . .	. 38
3.3 Testores Difusos . . . . .	. 39
Capítulo 4. El algoritmo RECPLUS para el cálculo de testores típicos . . . . .	.42
4.1 Formalización del Orden . . . . .	. 43
4.2 La función de error . . . . .	. 46
4.3 La función de recorrido . . . . .	. 47
4.5 Algoritmo RECDIF para el cálculo de los testores difusos Típicos . . . . .	. 53
Capítulo 5 Implantación computacional . . . . .	.61
REC_DIFL.CPP . . . . .	.61
REC_IMPR.CPP . . . . .	.62

RECLEE.CPP	.63
REC_ALGO.CPP	.66
RECDEF2.CPP	.72
Conclusiones .	.73
Anexo I. .	.74
Bibliografía. .	.75

CENTRO DE INVESTIGACION Y DE  
 ESTUDIOS AVANZADOS DEL  
 I. P. N.  
 BIBLIOTECA  
 INGENIERIA ELECTRICA

## INTRODUCCIÓN

La selección de variables es una parte importante dentro del proceso de Reconocimiento de Patrones. Una buena selección de variables es un factor determinante en la calidad del reconocedor.

Existen muchas técnicas para llevar a cabo esta selección. Algunas de ellas son técnicas matemáticas que se utilizan en el Reconocimiento de Patrones dentro del enfoque estadístico-probabilístico, y que exigen que el modelo al que se aplica, tenga tal o cual distribución de probabilidades dentro de su espacio de variables o que sus datos no estén relacionados o que se cumplan con las propiedades de un espacio vectorial. En el enfoque sintáctico estructural, se eligen los rasgos o “primitivas” a partir de una serie de criterios que jerarquizan el patrón a reconocer e identifican cuál es su estructura más simple para conformar el alfabeto.

Estas técnicas permiten resolver un sin número de problemas de selección de variables de manera bastante eficaz. Desgraciadamente existen problemas donde el usuario no puede asegurar que las variables siguen una determinada distribución de probabilidad o su no correlación, o es un espacio con distintos tipos de variables, entre otras cosas que no permite aplicar las técnicas anteriores. En estos casos se puede recurrir al enfoque lógico combinatorio, donde se puede llevar a cabo la selección de variables utilizando la teoría de Testores, y aún más, la teoría de Testores Difusos.

En el presente trabajo se toma como base un algoritmo llamado REC para el cálculo de testores típicos que fue propuesto por Rafael Morales Gamboa [12], se modifica y extiende de manera que su desempeño sea óptimo, que trabaje con clases no disjuntas, con el concepto de  $\epsilon$ -testores y con el concepto de comparación difusa ( semejanza ó diferencia difusa) [18] [10] [15].

En el capítulo 1, se hace una breve descripción de los métodos más mencionados en la literatura para llevar a cabo un proceso de selección de variables. Se hace énfasis en sus requerimientos. Esta revisión no pretende nunca ser exhaustiva, pero sí poner de relieve los principales limitaciones de cada uno de los métodos sin menospreciarlos. No se intenta subvalorar su utilidad, sino recalcar su campo de aplicación. No existe un método que solucione “todo”, cada método fue pensado para solucionar problemas de selección de variables bajo ciertas condiciones que no deben pasarse por alto y que imposibilitan la aplicación del método a ciertos problemas específicos de la realidad.

En el capítulo 2 se lleva a cabo una breve descripción de las capacidades del algoritmo RECPLUS, antecesor directo del algoritmo que se propone en este trabajo, el algoritmo RECDIF. RECPLUS es un algoritmo cuyo objetivo es encontrar todos los testores típicos de una matriz de aprendizaje<sup>1</sup> para aquellos problemas “supervisados”<sup>2</sup> en los que las clases en que se dividen los objetos son no disjuntas, donde además se mezclan rasgos de distintos tipos, digase booleanos, k-va- lentes, reales, continuos, discretos, lingüísticos, etc. Problemas donde se requiere manejar más de

<sup>1</sup> La definición de estos conceptos se da en el capítulo 2.

<sup>2</sup> Llamaremos problemas supervisados a aquellos donde se cuenta con información acerca de una clasificación inicial de objetos, es decir, se cuenta con una muestra de objetos de cada clase, de inicio.



un concepto de “ semejanza ” entre los rasgos. RECPLUS, además se formuló como un algoritmo que llevará a cabo la búsqueda de los testores típicos con un grado aceptable de cálculos y operaciones computacionales.

El capítulo 3, es una breve introducción a los subconjuntos difusos y testores difusos. Establecen las operaciones y relaciones entre los conjuntos difusos, establecidos por Zadeh. Se describen también los conceptos de testor difuso y testor difuso típico, trabajados por Goldman y Lazo [17]. Todo esto se establece como una base conceptual del trabajo realizado.

Finalmente el capítulo 4, presenta la propuesta de desarrollo del algoritmo RECDIF, que localiza los testores típicos a partir de una matriz de diferencias difusas<sup>3</sup> En el Reconocimiento de Patrones, desde el enfoque lógico combinatorio, un proceso de clasificación con aprendizaje se inicia representando la información de los objetos en n-uplos que contienen la información de los valores que toma el objeto para cada uno de sus rasgos. Estos n-uplos se agrupan en un determinado número de clases a partir de la información que el especialista nos proporciona. Tomando como información inicial lo anterior, el algoritmo presentado pretende ser una herramienta flexible para el cálculo de testores difusos típicos como elementos previos para la selección de variables en condiciones donde las descripciones de los objetos con los que se trabaja, están contemplados en clases no disjuntas, es decir, un objeto puede pertenecer a más de una clase al mismo tiempo, cosa que ocurre de manera frecuente en los problemas de clasificación o pronóstico en la medicina, la geología, y a el reconocimiento de imágenes. El algoritmo considera también los problemas en los cuales la comparación entre objetos no compromete igualdad absoluta, por ejemplo, en la medicina, la descripción de un paciente y sus síntomas son en la mayoría de los casos n-úplos que no son exactamente iguales y que sin embargo ambos pacientes pertenecen a la misma clase de enfermedad o al mismo tipo de paciente. Además RECDIF parte de que los rasgos con que se describen dichos objetos tienen distintos espacios de representación, y por tanto distintos criterios de comparación, para muestra, consideremos nuevamente el ejemplo anterior, y fijemos nuestra atención en el hecho de que la descripción de los objetos puede tener rasgos como “Edad”, “Sexo”, entre otras y en este caso, la variable edad, es una variable con un espacio de representación discreto si se contabiliza en días por ejemplo, y su criterio de comparación es por intervalos, es decir, si dos pacientes tienen una edad en el intervalo de 1 a 365 días se consideran “semejantes” para ese rasgo, por otra parte el “Sexo” es una etiqueta lingüística, a saber “Femenino” y “Masculino” cuyo criterio de comparación es de igualdad, es decir, Femenino sólo es igual a Femenino. Por último el algoritmo toma en cuenta que, como se vio en el ejemplo, los valores de dichos rasgos pueden no pertenecer a un espacio continuo, no se requiere asegurar su no correlación, ni requieren tener una distribución probabilística predeterminada entre los elementos. A RECDIF no le preocupa que el número inicial de rasgos sea demasiado grande, es precisamente su trabajo reducirlo, sin hacer una búsqueda exhaustiva para encontrar las combinaciones de rasgos que permitan trabajar de manera más óptima los datos, sin perder información, y preparando el terreno para reducir posibles errores en el proceso posterior de clasificación (si se utiliza como proceso intermedio).

<sup>3</sup> La definición de este concepto se da en el capítulo 4.

## Capítulo 1. SELECCIÓN DE VARIABLES

El Reconocimiento de Patrones es una importante herramienta para el procesamiento de datos, la toma de decisiones, la clasificación y el pronóstico.

El Reconocimiento de Patrones (R.P.) puede ser caracterizado como una reducción, mapeo o etiquetación de la información. Un patrón puede ser tan básico como un conjunto de mediciones u observaciones representadas en una  $n$ -ada. El conjunto de patrones puede a su vez ser representado por un conjunto de  $n$ -adas como renglones que conformen una matriz, que llamaremos Matriz de Aprendizaje. Estas mediciones u observaciones pueden ser discretas, continuas, booleanas,  $k$ -valentes o etiquetas lingüísticas. Debe considerarse incluso la ausencia de información, es decir, cuando se desconoce el valor de alguno (s) de los rasgos que describen al objeto.

Existen dos problemas básicos relacionados con la conformación de patrones: la extracción y la selección de los rasgos<sup>1</sup>. Se debe considerar que 1) los algoritmos y la extracción de la información sean computacionalmente realizables. 2) Conduzcan a un buen sistema de R.P.<sup>2</sup> 3) Reduzcan el problema de los datos “estériles”<sup>3</sup> dentro del cúmulo de información sin descartar la información vital [1],[2],[3].

Para Harry Andrew [4], conceptualmente el problema de Reconocimiento de Patrones puede ser descrito como una transformación del espacio de patrones  $P$  al espacio de características o rasgos  $F$  y finalmente al espacio de clasificación  $C$  ( $P \rightarrow F \rightarrow C$ ).

La selección de características es entonces, un proceso intermedio entre la formulación del espacio de patrones y el espacio de la clasificación, pero puede ser también un objetivo *per se*,  $P \rightarrow F$ . El trabajar con una buena selección de variables nos permitirá hacer una discriminación efectiva entre los distintos objetos, permitiendo además que los algoritmos de clasificación puedan ser efectivamente implementados al reducir o modificar con esta selección el espacio de representación.

Los objetivos de elegir el espacio de características (ó selección de variables) es reducir la dimensionalidad del espacio de patrones para simplificar, mantener el poder de discriminación con menos variables para los propósitos de una mejor clasificación, y/ó la determinación de un espacio de representación.

No existe hasta este momento un modelo matemático que iguale la efectividad de un humano para extraer las características necesarias para poder llevar a cabo un proceso de distinción entre patrones. Aunque hay que acotar también que mientras más grande es el espacio de representación inicial de los patrones, más difícil se augura el trabajo de selección de los rasgos y luego el de la reducción de ese espacio hecho por el hombre, de manera directa.

<sup>1</sup> Rasgo, característica y variable son sinónimos para efectos de este trabajo.

<sup>2</sup> La selección de variables afecta directamente el grado de error que se cometerá en el proceso de R.P.

<sup>3</sup> Se entenderá por datos estériles aquellos que no aportan verdadera información, o que aportan información redundante o innecesaria.

La filosofía subyacente de cualquier mecanismo de selección de variables podría ser la retención de la información discriminatoria entre las clases y la reducción de la información común entre las mismas. Esto debe ayudar a evitar posibles errores al discriminar y por lo tanto una mejor clasificación.

Ruiz Shulcloper [5] considera que uno de los problemas más frecuentes en algunas ciencias es la determinación del espacio de representación de los objetos o fenómenos sujetos a estudio es decir, del conjunto de características que los describen; se desea conocer cómo se vinculan entre sí algunas de dichas características; valorar ciertos conjuntos de rasgos respecto a otros; cómo se desea conocer la representatividad de ciertos objetos en el contexto de sus clases y muchas otras interrogantes que conducen directamente al mismo problema: la selección de rasgos.

Para Fu [6] la información de la divergencia y el promedio de las características es un criterio de calidad de las mismas y la selección entonces queda en términos de elegir las características con mayor "calidad". El concepto de Divergencia se relaciona con el poder de discriminación entre dos clases de patrones con valores en sus rasgos con una distribución gaussiana.

B.A. Zyrianov [7], asegura que el principal propósito para reducir la dimensionalidad de un espacio de rasgos es incrementar la calidad de la clasificación y la velocidad de los cálculos. Obtener un buen resultado en una aplicación con algún método, no posibilita recomendarlo para cualquier tarea.

Este último comentario es de suma importancia. Existen muchos métodos para llevar a cabo el proceso de selección de variables, pero es preciso que el método que elijamos para un problema en particular, sea el más adecuado. Para lograr esto, estamos obligados a verificar que las premisas que asume el método elegido, sean cumplidas por nuestro problema particular.

Aplicar un método sin verificar y tener en cuenta sus restricciones y requerimientos es un acto irresponsable que puede conducir a resultados erróneos, falsos y por lo tanto inútiles o peligrosos.

En los siguientes párrafos se pretende hacer una revisión de algunos de los métodos de selección de variables más utilizados, o más mencionados en la literatura, enfatizando los requerimientos y condiciones iniciales para ser empleados.

## **1. MÉTODOS DE SELECCIÓN DE VARIABLES.**

Para llevar a cabo esta descripción de los distintos métodos, se inicia proponiendo una división de dichos métodos, con base en lo dicho por Barandela[9] y Tou[10], y que se define en los siguientes términos:

### **I. Enmarcados dentro de alguna de las 3 líneas de Reconocimiento**

- I.A. Estadístico - Probabilístico: A su vez se pueden subdividir en
  - I.A.1. métodos basados en medidas de distancias probabilísticas
  - I.A.2. métodos basados en el desarrollo de Karhunen-Loève
  - I.A.3. métodos basados en matrices de dispersión

- I.A.4. métodos para el cálculo de la dimensionalidad intrínseca de una muestra de patrones.
- I.A.5. métodos para el agrupamiento de los patrones representados mediante la colección óptima de variables, mediante una búsqueda exhaustiva de las mismas.
- I.B. Lógico -Combinatorio: Se ha basado principalmente en la Teoría de Testores. Existen 3 clases de algoritmos para el cálculo de testores típicos:
  - I.B.1 De escala interior.
  - I.B.2 De escala exterior.
  - I.B.3 con operaciones lógicas.
- I.C. Sintáctico- Estructural: Se basan principalmente en la teoría de Lenguajes Formales y teoría de Autómatas.

## 1.1. TRANSFORMACIONES DE AGRUPAMIENTOS Y ORDENAMIENTO DE CARACTERÍSTICAS. [10]

### 1.1.1 Exposición Técnica

Los rasgos que describen a los objetos pueden ser representados por diferentes coordenadas  $x_k$  que no son igualmente importantes para definir la similaridad entre los patrones de una misma clase. El proceso de selección entonces, se trabaja en base a ponderar las características. El proceso de ponderación de las características puede ser realizado a través de una transformación lineal, la cual agrupe los puntos del espacio de objetos o patrones en un nuevo espacio.

Consideremos los vectores de objetos  $\mathbf{a}$  y  $\mathbf{b}$ , los cuales se transforman en los vectores  $\mathbf{a}^*$  y  $\mathbf{b}^*$  a través de la transformación  $W$ . Entonces:

$$\mathbf{a}^* = W\mathbf{a} \quad \text{y} \quad \mathbf{b}^* = W\mathbf{b}$$

donde

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix}$$

en la cual  $w_{kj}$  son coeficientes de ponderación. De este modo, tenemos

$$a_k^* - b_k^* = \sum_{j=1}^n w_{kj} (a_j - b_j)$$

Cada elemento del vector transformado es una combinación lineal de los elementos del vector del objeto original. La distancia Euclidiana entre  $\mathbf{a}^*$  y  $\mathbf{b}^*$  en el nuevo espacio esta dada nuevamente como:

$$D(\mathbf{a}^*, \mathbf{b}^*) = \sqrt{\sum_{k=1}^n (a_k^* - b_k^*)^2} = \sqrt{\sum_{k=1}^n \left[ \sum_{j=1}^n w_{kj} (a_j - b_j) \right]^2}$$

donde la transformación lineal involucra solo cambios de factores escalares de las coordenadas. Podemos tener una  $W$  que sea una matriz diagonal. Así la distancia Euclidiana se reduce a

$$D(a^*, b^*) = \sqrt{\sum_{k=1}^n w_{kk}^2 (a_k - b_k)^2}$$

donde  $w_{kk}$  representan los coeficientes del peso de las características. El problema de la transformación de un agrupamiento es determinar los coeficientes  $W_{kk}$  de manera que la distancia interconjuntos es minimizada, sujeta a las restricciones especificadas más adelante sobre  $W_{kk}$ . Entonces

$$\overline{D^2} = 2 \sum (w_{kk} \sigma_k)^2$$

El procedimiento de minimización considera 2 casos:

Caso 1: Restricción  $\sum_{k=1}^n w_{kk} = 1$ ,

Minimizando  $\overline{D^2}$  por multiplicadores de Lagrange, los coeficientes de ponderación son

$$w_{kk} = \frac{1}{\sigma_k^2 \sum_{k=1}^n \left( \frac{1}{\sigma_k^2} \right)} \quad \dots(1.1)$$

Esto implica en la medida de las distancias, que una ponderación pequeña es asignada para una característica de gran variación.

caso 2:

$$\prod_{k=1}^n w_{kk} = 1$$

$$w_{kk} = \frac{1}{\sigma_k} \left( \prod_{j=1}^n \sigma_j \right)^{\frac{1}{n}} \quad \dots(1.2)$$

donde  $w_{kk}$  se obtuvo de minimizar de igual manera  $\overline{D^2}$  con respecto a esta restricción, resultando una  $w_{kk}$  que es inversamente proporcional a la desviación estándar de la k-ésima medida. Las ecuaciones (1.1) y (1.2) determinan la matriz de transformación W bajo las restricciones especificadas

Si los vectores de objetos son transformados del espacio X al espacio X\* por la transformación:

$$x^* = Wx$$

la distancia interconjunto en el espacio X\* es minimizada. Ahora se quiere realizar una segunda transformación:

$$x^{**} = Ax^*$$

para hacer evidentes cuáles componentes tienen pequeñas (o grandes) varianzas, y así realizar el ordenamiento y selección de características.

El problema es determinar una expresión para A en términos de los valores propios de la matriz de covarianza la cual es conocida. Sea C la matriz de covarianzas de los objetos en el espacio X, y C\* y C\*\* las correspondientes matrices de covarianzas en los espacios X\* y X\*\* respectivamente. Sea m el vector de valores medios de los puntos del patrón en el espacio X y m\* y

$m^{**}$  los correspondientes vectores de valores medios en los espacios  $X^*$  y  $X^{**}$  respectivamente. Entonces tenemos:

$$m^* = Wm$$

y

$$x^* - m^* = W(x - m) = Wz$$

donde

$$m^* = E\{X^*\} \text{ y } z = X - m.$$

La matriz de covarianza en el espacio  $X^*$  es dada por  $C^* = E\{(x^* - m^*)(x^* - m^*)'\} = E\{Wz z' W'\} = WCW'$  entonces  $C = E\{zz'\}$ . Similarmente puede demostrarse que  $C^{**} = AC^*A' = AWCW'A'$  Entonces  $A'$  es ortogonal

$$AA' = I \text{ y } A' = A^{-1}$$

Entonces

$$C^{**} = AC^*A^{-1}$$

luego  $C^{**}$  y  $C^*$  son semejantes.

Es bien conocido que para transformaciones semejantes la matriz  $C^{**}$  será diagonal si  $A^{-1}$  es elegida como matriz modal de  $C^*$ , esto es, las columnas de  $A^{-1}$  son los vectores propios de  $C^*$ , entonces, las filas de  $A$  son los vectores propios de  $C^*$ . Esta transformación es de congruencia y no permite expresar fácilmente los vectores propios de  $WCW'$  en términos de aquellos de  $C$ .

Sean  $e_k$ ,  $k=1,2,\dots,n$  los vectores propios normalizados de  $C^*$  y  $\lambda_k$ ,  $k=1,2,\dots,n$ , los correspondientes valores propios. Entonces:

$$C^*e_k = \lambda_k e_k$$

y

$$(C^* - \lambda_k I)e_k = 0,$$

Se elige  $A$  de manera que  $A = \begin{pmatrix} e'_1 \\ e'_2 \\ \vdots \\ e'_n \end{pmatrix}$ ,  $A^{-1} = (e_1 \ e_2 \ \dots \ e_n)$ ,

entonces  $C^{**} = AC^*A^{-1} = \begin{pmatrix} e'_1 \\ e'_2 \\ \vdots \\ e'_n \end{pmatrix} C^* (e_1 \ e_2 \ \dots \ e_n)$ ,

$$C^{**} = \begin{pmatrix} \lambda_1 e'_1 e_1 & \lambda_2 e'_1 e_2 & \dots & \lambda_n e'_1 e_n \\ \lambda_1 e'_2 e_1 & \lambda_2 e'_2 e_2 & \dots & \lambda_n e'_2 e_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1 e'_n e_1 & \lambda_2 e'_n e_2 & \dots & \lambda_n e'_n e_n \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_n \end{pmatrix},$$

ya que  $e'_k e_k = 1$  y  $e'_k e_j = 0$  para  $k \neq j$ . Se puede mostrar que  $\lambda_k = \sigma_k^2$ .

La matriz de transformación  $A$  convierte la matriz de covarianza  $C^*$  en una matriz diagonal en términos de las varianzas insesgadas muestrales. Las medidas correspondientes a varianzas pequeñas son más realzadas y pueden ser consideradas como características más importantes.

### 1.1.2 Análisis de la Técnica:

**A) Presupuestos:** ya que los rasgos serán representados como coordenadas en un espacio, deben tener espacios de representación en los reales. Deben cumplir las propiedades de un espacio de representación vectorial. Esto es un presupuesto que limita notablemente el número de aplicaciones, debido a que muchos problemas reales no logran cubrir este requisito. Otro presupuesto importante es que asume que el especialista tiene el dominio suficiente del problema como para ponderar a priori la importancia de cada rasgo.

**B) Significación y Alcance:** Este método, explota el concepto de distancia entre los elementos de una misma clase y por lo tanto trabaja bastante bien cuando los elementos de la misma clase están suficientemente unidos y al mismo tiempo existe una buena separación entre las clases. La selección se basa en elegir aquellos rasgos que bajo una transformación evidencian la cercanía de ciertos cúmulos de objetos. Su área de aplicación es limitada pero eficiente.

## 1.2 AGRUPAMIENTO EN LA SELECCIÓN DE VARIABLES. [10]

La selección de variables puede ser estudiada como un problema de agrupamiento. Se usará una transformación lineal para agrupar puntos que representan objetos que pertenecen a la misma clase y para reducir la dimensionalidad del espacio.

### 1.2.1. Exposición Técnica

Consideremos una clase de objetos caracterizada por una población multivariada. Uno de sus miembros normalizados digamos  $z_1$ , es seleccionado arbitrariamente como una referencia sobre la secuencia de todas las distancias de los vectores de objetos normalizados adyacentes  $z$ . La elección del patrón  $z_1$  es asumido como independiente de la selección de otro patrón vector  $z$ . Así:

$$p(z, z_1) = p(z)p(z_1) = p p_1$$

entonces, la distancia interconjuntos de la población multivariada :

$$\overline{D^2} = E_p \{z'z\} + E_{p_1} \{z_1 z_1'\} = 2E_p \{zz'\} \quad \dots(1.2.1)$$

que en términos de la matriz de covarianza se expresa como:

$$C_z = E_p \{zz'\}$$

la ecuación 1.2.1 puede escribirse entonces como

$$\overline{D^2} = 2E_p \{tr zz'\} = 2 tr C_z$$

Introduciendo una matriz de transformación ortogonal  $A$  y una matriz de ponderación diagonal  $W$ , tenemos la matriz de covarianzas en el espacio transformado dado por :

$$C_z^{**} = WAC_z A'W'$$

entonces la distancia interconjunto en el espacio transformado es:

$$\overline{D^2} = 2tr(WAC_z A'W')$$

Sean  $e_1, e_2, \dots, e_n$  los vectores propios de la matriz de covarianza  $C_z$ , y sean  $\lambda_1, \lambda_2, \dots, \lambda_n$  los correspondientes valores propios. Entonces:

$$C_z e_k = \lambda_k e_k \quad \dots(1.2.2)$$

Los elementos de la transformación ortogonal  $A$  son elegidos de manera tal que la matriz de covarianza en el espacio transformado esté diagonalizada. Esto puede ser logrado al elegir  $m$  de los  $n$  valores propios traspuestos de  $C_z$  como los renglones de la matriz ortogonal  $A$ . Así:

$$A = \begin{pmatrix} e'_1 \\ e'_2 \\ \vdots \\ e'_m \end{pmatrix} \quad \dots(1.2.3)$$

La dimensionalidad del espacio transformado se reduce a  $m$ . Siguiendo de (1.2.2) y (1.2.3) se tiene que

$$C_z A' = (\lambda_1 e_1 \quad \lambda_2 e_2 \quad \dots \quad \lambda_m e_m)$$

En vista de las condiciones de ortonormalidad, la matriz  $AC_z A'$  se reduce a una matriz diagonal:

$$AC_z A = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \lambda_m \end{pmatrix} = \Lambda$$

Entonces la distancia interconjuntos puede ser escrita como

$$\overline{D^2} = 2 \operatorname{tr}(W \Lambda W') = 2 \sum \lambda_k w_{kk}^2 \quad \dots(1.2.4)$$

Ahora, nosotros queremos determinar la matriz de ponderaciones  $W$  así que  $\overline{D^2}$  es un extremo bajo una cierta restricción específica. Dos restricciones serán estudiadas. Primero se consideran las restricciones de manera tal que  $\prod_{k=1}^m w_{kk} = 1$ , esta restricción puede ser escrita como  $|W| - 1 = 0$ ,

la cual es elegida para producir la solución trivial  $W=0$ . Minimizar  $\overline{D^2}$  sujeto a la restricción de arriba es equivalente a minimizar

$$S = 2 \sum \lambda_k w_{kk}^2 - \gamma (\prod_{k=1}^m w_{kk} - 1) \quad \dots(1.2.5)$$

Tomando la derivada parcial de (1.2.5) con respecto a  $w_{kk}$  e igualando la ecuación a cero, obtenemos:

$$w_{kk} = \pm \frac{\sqrt{\gamma}}{2\sqrt{\lambda_k}} \quad \dots(1.2.6)$$

donde el multiplicador de Lagrange esta dado por

$$\gamma = 4 \left( \prod_{k=1}^m \sqrt{\lambda_k} \right)^{2/m} \quad \dots(1.2.7)$$

Combinando (1.2.6) en (1.2.7) se produce:

$$w_{kk} = \frac{1}{\sqrt{\lambda_k}} \left( \prod_{j=1}^m \sqrt{\lambda_j} \right)^{1/m} \quad \dots(1.2.8)$$

Así, la matriz de ponderaciones  $W$  está dada por



$$W = \left( \prod_{j=1}^m \lambda_j \right)^{1/2m} \begin{pmatrix} \lambda_1^{-1/2} & 0 & \dots & 0 \\ 0 & \lambda_2^{-1/2} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \lambda_m^{-1/2} \end{pmatrix} \quad \dots(1.2.9)$$

Sustituyendo (1.2.8) en (1.2.4) obtenemos la mínima distancia interconjuntos como

$$\overline{D^2} = 2m \left( \prod_{k=1}^m \lambda_k \right)^{1/m}$$

Esta ecuación implica que  $\overline{D^2}$  será un mínimo global si se utilizan los  $m$  valores propios más pequeños. Entonces, si queremos minimizar la distancia interconjuntos, los correspondientes vectores propios asociados a los más pequeños valores propios de la matriz de covarianzas  $C_x$  deben ser elegidos como los vectores de las características.

Por otro lado, al utilizar la restricción  $\sum_{k=1}^m w_{kk} = 1$ , se puede mostrar que los coeficientes de ponderación son [10]

$$w_{kk} = \left( \sum_{j=1}^m \frac{1}{\lambda_j} \right)^{-1} \frac{1}{\lambda_k}$$

Así  $\overline{D^2}$  tomará el mínimo global mientras se elijan los valores propios  $\lambda_j$  como los  $m$  más pequeños de los  $n$  valores propios de la matriz de covarianza  $C_x$ , y la matriz de transformación  $A$  es construida con los correspondientes  $m$  vectores propios.

### 1.2.2 Análisis de la Técnica

**a) Presupuestos:** Este método transforma el espacio original en otro, donde se evidencian los agrupamientos y se disminuye la dimensionalidad del espacio. Todo el análisis se basa en la matriz de covarianzas y por lo tanto parte del presupuesto de que los objetos están descritos en término de variables con espacio de representación en los reales.

**b) Significación y alcance:** La idea es ingeniosa, pero debe cuidarse la interpretación del resultado, ya que no se debe perder de vista que el conjunto de variables final, es una transformación del conjunto de variables inicial, y puede ocurrir que el significado inicial del rasgo no se mantenga en el nuevo espacio.

### 1.3 SELECCIÓN DE CARACTERÍSTICAS A TRAVÉS DE LA MINIMIZACIÓN DE LA ENTROPÍA. [10]

#### 1.3.1 Exposición de la técnica

La entropía es una medida estadística de incertidumbre. El concepto de entropía es usado como un criterio para una selección de características, eligiendo aquellas que permiten reducir la incertidumbre.

Consideremos M clases de objetos cuyas poblaciones se rigen por densidades de probabilidad  $p(\mathbf{x}/w_1), p(\mathbf{x}/w_2), \dots, p(\mathbf{x}/w_m)$ . La entropía de la i-ésima población de patrones es

$$H_i = - \int_x p(x/w_i) \ln p(x/w_i) dx \quad \dots(1.3.1)$$

donde la integral es tomada sobre el espacio de patrones. Notesé que si  $p(x/w_i)=1$ , indica que no existe incertidumbre, por lo tanto  $H_i = 0$ .

Con esta consideración en mente, la idea básica subrayada en este método consiste en determinar una matriz de transformación lineal  $\mathbf{A}$ , la cual opera sobre los vectores de patrones para producir nuevos vectores de menor dimensionalidad. Esta transformación puede ser escrita como

$$\mathbf{y} = \mathbf{A} \mathbf{x} \quad \dots(1.3.2)$$

donde la matriz de transformación se determina al minimizar la entropía poblacional de las clases de patrones bajo consideración. En la ecuación (1.3.2)  $\mathbf{x}$  es un n-ada,  $\mathbf{y}$  es un vector m-dimensional como imagen, de menor dimensionalidad que  $\mathbf{x}$ , y  $\mathbf{A}$  es una matriz  $m \times n$ . Los renglones de la matriz  $\mathbf{A}$  son los m vectores de características seleccionadas  $\mathbf{a}'_1, \mathbf{a}'_2, \dots, \mathbf{a}'_k, \dots, \mathbf{a}'_m$ , los cuales son vectores renglones. Así, la matriz  $\mathbf{A}$  esta dada por

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \\ \vdots \\ \mathbf{a}'_m \end{pmatrix} \quad \dots(1.3.3)$$

Asumiendo que todas las matrices de covarianzas son iguales y siendo  $\mathbf{C}_1 = \mathbf{C}_2 = \dots = \mathbf{C}_M = \mathbf{C}$ , podemos escribir la densidad de probabilidad normal de la i-ésima clase de objetos como

$$p(x/w_i) = \frac{1}{(2\pi)^{n/2} |\mathbf{C}|^{1/2}} \exp\left[-\frac{1}{2}(x - m_i)' \mathbf{C}^{-1} (x - m_i)\right] \quad \dots(1.3.4)$$

El vector de las medias del objeto imagen  $\mathbf{y}$ , denotado por  $\mathbf{m}_i^*$ , partiendo de la ec. (1.3.2) es

$$\mathbf{m}_i^* = \mathbf{A} \mathbf{m}_i \quad \dots(1.3.5)$$

Siendo  $\mathbf{z} = \mathbf{x} - \mathbf{m}_i$ , obtenemos de (1.3.5)

$$\mathbf{y} - \mathbf{m}_i^* = \mathbf{A} (\mathbf{x} - \mathbf{m}_i) = \mathbf{A} \mathbf{z} \quad \dots(1.3.6)$$

La matriz de covarianzas para los vectores de la imagen es entonces:

$$\mathbf{C}^* = E\{(\mathbf{y} - \mathbf{m}_i^*)(\mathbf{y} - \mathbf{m}_i^*)'\} \quad \dots(1.3.7)$$

$$= \mathbf{AE}\{\mathbf{zz}'\}\mathbf{A}' = \mathbf{ACA}' \quad \dots(1.3.8)$$

entonces  $E\{\mathbf{zz}'\} = E\{(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)'\} = \mathbf{C}$ . De las ecuaciones (1.3.5) y (1.3.8) la densidad de probabilidad de los patrones de imágenes es

$$p(y / w_i) = \frac{1}{(2\pi)^{m/2} |\mathbf{ACA}'|^{1/2}} \exp\left[-\frac{1}{2}(y - m_i^*)(\mathbf{ACA}')^{-1}(y - m_i^*)\right] \quad \dots(1.3.9)$$

La entropía de la imagen de objetos es entonces

$$H_i^* = - \int_y p(y / w) \ln p(y / w_i) dy \quad \dots(1.3.10)$$

Substituyendo en la ec. (1.3.9) dentro de la ec. (1.3.10) y minimizando con respecto a los vectores propios de  $\mathbf{C}$  se produce el siguiente resultado:

La función de entropía  $H_i^*$  es minimizada al formar la matriz de transformación  $\mathbf{A}$  de los  $m$  vectores propios normalizados asociados con los valores propios más pequeños de la matriz de covarianzas  $\mathbf{C}$ .

### 1.3.2 Análisis de la técnica

El problema es como seleccionar los  $m$  vectores de características de manera que el vector de medidas  $\mathbf{x}$  sea transformado al vector imagen  $\mathbf{y}$  mientras la función de entropía definida en la ecuación 1.3.1 sea minimizada.

**a)presupuestos:** Se asume que se trabaja con una muestra de objetos que tiene una distribución normal multivariada y por lo tanto su información es completamente caracterizada por su vector de medias y su matriz de covarianzas. Esta matriz es caracterizada entonces por sus vectores propios y sus valores propios. Lógicamente hablamos de espacios de representación en los reales.

**b)Significación y alcance:** Si la muestra de objetos cumple con los requerimientos descritos en el párrafo anterior, podemos entonces, tener claro la diferencia entre seleccionar rasgos y extraer rasgos. El proceso de selección consiste en elegir los  $m$  vectores propios de  $\mathbf{C}$  mientras se satisfacen las condiciones establecidas a lo largo de la exposición técnica del método, y por otra parte, el procedimiento de extracción consiste en la determinación de los valores propios y los vectores propios.

## 1.4 SELECCIÓN DE CARACTERÍSTICAS A TRAVÉS DE EXPANSIONES ORTOGONALES.[10]

### 1.4.1 Exposición de la técnica

#### Caso continuo

Consideremos  $M$  clases de objetos  $w_1, w_2, \dots, w_M$ , donde los objetos son descritos con funciones aleatorias reales y continuas, y sea  $x_i(t)$ ,  $T_1 \leq t \leq T_2$ ,  $i=1,2,\dots,M$ , representando

observaciones de alguna de las  $M$  clases. Entonces los  $x_i(t)$  pueden ser expandidos como una combinación lineal de funciones de base conocidas  $\phi(t)$  como sigue:

$$x_i(t) = \sum c_{ij} \phi_j(t), \quad T_1 \leq t \leq T_2, i=1,2,\dots,M \quad \dots(1.4.1)$$

donde las  $c_{ij}$  son coeficientes aleatorios que satisfacen la condición  $E\{c_{ij}\} = 0$ ; las funciones de base  $\phi(t)$  se asumen como un conjunto determinístico de funciones ortonormales sobre el intervalo  $T_1 \leq t \leq T_2$ .

La función de autocorrelación sobre las  $M$  clases de patrones se define como

$$R(t, s) = \sum_{i=1}^M p(w_i) E\{x_i(t)x_i(s)\} \quad \dots(1.4.2)$$

donde  $p(w_i)$  es la probabilidad a priori de ocurrencia de la  $i$ -ésima clase de patrones y  $E_i\{x_i(t)x_i(s)\}$  indica el operador de valor esperado sobre todas las observaciones de la clase. Como la función de autocorrelación se define usualmente por la expresión  $E_i\{x_i(t)x_i(s)\}$ , entonces la ecuación (1.4.2) representa una función de autocorrelación "promedio" la cual toma en consideración que las funciones aleatorias  $x_i(t)$  puede surgir de más de una fuente, esto es, hay  $M$  fuentes o clases de las cuales estas funciones pueden originarse.

Desde el punto de vista del Reconocimiento de Patrones, la formula dada en (1.4.2) se considera como la principal para tomar en cuenta la existencia de más de una clase de objetos, donde la cantidad  $E_i\{x_i(t)x_i(s)\}$  es confinada a las funciones aleatorias de un origen único en el sentido de Reconocimiento de Patrones.

La sustitución de la ecuación (1.4.1) dentro de la ecuación (1.4.2) produce la siguiente relación

$$R(t, s) = \sum_{i=1}^M p(w_i) E\left\{ \sum_{j=1}^{\infty} c_{ij} \phi_j(t) \sum_{k=1}^{\infty} c_{ik} \phi_k(s) \right\} \quad \dots(1.4.3)$$

Observe el cambio de índices en la expansión de  $x_i(s)$ . Entonces las funciones base son deterministas, la ecuación (1.4.3) puede ser escrita como

$$\begin{aligned} R(t, s) &= \sum_{i=1}^M p(w_i) \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \phi_j(t) \phi_k(s) E\{c_{ij} c_{ik}\} \\ &= \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \phi_j(t) \phi_k(s) \sum_{i=1}^M p(w_i) E\{c_{ij} c_{ik}\} \end{aligned} \quad \dots(1.4.4)$$

Se asume que los coeficientes aleatorios son estadísticamente independientes en el sentido de que

$$\sum_{i=1}^M p(w_i) E\{c_{ij} c_{ik}\} = \begin{cases} \lambda_j & \text{si } j = k \\ 0 & \text{si } j \neq k \end{cases} \quad \dots(1.4.5)$$

donde  $\lambda_i$  es una constante positiva. Bajo estas condiciones, la ecuación (1.4.4) llega a ser

$$R(t, s) = \sum_{j=1}^{\infty} \lambda_j \phi_j(t) \phi_j(s) \quad \dots(1.4.6)$$

Multiplicando ambos lados de la ecuación (1.4.6) por  $\phi_k(s)$  e integrando sobre el intervalo de ortonormalidad se tiene

$$\int_{T_1}^{T_2} R(t, s) \phi_k(s) ds = \int_{T_1}^{T_2} \sum_{j=1}^{\infty} \lambda_j \phi_j(t) \phi_j(s) \phi_k(s) ds \quad \dots(1.4.7)$$

Intercambiando el orden de las sumatorias y la integral se llega a

$$\int_{T_1}^{T_2} R(t, s) \phi_k(s) ds = \sum_{j=1}^{\infty} \lambda_j \phi_j(t) \int_{T_1}^{T_2} \phi_j(s) \phi_k(s) ds \quad \dots(1.4.8)$$

En vista de la ortonormalidad asumida para las funciones bases la ecuación (1.4.8) se reduce a la ecuación integral

$$\int_{T_1}^{T_2} R(t, s) \phi_k(s) ds = \lambda_k \phi_k(t) \quad \dots(1.4.9)$$

La expansión dada en la ecuación (1.4.1), donde las funciones base son determinadas por (1.4.8) o (1.4.9) y las funciones de autocorrelación son calculadas de acuerdo a la ecuación (1.4.2), y son conocidas como *expansión generalizada K-L (Karhunen-Loève Expansión)*. El término "generalizada" se usa para indicar que  $R(t, s)$  se calcula de la ecuación (1.4.2) más que de  $E\{x_i(t)x_i(s)\}$ , la cual es la definición normal de la función de autocorrelación.

Los procesos de la expansión K-L tienen las siguientes propiedades óptimas: (1) minimizan el error cuadrado medio siempre que se use un número finito de funciones base en la ecuación (1.4.1), y (2) se minimiza la función de Entropía definida en términos del promedio cuadrado de los coeficientes usados en la expansión. La primera propiedad es importante porque garantiza que ninguna expansión producirá una baja aproximación del error en el sentido del cuadrado medio. El significado de la segunda propiedad es que asocia con los coeficientes de la expansión una medida de minimización de la entropía o dispersión.

### Caso discreto

Si las funciones  $x_i(t)$  son muestras uniformes ejemplificadas en el intervalo  $T_1 \leq t \leq T_2$  pueden ser representadas de la siguiente manera vectorial

$$x_i = \begin{pmatrix} x_i(t_1) \\ x_i(t_2) \\ \vdots \\ x_i(t_n) \end{pmatrix} \quad \dots(1.4.10)$$

donde n es el número de ejemplares o pruebas tomadas en el intervalo de definición de  $x_i(t)$  La ecuación (1.4.1) se transforma en la suma finita

$$x_i = \sum_{j=1}^n c_{ij} \phi_j \quad \dots(1.4.11)$$

donde se asume que los coeficientes satisfacen  $E\{c_{ij}\}=0$ , y  $\phi_j$  es un vector

$$\phi_j = \begin{pmatrix} \phi_j(t_1) \\ \phi_j(t_2) \\ \vdots \\ \phi_j(t_n) \end{pmatrix} \quad \dots(1.4.12)$$

Si los coeficientes son representados en la siguiente forma vectorial:

$$c_i = \begin{pmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{in} \end{pmatrix} \quad \dots(1.4.13)$$

donde  $E\{c_i\}=0$ , la ec. (1.4.11) puede ser expresada más convenientemente en forma matricial,

$$x_i = \Phi c_i \quad \dots(1.4.14)$$

donde  $\Phi$  es la matriz

$$\Phi = (\phi_1 \phi_2 \dots \phi_n) \quad \dots(1.4.15)$$

La función de autocorrelación análoga de la ec. (1.4.2) para el caso discreto es la matriz de autocorrelación, definida como

$$R = \sum_{i=1}^M p(w_i) E\{x_i x_i'\} \quad \dots(1.4.16)$$

Substituyendo la ec. (1.4.14) por  $x_i$  se obtiene

$$\begin{aligned} R &= \sum_{i=1}^M p(w_i) E\{\Phi c_i c_i' \Phi'\} \\ &= \Phi \left( \sum_{i=1}^M p(w_i) E\{c_i c_i'\} \right) \Phi' \end{aligned} \quad \dots(1.4.17)$$

donde el segundo paso sigue de la naturaleza determinista de la matriz  $\Phi$ . Si requerimos que

$$\sum_{i=1}^M p(w_i) E\{c_i c_i'\} = D_\lambda \quad \dots(1.4.18)$$

donde  $D_\lambda$  es la matriz diagonal

$$D_\lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix} \quad \dots(1.4.19)$$

la ecuación (1.4.17) se reduce a

$$R = \Phi D_\lambda \Phi' \quad \dots(1.4.20)$$

Si los vectores bases  $\phi_j$  son asumidos como ortonormales, postmultiplicando la ec. (1.4.20) por la matriz  $\Phi$  se tiene

$$\begin{aligned} R\Phi &= \Phi D_\lambda \Phi' \Phi \\ &= \Phi D_\lambda \end{aligned} \quad \dots(1.4.21)$$

como  $\Phi' \Phi = I$  porque se asumió ortonormalidad de los vectores base que componen  $\Phi$ .

En vista de la ec. (1.4.21) es evidente que

$$R\phi_j = \lambda_j \phi_j \quad \dots(1.4.22)$$

la cual es la fórmula análoga discreta de (1.4.9).

De la ecuación (1.4.22) y la definición de valor propio y vector propio, vemos que el  $j$ -ésimo vector base usado en la expansión dada en la ec. (1.4.19) es simplemente el vector propio de la matriz de correlación correspondiente al  $j$ -ésimo valor propio. Entonces los vectores base son los vectores propios de la matriz simétrica real, siendo mutuamente ortogonales. Si en adición, son ortonormalizados, entonces

$$\phi_j' \phi_k = \begin{cases} 1 & \text{si } j = k \\ 0 & \text{si } j \neq k \end{cases} \quad \dots(1.4.23)$$

que fueron las condiciones mostradas en la ec. (1.4.21). Sobre la base de esta propiedad los coeficientes de la expansión pueden ser obtenidos como sigue:

$$\Phi c_i = x_i, \quad \Phi' \Phi c_i = \Phi' x_i, \quad c_i = \Phi' x_i \quad \dots(1.4.24)$$

Puede verificarse por sustitución directa que estos coeficientes satisfacen la condición establecida en la ec. (1.4.19). En adición, vemos de la ec. (1.4.24) que la condición  $E\{c_i\}=0$  tiene una interpretación alternativa

$$E\{c_i\} = E\{\Phi' x_i\} = \Phi' E\{x_i\} = 0$$

lo cual indica que la condición de que  $E\{c_i\}=0$  es automáticamente satisfecha si los diversos patrones de población son caracterizados por una media estadística igual a cero.

## APLICACIÓN DE LA EXPANSIÓN K-L A LA SELECCIÓN DE CARACTERÍSTICAS

La aplicación de la expansión discreta K-L a la selección de características puede ser vista como una transformación lineal. Si consideramos

$$\Phi = (\phi_1 \quad \phi_2 \quad \dots \quad \phi_m), \quad m < n \quad \dots(1.4.24)$$

como la matriz de transformación, entonces, de la ecuación (1.4.23) los objetos de la imagen son los coeficientes de la *expansión K-L*, esto es, para cualquier objeto  $x_i$  de la clase  $w_i$  sabemos por la ecuación (1.4.23) que

$$c_i = \Phi' x_i$$

Entonces  $\Phi'$  es una matriz de  $n \times m$  y  $x$  es un  $n$ -vector, vemos que, si  $m < n$ , las  $c_i$  son los vectores imagen de menor dimensionalidad.

Las propiedades de optimalidad de la expansión de K-L son satisfechas si las columnas de la matriz transformada  $\Phi$  se eligen como los  $m$  vectores propios *normalizados* para los valores propios más grandes de la matriz de correlación  $R$ .

La notación anterior puede ser expresada en la misma forma como se desarrolló en la sección anterior simplemente definiendo la matriz

$$A = \Phi = \begin{pmatrix} \phi_1' \\ \phi_2' \\ \vdots \\ \phi_m' \end{pmatrix} \quad (1.4.25)$$

donde los renglones de A son los vectores propios *normalizados* asociados a los valores propios más grandes de R. Si hacemos  $y = x$ , -  $\sum_{j=1}^m c_{ij} \phi_j$ ; esto es, igualando la imagen al error, entonces, para cualquier vector x, los vectores imágenes reducidos son dados por

$$y = Ax$$

como antes.

#### 1.4.2.1 Análisis de la técnica

**a) Presupuestos.** La condición  $Efc_j \neq 0$  o su equivalente,  $Efx_j = 0$ , debe ser satisfecha por la expansión K-L para producir un resultado óptimo. Esta condición es automáticamente satisfecha si los objetos de las clases son caracterizados por medias iguales a cero. Si este no es el caso, sólo puede ser esperado un resultado subóptimo de la expansión.

La principal ventaja de este método de selección es que no requiere conocer la densidad de probabilidad.

#### b) Significación y Alcance

Aunque la condición de que todos los objetos de la población deben tener una media idéntica es ciertamente una limitación de la expansión K-L, uno no debe concluir que el método queda sin ningún mérito. La propiedad de la minimización de la entropía tiene el efecto deseable del agrupamiento de objetos.

El método puede ser resumido en los siguientes pasos.

- 1.- Calcular la matriz de correlación R de los objetos del conjunto
- 2.- Obtener los valores propios R y sus correspondientes vectores propios.
- 3.- Formar la matriz de transformación  $\Phi$  de los  $m$  vectores propios correspondientes a los más grandes valores propios de R.
- 4.- Calcular los coeficientes de la expansión de la formula (1.4.23)



## 1.5. SELECCIÓN DE VARIABLES A TRAVÉS DE APROXIMACIÓN FUNCIONAL.[10]

### 1.5.1 Expansión funcional.

En este método se construye un polinomio de aproximación de la siguiente manera:

#### 1.5.1.1 Exposición técnica

Dadas  $M$  clases de objetos, sea  $f_i(x)$  la función que representa las características de la  $i$ -ésima clase. En la determinación de una función de aproximación  $f_i^{\wedge}(x)$ , el principal criterio a considerar es que la suma de los pesos multiplicados por el cuadrado de los errores en los puntos observados sea mínimo. Este criterio de error puede ser escrito como

$$e_i = \sum \{u_i(x_{ik}) [f_i(x_{ik}) - f_i^{\wedge}(x_{ik})]^2\}, \quad i = 1, 2, \dots, M \quad \dots(1.5.1)$$

donde  $x_{ik}$  representa el  $k$ -ésimo objeto de la  $i$ -ésima clase,  $N_i$  es el número de patrones en esa clase, y  $u_i(x_{ik})$  son algunas ponderaciones positivas asociadas con los vectores de objetos  $x_{ik}$ . Ahora el problema comienza con la determinación de una aproximación a la función de características  $f_i^{\wedge}(x)$  para cada objeto de la clase tal que las  $M$  funciones de error en la ecuación (1.5.1) sean minimizada.

La aproximación funcional  $f_i^{\wedge}(x)$  puede ser expresada como una combinación lineal de funciones base de la forma:

$$f_i^{\wedge}(x) = \sum_{j=1}^{m_i} c_{ij} \phi_{ij}(x) = c_i' \phi_i(x) \quad \dots(1.5.2)$$

En esta ecuación,

$$\phi_i(x) = \begin{pmatrix} \phi_{i1}(x) \\ \phi_{i2}(x) \\ \vdots \\ \phi_{ij}(x) \\ \vdots \\ \phi_{im_i}(x) \end{pmatrix} \quad \dots(1.5.3)$$

y

$$c_i = \begin{pmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{ij} \\ \vdots \\ c_{im_i} \end{pmatrix} \quad \dots(1.5.4)$$

donde  $m_i$  es el número de términos usados en la aproximación de la  $i$ -ésima función de características, y las  $\{\phi_{ij}(x)\}$  son funciones linealmente independientes sobre las observaciones discretas  $x_{i1}, x_{i2}, \dots, x_{iN_i}$ . En vista de la ecuación (1.5.2), el valor de la función de aproximación  $f_i^{\wedge}(x)$  depende de los coeficientes  $c_{ij}$ ,  $j=1, 2, \dots, m_i$ , con  $m_i < N_i$ . Si  $m_i$  es igual a  $N_i$ , el error podría

reducirse a cero, pero entonces el número de términos en la expansión debería ser igual al número de objetos en la clase  $w_i$ . El mínimo de  $e_i$  puede ser determinado tomando la derivada parcial con respecto a  $c_{ij}$ . La condición para un mínimo es:

$$\frac{\partial e_i}{\partial c_{ij}} = 0 \quad i=1,2,\dots,M; \quad j=1,2,\dots,m_i \quad \dots(1.5.5)$$

lo cual produce un conjunto de ecuaciones algebraicas para la determinación de los coeficientes de la expansión. Una vez que las funciones  $\phi_j(x)$  son preseleccionadas y los coeficientes calculados, la función de aproximación de características  $f_i^*(x)$  se sigue de la ecuación (1.5.1).

Sustituyendo la ecuación (1.5.2) dentro de la ecuación (1.5.1) y tomando la derivada parcial se obtiene que

$$\frac{\partial e_i}{\partial c_{ij}} = - \sum_{k=1}^{N_i} \left\{ 2u_i(x_{ik}) \left[ f_i(x_{ik}) - \sum_{j=1}^{m_i} c_{ij} \phi_j(x_{ik}) \right] \phi_j(x_{ik}) \right\} \quad \dots(1.5.6)$$

Igualando la derivada parcial a cero y simplificando se tiene:

$$\sum_{k=1}^{N_i} u_i(x_{ik}) \phi_j(x_{ik}) \left[ \sum_{j=1}^{m_i} c_{ij} \phi_j(x_{ik}) \right] = \sum_{k=1}^{N_i} u_i(x_{ik}) \phi_j(x_{ik}) f_i(x_{ik}), \quad i = 1,2,\dots,M \quad \dots(1.5.7)$$

Expresando la ecuación (1.5.7) en forma matricial resulta la siguiente condición para un mínimo:

$$\mathbf{B}_i \mathbf{c}_i = \mathbf{v}_i \quad \dots(1.5.8)$$

donde  $\mathbf{B}_i$  es una matriz simétrica definida positiva de  $m_i \times m_i$  con elementos

$$b_{lq} = \sum_{k=1}^{N_i} u_i(x_{ik}) \phi_l(x_{ik}) \phi_q(x_{ik}) \quad \dots(1.5.9)$$

y  $\mathbf{v}_i$  es un  $m_i$ -vector con componentes

$$v_{il} = \sum_{k=1}^{N_i} u_i(x_{ik}) \phi_l(x_{ik}) f_i(x_{ik}) \quad \dots(1.5.10)$$

Como se ha asumido que las funciones base  $\phi_j(x)$  son linealmente independientes sobre los objetos de la clase  $w_i$ , no es difícil mostrar que  $\mathbf{B}_i$  posee un inverso. Por lo tanto, los coeficientes de la expansión están dados por

$$\mathbf{c}_i = \mathbf{B}_i^{-1} \mathbf{v}_i, \quad i=1,2,\dots,M \quad \dots(1.5.11)$$

El cálculo de estos coeficientes puede ser simplificado eligiendo las funciones  $\phi_j(x)$  de manera que ellas sean ortogonales con respecto al factor de ponderación  $u_i(x_{ik})$ . Entonces estas funciones satisfacen la condición

$$\sum_{k=1}^{N_i} u_i(x_{ik}) \phi_l(x_{ik}) \phi_q(x_{ik}) = 0, \quad l \neq q \quad \dots(1.5.12)$$

y la matriz  $\mathbf{B}_i$  es entonces reducida a una forma diagonal, simplificando considerablemente los cálculos de  $\mathbf{B}_i^{-1}$ . Bajo la condición expresada en (1.5.12), los coeficientes de la expansión llegan a ser

$$c_{ij} = \frac{\sum_{k=1}^{N_i} u_i(x_{ik}) \phi_{il}(x_{ik}) f_i(x_{ik})}{\sum_{k=1}^{N_i} u_i(x_{ik}) \phi_{ij}^2(x_{ik})} \quad \dots(1.5.13)$$

o en forma vectorial,

$$c_i = \mathbf{B}_{iD}^{-1} \mathbf{v}_i \quad \dots(1.5.14)$$

donde  $\mathbf{B}_{iD}$  es una matriz diagonal con elementos

$$b_{ii} = \frac{1}{\sum_{k=1}^{N_i} u_i(x_{ik}) \phi_{il}^2(x_{ik})} \quad \dots(1.5.15)$$

Si en adición, las funciones  $\phi_{ij}(x)$  son elegidas como ortonormales con respecto al factor de ponderación, entonces

$$c_{ij} = \sum_{k=1}^{N_i} u_i(x_{ik}) \phi_{ij}(x_{ik}) f_i(x_{ik}) \quad \dots(1.5.16)$$

y, entonces  $\mathbf{B} = \mathbf{Y}$  bajo esta condición, tenemos de (1.5.11) que

$$c_i = \mathbf{I} \mathbf{v}_i = \mathbf{v}_i \quad \dots(1.5.17)$$

### 1.5.2.1 Análisis de la técnica

**a) Presupuestos:** Este tipo de selección es motivado por el teorema de aproximación de Weierstrass, el cual, establece que cualquier función que sea continua en un intervalo cerrado puede ser uniformemente aproximada con una tolerancia predefinida sobre ese intervalo por algún polinomio. Por supuesto, se asume entonces que los rasgos de los objetos de cada clase pueden ser caracterizados por una función la cual se determina sobre la base de los datos observados. El método por lo tanto se aplica sobre datos determinísticos.

**b) Significación y alcance:** En este caso, las características de cada clase son simplemente los vectores de coeficientes  $c_i$ ,  $i=1,2,\dots,M$ . El proceso de selección consiste en elegir suficientes coeficientes de modo tal que el error definido en (1.5.1) sea suficientemente pequeño. Si la suma de los cuadrados de los errores en los puntos de los objetos observados no es suficientemente pequeña, podemos obtener una aproximación de un orden mayor introduciendo términos extras  $c_{i,m+1} \phi_{i,m+1}(x)$ , donde  $\phi_{i,m+1}(x)$  es otra función ortonormal u ortogonal.

Además los coeficientes determinados previamente permanecen sin cambio. Uno puede observar sin embargo, que el error de aproximación no proporciona una indexación directa de cada una de las funciones de características resultantes. En muchas situaciones, errores relativamente grandes en la aproximación pueden ser tolerados, sin introducir una degeneración en la realización de los sistemas de reconocimiento de patrones.

## 1.6 SELECCIÓN DE CARACTERÍSTICAS A TRAVÉS DE LA MAXIMIZACIÓN DE LA DIVERGENCIA. [10]

### 1.6.1 EXPOSICIÓN TÉCNICA:

Considere dos poblaciones de patrones,  $w_1$  y  $w_2$ , gobernadas respectivamente por las siguientes funciones de densidad de probabilidad  $p_1(x) = p(x/w_1)$  y  $p_2(x) = p(x/w_2)$ . La divergencia entre las dos clases viene dada por:

$$J_{12} = \int_x [p_1(x) - p_2(x)] \ln \frac{p_1(x)}{p_2(x)} dx \quad \dots(1.6.1)$$

La divergencia en este caso es usada como una función criterio para generar un conjunto óptimo de características. Como en algunos de los métodos vistos antes, buscaremos una matriz de transformación  $\mathbf{A}$  que produzca una imagen de objetos con menor dimensionalidad. La imagen de cada objeto como antes, viene dada por

$$y = \mathbf{A}x \quad \dots(1.6.2)$$

donde  $y$  es un  $m$ -vector,  $x$  es un  $n$ -vector, y  $\mathbf{A}$  es una matriz  $m \times n$  cuyos renglones son vectores linealmente independientes  $\mathbf{a}_k$ ,  $k=1,2,\dots,m < n$ . La divergencia de la imagen de objetos viene dada por

$$J'_{12} = \int_y [p_1(y) - p_2(y)] \ln \frac{p_1(y)}{p_2(y)} dy \quad \dots(1.6.3)$$

Asumiendo que las clases  $w_1$  y  $w_2$  se encuentran normalmente distribuidas de acuerdo con  $N(\mathbf{m}_1, C_1)$  y  $N(\mathbf{m}_2, C_2)$  respectivamente. El vector de las medias de la imagen está dado por

$$\mathbf{m}_1^* = \mathbf{A}\mathbf{m}_1, \quad \mathbf{m}_2^* = \mathbf{A}\mathbf{m}_2 \quad \dots(1.6.4)$$

y las matrices de covarianza son

$$\mathbf{C}_1^* = \mathbf{A}\mathbf{C}_1\mathbf{A}', \quad \mathbf{C}_2^* = \mathbf{A}\mathbf{C}_2\mathbf{A}' \quad \dots(1.6.5)$$

Bajo estas condiciones la divergencia de la población de la imagen es dada por

$$J_{12}^* = \frac{1}{2} \text{tr}[(\mathbf{C}_2^*)^{-1}\mathbf{C}_1^* + (\mathbf{C}_1^*)^{-1}\mathbf{C}_2^*] - m + \frac{1}{2} \text{tr}\{[(\mathbf{C}_1^*)^{-1} + (\mathbf{C}_2^*)^{-1}] \delta^*(\delta^*)'\} \quad \dots(1.6.6)$$

donde

$$\delta^* = \mathbf{A}\delta = \mathbf{A}(\mathbf{m}_1 - \mathbf{m}_2) = \mathbf{m}_1^* - \mathbf{m}_2^* \quad \dots(1.6.7)$$

entonces la traza de la matriz es igual a la suma de los valores propios, teniendo entonces que

$$J_{12}^* = \frac{1}{2} \sum_{k=1}^m (\lambda_k + \lambda_k^{-1}) - m + \frac{1}{2} \lambda_{m+1} + \frac{1}{2} \lambda_{m+2} \quad \dots(1.6.8)$$

Como  $\lambda_k$  son los valores propios de  $(\mathbf{C}_2^*)^{-1}(\mathbf{C}_1^*)$ ,  $\lambda_{m+1}$  son los valores propios de  $(\mathbf{C}_1^*)^{-1}\delta^*(\delta^*)'$  y  $\lambda_{m+2}$  son los valores propios de  $(\mathbf{C}_2^*)^{-1}\delta^*(\delta^*)'$ . La diferencial de la ecuación (1.6.8) es

$$dJ'_{12} = \frac{1}{2} \sum_{k=1}^m (1 - \lambda_k^{-1}) d\lambda_k + \frac{1}{2} d\lambda_{m+1} + \frac{1}{2} d\lambda_{m+2} \quad \dots(1.6.9)$$

puesto que  $\lambda_k$  son los valores propios de  $(C_2^*)^{-1}(C_1^*)$ , estos satisfacen las siguientes relaciones:

$$(AC_2A')^{-1}(AC_1A')e_k = \lambda_k e_k \quad \dots(1.6.10)$$

ó

$$(AC_1A')e_k = \lambda_k (AC_2A')e_k \quad \dots(1.6.11)$$

donde  $e_k$  es un vector propio de  $(C_2^*)^{-1}(C_1^*)$  asociado con  $\lambda_k$ . Tomando la diferencial de (1.6.11) obtenemos

$$\begin{aligned} (dA)(C_1A' - \lambda_k C_2A')e_k + (AC_1 - \lambda_k AC_2)(dA')\lambda_k \\ = - (AC_1A' - \lambda_k AC_2A')d e_k + (d \lambda_k) AC_2A' e_k \end{aligned} \quad \dots(1.6.12)$$

puesto que  $C_1^*$  y  $C_2^*$  son simétricas, los valores propios serán mutuamente ortogonales con respecto a  $C_2^*$ . Podemos encontrar un conjunto completo de vectores propios, y  $de_k$  puede ser escrito como

$$de_k = \sum_{j=1}^m c_{jk} e_j \quad \dots(1.6.13)$$

donde los vectores propios  $e_j$  son normalizados con respecto a  $C_2^*$ . Substituyendo la ecuación (1.6.13) dentro de la ecuación (1.6.12) se produce

$$\begin{aligned} (dA)(C_1A' - \lambda_k C_2A')e_k + (AC_1 - \lambda_k AC_2)(dA')e_k = \\ \sum c_{jk} (\lambda_j - \lambda_k) AC_2A' e_j + AC_2A' e_k (d \lambda_k) \end{aligned} \quad \dots(1.6.14)$$

Premultiplicando por  $e_k'$  y haciendo uso de la condición

$$e_k' AC_2A' e_k = 1 \quad e_j' AC_2A' e_k = 0, \quad j \neq k \quad \dots(1.6.15)$$

resulta

$$\begin{aligned} d\lambda_k = e_k'(dA)(C_1A' - \lambda_k C_2A')e_k + e_k'(AC_1 - \lambda_k AC_2)(dA')e_k \\ = 2 e_k'(dA)(C_1A' - \lambda_k C_2A')e_k \end{aligned} \quad \dots(1.6.16)$$

Las diferenciales  $d\lambda_{m+1}$  y  $d\lambda_{m+2}$  pueden ser determinadas de manera similar. Puesto que  $\lambda_{m+1}$  es un valor propio de  $(C_1^*)^{-1}\delta^*(\delta^*)$ , se satisface la relación

$$(A\delta\delta'A)e_{m+1} = \lambda_{m+1}(AC_1A')e_{m+1} \quad \dots(1.6.17)$$

Tomando la diferencial de esta ecuación se obtiene:

$$\begin{aligned} (dA)(\delta\delta'A' - \lambda_{m+1} C_1A')e_{m+1} + (A\delta\delta' - \lambda_{m+1} AC_1)(dA')e_{m+1} \\ = - (A\delta\delta'A' - \lambda_{m+1} AC_1A')d e_{m+1} + AC_1A'e_{m+1} (d \lambda_{m+1}) \end{aligned} \quad \dots(1.6.18)$$

Partiendo del hecho que las matrices  $C_1^*$  y  $\delta^*(\delta^*)'$  son simétricas, todos los vectores propios son mutuamente ortogonales con respecto a  $C_1^*$ . Notamos sin embargo, que  $\delta^*(\delta^*)'$  es de rango 1 y así  $(C_1^*)^{-1}[\delta^*(\delta^*)']$  es también de rango 1. Nosotros podemos encontrar un conjunto completo de vectores propios  $e_{m+1}, \gamma_{21}, \dots, \gamma_{m1}$  de la matriz  $(C_1^*)^{-1}\delta^*(\delta^*)'$ . Estos vectores propios son ortogonales con respecto a  $C_1^*$ , de modo que

$$\begin{aligned} \gamma_{j1}' C_1^* \gamma_{k1} &= 0 \\ e'_{m+1} C_1^* \gamma_{k1} &= 0 \\ \gamma_{j1}' C_1^* \gamma_{j1} &= 1, \quad j \neq k; \quad k=2,3,\dots,m; \quad j=2,3,\dots,m. \\ e'_{m+1} C_1^* e_{m+1} &= 1 \end{aligned} \quad \dots(1.6.19)$$

Los valores propios de  $(C_1^*)^{-1}\delta^*(\delta^*)'$  correspondientes a los vectores propios aumentados  $\gamma_{k1}$ ,  $k=2, 3, \dots, m$ , son iguales a cero. Así, podemos expresar  $d\mathbf{e}_{m+1}$  como

$$d\mathbf{e}_{m+1} = c_1 \mathbf{e}_{m+1} + \sum_{k=2}^m c_k \gamma_{k1} \quad \dots(1.6.20)$$

Substituyendo la ecuación (1.6.20) dentro de la ecuación (1.6.18) y simplificando obtenemos

$$\begin{aligned} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+1} C_1 \mathbf{A}') \mathbf{e}_{m+1} + (\mathbf{A} \delta \delta^* - \lambda_{m+1} \mathbf{A} C_1) (d\mathbf{A}') \mathbf{e}_{m+1} \\ = - \left( \sum_{k=2}^m c_k \lambda_{m+1} \mathbf{A} C_1 \mathbf{A}' \gamma_{k1} \right) + \mathbf{A} C_1 \mathbf{A}' \mathbf{e}_{m+1} (d\lambda_{m+1}) \end{aligned} \quad \dots(1.6.21)$$

Para llegar a la ecuación (1.6.21) se utilizó la ecuación (1.6.17) y el hecho de que los valores propios correspondientes a  $\gamma_{k1}$  son cero. Premultiplicando la ecuación (1.6.21) por  $\mathbf{e}_{m+1}$  y utilizando la ecuación (1.6.19) obtenemos

$$\begin{aligned} d\lambda_{m+1} &= \mathbf{e}'_{m+1} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+1} C_1 \mathbf{A}') \mathbf{e}_{m+1} + \mathbf{e}'_{m+1} (\mathbf{A} \delta \delta^* - \lambda_{m+1} \mathbf{A} C_1) (d\mathbf{A}') \mathbf{e}_{m+1} \\ &= 2\mathbf{e}'_{m+1} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+1} C_1 \mathbf{A}') \mathbf{e}_{m+1} \end{aligned} \quad \dots(1.6.22)$$

Análogamente obtenemos

$$d\lambda_{m+2} = 2\mathbf{e}'_{m+2} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+2} C_2 \mathbf{A}') \mathbf{e}_{m+2} \quad \dots(1.6.23)$$

Insertando las ecuaciones (1.6.23), (1.6.22) y (1.6.16) dentro de la ecuación (1.6.9) se obtiene

$$\begin{aligned} dJ^*_{12} &= \sum_{k=1}^m (1 - \lambda_k^{-2}) \mathbf{e}'_k (d\mathbf{A})(C_1 \mathbf{A}' - \lambda_k C_2 \mathbf{A}') \mathbf{e}_k + \mathbf{e}'_{m+1} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+1} C_1 \mathbf{A}') \mathbf{e}_{m+1} \\ &= \mathbf{e}'_{m+1} (d\mathbf{A})(\delta\delta^* \mathbf{A}' - \lambda_{m+2} C_2 \mathbf{A}') \mathbf{e}_{m+2} \end{aligned} \quad \dots(1.6.24)$$

Expresando la ecuación (1.6.24) en términos de la traza tenemos

$$dJ^*_{12} = \text{tr}[(d\mathbf{A})\mathbf{G}] \quad \dots(1.6.25)$$

donde

$$\mathbf{G} = \sum_{k=1}^m (1 - \lambda_k^{-2}) (\mathbf{C}_1 \mathbf{A}' - \lambda_k \mathbf{C}_2 \mathbf{A}') \mathbf{e}_k + \mathbf{e}'_{m+1} (d\mathbf{A}) (\delta \delta' \mathbf{A}' - \lambda_{m+1} \mathbf{C}_1 \mathbf{A}') \mathbf{e}_{m+1} \\ + \mathbf{e}'_{m+1} (d\mathbf{A}) (\delta \delta' \mathbf{A}' - \lambda_{m+2} \mathbf{C}_2 \mathbf{A}') \mathbf{e}_{m+2} \quad \dots (1.6.26)$$

donde  $\lambda_k$  y  $\mathbf{e}_k$  son los valores propios y los vectores propios de  $(\mathbf{A} \mathbf{C}_2 \mathbf{A}')^2 (\mathbf{A} \mathbf{C}_1 \mathbf{A}')$ ;  $\lambda_{k+1}$ ,  $\mathbf{e}_{k+1}$  y  $\lambda_{k+2}$ ,  $\mathbf{e}_{k+2}$  son los valores propios y vectores propios de  $(\mathbf{C}_1 \mathbf{A}')^{-1} (\mathbf{A} \delta \delta' \mathbf{A}')$  y  $(\mathbf{C}_2 \mathbf{A}')^{-1} (\mathbf{A} \delta \delta' \mathbf{A}')$  respectivamente y  $\delta = \mathbf{m}_1 - \mathbf{m}_2$ .

Ya que  $d\mathbf{A}$  es arbitraria, una condición necesaria para que  $J^*_{12}$  sea un extremo es que la matriz  $\mathbf{G}$  sea igual a la matriz cero. El resultado anterior puede ser resumido como sigue:

Si dos poblaciones de objetos  $w_1$  y  $w_2$  se encuentran normalmente distribuidas de acuerdo a  $N(\mathbf{m}_1, \mathbf{C}_1)$  y  $N(\mathbf{m}_2, \mathbf{C}_2)$ , respectivamente, y si los objetos son mapeados dentro de un espacio de menor dimensionalidad por la transformación  $\mathbf{y} = \mathbf{A}\mathbf{x}$ , donde  $\mathbf{y}$  es un  $m$ -vector,  $\mathbf{x}$  es un  $n$ -vector y  $\mathbf{A}$  es una matriz  $m \times n$ , cuyos renglones son vectores de características  $\mathbf{a}_k$  linealmente independientes, una condición necesaria para que la divergencia  $J^*_{12}$  sea un extremo es que la matriz  $\mathbf{A}$  satisfaga la relación

$$\mathbf{G} = 0$$

Se pueden discutir 3 casos:

*Caso 1.*  $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}$ , y  $\mathbf{m}_1 \neq \mathbf{m}_2$ .

En este caso especial, la transformación de los vectores  $\mathbf{x}$  a los vectores  $\mathbf{y}$  no causa pérdida de información ya que  $J_{12} = J^*_{12}$

*Caso 2.*  $\mathbf{C}_1 \neq \mathbf{C}_2$ , y  $\mathbf{m}_1 = \mathbf{m}_2$ .

En vista de la ecuación (1.6.8), la divergencia  $J^*_{12}$  es maximizada si los vectores propios de  $\mathbf{C}_2^{-1} \mathbf{C}_1$  correspondientes a los valores propios  $\alpha_k$  para los cuales

$$(\alpha_k + \alpha_k^{-1}) \geq (\alpha_j + \alpha_j^{-1}), \quad j=1, \dots, n; \quad k=1, 2, \dots, m; \quad j \neq k \quad \dots (1.6.27)$$

se eligen como los renglones de la matriz de transformación  $\mathbf{A}$ .

*Caso 3.*  $\mathbf{C}_1 \neq \mathbf{C}_2 = \mathbf{C}$ , y  $\mathbf{m}_1 \neq \mathbf{m}_2$ .

En el caso general, la solución de la ecuación  $\mathbf{G}=0$  es considerablemente más difícil. De la ecuación (1.6.25),

$$dJ^*_{12} = \text{tr}[(d\mathbf{A})\mathbf{G}] = \sum_{k=1}^m da'_k g_k \\ = (da'_1, da'_2, \dots, da'_m) \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix} \quad \dots (1.6.28)$$

Así,  $(g_1', g_2', \dots, g_m')$  es el gradiente de  $J_{12}^*$  con respecto a las características  $\mathbf{a}_k$ ,  $k=1, \dots, m$ . Por lo tanto, si incrementamos  $\mathbf{A}$  por  $\theta \mathbf{G}$ , donde  $\theta$  es algún factor de convergencia realizable,  $J_{12}^*$  se

incrementará, entonces, podemos considerar algún método de ascenso acelerado con el cual podamos resolver el conjunto de las siguientes ecuaciones :

$$\mathbf{A}(s+1) = \mathbf{A}(s) + \theta \mathbf{G}(s) \quad \text{donde } s \text{ es un índice de iteración.} \quad \dots(1.6.29)$$

### 1.6.2 Análisis de la técnica:

**a) Presupuestos.** El análisis se hace exclusivamente para dos clases con una distribución normal. Los objetos deben poder ser mapeados a un espacio con distancias euclidianas. Entonces los valores con los que se está trabajando deben ser todos reales. Las dos clases además deben de ser separadas, en el sentido de que su intersección sea igual al conjunto vacío.

**b) Significación y Alcance:** Una manera de llevar a cabo la selección y extracción de variables consiste en generar un conjunto de características que tiendan a maximizar la separación entre las clases. Si uno puede derivar un conjunto de características que, sobre alguna combinación con los objetos de dos clases pueden dar como resultado una transformación, que genere un conjunto imagen de objetos que exhiban un incremento en la medida de la separación entre las clases, estas características pueden ser interpretadas como representativas de la disimilaridad entre las poblaciones. La separación entre las clases puede ser medida como una distancia Euclidea, sin embargo, un concepto más abstracto de distancia es el de la divergencia entre los objetos que es tratada como se desarrolló en este método.

## 1.7 ENFOQUE BÁSICO DE LA SEPARABILIDAD DE LAS CLASES.[10]

En este bloque de técnicas se busca el mejor subconjunto dentro de un conjunto original. Por mejor subconjunto se entiende el que logra el valor máximo de una determinada función de criterios  $J(x)$ . Cabe recalcar que la obtención de esta función puede ser completamente subjetiva o completamente compleja de obtener.

Las funciones de criterio miden la capacidad de no cometer errores de clasificación. Se considera entonces que la separación entre las clases influye en la clasificación, y a mayor distancia mejor clasificación. Puede entonces considerarse:

- ⊗ Distancia entre las clases.
- ⊗ Distancia probabilística.
- ⊗ Dependencia probabilística.
- ⊗ Minimización de la entropía.

Sobre la base de la función  $J(x)$  mencionada en el primer párrafo de este punto, se pueden proponer los siguientes algoritmos de búsqueda [9], los cuales se basan en la premisa de que la única vía para garantizar la optimalidad de un proceso de selección de variables es necesaria y además obligatoria una búsqueda exhaustiva (sic):

### 1.7.1. Las mejores variables

#### 1.7.1.1 Exposición de la Técnica:



Se seleccionan las  $d$  variables que resulten las mejores individualmente, denotaremos por  $x_i$ , las  $i$ -ésimas variables que describen a los objetos de nuestra muestra. Las variables se ordenan de forma que:

$$J(x_1) \geq J(x_2) \geq \dots \geq J(x_d)$$

y se toman las primeras  $d$ , con  $d \leq D$ . Es importante señalar que el usuario establece el valor de  $d$ .

### 1.7.1.2 Análisis de la técnica

**A) Presupuestos:** Se supone que se puede encontrar una función  $J$  capaz de cuantificar las ventajas de una rasgo sobre los demás. Ni en este método ni en los siguientes se especifica la manera de construir  $J$ . El usuario debe conocer la cantidad de rasgos que desea obtener en la reducción de la representación original. Se asume que la correlación entre las variables no tiene ningún efecto sobre este proceso.

**B) Significación y Alcance:** La deducción de  $J$  puede tornarse en un trabajo bastante complicado y muchas veces inalcanzable, lo que reduce y complica su aplicación. Simplemente es inaplicable cuando el usuario desconoce el tamaño que debe tener el espacio de representación resultante aun cuando siempre queda la posibilidad de “probar” para distintos valores de  $d$ .

## 1.7.2. Selección Secuencial Hacia Adelante (SFS):

### 1.7.2.1 Exposición Técnica:

Se tiene un subconjunto  $X_k$  formado por  $k$  variables o atributos del conjunto total, se ordenará los restantes atributos de modo que

$$J(X_k + x_1) \geq J(X_k + x_2) \geq \dots \geq J(X_k + x_{D-k}),$$

el nuevo conjunto se construye como  $X_{k+1} = X_k + x_1$ , y la inicialización se define  $X_0 = \emptyset$

### 1.7.2.2 Análisis de la Técnica:

**A) Presupuestos:** Nuevamente contar con  $J$ .

**B) Significación y Alcance:** A diferencia del anterior tiene la ventaja de considerar la calidad de los rasgos solos y luego en su asociación con otros. El problema es que el número de combinaciones crece exponencialmente, los cálculos se realizan sobre todas las combinaciones posibles y generalmente se analiza la misma combinación en más de una ocasión, sólo con el orden alterado de las variables analizadas.

## 1.7.3 Generalización Secuencial Hacia adelante GSFS(r):

Similar al anterior, pero aquí en lugar de añadir una variable cada vez se agregan  $r$  atributos al conjunto vigente en cada paso.

**1.7.4. Selección Secuencial Hacia Atras SBS:** En este caso se asume que  $X_0 = X_D$ . Para formar el conjunto  $X_k$  se eliminan k mediciones. Para llegar al subconjunto reducido  $X_{k+1}$  se ordenan los elementos  $x_i$  de  $X_k$  de modo tal que

$$J(X_k - x_1) \geq J(X_k - x_2) \geq \dots \geq J(X_k - x_{D-k}) \text{ y } X_{k+1} = X_k - x_1$$

este método es más caro en términos de tiempo de cálculo.

**1.7.5. Generalización de la Selección Secuencial Hacia Atras GSBS(l):** Es igual que el anterior pero aquí se eliminan l cada vez. Es inflexible en el sentido de que no se eligen cuales l variables, sino las últimas l.

**1.7.6. selección (l,r) "Plus l-take away r":** Se tiene  $X_k$  formado por k variables.

INICIALIZACIÓN: Si  $l > r$  el algoritmo es "de abajo hacia arriba" comenzar por el paso 1, con  $k=0$  y  $X_0 = \emptyset$ . Si  $l < r$  el algoritmo es de "arriba hacia abajo" comenzar por el Paso 2, con  $k=D$  y  $X_0 = X_D$ .

PASO 1. Usando el algoritmo SFS añadir l variables  $x_i$  del conjunto de mediciones disponibles  $X_D - X_k$  para formar el conjunto  $X_{k+l}$ . Poner  $k = k + l$ ,  $X_{D-k} = X_k$

PASO 2. Eliminar las r variables peores del conjunto  $X_{D-k}$  con el algoritmo SBS para formar el conjunto  $X_{D-k+r}$ . Poner  $k = k - r$ . Si  $k=d$  el algoritmo termina. Si no, poner  $X_k = X_{D-k}$  y volver al paso 1.

**1.7.7. Algoritmo Generalizado (l,r):** En lugar de SFS y SBS se emplean GSFS(l) y GSBS(r) donde se emplean varios números  $l_i$ ,  $i = 1, 2, \dots, z_1$ , y  $r_i$ ,  $i = 1, 2, \dots, z_2$ .

## 1.8 SELECCIÓN DE CARACTERÍSTICAS BINARIAS. [10]

### 1.8.1.1 Exposición Técnica:

Para N objetos binarios  $O_1, O_2, \dots, O_N$  las características en el iésimo paso en la k-ésima iteración a través de los objetos viene dado por

$$f_k(i) = \begin{cases} f_k(i-1) \cap O_i & \text{si } \|f_k(i-1) \cap O_i\| \geq \theta^* \\ f_k(i-1) & \text{en otro caso} \end{cases}, \quad i = 1, 2, \dots, N. \quad \dots(1.8.1)$$

en donde

$$\theta^* = \theta + \|f_1(M) \cap [f_k(i-1) \cap O_i]\| + \|f_2(M) \cap [f_k(i-1) \cap O_i]\| + \dots + \|f_{k-1}(M) \cap [f_k(i-1) \cap O_i]\| \quad \dots(1.8.2)$$

la descripción de  $O_j$  es el conjunto ordenado formado por los valores binarios de sus características

y

$f_k(\theta)$  es la descripción original del objeto  $O_j$

### 1.8.2.1 Análisis de la Técnica:

**a) Presupuestos:** Este algoritmo se enfoca a determinar un conjunto mínimo de características comunes a un grupo de objetos.

**b) Significación y alcances:** El problema básico consiste en seleccionar un conjunto mínimo de características binarias de la misma dimensión así como los objetos para los cuales basta con reconocer el patrón original, con el mínimo error posible. Aunque este problema no ha sido resuelto en general, el algoritmo presentado es una aproximación razonable (pero no siempre completamente exitosa) para generar características binarias útiles.

Básicamente el procedimiento consiste en establecer una variable como umbral y alterar las características durante cada iteración donde el umbral es excedido. Siempre quedan dos preguntas sin respuesta completamente satisfactoria en casos generales: ¿Cómo seleccionar el umbral  $\theta$ ? y ¿Cuántas características se requieren como resultado de la selección?

## 1.9 SELECCIÓN GRÁFICA DE VARIABLES. [9]

### 1.9.1 Exposición Técnica:

1.9.1.1 Proyección de Sammon: Indica la mayor separabilidad entre las clases de manera gráfica. Es una proyección que trata de preservar las distancias relativas de un espacio mayor a  $n$  al trasladarlo a un espacio de dimensión 2.

1.9.1.2 Proyecciones de Fisher: Trabaja con Glifos como los siguientes:



donde hay un rayo por cada rasgo, la longitud de cada rayo, esta asociada con el valor del rasgo, y existe un círculo por cada elemento.

### 1.9.2 Análisis de la Técnica

Sin lugar a dudas ambas técnicas son más ilustrativas que verdaderas soluciones al problema. En muy contadas ocasiones se cuenta con la información necesaria para poder llevar a cabo las gráficas, y no siempre se puede hacer el mapeo gráfico sin pérdida en la precisión de la comparación entre los rasgos. Pero cuando esto se logra, el proceso de selección resulta fácil y agradable.

## 2. RESUMEN DE REQUERIMIENTOS:

Se muestra ahora a grosso modo un resumen de los requerimientos más importantes de los métodos encontrados en la literatura:

♣ Minimización de la Entropía:

- requiere conocer probabilidades a priori
  - tener espacios métricos.
  - los rasgos o características deben seguir una distribución normal.
- ♣ Ordenamiento de las variables en dependencia de su divergencia:
- se asume independencia entre las variables.
  - se asume una medida de "calidad" a priori.
  - se asume que esta calidad es una función lineal cuando las variables se agrupan.
- ♣ Selección por una expansión de Karhunen-Loève
- Se requiere que los procesos sean aleatorios
  - Que garanticen periodicidad
  - Se aplica en procesos aleatorios cuyo periodo garantice que  $x_n$  y  $x_m$ ,  $n \neq m$ , son no correlacionados.
  - la función que determina el dominio de los objetos debe ser continua.
- ♣ Método de las mejores variables (Selección Secuencial Hacia Adelante, Hacia Atras, y la generalización de ambas):
- Se eligen las k mejores variables, a partir de una función de "optimalidad"
  - Se debe dar de entrada el valor de k.
  - La búsqueda es exhaustiva dentro de todas las combinaciones de variables.
- ♣ Medidas probabilísticas de Separabilidad
- Se define a priori una cota superior de error.
  - Se define sobre espacio métricos.
  - Para 2 clases
- ♣ Selección Gráfica de Variables (Proyección de Sammon, Proyecciones de Fisher)
- Se ve seriamente restringida a la resolución del equipo y capacidad visual.
  - presupone clases disjuntas.
  - puede resultar demasiado subjetiva.

Una de las principales motivaciones de este trabajo fue la de resolver problemas para los cuales la descripción de los objetos pueda estar dada en términos de variables de cualquier tipo, y no necesariamente reales, con lo que muchos de estos dejan de ser factibles. Partimos además del hecho de que podíamos tener más de 2 clases entre las cuales hubiera objetos en común, lo que descarta otros métodos. Asumimos que los rasgos o características no estaban subordinados a algún tipo de distribución probabilística de datos ni contábamos con un especialista que pudiera determinar el número de rasgos óptimos resultado de la selección. Quedaba pues, como única opción los métodos comprendidos en el enfoque lógico combinatorio. Quisiéramos recalcar, que no es esto una insinuación de que estos métodos pudieran ser superiores a los otros, mucho menos la panacea en la selección de variables, simplemente ofrecen posibilidades de solución a problemas donde no es posible asumir ciertos requisitos.

Después de haber analizado estos métodos, podemos describir ahora los métodos que se utilizan en el enfoque lógico combinatorio. En estos métodos se puede uno basar en el concepto de testor típico y una manera de cómo se aborda el problema de encontrar los testores típicos se explicita en el siguiente capítulo.

## Capítulo 2

### EL ALGORITMO RECPLUS

Para seleccionar variables en el enfoque lógico combinatorio, se parte primeramente del hecho que la información con la que se cuenta<sup>1</sup> se encuentra representada por una matriz de aprendizaje, de la cual se extrae una matriz de diferencias y la cual finalmente puede ser reducida a una matriz básica, con la cual se trabaja para encontrar los testores típicos, que son combinaciones de rasgos, variables o características<sup>2</sup> que permiten distinguir cualquier par de objetos de clases distintas. Las definiciones de estas matrices se enlistan a continuación:

**Matriz de Aprendizaje:** es una matriz como la siguiente:

$$\begin{array}{c}
 K'_1 \\
 \vdots \\
 O_h \\
 K'_2 \\
 \vdots \\
 O_r \\
 \vdots \\
 O_s \\
 K'_l \\
 \vdots \\
 O_m
 \end{array}
 \begin{bmatrix}
 X_1 & X_2 & \cdots & X_n \\
 X_1(O_1) & X_2(O_1) & \cdots & X_n(O_1) \\
 \vdots & \vdots & \ddots & \vdots \\
 X_1(O_h) & X_2(O_h) & \cdots & X_n(O_h) \\
 X_1(O_{h+1}) & X_2(O_{h+1}) & \cdots & X_n(O_{h+1}) \\
 \vdots & \vdots & \ddots & \vdots \\
 X_1(O_r) & X_2(O_r) & \cdots & X_n(O_r) \\
 \vdots & \vdots & \ddots & \vdots \\
 X_1(O_s) & X_2(O_s) & \cdots & X_n(O_s) \\
 \vdots & \vdots & \ddots & \vdots \\
 X_1(O_m) & X_2(O_m) & \cdots & X_n(O_m)
 \end{bmatrix}$$

En la cual existe un renglón para la descripción de cada objeto. Cada una de estas descripciones aparece en forma de un n-uplo, donde el i-ésimo lugar de ese n-uplo,  $X_j(O_i)$  contiene el valor del j-ésimo objeto en su i-ésimo rasgo. Estos valores pueden cambiar de dominio o conjunto de valores admisibles de un rasgo a otro. Se acepta que los rasgos sean booleanos, k-valentes, reales, o incluso difusos, puede además considerarse la ausencia de la información, es decir, puede no conocerse el valor para un determinado rasgo dado un objeto. Además los objetos se encuentran agrupados en  $l$  clases,  $K_1, K_2, \dots, K_l$ , de cada una de las cuales se cuenta con una muestra  $K'_b \subseteq K_b$ . Estas clases pueden tener objetos en común, es decir se permite que se solapen y por lo tanto será admisible la multclasificación.

**Ejemplo 1:** En la medicina ocurre de manera frecuente que un mismo paciente tiene más de una enfermedad, entonces, si se considerarán a las enfermedades como clases y a los pacientes como objetos, aquellos pacientes que tiene más de una enfermedad, serían objetos multclasificados, y serían el solapamiento de las clases.

**Matriz de Diferencias:** Se establecen criterios de comparación para cada rasgo, que pueden aplicarse a la matriz de aprendizaje para obtener otra matriz que condense la información que resulta de comparar los valores de los rasgos para cada par de objetos en otra matriz llamada Matriz de Diferencias (MD). Los

<sup>1</sup> Suponiendo que nos encontramos en un problema de clasificación Supervisada

<sup>2</sup> Rasgo, característica o variable son sinónimos en este trabajo.

criterios de comparación permiten definir la igualdad o diferencia entre cada par de objetos para un rasgo predefinido. Por ejemplo, sea  $\delta_i$  un criterio de comparación para el  $i$ -ésimo rasgo, entonces:

$$\delta_i(O_r, O_s) = \begin{cases} 0 & \text{si } X_i(O_r) = X_i(O_s) \\ 1 & \text{si } X_i(O_r) \neq X_i(O_s) \end{cases}$$

Es decir, el resultado de la comparación será igual a 0, si los objetos  $r$  y  $s$  son iguales en su  $i$ -ésimo rasgo, y será igual a 1, cuando estos rasgos sean distintos. La manera en que se define cada  $\delta_i$  depende de la modelación hecha con el especialista.

Los valores obtenidos al aplicar los distintos criterios de comparación entre cada par de objetos de la matriz de aprendizaje que pertenecían a clases distintas, da como resultado la matriz de diferencias:

$$MD = \begin{bmatrix} & X_1 & X_2 & \dots & X_n \\ \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \vdots & \vdots & & \vdots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mn} \end{bmatrix}$$

donde  $\alpha_{ji} = \delta_i(O_h, O_s)$ , donde  $O_h \in K'_p$  y  $O_s \in K'_q$  es el  $j$ -ésimo par de objetos de clases distintas ( $p \neq q$ ) y  $\delta$  el  $i$ -ésimo criterio de comparación o el criterio de comparación para el  $i$ -ésimo rasgo.  $\alpha_{ji} = 0$  si los valores se consideran semejantes y es uno si se consideran diferentes. En resumen, la matriz de diferencias condensa la información que resulta de aplicar los criterios de comparación booleanos  $\delta_i$ .

**Ejemplo 2:** Supongamos la descripción de dos pacientes de un hospital con labio y/o paladar hendido, en base a tres rasgos: edad, tipo de lesión, deformación dental. La segunda variable (tipo de lesión) puede tomar dos valores, primario y secundario, la tercera variable es una variables lingüística, cuyos valores pueden ser, severa, poco severa, poco perceptible, imperceptible. Al comparar las dos descripciones, se define un criterio de comparación para cada rasgo, en el caso de la primera variable se define un criterio, para el cual ambos valores son iguales si tiene una diferencia menor a un año, y diferentes en otro caso.

$$\delta_{\text{edad}}(O_1, O_2) = \begin{cases} 0 & \text{si } |O_1(\text{edad}) - O_2(\text{edad})| \leq 1 \text{ año} \\ 1 & \text{en otro caso} \end{cases}$$

para el segundo rasgo, el criterio de comparación es: son iguales si la lesión es la misma, son distintos en otro caso:

$$\delta_{\text{lesión}}(O_1, O_2) = \begin{cases} 0 & \text{si } O_1(\text{lesion}) = O_2(\text{lesion}) \\ 1 & \text{en otro caso} \end{cases}$$

finalmente, la última variable tendrá un criterio de comparación, en el cual dos valores serán iguales si la deformación dental es la misma y diferentes en otro caso.

$$\delta_{\text{deformacion}}(O_1, O_2) = \begin{cases} 0 & \text{si } O_1(\text{deformacion}) = O_2(\text{deformacion}) \\ 1 & \text{en otro caso} \end{cases}$$

Para los dos pacientes en turno las descripciones son las siguientes:

$O_1 = \{12 \text{ años, 5 meses; primario; severa}\}$

$O_2 = \{11 \text{ años, 10 meses; secundario; poco severa}\}$

entonces la comparación de estas dos descripciones producirá el siguiente renglón en la matriz de diferencias

$$\begin{array}{ccc} & 0 & 1 & 1 \\ \text{lo que se interpreta como que el par de objetos, son iguales en el primer rasgo y distintos en los dos últimos rasgos.} & & & \end{array}$$

A partir de la matriz de diferencias, se puede definir otra matriz que condensa la información de la primera, que se denomina matriz básica y que se define de la siguiente manera:

**Matriz Básica** Es la matriz formada exclusivamente por las filas básicas de la matriz de diferencias. Se dice que una fila  $i_i$  es básica, si y sólo si en M.D. no existe fila alguna  $i_p$ , tal que sea subfila de  $i_i$ . Finalmente se dice que  $i_p$  es una subfila de  $i_i$  si y sólo si:

$$\text{a) } \forall j (a_{ij} = 0 \Rightarrow b_{pj} = 0)$$

$$\text{b) } \exists j (a_{ij} = 1 \wedge b_{pj} = 0).$$

donde  $a_{ij}$  es el  $j$ -ésimo elemento del renglón  $i$ , y  $a_{pj}$  es el  $j$ -ésimo elemento del renglón  $p$ .

**Ejemplo 3:** Suponga la fila  $a = (0\ 0\ 1\ 0)$  y la fila  $b = (1\ 0\ 1\ 0)$ , se dice entonces, en base a la definición de matriz básica, que  $a$  es subfila de  $b$ , porque para todas las posiciones donde  $b$  tiene un cero, la fila  $a$  también tiene cero, y existe por lo menos una posición para la cual, la fila  $b$  tiene 1 y su correspondiente en  $a$  es cero. entonces si no existe ninguna otra fila que sea subfila de  $a$ , ésta será considerada como fila básica y por lo tanto formará parte de la Matriz Básica.

**Ejemplo 4.** Al eliminar de la siguiente matriz de diferencias

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

las filas no básicas (fila 2 y fila 3) produce la siguiente matriz básica, formada exclusivamente por filas básicas (fila 1 y fila 4):

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

El localizar las combinaciones de rasgos que permiten distinguir a cualquier par de objetos que pertenecen a distintas clases, es equivalente a encontrar los testores. Recordemos entonces su definición. Consideremos  $\mathcal{R} = \{x_1, x_2, \dots, x_n\}$  el conjunto de rasgo que describen a los objetos en la matriz básica.

**TESTOR:** Sea  $T \subset \mathcal{R}$ , entonces  $T$  es un testor si no existe en la matriz de diferencias definida por las columnas correspondientes a los rasgos que contiene  $T$ , ningún renglón completo de ceros.

**TESTOR TÍPICO:** Es aquel testor  $T$  al que no se le puede eliminar ningún rasgo sin que aparezca un renglón formado exclusivamente por ceros en la submatriz definida por las columnas de los rasgos correspondientes a  $T$ .

**Ejemplo 5:** Suponga la siguiente matriz básica:

$$MB = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Entonces se dice que el conjunto de variables  $\{x_1, x_3, x_4\}$  forma un testor porque la matriz que se define exclusivamente con la primera, la tercera y la cuarta columna, no tiene ningún renglón formado exclusivamente por ceros. No es testor típico, porque se puede eliminar un rasgo (una columna) sin que deje de ser testor, es decir sin que aparezca un renglón exclusivamente formado por ceros, por ejemplo la cuarta variable (la cuarta columna). Por otra parte  $\{x_1, x_3\}$  si es un testor típico, ya que es testor, pero no se puede eliminar ninguno de sus rasgos sin que deje de ser testor, dicho de otro modo, si elimino la variable  $x_3$  ó  $x_1$  del conjunto, la matriz definida por las columnas correspondientes a estos rasgos, contendrá un renglón exclusivamente formado por ceros (en este ejemplo en particular, la longitud del renglón de ceros es 1).

El algoritmo RECPLUS es un método para localizar los testores típicos en una matriz básica. La idea esencial es de poder contar con un mecanismo para encontrar los testores típicos en un espacio de búsqueda, determinado por todos los subconjuntos posibles del conjunto original de rasgos, es decir por el conjunto potencia  $\mathcal{P}(\mathcal{R})$ . Como este espacio de búsqueda es demasiado grande en la mayoría de los casos, deseamos encontrar los testores típicos, pero sin tener que recorrer todos los subconjuntos de manera exhaustiva. Es decir, poder determinar un orden entre ellos que nos permita descartar subconjuntos completos de combinaciones de rasgos.

Dentro de las características interesantes que presenta el algoritmo RECPLUS es que éste permite localizar los testores típicos (o las combinaciones minimales de rasgos que permiten la diferenciación entre clases):

- A) considerar muestras donde las clases pueden ser no disjuntas;
- B) trabajar sobre la matriz básica lo que lo hace más eficiente en la búsqueda;
- C) permitir utilizar un concepto más flexible de igualdad: los  $\epsilon$ -testores;
- D) no hacer un recorrido exhaustivo por todas las combinaciones posibles de rasgos;

Veamos rápidamente cada uno de estos incisos [11]:

A) Este punto se logra en base a una definición de error, que evidencia cuántas veces un subconjunto de rasgos no logra distinguir a elementos de clases distintas, considerando que estos elementos efectivamente son distintos, y salva el caso en que los elementos son indistinguibles por tratarse del mismo elemento clasificado en clases distintas. La definición dice lo siguiente:

**Definición 1.** Denotemos por  $ET(\mathcal{R})$  al multiconjunto de parejas de descripciones estándar semejantes, que pertenecen a clases distintas con respecto al conjunto completo de rasgos, es decir:

$$ET(\mathcal{R}) = \{ (I(O), I(O')) \mid I(O) \in K_i \wedge I(O') \in K_j, \text{ para } i < j \},$$

con  $i=1, \dots, l-1, j=i+1, \dots, l$  y  $\mathcal{R} = \{x_1, x_2, \dots, x_n\}$ , donde  $l$  es el número de clases distintas.

Sea  $X = \{x_{i_1}, x_{i_2}, \dots, x_{i_s}\}$  con  $s \leq n$ ,

$E(X)$  denota al conjunto de subdescripciones semejantes atendiendo sólo a los rasgos de  $X$  en clases distintas  $(I(O'), I(O))$ , es decir:



$$E(X) = \{ (I(O), I(O')) \mid I(O) \approx I(O'), I(O) \in K_i, I(O') \in K_j, i < j \}$$

donde:

$X$  se lee como "son semejantes bajo el subconjunto de rasgos  $x$ "

Diremos que el **error de  $X$**  es igual a la diferencia entre los cardinales de los conjuntos  $E(X)$  y  $ET(\mathcal{A})$ ).

B) Para ocupar la matriz básica en lugar de la matriz de aprendizaje y asegurar que la búsqueda de los testores típicos tendrá la misma efectividad se requirió demostrar las siguientes proposiciones:

Se define una función de la siguiente manera:

$$e_x: X \rightarrow \mathbb{N}$$

$$x_i \mapsto \|E(X_i)\|$$

es decir, la función que tiene como dominio un subconjunto de rasgos, y como contradominio  $\mathbb{N}$  el conjunto de los números naturales, el número  $\|E(X_i)\|$  simboliza la cantidad de pares de descripciones que se confunden con ese subconjunto de rasgos hasta su  $s$ -ésimo elemento. Si la función  $e_x$  aparece con los subíndices  $M_D$  y  $M_B$  indica la matriz sobre la cual fue aplicada.

**PROPOSICIÓN 1:**  $e_{x_{MA}}(X_i) = e_{x_{MD}}(X_i)$  y  $\|E_{MA}(X)\| = \|E_{MD}(X)\|$ , donde  $\|E_{MA}(X)\|$  es la cardinalidad del conjunto de la definición 1 dada en el inciso A sobre la matriz M.A. y  $\|E_{MD}(X)\|$  la cardinalidad el conjunto formado a partir de la matriz M.D.

**PROPOSICIÓN 2:**  $\|E_{MD}(X)\| \geq \|E_{MB}(X)\|$ .

**PROPOSICIÓN 3:** Si  $X$  es testor con respecto a M.D. entonces  $X$  es testor con respecto a M.B.

**PROPOSICIÓN 3<sup>4</sup>:** El conjunto de testores típicos encontrados en M.D. por RECPLUS es igual al conjunto de testores típicos encontrados en M.B. por RECPLUS.

C) Para hablar de una función de semejanza más flexible, de una expresión matemática que nos permita establecer la semejanza o diferencia entre pares de objetos de clases distintas, podemos referirnos a la expresión más sencilla, la función de diferencia booleana:

$$f(O_i, O_j) = \begin{cases} 0 & \text{si } s = n \\ 1 & \text{en otro caso} \end{cases}$$

donde la función toma el valor 0 si los objetos  $O_i$  y  $O_j$  descritos con  $n$  rasgos son semejantes y 1 si son diferentes,  $s$  es el número de rasgos donde  $O_i$  y  $O_j$  deben coincidir.

Al hablar de  $\epsilon$ -testores, hablamos de descripciones que se consideran diferentes, sólo si el número  $s$  de rasgos en que no coinciden un par de objetos es mayor que un  $\epsilon$  preestablecido, es decir:

<sup>4</sup> La demostración de estas 4 proposiciones se encuentra en [11]

$$f(O_i, O_j) = \begin{cases} 0 & \text{si } s' \leq \varepsilon \\ 1 & \text{si } s' > \varepsilon \end{cases}$$

entonces para diferenciar un par de objetos, se deben encontrar por lo menos  $\varepsilon+1$  rasgos diferenciantes en cada renglón de M.B. y entonces no considerar que el subconjunto de rasgos en turno comete un error. Entonces la definición de error considerada en el inciso A de este capítulo debe contabilizar los errores en función de la siguiente definición:

Sea  $X \subseteq R = \{x_1, x_2, \dots, x_n\}$ . Denotemos por  $\mathcal{E}E(X)$  al conjunto de renglones de M.B. que en la submatriz definida por  $X$  no contengan por lo menos  $\varepsilon+1$  rasgos diferenciantes. Se dirá que el error del renglón  $j$ -ésimo con respecto a  $X_j$  queda definido de la siguiente manera:

$$eren(j, X_{i_k}) = \begin{cases} \varepsilon + 1 - \sum_{h=1}^k a_{ji_h} & \text{si } \sum_{h=1}^k a_{ji_h} \leq \varepsilon + 1 \\ 0 & \text{en otro caso} \end{cases}$$

donde  $X_i$  representa el  $i$ -ésimo subconjunto de rasgos, y el subíndice  $k$  indica que se está tomando de ese subconjunto, hasta el  $k$ -ésimo rasgo. Las letras  $a$  representan los unos del renglón  $j$  en turno.

Se dirá que el épsilon error de  $X$ , denotado por  $\mathcal{E}e_X(x)$  es:

$$\mathcal{E}e_X(x) = \sum_{j=1}^S eren(j, X)$$

donde  $S$  es el número de renglones de la matriz en cuestión.

D) Por último, para no llevar a cabo una búsqueda exhaustiva, se definió un orden entre los elementos del conjunto potencia del conjunto inicial, y se utilizaron las funciones de recorrido definidas por Morales Gamboa [12] que se enlistan a continuación:

DEFINICIÓN [12] : Sea  $\rho : \mathcal{P}(R) \rightarrow \mathcal{P}(R)$ . Entonces

- a) Si  $e_X$  no es inyectiva y  $X$  es testor entonces  $\rho(X) = \sigma(X)$
- b) Si  $e_X$  es inyectiva y  $X$  es testor entonces  $\rho(X) = \beta(X)$
- c) En otro caso, es decir, cuando  $X$  no es testor  $\rho(X) = \alpha(X)$

donde:

- 1)  $\sigma : \mathcal{P}(R) \rightarrow \mathcal{P}(R)$ , tal que si  $X = \{x_1, \dots, x_i\}$ , entonces
- $$\sigma(X) = \{x_i \in X \mid \forall x \in X, \text{ si } x < x_i \text{ entonces } e_X(x) > e_X(x_i)\}$$

y

$$\sigma(\emptyset) = \emptyset$$

- 2)  $\beta : \mathcal{P}(R) \rightarrow \mathcal{P}(R)$ , tal que si  $X = \{x_1, \dots, x_i\}$ , entonces

$$\beta(X) = \{x_1, \dots, x_{i-1}\} \cup \{r \in R \mid r > x_i\}$$

y

$$\beta(\emptyset) = \emptyset$$

- 3)  $\alpha : \mathcal{P}(R) \rightarrow \mathcal{P}(R)$ , tal que si  $X = \{x_1, \dots, x_i\}$ , entonces

$$\alpha(X) = \left\{ x_{i_1}, \dots, x_{i_{j-1}} \right\} \cup \{ r \in \mathbb{R} \mid r > x_{i_j} \}$$

$$\text{Si } x_{i_j} = \max \{ x \in X \mid \exists r \in \mathbb{R} \setminus X, \text{ con } x < r \}$$

y

$$\alpha(X) = \emptyset \quad \text{Si } r < x_{i_1} \quad \forall r \in \mathbb{R} \setminus X$$

El algoritmo donde se empleó esta función de recorrido para encontrar  $\epsilon$ -testores es entonces el siguiente:

Entrada:  $\epsilon$  : el  $\epsilon$ -valor, para determinar los  $\epsilon$ -testores.

$X = \{x_1, \dots, x_n\}$  : el conjunto inicial de rasgos.

MA: la matriz de aprendizaje

PASO 0. Identificación de posibles objetos multclasificados

PASO 1. Construcción de la matriz básica a partir de la matriz de aprendizaje.

PASO 2. Mientras X sea diferente del conjunto vacío hacer

PASO 3. Si  $e_X(X) = 0$  entonces paso 3.1

PASO 3.1. Si  $e_X$  es inyectiva entonces paso 3.1.1.1 y 3.1.1.2.

PASO 3.1.1.1 Guarda el conjunto X en la lista de Testores Típicos.

PASO 3.1.1.2. Aplica  $\beta(X)$ .

si no PASO 3.1.2. Aplica  $\sigma(X)$

si no PASO 3.2. Aplica  $\alpha(X)$

PASO 4. regresar al paso 2.

PASO 5. FIN.

Hasta este punto el algoritmo RECPLUS resolvía el problema de la localización de todos los testores típicos en situaciones en las situaciones descritas, pero empezaron a surgir nuevos problemas con nuevos retos. En el campo de la medicina, nos encontramos con que en muchas ocasiones los médicos definen criterios de comparación que no caben en la lógica clásica o bivalente, es decir, bajo muchas situaciones, el medico no podía afirmar si dos valores eran simplemente iguales o diferentes, sino hablaba de diferencias en cierto grado, de semejanzas hasta cierto punto. La lesión de un labio hendidado era "ligeramente" parecida a otra, o resultaban "casi" iguales. El dolor provocado por una infección era "muy semejante" a otro, aunque no igual. Este tipo de situaciones no es exclusivo de la medicina, se presentaba en muchas otras disciplinas como la sociología, la geología, la psicología, entre otras. ¿Cómo comparar la capacidad intelectual de dos niños?, reducir a se parecen o no se parecen, no resulta ser la mejor solución del problema. En cambio, podemos hablar del parecido "en cierto grado". ¿Es conveniente hablar de la igualdad estricta al comparar el grado de influencia de los medios de publicidad sobre un par de individuos?

Al enfrentarse con este tipo de nuevos problemas, surge la necesidad de localizar testores difusos, al trabajar con matrices de diferencia difusas, en las cuales la semejanza o diferencia entre los rasgos de cada par de objetos no es absoluta, es un concepto relativo, es decir, existe una "gradación" de la diferencia o parecido entre los valores.

Es entonces, que se plantea el trabajo con conjuntos difusos, para la localización de todos los testores difusos en una matriz de diferencias o en una matriz básica difusa. Este será el objetivo a trabajar en los dos siguientes capítulos.

### Capítulo 3

## SUBCONJUNTOS Y TESTORES DIFUSOS

En algunas ciencias, tales como la Ingeniería, la Química o la Física, se construyen modelos matemáticos exactos de los fenómenos reales. Sin embargo la noción de “perfección” o “concepto exacto”, muchas veces no es alcanzado en el proceso de modelado del “mundo real”. Es muy amplia la cantidad y variedad de situaciones en las cuales se encuentra implícitas cierta ambigüedad o difusidad en el proceso de su modelado.

Esta difusidad en el contexto de Reconocimiento de Patrones se puede manifestar como cierta ambigüedad para determinar los límites de un conjunto o clase de objetos. Puede presentarse también, y de manera muy frecuente en el proceso de comparar un par de valores para un mismo rasgo de un par de objetos, que esta diferencia ó semejanza sea relativa ó poco precisa, es decir, el resultado de la comparación no se resume en “ se parecen ó no se parecen” por lo que la lógica bivalente deja de ser una herramienta de solución para su tratamiento.

Esencialmente la difusión es un tipo de inexactitud que al agrupar elementos dentro de un (asi llamado) conjunto difuso, no define únicamente la pertenencia ó no del elemento al conjunto. Se habla entonces del grado de pertenencia del elemento al conjunto. Aquellos elementos que pertenecen completamente al conjunto tienen pertenencia uno. Aquellos elementos que de manera precisa no se consideran en el conjunto se dice que tienen pertenencia 0, y por último el resto de los elementos tiene un grado de pertenencia definido dentro del intervalo  $(0,1)$  en función del grado de integración al conjunto [13].

Existe un gran número de problemas cuya formulación matemática es muy compleja y poco precisa, por lo que requieren un planteamiento en términos de conjuntos difusos, lo que representa un desarrollo natural en la evolución de las ideas científicas. Es por demás decir que se han obtenido mejores resultados para este tipo de problemas “imprecisos” cuando su tratamiento se da desde el punto de vista de los conjuntos difusos. Cuando las respuestas SI ó NO se amplían de un completo SI y de manera continua llegan a un MAS O MENOS, siguiendo de manera continua hasta un NO, se permite construir modelos más apegados a la realidad y por lo tanto más fieles. Es decir, gracias a admitir cierta imprecisión se vuelven más precisos.

Para los fines de este trabajo, el concepto de difusión surge como una necesidad de acercarse a modelos en los cuales el proceso de comparación entre los objetos o entre los rasgos que describen a los objetos, no es de igualdad exacta, y deben considerarse entonces los conceptos de analogía, ó semejanza en cierto grado.

La selección de variables entonces, queda supeditada a un proceso de selección de aquellos rasgos que logran distinguir entre los objetos de distintas clases aún cuando la diferencia entre estos objetos pueda ser gradual ó que se pueda hablar de un “grado de

semejanza” entre los valores de los rasgos que describen a los objetos de la matriz de apendizaje y que lo ameriten surgiendo entonces el concepto de testor difuso y testor difuso típico.

Lo que sigue ahora es dar una introducción más formal de los conjuntos difusos y sus operaciones básicas como base para plantear después los testores difusos.

### 3.1 CONJUNTOS DIFUSOS.

La teoría de los conjuntos difusos [15] trabaja con un subconjunto  $A$  dentro de un universo de discurso  $X$ , donde la transición entre un miembro y un no miembro es gradual. El objeto que pertenece “completamente” al conjunto tendrá un grado de pertenencia al conjunto de 1, el objeto que no pertenece al conjunto tendrá un grado de pertenencia 0 y la función de pertenencia será denotada por  $\mu_A(x)$

$$\begin{aligned}\mu_A(x) &= 1 & \text{si } x \in A \\ \mu_A(x) &= 0 & \text{si } x \notin A\end{aligned}$$

La pertenencia difusa y la no pertenencia difusa pueden ser representadas por  $\in$  y  $\notin$  respectivamente.

Intuitivamente, un conjunto difuso es un conjunto que admite la posibilidad de pertenencia parcial en él. Sea  $X=\{x_1, x_2, \dots, x_n\}$  denotando un espacio de objetos. Entonces un conjunto difuso en  $X$  es un conjunto de pares ordenados

$$A=\{(\mu_A(x_i), x_i)\}, \quad x_i \in X,$$

donde  $\mu_A(x)$  es denominado “el grado de pertenencia de  $x_i$  en  $A$ ” y el conjunto formado por los  $x_i$  cuyos grados de pertenencia asociado es diferente de cero, como el soporte de  $A$  denotado por  $\text{sop}(A)$ . Se asumirá para simplificar que  $\mu_A(x)$  es un número en el intervalo  $[0,1]$ , con el grado 1 y 0 representado respectivamente, la completa pertenencia y la no pertenencia en un conjunto difuso, como se discutió antes, y también será denotado por  $A=\{\mu_A(x_i) \mid x_i \}_{x_i \in X}$ .

En general se distinguen tres tipos de inexactitud: *generalidad*, cuando un concepto se aplica a una variedad de situaciones; *ambigüedad*, cuando describe mas de un concepto indistinguible; y *vaguedad*, que precisa bordes no definidos. Los tres tipos de inexactitud son representados en los conjuntos difusos.

**DEFINICIÓN 3.1.** Un subconjunto difuso,  $A$ , de  $X$  es de tipo 1 si su función de pertenencia,  $\mu_A(x)$ , es un mapeo de  $X$  a  $[0,1]$ ; y  $A$  es de tipo  $K$ ,  $K=2,3,4,\dots$ , si  $\mu_A(x)$  es un mapeo de  $X$  al conjunto de subconjuntos difusos de tipo  $(K-1)$ . Por simplicidad, siempre entenderemos que  $A$  es de tipo 1 si no es especificado lo contrario.

### 3.2. OPERACIONES Y RELACIONES ENTRE CONJUNTOS DIFUSOS.

Sea A y B conjuntos difusos en X. Se pueden realizar las siguientes operaciones sobre los conjuntos difusos.

1. Se dice que los dos conjuntos difusos son iguales ( $A=B$ ) si y sólo si

$$\forall x \in A \quad \mu_A(x) = \mu_B(x) \wedge \forall x \in B \quad \mu_B(x) = \mu_A(x)$$

2. A está contenido en B ( $A \subseteq B$ ) si y sólo si

$$\forall x \in A, \quad \mu_A(x) \leq \mu_B(x).$$

3. La unión de dos conjuntos difusos A y B, denotada por  $A \cup B$ , esta definida como

$$A \cup B = \cup_{x \in X} \{ \max(\mu_A(x), \mu_B(x)) | x \}$$

4. La intersección de dos conjuntos difusos A y B, denotada por  $A \cap B$ , es definida como

$$A \cap B = \cup_{x \in X} \{ \min(\mu_A(x), \mu_B(x)) | x \}$$

5.  $sop(A \cup B) = sop(A) \cup sop(B)$

6.  $sop(A \cap B) = sop(A) \cap sop(B)$

Llamemos conjuntos elementales a los conjuntos A y B. Definamos como conjuntos compuestos a los conjuntos que resultan de aplicar cualquiera de las operaciones anteriores a los conjuntos A y B. Kandel [14], Kaufman [15], Fu y Fung [16], demostraron que se cumplen las siguientes propiedades en las operaciones hasta ahora descritas:

- |  |                       |
|--|-----------------------|
| (1) $A \cap B = B \cap A$                            | leyes conmutativas    |
| (2) $A \cup B = B \cup A$                            |                       |
| (3) $(A \cap B) \cap C = A \cap (B \cap C)$          | leyes asociativas     |
| (4) $(A \cup B) \cup C = A \cup (B \cup C)$          |                       |
| (5) $A \cap A = A$                                   | leyes de idempotencia |
| (6) $A \cup A = A$                                   |                       |
| (7) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ | leyes distributivas   |
| (8) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ |                       |

(9)  $A \cap \emptyset = \emptyset$                       donde  $\emptyset$  es el conjunto vacío tal que  $\forall x \in X: \mu_{\emptyset}(x) = 0$

(10)  $A \cup \emptyset = A$

(11)  $A \cap X = A$

(12)  $A \cup E = E$                       donde E es el conjunto Universo.

5. El *complemento* de A es denotado por  $A^c$  y es definido como

$$A^c = \bigcup_{x \in X} \{(1 - \mu_A(x)) | x\}$$

entonces  $\mu_{A^c}(x) = 1 - \mu_A(x)$ .

(13)  $(A^c)^c = A$     involución

(14)  $(A \cap B)^c = A^c \cup B^c$

Leyes de De Morgan para el caso difuso.

(15)  $(A \cup B)^c = A^c \cap B^c$

lo que no se cumple en los subconjunto difusos es

(16)  $A \cup A^c \neq X$

(17)  $A \cap A^c \neq \emptyset$

Para los subconjuntos difusos, el conjunto potencia o “conjunto de subconjuntos difusos” es infinito, pero puede considerarse un número de valores de pertenencia permisibles y finito, para que de este modo, el conjunto potencia de un conjunto difuso pueda ser finito, para ser usado en el siguiente capítulo y que será definido entonces en los siguientes términos:

DEFINICION 3.2.1: El conjunto potencia de un conjunto difuso será considerado como el conjunto de subconjuntos posibles de  $A = \{(\mu_A(x), x_i)\}$ .

### 3.3 TESTORES DIFUSOS.

El concepto de testor difuso resulta una herramienta que acomoda con bastante precisión a nuestro propósito de seleccionar variables en medios difusos. La idea de los testores difusos se debe en primera instancia a Goldman en 1980, que considera valores infinito-valentes para la comparación de cada rasgo (en su expresión más simple estos valores se encuentran en el intervalo [0,1]).

Formalicemos ahora el concepto de testor difuso. Sea  $\mathcal{R} = \{x_1, x_2, \dots, x_n\}$  el conjunto de rasgos que describen a los objetos a ser comparados.

Sea  $T = \{\mu_{r_1}(x_{r_1})|x_{r_1}, \dots, \mu_{r_k}(x_{r_k})|x_{r_k}\}$  un subconjunto difuso de  $\{x_1, \dots, x_n\}$  tal que  $\forall p \in \{1, \dots, r\} \mu_{r_p}(x_{r_p}) \neq 0$  ... (3.1)

A partir de este convenio Goldman en 1980 propone la siguiente definición de testor típico difuso:

**DEFINICION 3.3.1:**

“ El conjunto  $T$  es un testor difuso con respecto a  $MB$  si

$$\forall A_i \in MB \exists \mu_T(x_{r_p})|x_{r_p} \in T \mid \mu_T(x_{r_p})|x_{r_p} \leq \mu_{A_i}(x_{r_p})|x_{r_p} ” \dots(3.2)$$

donde  $A_i$  denota el  $i$ -ésimo renglón de la matriz básica. de esta definición se desprende los siguientes resultados:

**COROLARIO 3.1[17]:** Sea  $T = \{\mu_{i_1}(x_{i_1})|x_{i_1}, \dots, \mu_{i_s}(x_{i_s})|x_{i_s}\}$  un testor difuso y  $\{\mu_{r_1}(x_{r_1})|x_{r_1}, \dots, \mu_{r_k}(x_{r_k})|x_{r_k}\}$  un subconjunto difuso de  $\mathcal{R}$  que satisface (3.1) y  $\{x_{i_1}, \dots, x_{i_s}\} \cap \{x_{r_1}, \dots, x_{r_k}\} = \emptyset$

entonces  $T' = \{\mu_{i_1}(x_{i_1})|x_{i_1}, \dots, \mu_{i_s}(x_{i_s})|x_{i_s}, \mu_{r_1}(x_{r_1})|x_{r_1}, \dots, \mu_{r_k}(x_{r_k})|x_{r_k}\}$  es también un testor difuso.

**COROLARIO 3.2[17]:** Sea  $T = \{\mu_{r_1}(x_{r_1})|x_{r_1}, \dots, \mu_{r_k}(x_{r_k})|x_{r_k}\}$  un testor difuso y  $T' = \{\mu'_{r_1}(x_{r_1})|x_{r_1}, \dots, \mu'_{r_k}(x_{r_k})|x_{r_k}\}$  tal que satisface (3.1) y  $\forall p \mu'_{r_p} \leq \mu_{r_p}$  con  $p=1, \dots, k$ ; entonces  $T'$  es también un testor difuso. (En este caso se tiene que  $T' \subset T$  y  $sop(T') = sop(T)$ ).

De la definición 3.3.1 y sus corolarios, se deduce que el conjunto de testores difusos de una matriz de aprendizaje resulta ser infinito. Para tratar de salvar este inconveniente, Lazo y Barreto[19] en 1994, sugirieron sustituir la condición (3.1) por la siguiente condición:

$$\forall p(\mu_{r_p}(x) \neq 0 \wedge \exists A_i \in MD \mid \mu_{r_p}(x_{r_p}) \in A_i) \quad p=1, \dots, k. \quad \dots(3.3)$$

y plantean entonces las siguientes definiciones:

“ El conjunto  $T$  es un testor difuso restringido respecto a  $MB$  si satisface (3.3) y (3.1). ”

los corolarios 3.1 y 3.2 son validos para testores restringidos.



Como en el caso clásico, los testores típicos difusos son testores difusos especiales, testores difusos irreducibles, testores difusos que guardan ciertas características que describió Goldman en 1980 y que dice:

“ El conjunto  $T$  es un testor difuso típico respecto a  $MB$  si:

- a)  $T$  es un testor difuso con respecto a  $MB$
- b) cualquier conjunto difuso  $T'$  Formado por subconjuntos puntuales difusos de  $T$ , ( $T' \neq T$ ) no es un testor difuso respecto a  $MB$ .
- c) si  $T \subset T'$  y se tiene que  $sop(T) = sop(T')$  entonces  $T'$  no es un testor difuso respecto a  $MB$ . ”

Kotelnikov por su parte enuncia la siguiente propiedad, para caracterizar a los testores difusos típicos:

“ El conjunto  $T$  es un testor difuso típico respecto a  $MB$  si es un testor difuso respecto a  $MB$ ” y

$$\forall \mu_{r_p}(x_{r_p}) \Big| x_{r_p} \in T \exists A_i \in MD (\mu_{r_p}(x_{r_p}) \Big| x_{r_p} \in A_i \wedge \forall t \in \{1, \dots, k\} \dots (3.4)$$

$$(t \neq p \Rightarrow \mu_T(x_{r_t}) > \mu_{A_i}(x_{r_t})))$$

lo que significa que un testor difuso típico tiene todos sus grados de pertenencia iguales a los grados de pertenencia de cada renglón de la  $MB$  y sólo en uno mayor.

Por lo tanto un testor difuso típico es aquel al cual no puede aumentarse el grado de pertenencia de ninguno de sus rasgos y no se puede eliminar tampoco ninguno de sus rasgos.

## Capítulo 4 RECDIF

### EL ALGORITMO REC EN EL CALCULO DE LOS TESTORES TIPICOS

Establezcamos ahora la definición del algoritmo RECDIF para el cálculo de todos los testores típicos en una matriz de diferencia difusa, para lo cual debemos primero establecer la siguiente notación:

La matriz de diferencias difusas tiene la siguiente forma:

$$MD = \begin{bmatrix} & X_1 & X_2 & \dots & X_n \\ \mu_1(x_1) & \mu_1(x_2) & \dots & \mu_1(x_n) \\ \mu_2(x_1) & \mu_2(x_2) & \dots & \mu_2(x_n) \\ \vdots & \vdots & & \vdots \\ \mu_m(x_1) & \mu_m(x_2) & \dots & \mu_m(x_n) \end{bmatrix}$$

donde  $\mu(x_j)$  es el grado de pertenencia al conjunto “es diferente” del el  $i$ -ésimo par de objetos en su valor correspondiente al  $j$ -ésimo rasgo o variable. A partir de esta matriz introduzcamos el concepto de matriz básica que simplifica la búsqueda de los testores típicos:

**DEFINICIÓN 1.** Diremos que  $A_i$  es subfila de  $A_j$  si

$$\exists t \left[ \mu_{A_i}(x) < \mu_{A_j}(x_t) \right] \wedge \left[ \forall r [r \neq t] \Rightarrow \left[ \mu_{A_i}(x_r) \leq \mu_{A_j}(x_r) \right] \right]$$

**DEFINICION 2.**  $A_i$  es una fila básica de MD si no existe en MD fila alguna que sea subfila de de  $A_i$ , siendo  $A_i \neq A_j$ .

**DEFINICION 3.** Llamaremos matriz básica de MD (o también de MA) a la matriz MB formada exclusivamente por las filas básicas de MD, sin repetir filas.

Consideremos los subconjuntos difusos de  $\mathcal{B}$ ,  $X = \{ \mu_X(x_i) | x_i, \dots, \mu_X(x_i) | x_i \}$  tales que  $\forall s, 1 \leq s \leq r \exists A_k \in MB / \mu_X(x_i) = \mu_{A_k}(x_i)$ . Es decir, se considera el conjunto de todos los subconjuntos difusos de  $\mathcal{B}$ , con todas sus variantes en cuanto a grados de pertenencia, siempre y cuando esos grados existan en MB.

En  $X$  se tendrá  $i_1 < \dots < i_r$ , es decir, las variables o rasgos del soporte del conjunto difuso estarán ordenadas en forma ascendente respecto a su subíndice. A la familia de tales subconjuntos difusos de  $\mathcal{B}$  la denotaremos  $\mathcal{M}(\mathcal{B})$ . Además adoptaremos la siguiente notación para simplificar:

$$X = \{ \mu_X(x_{i_1}) | x_{i_1}, \dots, \mu_X(x_{i_r}) | x_{i_r} \} = \{ \mu_X(x_{i_1}) | x_{i_1} \| X^t \}$$

siendo  $X^t = \{ \mu_X(x_{i_2}) | x_{i_2}, \dots, \mu_X(x_{i_r}) | x_{i_r} \}$ .

Dicho de otra manera, podemos expresar el conjunto  $X$ , explicitando el primer elemento, y denotando el resto del conjunto por  $X^t$ .

En ocasiones utilizaremos una notación más simplificada:

$$X = \{ \mu_X(x_{i_1}) | x_{i_1}, \dots, \mu_X(x_{i_r}) | x_{i_r} \} = \{ \mu_{i_1} | x_{i_1}, \dots, \mu_{i_r} | x_{i_r} \}$$

### 4.1. FORMALIZACIÓN DEL ORDEN

Para localizar todos los testores típicos de una matriz básica, se podría analizar uno a uno todos los subconjuntos posibles del conjunto original de rasgos, de modo que se vayan localizando todos y cada uno de los testores típicos difusos. Este procedimiento naturalmente es bastante engorroso, poco eficiente y por lo tanto caro. En su lugar se desea establecer un orden entre los elementos del conjunto potencia<sup>1</sup> del conjunto original de rasgos, de modo tal, que a partir del análisis de un subconjunto en particular se puedan establecer las características de los conjuntos que le siguen, en relación con su capacidad o no de ser testor típico.

Dado que estamos trabajando sobre una matriz básica, el conjunto de combinaciones posibles del conjunto de rasgos original tomando los grados de pertenencia de esa matriz es un retículo ya que guarda orden parcial, cosa que no favorece nuestros propósitos, porque a partir de una combinación dada, siempre se podría elegir más de un conjunto como “siguiente”

Para ilustrar lo anterior, considérese el siguiente ejemplo:

$$\begin{array}{c}
 X_1 \quad X_2 \quad X_3 \\
 MB = \begin{bmatrix}
 .31 x_1 & .45 x_2 & .81 x_3 \\
 .70 x_1 & .13 x_2 & .63 x_3 \\
 .54 x_1 & .60 x_2 & .23 x_3
 \end{bmatrix}
 \end{array}$$

la porción de el retículo que muestra el número de combinaciones posibles sólo para aquellos subconjuntos con tres rasgos se muestra en el anexo I. A partir de cualquiera de estas opciones hay por lo menos 3 combinaciones posibles. En total el conjunto de subconjuntos posibles tomando sólo los valores de pertenencia que aparecen en la matriz de diferencias tiene una cardinalidad de

$$\sum_{i_1=1}^n V_{i_1} + \sum_{i_1=1}^{n-1} \sum_{i_2=i_1+1}^n V_{i_1} V_{i_2} + \sum_{i_1=1}^{n-2} \sum_{i_2=i_1+1}^{n-1} \sum_{i_3=i_2+1}^n V_{i_1} V_{i_2} V_{i_3} + \dots \sum_{i_1=1}^1 \dots \sum_{i_n=i_{n-1}}^n V_{i_1} \dots V_{i_n}$$

contra los 2<sup>n</sup> elementos que tiene el conjunto potencia en el caso duro. Esta formula resulta de formar los 2<sup>n</sup> subconjuntos posibles del conjunto original de rasgos involucrados, pero a esta combinaciones se agrega para cada rasgo tantos grados de diferencias distintos como haya en la MB, para esa variable. Así, la primera sumatoria contabiliza los subconjuntos de un solo rasgo para los distintos grados de pertenencia, el segundo termino (la doble sumatoria) contabiliza los subconjuntos de dos rasgos con los distintos grados de pertenencia y así sucesivamente, hasta que finalmente el último termino (las n sumatorias consecutivas) contabiliza el numero de subconjuntos posibles con n rasgos para todos los grados de pertenencia permitidos.

Para convertir este orden parcial en un orden total se propone la siguiente definición:

**DEFINICIÓN 4:** Sea  $X_1 = \{\mu_{i_1} | x_{i_1} \| X^{i_1}\}$ ,  $X_2 = \{\mu_{j_1} | x_{j_1} \| X^{i_2}\}$  y sea  $\prec$  una relación tal que  $X_1 \prec X_2$  ( $X_1$  antecede a  $X_2$ )<sup>2</sup> si :

<sup>1</sup> El conjunto potencia para un conjunto difuso fue definido en el capítulo anterior.

<sup>2</sup> Se define además  $\prec \bullet$  como la relación  $X_1$  antecede o es igual a  $X_2$ .

- a)  $X_2 = \emptyset$ .
- b)  $i_i < j_i$ ,
- c)  $i_i = j_i$ , y  $\mu_{i_i}(x_{i_i}) < \mu_{j_i}(x_{j_i})$ ,
- d)  $i_i = j_i$ ,  $\mu_{i_i}(x_{i_i}) = \mu_{j_i}(x_{j_i})$ , y  $X^{t_i} < X^{t_j}$

Esta orden es exactamente inverso al orden propuesto por Lazo[17] en los testores típicos, debido a que se requería poder inferir subconjuntos del conjunto en turno:

**DEFINICIÓN 5.** [17](Lazo). Sea  $\Psi(\text{MB})$  la familia de testores difusos de Goldman de una cierta matriz MB; sean  $t_i (i \in I)$  los elementos de  $\Psi(\text{MB})$  y definamos la siguiente relación sobre los elementos de  $\Psi(\text{MB})$ :

$$t_1 \xi t_2 \leftrightarrow D(t_1, t_2) = (t_1 \cap t_2) \cup ((\text{sup } t_1 \setminus \text{sup } t_2) \cap t_1) \cup ((\text{sup } t_2 \setminus \text{sup } t_1) \cap t_2) = t_2$$

De  $t_1 \xi t_2$  se tiene que  $(\text{sup } t_1 \setminus \text{sup } t_2) = \emptyset$ ; es decir  $(\text{sup } t_1 \subseteq \text{sup } t_2)$ . Si  $(\text{sup } t_1 = \text{sup } t_2)$  entonces  $t_1 \xi t_2 \leftrightarrow t_2 \subseteq t_1$ .

**PROPOSICIÓN 1.**(Lazo)  $\xi$  es una relación de orden.

**DEMOSTRACIÓN:**  $\xi$  es trivialmente reflexiva, para todo  $t \in \Psi$  se tiene que  $t \xi t$ . Fácilmente se prueba que  $\xi$  es antisimétrica a partir de la conmutatividad de  $\cap$  y la asociatividad de  $\cup$ . Demostremos ahora que  $\xi$  es transitiva.

Sean  $t_1, t_2, t_3 \in \Psi$  tales que  $t_1 \xi t_2$  y  $t_2 \xi t_3$ . De  $t_1 \xi t_2$  se tiene que  $\text{sup } t_1 \subseteq \text{sup } t_2$  y de  $t_2 \xi t_3$  se tiene que  $\text{sup } t_2 \subseteq \text{sup } t_3$  de donde resulta que  $\text{sup } t_1 \subseteq \text{sup } t_3$  y entonces  $\text{sup } t_3$  se puede particionar en  $\text{sup } t_1$  y  $\text{sup } t_3 \setminus \text{sup } t_1$ . Sea

$$D(t_1, t_3) = (t_1 \cap t_3) \cup ((\text{sup } t_1 \setminus \text{sup } t_3) \cap t_1) \cup ((\text{sup } t_3 \setminus \text{sup } t_1) \cap t_3)$$

como  $\text{sup } t_1 \subseteq \text{sup } t_3$  entonces

$$D(t_1, t_3) = (t_1 \cap t_3) \cup ((\text{sup } t_3 \setminus \text{sup } t_1) \cap t_3).$$

Sea  $x \in \text{sup } t_1$ , de  $t_1 \xi t_2$ , resulta que  $\mu_{t_1}(x) \geq \mu_{t_2}(x)$  y de  $t_2 \xi t_3$  que  $\mu_{t_2}(x) \geq \mu_{t_3}(x)$ . Luego  $\mu_{t_1}(x) \geq \mu_{t_3}(x)$  y se tiene que  $\mu_D(x) = \mu_{t_3}(x)$ . Sea ahora  $x \in (\text{sup } t_3 \setminus \text{sup } t_1)$ , por la definición de D resulta que  $\mu_D(x) = \mu_{t_3}(x)$ . Luego  $\forall x \in \text{sup } t_3$ ,  $\mu_D(x) = \mu_{t_3}(x)$ , de donde  $D(t_1, t_3) = t_3$  con lo que se demuestra que  $t_1 \xi t_3$  y con ello que  $\xi$  es transitiva. Por lo tanto  $\xi$  es una relación de orden. ■

**PROPOSICIÓN 2.**  $<$  es un relación de orden total.

**DEMOSTRACIÓN:** Demostremos primero que es una relación de orden parcial. Sean  $t_1 = \{\mu_{i_i}(x_{i_i}) | x_{i_i} \| X^{t_i}\}$  y  $t_2 = \{\mu_{j_j}(x_{j_j}) | x_{j_j} \| X^{t_j}\}$ .

A)  $<$  es reflexiva ( $t < \bullet t$ ). Si  $t = \emptyset$ , entonces  $\emptyset < \bullet \emptyset$  por el inciso a) de la definición 1. Si  $t \neq \emptyset$ , sea  $t = \{\mu_{i_i}(x_{i_i}) | x_{i_i} \| X^{t_i}\}$  entonces  $t < \bullet t$  si y sólo si  $X < \bullet X$  y luego de un número finito de pasos se obtiene que  $t < \bullet t$  si y sólo si  $\emptyset < \bullet \emptyset$ , lo cual se tiene.

B)  $\prec$  es antisimétrica:  $((t_1 \prec t_2 \text{ y } t_2 \prec t_1) \Rightarrow t_1 = t_2)$ . Sean  $t_1, t_2 \in \mathcal{M}(\mathcal{X})$  tales que  $t_1 \prec t_2$  y  $t_2 \prec t_1$ . Si  $t_1 = \emptyset$ , por la definición de orden  $t_2$  no puede ser otra cosa que  $t_2 = \emptyset$ , entonces, como sus variables están en orden ascendente, para que ocurra que  $t_1 \prec t_2$  y  $t_2 \prec t_1$ , debe ocurrir que  $i_1 = j_1$ ,  $\mu_{t_1}(x_{i_1}) = \mu_{t_2}(x_{j_1})$  y que  $X^{t_1} \prec X^{t_2}$  y  $X^{t_2} \prec X^{t_1}$ , siguiendo recursivamente el razonamiento, llegaremos a la conclusión de que  $\forall i, j, \mu_1(x_i) | x_i = \mu_2(x_j) | x_j \therefore t_1 \prec t_2$ .

C)  $\prec$  es transitiva.  $((t_1 \prec t_2 \wedge t_2 \prec t_3) \Rightarrow t_1 \prec t_3)$ . Sean  $t_1, t_2, t_3 \in \mathcal{M}(\mathcal{X})$ , tales que  $t_1 \prec t_2$  y  $t_2 \prec t_3$ .

Si  $t_3 = \emptyset$ , por el inciso a) de la definición 1  $t_1 \prec t_3$ .

Si  $t_2 = \emptyset$ , entonces  $t_3 = \emptyset$ ,  $\therefore t_1 \prec t_3$  nuevamente.

Si  $t_1 = \emptyset$  entonces necesariamente ocurre que  $t_1 = \emptyset$ , y luego como se vió en la consideración anterior  $t_3 = \emptyset$ ,  $\therefore t_1 \prec t_3$ .

Sea ahora  $t_1 \neq \emptyset, t_2 \neq \emptyset, t_3 \neq \emptyset$  y  $t_1 = \{\mu_{t_1}(x_{i_1}) | x_{i_1} | X^{t_1}\}$ ,

$t_2 = \{\mu_{t_2}(x_{j_1}) | x_{j_1} | X^{t_2}\}$  y  $t_3 = \{\mu_{t_3}(x_{k_1}) | x_{k_1} | X^{t_3}\}$

♦ Si  $i_1 < j_1$ , entonces como  $j_1 \leq k_1$ , se tiene que  $i_1 < k_1$  y por el inciso b) de la definición de  $\prec$ ,  $t_1 \prec t_3$ .

♦ Si  $i_1 = j_1$ , y  $\mu_{t_1}(x_{i_1}) \prec \mu_{t_2}(x_{j_1})$ , entonces puede ocurrir dos cosas:

i) Si  $j_1 < k_1$  entonces  $i_1 < k_1$  y por lo tanto  $t_1 \prec t_3$ .

ii) Si  $j_1 = k_1$ , entonces  $\mu_{t_2}(x_{j_1}) \leq \mu_{t_3}(x_{k_1})$ , luego  $\mu_{t_1}(x_{i_1}) \leq \mu_{t_3}(x_{k_1})$  y por lo tanto  $t_1 \prec t_3$ .

♦ Si  $i_1 = j_1$  y  $\mu_{t_1}(x_{i_1}) = \mu_{t_2}(x_{j_1})$  entonces se pueden presentar tres casos:

i) Si  $j_1 < k_1 \Rightarrow t_1 \prec t_3$ .

ii) Si  $j_1 = k_1$  y  $\mu_{t_2}(x_{j_1}) \prec \mu_{t_3}(x_{k_1})$  se tiene que  $\mu_{t_1}(x_{i_1}) \prec \mu_{t_3}(x_{k_1})$  por lo tanto  $t_1 \prec t_3$ .

iii) Si  $j_1 = k_1$  y  $\mu_{t_2}(x_{j_1}) = \mu_{t_3}(x_{k_1})$ , entonces se prosigue el análisis de manera recursiva con  $X^{t_1} \prec X^{t_2}$  y  $X^{t_2} \prec X^{t_3}$ .

Hemos demostrado que es una relación de orden, ahora demostraremos que este orden es total, es decir, no importa quien sea  $t_1, t_2 \in \mathcal{M}(\mathcal{X})$ , siempre podremos decir si  $t_1 \prec t_2$  o  $t_2 \prec t_1$ .

Si uno de los conjuntos es el vacío, denotamos por  $t_1$ , entonces  $t_2 \prec t_1$ , por el inciso a) de la definición de  $\prec$ . Si  $t_2 \neq \emptyset$  y  $t_1 \neq \emptyset$ , entonces:

♦ Si  $i_1 < j_1 \Rightarrow t_1 \prec t_2$  y si  $i_1 > j_1 \Rightarrow t_2 \prec t_1$ .

♦ Si  $i_1 = j_1$ , entonces se analizan las siguientes dos situaciones:

i)  $\mu_{t_1}(x_{i_1}) \prec \mu_{t_2}(x_{j_1}) \Rightarrow t_1 \prec t_2$  y si  $\mu_{t_1}(x_{i_1}) \succ \mu_{t_2}(x_{j_1}) \Rightarrow t_2 \prec t_1$

ii)  $\mu_{t_1}(x_{i_1}) = \mu_{t_2}(x_{j_1})$ , entonces se procede recursivamente. ■

Hemos definido un orden parcial  $\xi$  en la familia de los testores difusos de MB ( $\psi(\text{MB})$ ) y un orden total  $\prec$  en  $\mathcal{M}(\mathcal{X})$ . Evidentemente  $\psi(\text{MB}) \subset \mathcal{M}(\mathcal{X})$  de modo que para los elementos de  $\psi(\text{MB})$  podemos considerar ambas relaciones de orden, resulta entonces que :

**PROPOSICIÓN 3.** Sean  $t_1, t_2 \in \psi(\text{MB})$ ; se tiene que  $t_1 \xi t_2 \Rightarrow t_1 \prec t_2$ .

**DEMOSTRACIÓN:** Sean  $t_1 = \{\mu_{t_1}(x_i) | x_i \in X^{t_1}\}$ ,  $t_2 = \{\mu_{t_2}(x_j) | x_j \in X^{t_2}\}$  y

$$D(t_1, t_2) = (t_1 \cap t_2) \cup ((\text{sop } t_1 \setminus \text{sop } t_2) \cap t_1) \cup ((\text{sop } t_2 \setminus \text{sop } t_1) \cap t_2)$$

Si  $t_1 = t_2$  el resultado es inmediato. Consideremos entonces que  $t_1 \neq t_2$ . Procederemos por reducción al absurdo, supongamos entonces que  $t_1 < t_2$ , esto se puede justificar bajo alguna de las siguientes 4 razones:

A)  $t_2 = \emptyset$ , pero resulta que  $D(t_1, t_2) = \emptyset \cup (t_1) \cup \emptyset \neq t_2$  lo que contradice que  $t_1 \xi t_2$ .

B)  $i_1 < j_1$ ; de donde  $x_{i_1} \in \text{sop } t_1$  y  $x_{i_1} \notin \text{sop } t_2$ . Entonces  $\mu_{t_1 \cap t_2}(x_{i_1}) = 0$ ,  $x_{i_1} \in (\text{sop } t_1 \setminus \text{sop } t_2)$  y  $x_{i_1} \notin (\text{sop } t_2 \setminus \text{sop } t_1)$ , se tiene que  $\mu_D(x_{i_1}) = \mu_{t_1}(x_{i_1})$  y entonces  $D(t_1, t_2) \neq t_2$  lo que contradice la hipótesis inicial.

C)  $i_1 = j_1$  y  $\mu_{t_1}(x_{i_1}) < \mu_{t_2}(x_{j_1})$ , esto hace que  $\mu_{t_1 \cap t_2}(x_{i_1}) = \mu_{t_1}$ , por la definición (de intersección en difusos), entonces  $D(t_1, t_2) \neq t_2$ , lo que contradice la suposición inicial.

D)  $i_1 = j_1$ ,  $\mu_{t_1}(x_{i_1}) = \mu_{t_2}(x_{j_1})$ , entonces debe ocurrir que  $X^{t_1} < X^{t_2}$ , lo que nos conduce a uno de los incisos anteriores. ■

## 4.2 LA FUNCION DE ERROR

Establecido un orden para recorrer el conjunto de combinaciones de rasgos con grados de pertenencia, continuaremos ahora con la definición de error, que será un concepto en el que se basarán más adelante las funciones de recorrido que nos permitirán “saltar” grupos de subconjuntos en la búsqueda de los testores típicos difusos. Un subconjunto de rasgos “comete un error” cada vez que es incapaz de diferenciar un par de objetos de clases distintas, dicho de otro modo, los grados de pertenencia del subconjunto son todos mayores a sus correspondientes en MB. Esto se formaliza en la siguiente

**DEFINICIÓN 6.** Sea  $X \in \mathcal{M}(\mathcal{Q})$  con  $X = \{\mu_X(x_i) | x_i, \dots, \mu_X(x_s) | x_s\}$ . Denotemos por  $EDIF(X)$  al conjunto:

$$EDIF(X) = \left\{ A_k \in MB \mid \mu_X(x_{j_1}) > \mu_{A_k}(x_{j_1}) \right\} \quad \forall j \quad 1 \leq j \leq s.$$

Se dirá que el error de X es igual al cardinal del conjunto  $EDIF(X)$ .

Entonces  $EDIF(X)$  es precisamente el conjunto de filas de MB para las cuales no se cumple la condición de la definición de testor difuso típico, y el *error de X* la cantidad de estas filas. Resulta entonces que:

**PROPOSICIÓN 4.**  $X$  es un testor si y sólo si el error de  $X$  es igual a cero.

**DEMOSTRACIÓN:** Es inmediata a partir de la definición de testor difuso. ■

**DEFINICIÓN 4.** La función de error acumulado será definida de la siguiente manera:

$$X = \{\mu_{i_1} | x_{i_1}, \dots, \mu_{i_s} | x_{i_s}\}, \quad T = \{\mu_{i_1} | x_{i_1}, \dots, \mu_{i_k} | x_{i_k}\}, \quad 1 \leq k \leq s,$$

$$\begin{aligned} edif_X: \text{sop } X &\longrightarrow \mathbf{N} \\ x_{i_k} &\longrightarrow \|\text{EDIF}(T^k)\|. \end{aligned}$$

Esto es, el error del subconjunto  $X$  hasta su  $k$ -ésimo elemento que es igual al cardinal de  $EDIF(T^k)$ , que es el número de renglones en que no se cumple la condición de la definición de testor difuso típico en la

submatriz definida por las primeras  $k$ -ésimas columnas de MB correspondientes al *sop*  $X$ , y que a partir de ahora denominaremos “error”.

**PROPOSICIÓN 5.**  $edif_X$  es una función no creciente

**DEMOSTRACIÓN:** Sea  $X = \{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_k}|x_{i_k}\}$ , a partir de este conjunto difuso se define  $T^k = \{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_k}|x_{i_k}\}$ , como un subconjunto de  $X$ , formado por sus primeros  $k$ -ésimos rasgos. Si  $k < k'$  entonces  $EDIF(T^k) \subseteq EDIF(T^{k'})$ . ■

**PROPOSICIÓN 6.** Si  $T$  es un testor difuso típico entonces  $edif_T$  es inyectiva.

**DEMOSTRACIÓN:** Sea  $T^k = \{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_k}|x_{i_k}\}$ , un testor difuso típico. Supongamos que de  $\mu_T(x_{i_k})|x_{i_k}$ , hasta  $\mu_T(x_{i_j})|x_{i_j}$  la función decrece (proposición 5) y  $\mu_T(x_{i_{j+1}})|x_{i_{j+1}}$  no hace que el error disminuya con respecto al rasgo anterior. Según la caracterización de los testores difusos existe una fila en MB, sea esta fila  $A_r$ , tal que  $\mu_{A_r}(x_{i_{j+1}}) = \mu_T(x_{i_{j+1}})$  y  $\forall k < j$ , ocurre  $\mu_{A_r}(x_{i_k}) = \mu_T(x_{i_k})$ , luego  $A_r \notin EDIF(T^{j+1})$ ; pero  $A_r \in EDIF(T^j)$ . Como  $EDIF(T^{j+1}) \subseteq EDIF(T^j)$  se tiene que  $EDIF(T^{j+1}) \subset EDIF(T^j)$  y  $edif_T(x_{i_{j+1}}) < edif_T(x_{i_j})$  por lo que podemos afirmar que si  $T$  es un testor típico difuso entonces  $edif_T$  es estrictamente decreciente y por lo tanto inyectiva. ■

### 4.3 LA FUNCION DE RECORRIDO

Por la manera en que se han ordenado los conjuntos, podemos a partir de que encontramos un conjunto que es testor difuso, que no es típico, eliminar de él aquellos rasgos donde la función de error no es inyectiva ya que esos rasgos no se necesitan para que el conjunto distinga a los elementos de MB. Cuando encontramos un conjunto que es testor difuso típico, podemos saltar al grupo de conjuntos que le siguen y que son subconjuntos de él, ya que por la definición de testor típico difuso ninguno de sus subconjuntos es testor difuso típico. Por último si tenemos un conjunto que no es testor difuso, ninguno de los subconjuntos que le siguen será testor difuso, por lo tanto igual que en el punto anterior podemos saltarlos. Para llevar a cabo estas tres acciones se define la siguiente función de recorrido entre los conjuntos difusos:

**DEFINICIÓN 7:** Sea  $\rho dif: \mathcal{M}(\mathcal{R}) \rightarrow \mathcal{M}(\mathcal{R})$  y  $X = \{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_k}|x_{i_k}\}$  Entonces:

A) Si  $edif_X$  no es inyectiva y  $X$  es testor entonces  $\rho dif(X) = \sigma dif(X)$ .

B) Si  $edif_X$  es inyectiva y

i)  $X$  es testor pero no típico  $\rho dif(X) = \beta dif(X)$

ii)  $X$  es testor típico  $\rho dif(X) = \alpha dif(X)$ .

C) Si  $X$  no es testor, entonces  $\rho dif(X) = \alpha dif(X)$ .

donde:

**PRIMER CASO:**

1)  $\sigma dif: \mathcal{M}(\mathcal{R}) \rightarrow \mathcal{M}(\mathcal{R})$ , entonces

$$\sigma dif(X) = \{sop X \setminus \{x_{i_j} | edif_X(x_{i_j}) = edif_X(x_{i_{j-1}})\}\} \cap X,$$

$$\sigma dif(\emptyset) = \emptyset.$$

dicho de otro modo, la función  $\alpha dif$  elimina del conjunto  $X$ , todas las variables para las cuales el error que se produce no varía con respecto a la variable que la antecede.

SEGUNDO CASO:

2)  $\beta dif(X): \mathcal{M}(\mathcal{X}) \rightarrow \mathcal{M}(\mathcal{X})$ . Sea :

$$\beta dif(X) = Y \cup Z \cup H$$

definiendo como sigue:

$$Y = \begin{cases} \left\{ \mu_{i_1} | x_{i_1}, \dots, \mu_{i_{s-1}} | x_{i_{s-1}} \right\} & \text{si } s \geq 2 \\ \emptyset & \text{si } s = 1 \end{cases}$$

$$Z = \begin{cases} \emptyset & \text{si } \forall A_k \mu_{A_k}(x_{i_s}) \leq \mu_X(x_{i_s}) \\ \left\{ \mu^*_{i_s} | x_{i_s} \right\}, e.o.c. \text{ Siendo } \mu^*_{i_s} = \min_{A_k \in MB} \left\{ \mu_{A_k}(x_{i_s}) \right\} \mu_X(x_{i_s}) \end{cases}$$

o sea,  $Y$  introduce en el nuevo conjunto las mismas variables que tenía el conjunto original hasta la penúltima, siempre y cuando el conjunto tenga una longitud mayor o igual a dos<sup>3</sup>, por otra parte, el conjunto  $Z$  toma el último elemento del conjunto  $X$  e incrementa su grado de pertenencia asociado al grado de pertenencia siguiente, en dependencia de los valores que este toma en la MB.

$$H = \left\{ \mu_r | x_r \mid \mu_r = \min_{A_k \in MB} \left\{ \mu_{A_k}(x_r) \right\} \text{ con } s(r \leq n) \right\}$$

es decir,  $H$  es el conjunto difuso formado por todos los rasgos a la derecha del  $s$ -ésimo, con el mínimo grado de pertenencia con respecto a cada variable.

Finalmente:

$$\beta dif(\emptyset) = \emptyset.$$

TERCER CASO:

3)  $\alpha dif: \mathcal{M}(\mathcal{X}) \rightarrow \mathcal{M}(\mathcal{X})$ . Entonces

$$\alpha dif(X) = Y \cup Z \cup H$$

donde  $Y, Z, H$  se definen según se presenten los siguientes casos:

- ♦ Si  $i_s = n$  y  $s=1$ , entonces  $\alpha dif(X) = \emptyset$
- ♦ Si  $i_s = n$ :

$$Y = \begin{cases} \left\{ \mu_{i_1} | x_{i_1}, \dots, \mu_{i_{s-2}} | x_{i_{s-2}} \right\} & \text{si } s > 2 \\ \emptyset & \text{si } s \leq 2 \end{cases}$$

o sea que el conjunto  $Y$  toma los mismos elementos que contenía el conjunto  $X$ , hasta el antepenúltimo elemento.

$$Z = \begin{cases} \emptyset & \text{si } \forall A_k \mu_{A_k}(x_{i_{s-1}}) \leq \mu_X(x_{i_{s-1}}) \\ \left\{ \mu^*_{i_{s-1}} | x_{i_{s-1}} \right\}, e.o.c. \text{ Siendo } \mu^*_{i_{s-1}} = \min_{A_k \in MB} \left\{ \mu_{A_k}(x_{i_{s-1}}) \right\} \mu_X(x_{i_{s-1}}) \end{cases}$$

es decir, toma el penúltimo elemento e incrementa su grado de pertenencia al siguiente grado que exista en la MB asociado a ese rasgo.

Finalmente  $H$  es el conjunto de los rasgos a la derecha del elemento  $s-1$ -ésimo, con sus grados de pertenencia más pequeños:

<sup>3</sup> Tomaremos como longitud del conjunto a la cardinalidad del mismo.



$$H = \left\{ \mu_r | x_r \parallel \mu_r = \min_{A_k \in MB} \{ \mu_{A_k}(x_r) \} \text{ con } (s-1) < r \leq n \right\}$$

♦ Si  $i_s < n$ :

$$Y = \begin{cases} \{ \mu_{i_1} | x_{i_1}, \dots, \mu_{i_{s-1}} | x_{i_{s-1}} \} & \text{si } s \geq 2 \\ \emptyset & \text{si } s = 1 \end{cases}$$

$$Z = \begin{cases} \emptyset & \text{si } \forall A_k \mu_{A_k}(x_{i_s}) \leq \mu_X(x_{i_s}) \\ \{ \mu^*_{i_s} | x_{i_s} \}, \text{ e. o. c. Siendo } \mu^*_{i_s} = \min_{A_k \in MB} \{ \mu_{A_k}(x_{i_s}) \} \mu_X(x_{i_s}) \end{cases}$$

$$H = \left\{ \mu_r | x_r \parallel \mu_r = \min_{A_k \in MB} \{ \mu_{A_k}(x_r) \} \text{ con } s < r \leq n \right\}$$

Entonces  $Y$  se encarga de la primera parte del conjunto que no cambia con respecto a  $X$ .  $Z$  es la parte que cambia (avanza al siguiente valor de grado de pertenencia asociado a esa variable).  $H$  se encarga de los que siguen al rasgo que modificó  $Z$ , y que aparecerán con su grado de pertenencia más pequeño asociado. Finalmente

$$\text{adif}(\emptyset) = \emptyset.$$

Nos interesa ahora, que en base a la definición de orden establecida en nuestro conjunto potencia y a la definición de recorrido anterior, se pueda asegurar que siempre “avanzaremos” sobre los conjuntos, es decir, que siempre a partir de un conjunto la aplicación de la función de recorrido nos llevará a un conjunto “posterior”, entendido en términos del orden establecido. Esto es lo que se establece en la proposición siguiente:

**PROPOSICIÓN 7.** Para todo  $X \in \mathcal{M}(\mathcal{R})$  se cumple que:

$$X \prec \bullet \text{adif}(X),$$

$$X \prec \bullet \beta\text{dif}(X),$$

y 
$$X \prec \bullet \text{adif}(X).$$

**DEMOSTRACIÓN:** Si  $X = \emptyset$ , la demostración es obvia, ya que por definición 1 de orden  $\text{adif}(X) \prec \bullet \beta\text{dif}(X) \prec \bullet \text{adif}(X) = \emptyset$ .

Si  $X \neq \emptyset$ :

a) De la Definición 2:

$$\begin{aligned} D(\text{adif}(X), X) &= (\text{adif}(X) \cap X) \cup ((\text{sop } \text{adif}(X) \setminus \text{sop } X) \cap \text{adif}(X)) \cup ((\text{sop } X \setminus \text{sop } \text{adif}(X)) \cap X) \\ &= \text{adif}(X) \cup \emptyset \cup ((\text{sop } X \setminus \text{sop } \text{adif}(X)) \cap \text{sop } X) = X \end{aligned}$$

∴  $\text{adif}(X) \xi X$  y por la proposición 3,  $X \prec \text{adif}(X)$ .

b) Sea  $X = \{ \mu_{i_1} | x_{i_1}, \dots, \mu_{i_s} | x_{i_s} \}$ ,  $Y$ ,  $Z$  y  $H$  como en el inciso 2 de la definición 5.  $\beta\text{dif}(X) = Y \cup Z \cup H$  hasta el rasgo  $x_{i_s}$  ( $Y, X$  es igual al conjunto  $\beta\text{dif}(X)$ ). Pero si  $Z = \{ \mu^*_{i_s} | x_{i_s} \}$  donde  $\mu^*_{i_s}$  por construcción es mayor que  $\mu_{i_s}$  y por el inciso c) de la definición 1,  $X \prec \beta\text{dif}(X)$ . Si  $Z = \emptyset$ , entonces por el inciso c) de la definición 1,  $X \prec \beta\text{dif}(X)$ .

- c) Si  $X = \{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_r}|x_{i_r}\}$ ,  $Y$ ,  $Z$  y  $H$  como en el inciso 3 de la definición 5.  $\text{adif}(X) = Y \cup Z \cup H$ .  
 Como en el caso anterior  $X$  es igual a  $\text{adif}(X)$ , en los elementos de  $Y$ .  
 Si  $Z = \emptyset$ , entonces por el inciso b) de la definición 1,  $X \prec \text{adif}(X)$ .  
 si  $i_r = n$ ,  $Z = \{\mu_{i_{r-1}}^*|x_{i_{r-1}}\}$ , entonces por construcción y por el inciso c) de la definición 1  $X \prec \text{adif}(X)$ .  
 si  $i_r < n$ ,  $Z = \{\mu_{i_r}^*|x_{i_r}\}$ , y por la misma razón que en el párrafo anterior  $X \prec \text{adif}(X)$ .

Establecido lo anterior veamos otras características de esta función de recorrido, por ejemplo, en el caso particular de la función  $\sigma\text{dif}(X)$ , podemos decir que el grupo de elementos de clases distintas que no se logran distinguir con el conjunto de rasgos  $X$  es el mismo grupo que el que produce sin poder distinguir  $\sigma\text{dif}(X)$ , como se formaliza en la proposición siguiente:

**PROPOSICIÓN 8.** Sea  $X \in \mathcal{M}(\mathcal{R})$  y  $\text{EDIF}(X)$  como en la definición 3. Entonces:

- i)  $\text{EDIF}(\sigma\text{dif}(X)) = \text{EDIF}(X)$  y
- ii)  $\text{edif}_{\sigma\text{dif}(X)}(x) = \text{edif}_X(x)$  para cualquier  $x \in \text{sup } \sigma\text{dif}(X)$ .

**DEMOSTRACIÓN:**  $\text{edif}_X$  puede ó no ser inyectiva sobre  $X$ . Si es inyectiva, entonces  $\text{edif}_X(x_{i_j}) > \text{edif}_X(x_{i_{j+1}})$  para toda  $i$ , por lo tanto  $\sigma\text{dif}(X)$  no eliminará ningún rasgo, dando como resultado que  $\sigma\text{dif}(X) = X$ , entonces es inmediato que  $\text{EDIF}(\sigma\text{dif}(X)) = \text{EDIF}(X)$ .

Si no es inyectiva:  $\sigma\text{dif}(X)$  elimina aquellos rasgos que  $\text{edif}_X(x_{i_j}) = \text{edif}_X(x_{i_{j+1}})$ ; pero  $\mu_{i_{j+1}}|x_{i_{j+1}}$  no lograba distinguir más pares de objetos de los que distinguía  $X$  hasta el rasgo  $\mu_{i_j}|x_{i_j}$  por lo tanto  $\text{EDIF}(\sigma\text{dif}(X))$  es el mismo conjunto que  $\text{EDIF}(X)$ . Para el inciso ii) se tiene  $\|\text{EDIF}(\{\mu_{\sigma\text{dif}(X)}|x_{i_1}, \dots, \mu_{\sigma\text{dif}(X)}|x_{i_k}\})\| = \|\text{EDIF}(\{\mu_{i_1}|x_{i_1}, \dots, \mu_{i_k}|x_{i_k}\})\|$  entonces

$$\text{edif}_X(x_{i_k}) = \text{edif}_{\sigma\text{dif}(X)}(x_{i_k}),$$

para todo  $x_{i_k} \in \text{sup } \sigma\text{dif}(X)$ . ■

Otra propiedad importante, de esta función, es que si se aplica sobre un conjunto de rasgos que es testor, el nuevo conjunto obtenido también es testor difuso:

**PROPOSICIÓN 9.** Si  $X \in \mathcal{M}(\mathcal{R})$  es testor, entonces  $\sigma\text{dif}(X)$  también es testor difuso.

**DEMOSTRACIÓN:** Es inmediata a partir de la proposición 5 y la definición 8.

Existe una propiedad que nos será de utilidad en lo que sigue, y que nos permite asegurar que para cada par de conjuntos que coinciden en sus primeros  $k$  elementos, todos los conjuntos intermedios entre ellos cuentan también con esos mismos  $k$  elementos. Para esto se establece la siguiente

**PROPOSICIÓN 10.** Sea  $X_1 = \{\mu_1|x_{i_1}, \dots, \mu_1|x_{i_r}\}$  y  $X_2 = \{\mu_2|x_{i_1}, \dots, \mu_2|x_{i_k}\}$ , si  $X_1 \prec X_2$ , y ambos conjuntos coinciden en sus primeros  $r$  elementos con los mismos grados de pertenencia, entonces cualquier  $W \in \mathcal{M}(\mathcal{R})$ , tal que  $X_1 \prec W \prec X_2$ , tiene esos mismos elementos con los mismos grados.

**DEMOSTRACIÓN:** Hagamos la demostración por contradicción, es decir, supongamos que existe un elemento  $\mu_w | x_{i_k}$ , tal que  $1 \leq k \leq r$ , dicho de otro modo, es un rasgo que existe dentro de los primeros  $r$  elementos de  $W$ , pero  $\mu_w | x_{i_k} \neq \mu_{X_1} | x_{i_k}$ , o  $\mu_w | x_{i_k} \neq \mu_{X_2} | x_{i_k}$ . Como los primeros  $r$  elementos de  $X_1$  y  $X_2$  son los mismos, bastará decir que  $\mu_w | x_{i_k} \neq \mu_{X_1} | x_{i_j}$  con  $k$  como el lugar correspondiente a  $j$  en  $X_1$  y  $X_2$ . Esta situación sólo se puede dar por alguna de las siguientes causas:

♦ Si  $i_k < i_j$ , entonces por el inciso b) de la definición 1,  $X_2 < W$  lo que contradice la premisa inicial de orden.

♦ Si  $i_k > i_j$ , entonces por la misma razón que en el inciso anterior,  $W < X_1$ , lo que nuevamente contradice la suposición inicial de orden.

♦ Si  $i_k = i_j$ , entonces puede ocurrir que:

i)  $\mu_w | x_{i_k} < \mu_{X_1} | x_{i_j}$ , en este caso por el inciso c) de la definición 1,  $W < X_1$ , que contradice el orden supuesto.

ii)  $\mu_w | x_{i_k} > \mu_{X_1} | x_{i_j}$ , entonces  $X_2 < W$  que también es una contradicción con el orden inicial.

Por tanto, no puede existir  $\mu_w | x_{i_k}$  en  $W$ , dentro de los  $r$  primeros rasgos sin que sea igual a sus correspondientes rasgos en  $X_1$  y  $X_2$  ■

Establecido lo anterior, toca ahora demostrar que en base al orden establecido sobre el conjunto potencia  $\mathcal{M}(\mathcal{E})$  y la función de recorrido  $\rho dif_X$  no se dejará de encontrar ningún testor difuso típico. No existirá la posibilidad de que una aplicación de la función de recorrido “brinque”<sup>4</sup> a algún testor difuso típico.

**PROPOSICIÓN 11.** No existe testor difuso típico  $T \in \mathcal{M}(\mathcal{E})$  tal que  $\rho dif^*(X) < T < \rho dif^{**}(X)$  siendo  $X = \{\mu_1^* | x_1, \dots, \mu_n^* | x_n\}$ , y  $\mu_n = \min_{A_k \in MB} \{\mu_{A_k}(x_n)\}$ .

**DEMOSTRACIÓN:** Denotemos por  $X_1 = \rho dif^*(X) = \{\mu_{i_1} | x_{i_1}, \dots, \mu_{i_r} | x_{i_r}\}$ , y supongamos que no se cumple la proposición, es decir, que existe un testor difuso típico  $T$  que no está incluido en el recorrido definido por  $\rho dif$ . Esto sólo puede pasar bajo alguna de las siguientes 3 situaciones:

A)  $edif_{X_{11}}$  no es inyectiva y  $X_1$  es testor difuso: Bajo estas condiciones  $\rho dif^*(X_1) = \sigma dif(X_1)$ .  $X_1 < T < \sigma dif(X_1)$ . Hay elementos  $x_{i_j} \in \text{sop } X_1$  y  $x_{i_j} \notin \sigma dif(X_1)$ .  $x_{i_j}$  puede o no pertenecer al  $\text{sop } T$ :

♦ Si  $x_{i_j} \in T$ , puede o no tener el mismo grado de pertenencia que en  $X_1$ .

\*Si  $\mu_{X_1}(x_{i_j}) = \mu_T(x_{i_j})$ , entonces no disminuye el error obtenido hasta el rasgo  $x_{i_{j-1}}$  por  $edif_{X_1}$  y por lo tanto puede ser eliminada del conjunto difuso, sin que cambie el error total. Por lo tanto  $T$  no es testor difuso típico, ya que se puede eliminar uno de sus rasgos sin que deje de ser testor difuso.

\*Si  $\mu_{X_1}(x_{i_j}) \neq \mu_T(x_{i_j})$ , puede ocurrir que  $\mu_T(x_{i_j})$  sea mayor o menor que su correspondiente  $\mu_{X_1}(x_{i_j})$ :

• Si  $\mu_{X_1}(x_{i_j}) < \mu_T(x_{i_j})$ , entonces podemos asegurar que  $\mu_{X_1}(x_{i_j})$  no

<sup>4</sup> tómesese “brincar”, como sinónimo de no ser considerado por la función de recorrido.

lograba disminuir el error encontrado por  $edif_{X_1}$ , cualquier valor de pertenencia mayor al de él, tampoco lo hará, por lo tanto también se puede eliminar de  $T$  sin que  $T$  deje de ser testor difuso, lo que contradice que  $T$  sea testor difuso típico.

- Si  $\mu_{X_1}(x_{ij}) > \mu_T(x_{ij})$ ,  $T$  no puede ser testor difuso típico ya que su variable

$x_{ij}$

tiene un grado de pertenencia que puede aumentar sin dejar de ser testor difuso.

- ♦ Si  $x_{ij} \notin T$ . Por construcción, toda  $T$  que se encuentra entre  $X_1$  y  $adif(X_1)$  tiene o los mismos rasgos en el soporte o más.
  - Si son más rasgos,  $T$  no puede ser testor difuso típico porque existe  $adif(X_1)$  que es testor con menos rasgos.
  - Si tiene exactamente el mismo soporte, debemos analizar si los grados son mayores o menores
    - Si los grados de pertenencia son mayores no pueden distinguir más de los que distinguía  $X$  por lo tanto no puede ser considerado testor difuso típico difuso.
    - Si los grados de pertenencia son menores, entonces existe  $adif(X_1)$  que es testor difuso con grado de pertenencia menor por lo tanto  $T$  no puede ser testor típico difuso.

B)  $edif_{X_1}$  es inyectiva y  $X_1$  testor difuso pero no típico. Se aplica  $\beta dif(X_1)$  y por lo tanto la función avanza al conjunto inmediato. No hay nadie en medio, por lo tanto  $T$  no puede existir entre ambos.

C)  $edif_{X_1}$  es inyectiva y  $s$  testor difuso típico. Se aplica  $adif(X_1)$ . Como tenemos una combinación que es testor difuso típico, cualquiera que le siga con menos variables o mayor grado de pertenencia, puede que sea testor difuso, pero seguro no es típico. Esto se observa bajo dos condiciones distintas:

- ♦  $i_1 < n$ ,  $adif(X_1)$  lleva al conjunto siguiente, por lo tanto no hay manera de que exista  $X_1 \prec T \prec adif(X_1)$ .

- ♦ Si  $i_1 = n$ , y suponemos que  $X_1 \prec T \prec adif(X_1)$ , entonces  $\mu_{X_1}(x_{ij}) < \mu_T(x_{ij})$  por construcción, pero como  $X_1$  es testor difuso típico, entonces ningún conjunto con el mismo soporte y mayor grado de pertenencia en alguno de sus rasgos puede ser testor difuso y lo mismo ocurre si  $Z = \emptyset$ , ya que  $T$  tendrá menos rasgos que  $X_1$ , como lo dicta la definición de testor difuso típico.

D)  $X_1$  no es testor. Se aplica  $adif(X_1)$ . Como no es testor difuso la función lo que hace es saltar a los siguientes conjuntos que tengan menos variables o mayor grado de pertenencia, porque tampoco serán testores difusos. Entonces provemos que salta efectivamente a los que tiene menos variables o mayor grado de pertenencia en las variables correspondientes al conjunto  $X_1$ . Como en el inciso anterior se pueden dar dos casos al aplicar  $adif(X_1)$ :

- ♦  $i_1 < n$ ,  $adif(X_1)$  lleva al conjunto siguiente, por lo tanto no existe  $T$  tal que  $X_1 \prec T \prec adif(X_1)$ .
- ♦  $i_1 = n$ , y suponemos que  $X_1 \prec T \prec adif(X_1)$ . Pero como  $X_1$  no es testor difuso, es decir  $T$ , al aumentar el grado de uno de sus rasgos no obtendremos un testor difuso, al aumentar el grado de uno de sus rasgos no obtendremos un testor difuso, es decir  $T$  no podrá ser testor difuso. Por otro lado, si  $Z = \emptyset$ , entonces si  $X_1$  no era testor difuso, le estamos quitando un rasgo lo que imposibilita aun más que el nuevo conjunto sea siquiera testor difuso, por lo tanto  $T$  no puede ser testor difuso típico ■.

#### 4.5 ALGORITMO RECDIF PARA EL CALCULO DE LOS TESTORES TÍPICOS.

Se describe ahora el algoritmo que resume los resultados del presente trabajo, al localizar testores típicos difusos partiendo de una matriz básica (MB), generando para analizar sólo aquellas combinaciones con posibilidades de ser testor difuso, de modo tal que la búsqueda no sea exhaustiva y no se requiera tener almacenado de antemano toda la lista de combinaciones posibles de subconjuntos del conjunto difuso.

**ENTRADA:** existen dos posibles entradas a) Una matriz de aprendizaje o b) una matriz de diferencias. En el primer caso se debe introducir también los criterios de comparación de cada rasgo.

**SALIDA:** El conjunto de testores típicos difusos de la MB correspondiente.

**PASO 1:** Se considera  $X$  el conjunto difuso de todos los rasgos del conjunto  $R$  con sus mínimos grados de pertenencia.  $X = \{\mu_1(x_1), \dots, \mu_n(x_n)\}$ .

**PASO 2.** Mientras  $X \neq \emptyset$  repetir los pasos 3 y 4.

**PASO 3.** Se calcula el error  $edif_x$  del conjunto difuso  $X$ .

**PASO 4.** Si  $edif_x = 0$ , entonces

Si  $edif_x$  es inyectiva entonces

Si  $X$  es testor típico entonces

Se anota  $X$  en la lista de testores típicos

Se aplica  $\alpha dif(X)$ .

Si no:

Se aplica  $\beta dif(X)$ .

Si no:

Se aplica  $\sigma dif(X)$

Si no:

Se aplica  $\alpha dif(X)$ .

**PASO 5.** Salida: Imprimir la lista de testores.

**PASO 6.** Terminar.

##### 4.5.1 Ejemplo

Analicemos ahora un ejemplo muy sencillo de la manera en la cual el algoritmo anterior encuentra los testores difusos típicos sobre una matriz básica con tan sólo 4 rasgos y 3 renglones. Consideramos la matriz :

$$\text{MB} = \begin{matrix} & \begin{matrix} x1 & x2 & x3 & x4 \end{matrix} \\ \begin{bmatrix} 0.58 & 0.52 & 0.09 & 0.40 \\ 0.59 & 0.42 & 0.98 & 0.21 \\ 0.90 & 0.38 & 0 & 0.13 \end{bmatrix} \end{matrix}$$

La lista de subconjuntos posibles a analizar a partir de esta matriz, ordenados en base a la definición 3 de orden es la siguiente:

combinación	$e_X$				testor?	inyectivo?	testor típico?	función
	$x_1$	$x_2$	$x_3$	$x_4$				
que se aplica								
1.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	0	0	0	0	si	no	--	$\sigma dif(X)=49$
2.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
3.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
4.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> }								
5.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
6.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
7.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
8.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> }								
9.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.13x <sub>4</sub> }								
10.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.21x <sub>4</sub> }								
11.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> , 0.40x <sub>4</sub> }								
12.-{0.58x <sub>1</sub> , 0.38x <sub>2</sub> }								
13.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }								
14.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
15.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
16.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> }								
17.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
18.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
19.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
20.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> }								
21.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }								
22.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.21x <sub>4</sub> }								
23.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.40x <sub>4</sub> }								
24.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }								
25.-{0.58x <sub>1</sub> , 0.42x <sub>2</sub> }								
26.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }								
27.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
28.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
29.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> }								
30.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
31.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
32.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
33.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0}								
34.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.13x <sub>4</sub> }								
35.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.21x <sub>4</sub> }								
36.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> , 0.40x <sub>4</sub> }								
37.-{0.58x <sub>1</sub> , 0.52x <sub>2</sub> }								
38.-{0.58x <sub>1</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }								
39.-{0.58x <sub>1</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
40.-{0.58x <sub>1</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
41.-{0.58x <sub>1</sub> , 0.09x <sub>3</sub> }								
42.-{0.58x <sub>1</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
43.-{0.58x <sub>1</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
44.-{0.58x <sub>1</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
45.-{0.58x <sub>1</sub> , 0.98x <sub>3</sub> }								
46.-{0.58x <sub>1</sub> , 0.13x <sub>4</sub> }								
47.-{0.58x <sub>1</sub> , 0.21x <sub>4</sub> }								
48.-{0.58x <sub>1</sub> , 0.40x <sub>4</sub> }								
49.-{0.58x <sub>1</sub> }	0	--	--	--	si	si	si	$\sigma dif(X)=50$
50.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	1	0	0	0	si	no	--	$\sigma dif(X)=61$

$$51.-\{0.59x_1, 0.38x_2, 0.09x_3, 0.21x_4\}$$

combinación	$\epsilon_x$				testor?	inyectivo?	testor típico?	función
	$x_1$	$x_2$	$x_3$	$x_4$				
52.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
53.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> }								
54.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
55.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
56.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
57.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> }								
58.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.13x <sub>4</sub> }								
59.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.21x <sub>4</sub> }								
60.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> , 0.40x <sub>4</sub> }								
61.-{0.59x <sub>1</sub> , 0.38x <sub>2</sub> }	1	0	--	--	si		si	no
$\beta dif(X)=62$								
62.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	1	0	0	0	si		no	--
$\alpha dif(X)=74$								
63.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
64.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
65.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> }								
66.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
67.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
68.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
69.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> }								
70.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }								
71.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.21x <sub>4</sub> }								
72.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.40x <sub>4</sub> }								
73.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }								
74.-{0.59x <sub>1</sub> , 0.42x <sub>2</sub> }	1	0	--	--	si		si	no
$\beta dif(X)=75$								
75.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	1	0	0	0	si	no	--	$\alpha dif(X)=86$
76.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
77.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
78.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> }								
79.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }								
80.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
81.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
82.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0}								
83.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.13x <sub>4</sub> }								
84.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.21x <sub>4</sub> }								
85.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> , 0.40x <sub>4</sub> }								
86.-{0.59x <sub>1</sub> , 0.52x <sub>2</sub> }	1	0	--	--	si		si	si
$\alpha dif(X)=87$								
87.-{0.59x <sub>1</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	1	--	0	0	si	no	--	$\alpha dif(X)=90$
88.-{0.59x <sub>1</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }								
89.-{0.59x <sub>1</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }								
90.-{0.59x <sub>1</sub> , 0.09x <sub>3</sub> }	1	--	0	--	si	si	no	$\beta dif(X)=91$
91.-{0.59x <sub>1</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }	1	--	1	0	si	no	--	$\alpha dif(X)=95$
92.-{0.59x <sub>1</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }								
93.-{0.59x <sub>1</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }								
94.-{0.59x <sub>1</sub> , 0.98x <sub>3</sub> }								
95.-{0.59x <sub>1</sub> , 0.13x <sub>4</sub> }	1	--	--	0	si	si		no
$\beta dif(X)=96$								
96.-{0.59x <sub>1</sub> , 0.21x <sub>4</sub> }	1	--	--	0	si	si		no
$\alpha dif(X)=97$								

97.-{0.59x <sub>1</sub> , 0.40x <sub>4</sub> }	1	--	--	0		si	si		si
$\beta dif(X)=99$									
98.-{0.59x <sub>1</sub> }									
99.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	2	0	0	0	0	si	no	--	$\alpha dif(X)=110$
100.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
101.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
102.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.09x <sub>3</sub> }									
103.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }									
104.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }									
105.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
106.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.98x <sub>3</sub> }									
107.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.13x <sub>4</sub> }									
108.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.21x <sub>4</sub> }									
109.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , 0.40x <sub>4</sub> }									
110.-{0.90x <sub>1</sub> , 0.38x <sub>2</sub> , }	2	0	--	--		si		si	no
$\beta dif(X)=111$									
111.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	2	0	0	0	0	si	no	--	$\alpha dif(X)=123$
112.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
113.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
114.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.09x <sub>3</sub> }									
115.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }									
116.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }									
117.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
118.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.98x <sub>3</sub> }									
119.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }									
120.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.21x <sub>4</sub> }									
121.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.40x <sub>4</sub> }									
122.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }									
123.-{0.90x <sub>1</sub> , 0.42x <sub>2</sub> }	2	0	--	--		si	si		si
$\alpha dif(X)=124$									
124.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	2	1	0	0	0	si	no	--	$\alpha dif(X)=127$
125.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
126.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
127.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.09x <sub>3</sub> }	2	1	0	--		si	si		no
128.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }	2	1	0	0	0	si	no		--
$\alpha dif(X)=131$									
129.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }									
130.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
131.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.98x <sub>3</sub> }	2	1	0	--		si	si	si	$\alpha dif(X)=132$
132.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.13x <sub>4</sub> }	2	1	--	0		si	si		no
$\beta dif(X)=133$									
133.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.21x <sub>4</sub> }	2	1	--	0		si	si		no
$\beta dif(X)=134$									
134.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , 0.40x <sub>4</sub> }	2	1	--	1		no	--		--
$\alpha dif(X)=136$									
135.-{0.90x <sub>1</sub> , 0.52x <sub>2</sub> , }									
136.-{0.90x <sub>1</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	2	--	0	0		si	no		--
$\alpha dif(X)=139$									
137.-{0.90x <sub>1</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
138.-{0.90x <sub>1</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
139.-{0.90x <sub>1</sub> , 0.09x <sub>3</sub> }	2	--	0	--		si	si		si
$\alpha dif(X)=140$									
140.-{0.90x <sub>1</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }	2	--	1	0		si	si		no
$\beta dif(X)=141$									



141.-{0.90x <sub>1</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }	2	--	1	0		si	si		no
$\beta dif(X) = 142$									
142.-{0.90x <sub>1</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }	2	--	1	0		si	si		si
$\alpha dif(X) = 144$									
143.-{0.90x <sub>1</sub> , 0.98x <sub>3</sub> }									
144.-{0.90x <sub>1</sub> , 0.13x <sub>4</sub> }	2	--	--	0		si	si		no
$\beta dif(X) = 145$									
145.-{0.90x <sub>1</sub> , 0.21x <sub>4</sub> }	2	--	--	0		si	si		si
$\beta dif(X) = 148$									
146.-{0.90x <sub>1</sub> , 0.40x <sub>4</sub> }									
147.-{0.90x <sub>1</sub> }									
148.-{ 0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	--	0	0	0	si	no		--	$\alpha if(X) = 159$
149.-{0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
150.-{0.38x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
151.-{ 0.38x <sub>2</sub> , 0.09x <sub>3</sub> }									
152.-{0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }									
153.-{0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }									
154.-{ 0.38x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
155.-{ 0.38x <sub>2</sub> , 0.98x <sub>3</sub> }									
156.-{ 0.38x <sub>2</sub> , 0.13x <sub>4</sub> }									
157.-{ 0.38x <sub>2</sub> , 0.21x <sub>4</sub> }									
158.-{ 0.38x <sub>2</sub> , 0.40x <sub>4</sub> }									
159.-{ 0.38x <sub>2</sub> }	--	0	--	--	si	si		si	$\alpha if(X) = 160$
160.-{ 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	--	1	1	0	si	no		--	$\alpha if(X) = 168$
161.-{ 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }									
162.-{ 0.42x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
163.-{ 0.42x <sub>2</sub> , 0.09x <sub>3</sub> }									
164.-{ 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }									
165.-{ 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }									
166.-{ 0.42x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
167.-{ 0.42x <sub>2</sub> , 0.98x <sub>3</sub> }									
168.-{ 0.42x <sub>2</sub> , 0.13x <sub>4</sub> }	--	1	--	0		si	si		no
$\beta dif(X) = 169$									
169.-{ 0.42x <sub>2</sub> , 0.21x <sub>4</sub> }	--	1	--	0		si	si		no
$\beta dif(X) = 170$									
170.-{ 0.42x <sub>2</sub> , 0.40x <sub>4</sub> }	--	1	--	1		no	--		--
$\alpha if(X) = 172$									
171.-{ 0.42x <sub>2</sub> }									
172.-{ 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	--	2	1	0		si	si		no
$\beta dif(X) = 173$									
173.-{ 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }	--	2	1	1		no	--		--
$\alpha if(X) = 176$									
174.-{ 0.52x <sub>2</sub> , 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }									
175.-{ 0.52x <sub>2</sub> , 0.09x <sub>3</sub> }									
176.-{ 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }	--	2	1	0		si	si		no
$\beta dif(X) = 177$									
177.-{ 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }	--	2	1	1	no	--		--	$\alpha dif(X) = 180$
178.-{ 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }									
179.-{ 0.52x <sub>2</sub> , 0.98x <sub>3</sub> , 0 }									
180.-{ 0.52x <sub>2</sub> , 0.13x <sub>4</sub> }	--	2	--	0		si	si		no
$\beta dif(X) = 181$									
181.-{ 0.52x <sub>2</sub> , 0.21x <sub>4</sub> }	--	2	--	1		no	--		--
$\alpha dif(X) = 184$									
182.-{ 0.52x <sub>2</sub> , 0.40x <sub>4</sub> }									
183.-{ 0.52x <sub>2</sub> , }									

184.-{ 0.09x <sub>3</sub> , 0.13x <sub>4</sub> }	--	--	1	0	si	si	no
βdif(X) = 185							
185.-{ 0.09x <sub>3</sub> , 0.21x <sub>4</sub> }	--	--	1	1	no	--	--
αdif(X) = 188							
186.-{ 0.09x <sub>3</sub> , 0.40x <sub>4</sub> }							
187.-{ 0.09x <sub>3</sub> }							
188.-{ 0.98x <sub>3</sub> , 0.13x <sub>4</sub> }	--	--	2	0	si	si	no
βdif(X) = 189							
189.-{ 0.98x <sub>3</sub> , 0.21x <sub>4</sub> }	--	--	2	1	no	--	--
αdif(X) = 192							
190.-{ 0.98x <sub>3</sub> , 0.40x <sub>4</sub> }							
191.-{ 0.98x <sub>3</sub> }							
<b>192.-{ 0.13x<sub>4</sub> }</b>	--	--	--	0	si	si	si
αdif(X) = 195							
193.-{ 0.21x <sub>4</sub> }							
194.-{ 0.40x <sub>4</sub> }							
195.-{ }							

Aunque en realidad el algoritmo sólo generará para su análisis los subconjuntos que tienen datos a su izquierda hemos decidido poner toda la lista para observar los saltos. Estos datos son el resultado de las consultas del algoritmo:

- 1) Las primeras 4 columnas tienen la evaluación de la función  $e_x$  sobre el subconjunto en turno.
- 2) La quinta columna representa la respuesta a la pregunta: ¿El subconjunto es testor?. Si la respuesta es negativa aplica la función  $\alpha dif(X)$  sin hacer ninguna otra pregunta.
- 3) La sexta columna contiene a respuesta a la pregunta ¿La función de error es inyectiva?. Si la respuesta es negativa se aplica  $\alpha dif(X)$  sin hacer ninguna otra pregunta.
- 4) La séptima columna tiene la respuesta a la pregunta ¿El conjunto es testor típico?. Si la respuesta es “si” se aplica al conjunto la función  $\alpha dif(X)$ . Si la respuesta es “no” se aplica la función  $\beta dif(X)$ .
- 5) El número que sigue a la función aplicada, señala el número del conjunto que se obtiene como resultado de la función.

Los testores típicos encontrados fueron encerrados en recuadros. Nótese que el número de conjuntos generados por el algoritmo (48) es muy inferior al conjunto completo de combinaciones (173), menos del 30%, y conforme crecen las dimensiones de la matriz la proporción de conjunto generados es menor. Por ejemplo para la matriz de 8 renglones por 7 rasgos

$$MB = \begin{bmatrix} .05 & 0.21 & 0.07 & 0.75 & 0.93 & 0.77 & 0.42 \\ .02 & 0.78 & 0.93 & 0.42 & 0.47 & 0.31 & 0.35 \\ 0.07 & 0.50 & 0.32 & 0.25 & 0.81 & 0.64 & 0.72 \\ 0.49 & 0.73 & 0.18 & 0.16 & 0.25 & 0.69 & 0.66 \\ 0.60 & 0.31 & 0.80 & 0.29 & 0.74 & 0.90 & 0.82 \\ 0.74 & 0.12 & 0.79 & 0.25 & 0.47 & 0.94 & 0.04 \\ 0.92 & 0.58 & 0.49 & 0.64 & 0.86 & 0.31 & 0.91 \\ 0.15 & 0.74 & 0.19 & 0.40 & 0.81 & 0.46 & 0.84 \end{bmatrix}$$

existen 851,641 subconjuntos difusos posibles, sin embargo, el algoritmo solamente construye 48,057, que representa únicamente el 5.64%.

## capítulo 5 Implantación Computacional

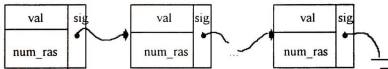
Los resultados alcanzados en la sección anterior, fueron implantados en un algoritmo computacional, desarrollado en lenguaje C++. La decisión de utilizar este lenguaje obedece a la gran capacidad del lenguaje para ser trasladado, por su fácil manejo de memoria de forma dinámica, entre otras cosas.

El programa se dividió en 5 subprogramas para un manejo más sencillo: REC\_DEFI.CPP, REC\_IMPR.CPP, RECLEE.CPP, REC\_ALGO.CPP y RECDIF2.CPP, que serán descritos a continuación.

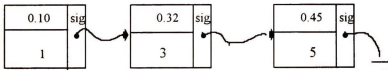
### REC\_DEFI.C

Este programa se encarga de definir las estructuras principales y las variables globales que se usaran en el resto de los programas. Básicamente las estructuras más importantes del programa son:

lista\_rasgos: es una lista simplemente ligada que guarda un conjunto de rasgos a la vez.

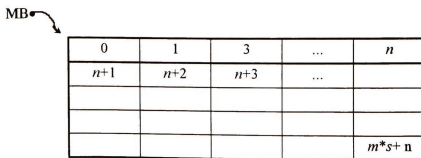


Por ejemplo el conjunto  $\{0.10x_1, 0.32x_3, 0.45x_5\}$  quedaría expresado en la siguiente lista ligada:



La matriz básica y la matriz básica ordenada son almacenadas en un vector de valores reales:

float \* MB;



donde:

$n$  es el tamaño del renglón (número de rasgos)

$m$  es el número de renglones (número de pares de comparaciones)

Este vector recibe un tratamiento de matriz auxiliándose de los valores de  $m$  y  $n$ .

El resultado de aplicar la función de error  $edif(X)$  se almacena en un vector de enteros:

funcion\_e



```

struct list_rasgos{
    float val;           : esta es la estructura que guardará los subconjuntos de rasgos
                        : valor de l rasgo (grado de pertenencia)
    int num_ras;        : número del rasgo(posición en el conjunto)
    struct list_rasgos *sig; : liga al siguiente rasgo
};

float *MB, *ordenMB;   : Apuntadores a la matriz Básica y una matriz con los
                        : datos de la matriz básica pero ordenados en forma
                        : ascendente

int m,n, longi;        : m: número de renglones;
                        : n: número de columnas;
                        : longi: longitud del conjunto que se analiza cada vez

int *funcion_e;        : apuntador al vector que guarda el resultado de la
                        : aplicación de la función de error.

typedef struct list_rasgos NODO;

NODO *Primero,*X;     : apuntador al primer rasgo
  
```

## REC\_IMPR.CPP

Este programa lleva a cabo la impresión de los resultados.

**Esta función imprime una lista ligada:**

```

void IMPRIME_LISTA(NODO *lista){
    if(lista){
        printf("%.2f x%d ",lista->val,lista->num_ras);
        IMPRIME_LISTA(lista->sig);
    }
}
  
```

**Esta función imprime un vector o arreglo de enteros:**

```
void IMPRIME_VECTOR(int *vector){
int i;

for(i=0; i<longj;i++)
    printf("\n error(x%d) =%d",i,vector[i]);
printf("\n");
}
```

**Esta función imprime matrices:**

```
void IMPRIME(float *matriz,int ren, int col){
int i,j;

for(i=0;i<ren;i++){
    for(j=0;j<col;j++)
        printf(" %5.2f ",matriz[i*col+j]);
    printf("\n");
}
}
```

## RECLEE.CPP

Este programa lee la matriz básica y se encarga de ordenarla:

**función:** PRUEBA\_LECTURA

**entrada:** entrada: un apuntador a archivo

s : una cadena de caracteres con el nombre del archivo

**salida:** 0 si el archivo esta vacío.

1 en otro caso

**Objetivo:** Verifica que el archivo con el que se trabajará no este vacío.

```
int PRUEBA_LECTURA(FILE *entrada, char s[]){
char micar;

micar = getc(entrada);
if(!feof(entrada) ){
    ungetc(micar, entrada);
    return(1);
}
else{
    printf("\n El archivo: %s esta vacio", s);
    return(0);
}
}
```

**función:** LEE\_ARCHI

**entrada:** no hay

**salida:** no hay

**Objetivo:** Se encarga de llevar a cabo la lectura de la matriz básica de un archivo, suponiendo que ese archivo tiene en sus dos primeros renglones dos valores enteros, el primero correspondiente al número de renglones y segundo correspondiente al número de columnas de la matriz que se leerá.

```
void LEE_ARCHI(){
int i,j;
FILE *archivo;
char nombre[12];

    clrscr();
    gotoxy(13,10);printf("Dame el nombre del archivo : ");
    scanf("%s",nombre);
    if( (archivo=fopen(nombre,"r"))==NULL){
        printf("no se pudo habrir el archivo");
        exit(1);
    }
    else
    if (PRUEBA_LECTURA(archivo,nombre)){
        fscanf(archivo,"%d",&m);
        fscanf(archivo,"%d",&n);
        gotoxy(printf("La matriz leida del archivo es:\n"));
        MB=(float *) malloc((m*n) * sizeof(float));
        for(i=0;i<m;i++){
            for(j=0;j<n;j++){
                fscanf(archivo,"%f",&MB[i*n+j]);
                ordenMB[i*n+j]=MB[i*n+j];
                printf(" %f ",MB[i*n+j]);
            }
            printf("\n");
        }
    }
}

free(MB);
}
```

**función:** LECTURA

**entrada:** no hay

**salida:** no hay

**Objetivo:** preguntar al usuario si los datos serán introducidos por medio de un archivo o se introducirán del teclado. Para el primera caso manda llamar a la función LEE\_ARCHI, para el segundo caso los lee directamente aquí.

```
void LECTURA()
{
int i,j;
char res, nombre[12];
FILE *archivo;

longi=0;
clrscr();
gotoxy(10,10);printf("la lectura puede ser T:teclado, A:archivo");
gotoxy(12,12);printf("opcion");
scanf("%c",&res);
if(toupper(res)=='T'){
    printf("número de renglones = ");
    scanf("%d",&m);
```

```

printf("número de columnas = ");
scanf("%d",&n);

MB=(float *) malloc((m*n) * sizeof(float));
ordenMB=(float *) malloc((m*n) * sizeof(float));

for(i=0;i<m;++i){
    for(j=0;j<n;++j)
        {
            printf(" p[%d][%d] = ",i,j);
            scanf("%f",&MB[i*n+j]);
            ordenMB[i*n+j]=MB[i*n+j];
        }

    printf("\n");
}
}
else
if(toupper(res)=='A')
    LEE_ARCHI();
else{
    printf("No se recibio una A o una T");
    exit(1);
}
}

```

**Función: PARTICION**

**Entrada:** Un vector de valores reales  $a$ ;  
 dos enteros  $l$ : inicio de la lista.  
 $r$ : final de la lista.

**Salida:**  $j$  : el nuevo final de la lista.

**Descripción:** Busca un pivote, y en base a este valor divide los elementos de la lista en dos nuevas sublistas.

```

int Particion(float a[ ], int l, int r){
float x,aux;
int i, j;

x=a[l];
i=l-1;
j=r+1;
while(1){
do
    j=j-1;
while(a[j]>x);
do
    i=i+1;
while(a[i]<x);
if(i<j){
    aux=a[i];
    a[i]=a[j];
    a[j]=aux;
}
else

```



```

        return(j);
    }
}

```

**Función:** QuickSort

**Entrada:** a: un vector de valores reales  
 l: el inicio de la lista  
 r: el final de la lista

**Salida:** lista ordenada a.

**Descripción:** Ordena la lista mediante llamados recursivos.

```

void QuickSort(float a[ ], int l, int r){
int q;

    if(l<r){
        q=Particion(a,l,r);
        QuickSort(a,l,q);
        QuickSort(a,q+1,r);
    }
}

```

**Función:** ORDENA

**Entrada:** no hay

**Descripción:** Se encarga de ir tomando cada una de las columnas de la Matriz Básica e ir las enviando una por una a la función QUICKSORT, para que sean ordenados sus valores y de esta manera ir construyendo la matriz OrdenMB.

```

void ORDENA(){
float *a;
int i,j;

a=(float *) malloc(m * sizeof(float));
for(j=0;j<n;j++){
    for(i=0; i<m;i++){
        a[i]=MB[i*n+j];
        QuickSort(a,0,m-1);
        for(i=0;i<m;i++){
            ordenMB[i*n+j]=a[i];
        }
        free(a);
    }
}

```

## REC\_ALGO.CPP

Este programa esta formado por las funciones que realizan la aplicación en sí del algoritmo RECDIF.

**Función:** INICIO

**Entrada:** X una apuntador a la lista de rasgos.

col: un entero que indica la columna a partir de la cual se inicia

**Descripción:** Esta función se encarga la primera vez de formar el conjunto de rasgos inicial con su mínimo grado asociado. Después es invocada por las funciones de recorrido para generar el conjunto H que se describió en la función de recorrido  $\rho_{dif}(X)$ , es decir el conjunto de rasgos con su mínimo grado a la derecha de un j-ésimo valor. Se auxilia de ordenMB, para saber quienes son los mínimos grados con respecto a cada rasgo.

```

NODO *INICIO(NODO *X,int col){
int i=0;
while(col<n){
if (X == NULL)
{
X = (NODO *) malloc( sizeof(NODO) );
longi++;
do{
X->val=ordenMB[i*n + col];
i++;
}while(X->val == 0 && i<m);
X->num_ras=col;
X->sig=NULL;
}
else
{
col++;
X->sig = INICIO(X->sig,col);
}
}
return(X);
}

```

**Función: ERROR**

**Entrada: X** : un apuntador a la lista del conjunto de rasgos que será analizado.

**Descripción:** Esta función es la implantación computacional de la función  $edif(X)$ .

```

void ERROR(NODO *X){
NODO *Aux;
int error,j,i,k,bandera;

Aux = X;
j=0;
error =-1;
while(error != 0 && j<longi){
error = 0;
for(i=0; i<m;i++){
bandera = 1;
Aux = X;
for(k=0; k<=j; k++){
if(Aux->val<=MB[i*n+Aux->num_ras])
bandera=0;
Aux = Aux->sig;
}
if(bandera)
error++;
}
}
}

```

```

        funcion_e[j]=error;
        j++;
    }
    for(j<n;j++)
        funcion_e[j]=0;
}

```

**Función: TESTOR\_TIP****Entrada:** X: Apuntador a la lista del conjunto de rasgos**Salida:** 1: si el conjunto de rasgos es testor típico  
0: si el conjunto de rasgos no es típico.**Descripción:** Esta función verifica si el conjunto de rasgos X es un testor típico, buscando para cada elemento del conjunto candidato a testor típico, un renglón en el cual aparezca, y que en ese mismo renglón el resto de los elementos del testor sean superiores a los que se encuentran en la matriz.

```

int TESTOR_TIP(NODO *X){
    NODO *aux1=NULL, *aux2=NULL;
    int i,j;

    aux1=X;
    while(aux1){
        i=0;
        while(aux1->val != MB[i*n+aux1->num_ras] && i<m)
            i++;
        if(i==m)
            return(0);
        aux2=X;
        for(j=0;j<longi;j++){
            if(aux1!=aux2)
                if(aux2->val <= MB[i*n+aux2->num_ras])
                    return(0);
            aux2=aux2->sig;
        }
        aux1=aux1->sig;
    }
    return(1);
}

```

**Función: INYECTIVA****Entrada:** no hay**Descripción:** Verifica si la función de error aplicada sobre el conjunto de rasgos es inyectiva o no, auxiliándose de la información que se encuentra en el vector *funcion\_e*.**Salida:** 0 si la función *edif(X)* no es inyectiva.  
1 si la función *edif(X)* es inyectiva.

```

int INYECTIVA(){
    int i;

    for(i=1;i<longi;i++)
        if(funcion_e[i-1]==funcion_e[i])

```

```

        return(0);
    }
    return(1);
}

```

**Función: ELIMINA****Entrada:** X: Apuntador a la lista del conjunto de rasgos**Descripción:** Elimina un nodo de la lista de rasgos**Salida:** devuelve un apuntador al nodo siguiente del que fue borrado o NULL si este nodo era el último.

```

NODO *ELIMINA(NODO *X){
NODO *Aux, *Aux2;

    Aux = X;
    Aux2= Aux->sig;
    free(Aux2);
    return(Aux2->sig);
}

```

**Función: GUARDA****Entrada:** no hay**Descripción:** Imprime en la pantalla una conjunto de rasgos.

```

void GUARDA(NODO *X){
NODO *Aux;

    Aux=X;
    printf("\n { ");
    while(Aux){
        printf("%.2f x%d",Aux->val,Aux->num_ras+1);
        Aux=Aux->sig;
        if(Aux) printf(", ");
    }
    printf("\n");
}

```

**Función: SIGMADIF****Entrada:** X : un apuntador a la lista del conjunto de rasgos que será analizado.**Descripción:** Esta función es la implantación computacional de la función *odif(X)* .

```

void SIGMADIF(NODO *X){
NODO *Aux;
int elimin=0;
int i;

    Aux = X;
    for(i=0;i<longi-1;i++){
        if(funcion_e[i]==funcion_e[i+1]){
            Aux->sig = ELIMINA(Aux);
            elimin++;
        }
        else
            Aux=Aux->sig;
    }
    longi-=elimin;
}

```

```
}

```

**Función: BETADIF**

**Entrada:** X : un apuntador a la lista del conjunto de rasgos que será analizado.

Ante: un apuntador al rasgo anterior al primero que entra en la función.

**Descripción:** Esta función es la implantación computacional de la función  $\beta dif(X)$ .

```
void BETADIF(NODO *X,NODO *Ante){
NODO *aux;
int i;

aux = X;

if(aux->sig)
    BETADIF(aux->sig,aux);
else
    {
    for(i=0; aux->val >= ordenMB[i * n + aux->num_ras] && i<m;i++);
    if(i>=m)
        {
        if(aux->num_ras >= n-1)
            {
            aux=NULL;
            Ante->sig=NULL;
            free(aux);
            longi--;
            }
        else
            {
            aux->val=0;
            i=0;
            aux->num_ras++;
            for(i=0; aux->val >= ordenMB[i * n + aux->num_ras] && i<m;i++);
            aux->val = ordenMB[i * n + aux->num_ras];
            aux->sig = INICIO(aux->sig,aux->num_ras+1);
            }
        }
    else
        {
        aux->val = ordenMB[i * n + aux->num_ras];
        aux->sig = INICIO(aux->sig,aux->num_ras+1);
        }
    }
}

```

**Función: ALFADIF**

**Entrada:** X : un apuntador a la lista del conjunto de rasgos que será analizado.

Ante: un apuntador al rasgo anterior al primero que entra en la función.

**Descripción:** Esta función es la implantación computacional de la función  $\alpha dif(X)$ .

```
void ALFADIF(NODO *X, NODO *Ante){
NODO *Aux;

Aux=X;

```

```

if(Aux->sig)
    ALFADIF(Aux->sig, Aux);
else
{
    if(Aux->num_ras==(n-1)){
        if(longi==1){
            Aux=NULL;
            free(Aux);
            longi--;
        }
        else
        {
            Aux=NULL;
            free(Aux);
            longi--;
            Ante->sig=NULL;
            BETADIF(Ante,Ante);
        }
    }
    else
        BETADIF(Aux,Ante);
}
}
}

```

**Función: RECDIF**

**Entrada:** X : un apuntador a la lista del conjunto de rasgos que será analizado.

**Descripción:** Esta función es propiamente la implantación computacional del algoritmo RECDIF.

```
void RECDIF(NODO *X){
```

```

    clrscr();
    IMPRIME(MB,m,n);
    gotoxy(10,10);printf(" lista de testores difusos tipicos:");
    gotoxy(10,11);printf("=====\n");
    funcion_e=(int *)malloc(n*sizeof(int));
    while(longi>0){
        ERROR(X);
        if(funcion_e[longi-1]==0){
            if(INYECTIVA()){
                if(TESTOR_TIP(X)){
                    GUARDA(X);
                    ALFADIF(X,NULL);
                }
                else
                    BETADIF(X,NULL);
            }
            else
                SIGMADIF(X);
        }
        else
            ALFADIF(X, NULL);
    }
}
}

```

## RECDIF2.CPP

Este es el programa principal que contiene la función principal main. Se encarga de ir llamando las funciones y de libera la memoria solicitada.

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include "rec_defi.cpp"
#include "rec_impr.cpp"
#include "reclee.cpp"
#include "rec_algo.cpp"

NODO *lib(NODO *X){

    if(X->sig)
        X->sig=lib(X);
    free(X);
    return(NULL);
}

void LIBERA(NODO *primero){
    free(ordenMB);
    free(MB);
    free(funcion_e);
    lib(primero);
}

void main( ){
    //programa principal
    LECTURA();
    ORDENA();
    Primero =NULL;
    Primero= INICIO(Primero,0);
    IMPRIME_LISTA(Primero);
    RECDIF(Primero);
    LIBERA();
}

```

## CONCLUSIONES

Existen muchos métodos para llevar a cabo la selección de variables. Algunos de ellos son bastante eficientes, bajo ciertas condiciones que se establecen claramente en los lugares donde se describen. Si el problema que deseamos resolver, cumple los requerimientos establecidos por algún método eficiente de selección de variables, sólo hay que aplicarlo.

Sin embargo, en muchas ocasiones el problema que tenemos enfrente no cumple con las condiciones establecidas para los métodos descritos en la primera parte de este trabajo. A esto habría que agregar los problemas para los cuales la solución tiene una buena carga de difusión.

El algoritmo aquí presentado, permite localizar los testores en medios difusos (testores de Goldman), sin ser muy estricto en sus requerimientos. Básicamente se necesita contar con una clasificación a priori de un conjunto de objetos, y poder expresar esta información en términos de una matriz de aprendizaje, una matriz de diferencias o una matriz básica.

El presente trabajo extiende un algoritmo que permite calcular los testores difusos típicos de Goldman sobre una Matriz básica, clases disjuntas y conceptos más flexibles de semejanza. Los testores difusos típicos, entre otras cosas, pueden ser empleados para llevar a cabo una selección ó determinación de variables, ya sea como un objetivo *per se* ó como proceso previo a la clasificación de objetos. Los conjuntos de objetos tratados pueden no tener clases disjuntas, no manejar un concepto de igualdad estricto y no requerir una distribución probabilística predeterminada entre los elementos. Los rasgos que describen a los elementos, pueden tener distintos espacios de representación sin requerir necesariamente la autocorrelación de los datos o de los rasgos ó presentar un comportamiento periódico.

Los obstáculos que involucraba extender el algoritmo RECPLUS para el caso difuso, se presentar desde el momento de comprobar matemáticamente que el algoritmo localiza todos los testores típicos de la matriz básica, y que lo hace sin llevar a cabo una búsqueda exhaustiva entre las combinaciones de rasgos, que dicho sea, crece exponencialmente cuando crece el número de rasgos involucrados. La efectividad del algoritmo crece conforme crece el número de rasgos, como se menciona en el capítulo 4., donde para una matriz de 8 renglones y 7 rasgos, que produciría 851,641 combinaciones distintas, el algoritmo genera 48,057 que representa el 5.64%, solamente.

Este trabajo ha sido presentado en 3 eventos internacionales:

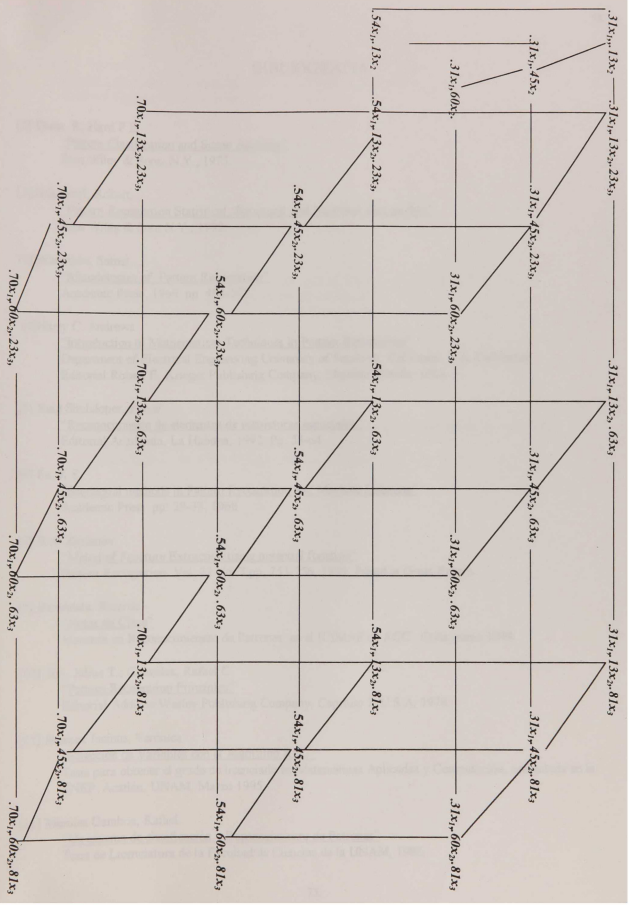
- En "Geoinfo 94" organizado por el Instituto de Geofísica y Astronomía, y que se llevo a cabo en el Capitolio de la Ciudad de la Habana, en Agosto de 1994.
- En el congreso Internacional "CIMAF 95" organizado por el Instituto de Cibernética, Matemática y Física de la Academia de Ciencias de Cuba, en el Palacio de las Convenciones de la Cd. de la Habana, Cuba, en Enero de 1995.
- En el Simposium Internacional de Computación "Tendencias de la Computación hacia el nuevo milenio", organizado por el Centro Nacional de Cálculo, en el centro cultural "Jaime Torres Bodet", en Noviembre de 1995.

El trabajo fue seleccionado para publicarse en las memorias de los dos últimos eventos.

Se cuenta entonces ahora otra nueva herramienta para el cálculo de testores típicos difusos. Las futuras aplicaciones incluyen el estudio de la relevancia de los rasgos para el reconocimiento de imágenes planas, y la selección de rasgos para la clasificación de la complejidad de las operaciones en niños con labio paladar endido.



## ANEXO I.



## BIBLIOGRAFÍA

- [1] Duda, R; Hard P.E.  
"Pattern Clasification and Scene Analisis"  
 Jhon Wiley & Sons, N.Y., 1973
- [2] Schalkoff, Robert.  
"Pattern Recognition Statistical, Structural and Neuronal Approaches"  
 Jhon Wiley & Son, N.Y., 1992.
- [3] Watanabe, Satoshi.  
"Metodologies of Pattern Recognition"  
 Academic Press. 1969 pp. 493-508.
- [4] Harry C. Andrews.  
"Introduction to Mathematical Techniques in Pattern Recognition"  
 Department of Electrical Engineering University of Southern. California. L.A. California  
 Editorial Robert E. Krieger Publishing Company, Malabar Florida. 1983
- [5] Ruiz Shulcloper, J *et al.*  
"Reconocimiento de elementos de estructuras espaciales"  
 Editorial Academia, La Habana, 1992. Pp. 53-64.
- [6] Fu, K.S.  
"Sequential methods in Pattern Recognition and Machine Learning"  
 Academic Press. pp: 29-33. 1968
- [7] B.A. Zyrianov  
"Metod of Feacture Extraction using potential function"  
 Pattern Recognition. Vol. 23 No. 7 pp. 753-756, 1990. Printd in Great Britain.
- [9] Barandela, Ricardo  
"Notas de Clase"  
 Maestría en Reconocimiento de Patrones. en el ICIMAF de ACC. Cuba, junio 1994.
- [10] Tou, Julius T.; González, Rafael C.  
"Pattern Recognition Principles"  
 Editorial Addison Wesley Publishing Company. Capitulo 7. U.S.A. 1974.
- [11] Jiménez Jacinto, Verónica.  
"Selección de Variables con el Algoritmo REC"  
 Tesis para obtener el grado de licenciada en Matemáticas Aplicadas y Computación, presentada en la  
 ENEP Acatlán. UNAM. Marzo 1995.
- [12] Morales Gamboa, Rafael.  
"Un sistema de clasificación y Reconocimiento de Patrones"  
 Tesis de Licenciatura de la Facultad de Ciencias de la UNAM, 1988.

- [13] Kurt J. Schmuker  
“Fuzzy Sets, Natural Language Computation and Risk Analysis”  
 Editorial Computer Science Press. 1984. pp.1-18
- [14] Kaufmann, A.  
“Introduction to the Theory of Fuzzy Subsets”  
 Volumen I. Academic Press. London 1975. 1-38
- [15] Kandel, Abraham.  
“Fuzzy Mathematical Techniques with Applications”  
 Addison-Wesley Publishing Company U.S.A. 1986. pp. 1-71.
- [17] Lazo Cortés, Manuel.  
“Modelos basados en la Teoría de Testores para la Selección de Rasgos y la clasificación supervisada con descripciones no clásicas de Objetos”.  
 Tesis Doctoral de la Universidad de las Villas, Santa Clara, Cuba, 1994.
- [18] Ruiz Shulcloper, J; Lazo Cortés, M.  
“Reconocimiento de Patrones II”  
 BUAP, FCFM. Colegio de Computación. Diplomado en computación. 1994.
- [19] Lazo Cortés, M y Barreto Fiu EE  
“Testores Difusos. Un algoritmo para determinar los testores difusos típicos de una matriz de Aprendizaje” en “Temas acerca de la Teoría de Testores”  
 Serie Amarilla. No. 134. CINVESTAV México, 1994. pp.29-42

Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará la Lic. Verónica Jiménez Jacito, declaramos que hemos revisado, declaramos que hemos revisado la tesis titulada:

**“EXTENSIÓN DEL ALGORITMO REC PARA EL CALCULO DE TESTORES TIPICOS”**

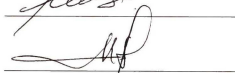
y consideramos que cumple con los requisitos para obtener el grado de Maestro en Ciencias, con especialidad en Ingeniería Eléctrica.

**Atentamente**

Dr. José Ruiz Shulcloper



Dr. Manuel Sabino Lazo Cortés



Dr. Humberto Sossa Azuela



M. en C. Juan José Montellano Ballesteros



CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL  
INSTITUTO POLITECNICO NACIONAL

**BIBLIOTECA DE INGENIERIA ELECTRICA**  
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro  
antes del vencimiento de préstamo señalado  
por el último sello.

DEVOLUCION



