Instituto Politécnico Nacional
Escuela Superior de Física y Matemáticas

Departamento de Matemáticas

Reconocimiento de Patrones, El Enfoque Difuso

Tesis que para obtener el título de Ingeniera Matemática
Presenta:

Elizabeth Herrera Ocaranza[1]

Director de Tesis: Dr. César Alberto Escobar Gracia

[1]Esta tesis forma parte del proyecto de investigacin SAPPI 20082589. Con Apoyo pacial de CONACyT y COFAA

Resumen

Objetivo: Esta tesis tiene como objetivo el mostrar la implementación de las herramientas matemáticas que se me han proporcionado durante mis estudios correspondientes a la Ingeniería Matemática, en el caso concreto de la clasificación mediante la lógica difusa.

El contenido de la tesis se desarrollará de la siguiente manera:

En el primer capítulo se da el marco teórico y justificaciones del estudio de la lógica difusa. Se introduce el objeto de estudio general, explicando en que consiste la clasificación sin aprendizaje aclarando, además, los distintos enfoques en como se puede abordar dicho problema.

En el capítulo dos se presentan los conceptos básicos de matemáticas empleados a lo largo de la tesis, introduciendo ejemplos para una mayor claridad de los mismos. Se dan los conceptos de métrica, norma, espacio métrico, espacio normado y conceptos básicos de la lógica difusa como: función de pertenencia, conjunto difuso, un α-corte, etc. También se analizan los elementos de una clasificación sin aprendizaje: función de similaridad, Espacio de Representación Inicial y los criterios de agrupamiento.

En el tercer capítulo, por sus importantes características y aplicaciones, se dedica un espacio para estudiar con mayor detalle el enfoque estadístico bajo el cual se puede determinar el Espacio de Representación Inicial; aunque, es importante mencionar que el algoritmo c-means difuso, que se estudia en el capítulo 4, no considera un espacio bajo el enfoque estadístico. En la segunda parte de este capítulo se presenta ya una forma de clasificar. Se estudia el algoritmo C-means, que se presentara posteriormente pero desde el punto de vista difuso.

En el cuarto capítulo se estudia el algoritmo c-means difuso bajo el enfoque difuso de la clasificación y se finaliza presentando los códigos de programación del algoritmo c-means y c-means difuso.

Finalmente, se mencionan algunas ventajas y desventajas de los algoritmos estudiados según las características que presenta cada uno.

Dedicatoria

Este trabajo de tesis lo dedico a mi madre por su cariño y comprensión,

a mi padre por su ejemplo de superación y esfuerzo,

a mis hermanos por su cariño y compañía.

A todos ellos que son mi fortaleza y que siempre han estado a mi lado.

Elizabeth Herrera Ocaranza
2008

Contenido

Capítulo 1

Introducción

En la vida diaria nos encontramos ante situaciones en las que no se puede decir *si* o *no*, *frío* o *caliente*, alto o bajo. Lo más común es encontrarse con términos como: muy caliente, muy frío, temperatura media, temperatura baja, edad media, alto, mediano, etc. Los objetos que se estudian con este tipo de etiquetas resultan difíciles de clasificar o agrupar. La lógica difusa (borrosa o fuzzy) es una herramienta matemática que permite analizar objetos bajo información de este tipo, información imprecisa.

El conocimiento común, que es principalmente del tipo lingüístico cualitativo y no necesariamente cuantitativo, puede representarse matemáticamente mediante la lógica difusa utilizando conjuntos difusos y funciones asociadas a ellos. Los términos lingüísticos son menos precisos que los términos matemáticos pero resultan ser de mayor utilidad en la realidad. Por ejemplo, si se considera un conjunto de personas que se desean clasificar en dos conjuntos, un grupo de personas altas y otro de personas de estatura baja considerando como personas altas a las de estatura mayor o igual a 1.70m y personas de estatura baja a las que miden menos de 1.70m. ¿Sería correcto decir que una persona de 1.70m es alta mientras que la de 1.69 no lo es? Tal vez sea correcto bajo la definición dada pero la diferencia entre ambas estaturas es muy pequeña; sin embargo, una persona de 1.70m y una de 1.85m de estatura serían consideradas altas a pesar de que la diferencia entre la estatura de estas personas es mayor que en el primer caso. Con la lógica difusa una forma de clasificarlos sería indicando en que grado cada persona puede ser considerada alta o no. Así, podría decirse que la persona de 1.69m es alta en un grado de 0.1, la de 1.70m en 0.2 y la de 1.85m en 0.8. A los conjuntos resultantes se les llama "conjuntos difusos", donde cada persona

pertenece a cada agrupamiento pero con cierto grado.

1.1 Lógica difusa: Breve historia y aplicaciones

El término "difuso" lo introduce por primera vez el ingeniero Lotfi A. Zadeh, de la Universidad de California en Berkeley en el artículo "Fuzzy Sets" (Conjuntos difusos), que publicó en 1965. A partir de dicho artículo muchos otros matemáticos e ingenieros se ocuparon en el estudio de este tipo de conjuntos y las aplicaciones comenzaron a surgir. La aplicación industrial considerada como la primera fue en 1978 por la empresa F. L. Smidt que construyó un controlador de un horno de cemento. En la actualidad la lógica difusa tiene aplicaciones en el campo científico-técnico.

Con base en la lógica difusa se desarrolló un algoritmo que permite formar conjuntos difusos, este algoritmo se conoce como c-means difuso. El algoritmo c-means difuso es actualmente utilizado en diversos campos de estudios tales como: medicina, geografía, economía, robótica, etc. Este ha sido empleado para la elaboración de mapas que permiten delimitar áreas geográficas; en la economía, se han detectado fraudes; en la industria, se ha empleado el algoritmo para la programación de maquinaria y aparatos electrodomésticos, etc.

En asuntos más concretos se puede mencionar la aplicación del algoritmo para predecir la concentración de ozono en la ciudad de México y para el control de temperatura en ciertos sistemas. En el metro de Senadi, Japón, se obtuvo una aceleración y un frenado más suaves. La compañía LG lanzo al mercado una lavadora automática que cuenta con un sistema de controlador difuso el cual hace más sencillo el trabajo de lavado, ya que ella toma la decisión, según la calidad del agua que desecha, de pasar a un ciclo más largo o de mayor intensidad.

1.2 Justificaciones

Por la importancia que presenta la lógica difusa en aplicaciones modernas, en este trabajo de tesis se presentan dos algoritmos para formar agrupamientos bajo ciertas características: el algoritmo **c-means** y el **c-means difuso** bajo el **enfoque difuso**. Este enfoque es en base a la lógica difusa y su algoritmo,

el c-means difuso, permite formar agrupamientos bajo las características de esta lógica. Mientras que el algoritmo c-means permite formar conjuntos "duros", es decir, conjuntos de la Teoría clásica de conjuntos. Analizar el algoritmo c-means nos permitirá analizar el enfoque difuso, y por lo tanto el algoritmo c-means difuso, por la semejanza de conceptos y herramientas matemáticas de ambos.

La Clasificación con aprendizaje, Clasificación con aprendizaje parcial y Clasificación sin aprendizaje son los casos posibles bajo los cuales se pueden formar agrupamientos. El caso que nos ocupa en este texto y que nos permitirá analizar los enfoques métrico y difuso es la Clasificación sin aprendizaje (CSA).

1.3 El problema de la CSA

El objetivo de la CSA (clasificación sin aprendiaje) es resolver un problema que consiste en esencia, en hallar la estructura interna de un conjunto de descripciones de objetos en el espacio de representación. Esta estructura interna obviamente depende, en primera instancia, de la selección del propio espacio de representación y de la forma en que los objetos se comparen, es decir del concepto de similaridad que se utilice y de la forma en que éste se emplee. De dicha estructuración, en términos generales pudiéramos decir que:

1. Se sabe o se desea que se realice un número dado de agrupaciones.

2. Se desconoce en cuántas agrupaciones se estructurará el conjunto de objetos una vez definidos el espacio de representación y los conceptos de similaridad y la forma de usarlos.

En cualquiera de los dos casos, un problema de clasificación sin aprendizaje consiste en hallar un procedimiento por el cual se pueda conocer la estructura interna del conjunto de descripciones de objetos dado. Para encontrar esa estructura existen tres formas generales de hacerlo, a saber:

1. El paradigma del conjunto cociente.

2. El paradigma del solapamiento.

3. El paradigma difuso.

1.4 Paradigmas de la CSA.

El paradigma del conjunto cociente, consiste en la formación de una partición del conjunto de objetos dado, bajo el supuesto que los mismos serán conjuntos en el sentido clásico de la Teoría de conjuntos. En otras palabras, de lo que se trata es de hallar el conjunto cociente del dado en el espacio de representación en cuestión. Esto supone que los agrupamientos serán ajenos. Aquí las propiedades que caracterizan a un agrupamiento dado contradicen las propiedades que caracterizan a cualquier otro de los restantes agrupamientos obtenidos.

El paradigma del solapamiento permite que las agrupaciones tengan elementos comunes; es decir, se trata de hallar un cubrimiento del conjunto de descripciones de objetos dado por subconjuntos (también en el sentido clásico) no necesariamente ajenos. Las propiedades que caracterizan a un agrupamiento dado pudieran ser satisfechas por otro de los agrupamientos restantes.

El paradigma difuso, sin embargo, parte de una suposición conceptual diferente: de los objetos no podemos afirmar categóricamente que pertenecen o no a un conjunto dado, sólo podemos hablar de grados de pertenencia (como es característico de la Teoría de los Subconjuntos Difusos, creada por Lotfi A. Zadeh en 1965). Así, el problema de la clasificación sin aprendizaje bajo estas suposiciones consiste en hallar una partición o bien un cubrimiento difuso del conjunto de objetos dado. Es claro que en estos casos las propiedades que caracterizan a los conjuntos, subconjuntos difusos del universo en cuestión, son satisfechas en cierto grado por todos los objetos del universo de estudio.

En todos estos paradigmas hay factores comunes. Uno de ellos esencial para la solución del problema de la clasificación sin aprendizaje, es la selección del *criterio de agrupamiento*.

La selección de un criterio de agrupamiento (se definirá mas adelante con precisión) puede realizarce de maneras diferentes.

Se llega al criterio de agrupamiento mediante la modelación matemática del problema y por la misma vía se llega al enfoque de realización de la estructuración del conjunto de objetos. Suponemos el enfoque, es decir, lo imponemos, y condicionamos el criterio de agrupamiento de modo tal que resulte una estructura acorde con el enfoque seleccionado.

El estudio de todos estos enfoques se puede hacer bajo dos ópticas, que aunque muy relacionadas poseen diferencias, en apariencia sutiles, consideradas de suma importancia para los análisis posteriores. Nos referimos a lo

que pudiéramos denominar una óptica *clasificatoria* y una óptica *conjuntual*.

En el enfoque clasificatorio se tiene un universo de objetos y se necesita agruparlos de modo tal que los objetos del mismo agrupamiento se parezcan más entre sí que con objetos de otros agrupamientos.

En el enfoque conjuntual se tiene un universo de objetos y se necesita agruparlos de modo tal que los objetos que estén en el mismo agrupamiento cumplan (en cierto grado) la propiedad que caracteriza al agrupamiento (como conjunto en su determinación intencional).

El objetivo fundamental del problema de clasificación sin aprendizaje es el de conocer la estructura interna de una población de objetos dada. Esa población pudiera ser una clase de objetos en un problema de clasificación con aprendizaje o con aprendizaje parcial.

El interés en lograr esa estructura puede ser porque se desea posteriormente clasificar nuevos objetos ya que la "población" a la que se está haciendo referencia no es todo el universo de objetos de un problema en cuestión. Por ejemplo puede ser el interés, la necesidad de analizar los datos con fines de depuración, de seleccionar la mejor representación de la clase en cuestión, etc.

A todos estos intereses asiste un factor común: las causas intrínsecas de la agrupación responden a la similitud en el enfoque clasificatorio, al cumplimiento de una propiedad en el enfoque conceptual.

Todos estos problemas serán analizados en el espacio de representación inicial (ERI) que en este enfoque no es más allá de un producto cartesiano de los valores admisibles de las variables en términos de las cuales se describen todos los objetos del universo en cuestión.

Un elemento substancial en los problemas de clasificación sin aprendizaje lo constituye el concepto de función de semejanza, junto a este factor habrá que considerar la forma en que esta función de semejanza es usada para realizar la estructuración del ERI.

1.5 Objetivo de la tesis

La presente tesis tiene como objetivo el estudio del **enfoque difuso** y de su algoritmo, c-means difuso. El algoritmo c-means se desarrollo bajo el paradigma del conjunto cociente y la óptica clasificatoria. Es decir, dado un universo de objetos se desea clasificarlos en **c** agrupamientos de manera tal que sean conjuntos ajenos y que los elementos de cada conjunto se parezcan

entre sí más de lo que pudieran parecerse con objetos de los otros conjuntos. El enfoque difuso se basa en el paradigma difuso y bajo la óptica clasificatoria. Los agrupamientos formados bajo este enfoque se pretende que cumplan con las características de los agrupamientos formados con el algoritmo c-means, solo que no se formarán agrupamientos ajenos y en su lugar se hablará de grados de pretenecia de cada elemento a cada conjunto.

Capítulo 2

Conceptos básicos

En este capítulo se presentan conceptos básicos de matemáticas que permitirán analizar los algoritmos mencionados anteriormente.

2.1 Distancia

De manera intuitiva, *espacio métrico* se define como un conjunto en el que se puede hablar de *distancia* entre sus elementos. Sin embargo, esta definición requiere a su vez conocer el concepto de *distancia*. En primer instancia se define como una función que asocia un número real positivo a todo par de elementos de un conjunto.

Definición 2.1.1 *Una* ***métrica*** *(distancia) es una función*

$$d : E \times E \to \mathbb{R}$$

que tiene las siguientes propiedades:

1. $\forall$ x, $y \in$ E : $d(x,y) \geq 0$ (positividad)
2. Para x, $y \in$ E : $d(x,y) = 0 \Leftrightarrow$ x=y. (precisión)
3. $\forall$ x, $y \in$ E : $d(x,y) = d(y,x)$ (simetría).
4. $\forall$ x, y, $z \in$ E : $d(x, y) \leq d(x, z) + d(y, z)$ (desigualdad triangular).

Definición 2.1.2 *Un* ***espacio métrico*** *es un par (E, d) constituido por el conjunto E y una métrica d definida sobre E.*

Debe mencionarse que se pueden definir más de una métrica sobre un mismo espacio, con lo que se obtienen diferentes espacios métricos.

El siguiente lema es un resultado de las propiedades que definen una métrica y que será de mucha utilidad en aplicaciones posteriores.

Lema 2.1.3 *En un espacio métrico* (E, d) *se verifica*

$$\forall\ x, y, z, t \in E : |d(x, y) - d(z, t)| \leq d(x, z) + d(y, t).$$

En particular:

$$\forall\ x, y, z \in E : |d(x, z) - d(y, z)| \leq d(x, y).$$

Demostración. Sean x, y, z, t $\in$ E y considerando que d es una métrica sobre E, entonces se cumple:

$$\begin{aligned} d(x, y) &\leq d(x, z) + d(y, z) \\ &\leq d(x, z) + d(y, t) + d(z, t) \end{aligned}$$

luego,

$$d(x, y) - d(z, t) \leq d(x, z) + d(y, t)...(1)$$

Por otro lado:

$$\begin{aligned} d(z, t) &\leq d(z, x) + d(t, x) \\ &= d(x, z) + d(t, x) \\ &\leq d(x, z) + d(t, y) + d(x, y) \\ &= d(x, z) + d(y, t) + d(x, y) \end{aligned}$$

de donde

$$-d(x, z) - d(y, t) \leq d(x, y) - d(z, t)$$

$$-(d(x,z)+d(y,t)) \leq d(x,y)-d(z,t)...(2)$$

Por las desigualdades (1) y (2) y las propiedades de valor absoluto se obtiene:

$$|d(x,y)-d(z,t)| \leq d(x,z)+d(y,t)$$

■

Se presentan ejemplos con el objetivo de entender con mayor claridad la definición de una métrica.

Ejemplo 1 *Sea E un conjunto cualquiera no vacío. Y se define la función*

$$d : E \times E \to \mathbb{R}$$

tal que: $\forall\ x, y \in E$:

$$d(x,y) = 1, \quad si \ \ x \neq y;$$

$$d(x,y) = 0, \quad si \ \ x = y.$$

Demostrar que la función d es una métrica en E.

Solución. Véase *d(x,y)*≥ 0 para todo $x,\ y \in \mathbb{R}$, cumpliendose la igualdad

cuando x=y, por lo tanto las propiedades de positividad y precisión se satisfacen, respectivamente.

Para verificar que se cumple la propiedad de simetría veáse que, según la definición de la función d, se cumple

$$d(y,x) = 1, \text{ si } y \neq x;$$

$$d(y,x) = 0, \text{ si } y = x.$$

Ahora, considere x, y, $z \in$ E, entonces

$$0 \leq d(x,z) + d(y,z) \leq 2$$

el mínimo valor de la suma es 0 y esto pasa para cuando los tres elementos de E son iguales, y el máximo valor es dos, para cuando los tres elementos son diferentes. En ambos casos la desigualdad triangular se cumple.

$$0 = d(x,y) \leq d(x,z) + d(y,z) = 0$$

o bien

$$1 = d(x,y) \leq d(x,z) + d(y,z) = 2$$

Ahora supóngase $x = y$, $x \neq z$ y $z \neq y$

$$0 = d(x,y) \leq d(x,z) + d(y,z) = 2$$

$$1 = d(x,z) \leq d(x,y) + d(y,z) = 1$$

En cualquier caso la desigualdad triangular se satisface, por lo tanto d es una métrica. ■

Ejemplo 2 *Considere el conjunto* $\mathbb{R}$ *y la función* $d(x,y)=|x-y|$ *con* x, $y \in \mathbb{R}$, *verificar que* d *es una métrica.*

Solución. Por definición de valor absoluto se sabe que

$$|x-y| \geq 0 \, \forall x, \, y \in \mathbb{R}$$

La igualdad se satisface para $x=y$, por lo tanto las porpiedades de positividad y precisión de una métrica se satisfacen. La propiedad de simetría se cumple y se verifica como sigue

$$\begin{aligned} d(x,y) &= |x-y| \\ &= |-(y-x)| \\ &= |y-x| \\ &= d(y,x) \end{aligned}$$

Para verificar que la función d también satisface la desigualdad triangular véase que dados x, y, $z \in \mathbb{R}$ se cumple

$$-|x-y| \leq x-y \leq |x-y|$$

y

$$-|y-z| \leq y-z \leq |y-z|$$

luego entonces

$$-|x-y|-|y-z| \leq (x-y)+(y-z) \leq |x-y|+|y-z|$$

$$\Rightarrow -(|x-y|+|y-z|) \leq x-z \leq |x-y|+|y-z|$$

$$\Rightarrow |x-z| \leq ||x-y|+|y-z||$$

$$\Rightarrow |x-z| \leq |x-y|+|y-z|$$

$$\Rightarrow |x-z| \leq |x-y|+|z-y|$$

$$d(x,z) \leq d(x,y)+d(z,y)$$

Como la función d cumple con las propiedades de una métrica entonces d definida como $d(x,y)=|x-y|$ es una métrica. ■

Ejemplo 3 *Considere el conjunto* $\mathbb{R}^2$ *y la función*

$$d((x_1,y_1),(x_2,y_2)) = |x_1-x_2|+|y_1-y_2|$$

definida en el conjunto. Demuestre que d es una métrica.

Solución. Dado que $|x_1-x_2| \geq 0$ y $|y_1-y_2| \geq 0$ y la igualdad se cumple para $(x_1,y_1)=(x_2,y_2)$ entonces $d((x_1,y_1),(x_2,y_2))$ cumple con las propiedades de positividad y precisión de una métrica. Ahora,

$$\begin{aligned} d((x_1,y_1),(x_2,y_2)) &= |x_1-x_2|+|y_1-y_2| \\ &= |x_2-x_1|+|y_2-y_1| \\ &= d((x_2,y_2),(x_1,y_1)) \end{aligned}$$

por lo tanto se cumple la propiedad de simetría. Para los puntos (x_1, y_1), (x_2, y_2), (x_3, y_3), se tiene:

$$d((x_1, y_1), (x_3, y_3)) + d((x_3, y_3), (x_2, y_2))$$
$$\Rightarrow | x_1 - x_3 | + | y_1 - y_3 | + | x_3 - x_2 | + | y_3 - y_2 |$$

Por la propiedad de la desigualdad del triángulo para cualesquiera $a, b \in \mathbb{R}$ se tiene $| a + b | \leq | a | + | b |$, entonces

$$(| x_1 - x_3 | + | x_3 - x_2 |) + (| y_1 - y_3 | + | y_3 - y_2 |) \geq$$
$$| x_1 - x_3 + x_3 - x_2 | + | y_1 - y_3 + y_3 - y_2 |$$
$$\Rightarrow d((x_1, y_1), (x_3, y_3)) + d((x_3, y_3), (x_2, y_2)) \geq | x_1 - x_2 | + | y_1 - y_2 |$$
$$\Rightarrow \; d((x_1, y_1), (x_3, y_3)) + d((x_3, y_3), (x_2, y_2)) \geq d((x_1, y_1), (x_2, y_2))$$

Por lo tanto la desigualdad del triángulo para una métrica se satisface. así que se concluye que $d((x_1, y_1), (x_2, y_2)) = | x_1 - x_2 | + | y_1 - y_2 |$ es una métrica. ■

Ejemplo 4 *Sea* $d((x_1, y_1), (x_2, y_2)) = max\{|x_1 - x_2|, |y_1 - y_2|\}$ *una función definida sobre* $\mathbb{R}^2$. *Se quiere verificar que la función d es una métrica, a la cual se le conoce comúnmente como distancia del camión.*

Solución. Las dos primeras propiedades de una métrica se satisfacen ya que $d((x_1,\ y_1),(x_2,\ y_2)) \geq 0$ y se satisface la igualdad para $x{=}y$.

Veáse que:

$$\begin{aligned} d((x_1, y_1), (x_2, y_2)) &= \max\{|x_1 - x_2|, |y_1 - y_2|\} \\ &= \max\{|x_2 - x_1|, |y_2 - y_1|\} \\ &= d((x_2, y_2), (x_1, y_1)) \end{aligned}$$

entonces se satisface la propiedad de simetría. Dados (x_1, y_1), (x_2, y_2), (x_3, y_3) $\in \mathbb{R}^2$ se tiene

$$d((x_1, y_1), (x_3, y_3)) + d((x_2, y_2), (x_3, y_3)) =$$
$$\max\{| x_1 - x_3 |, | y_1 - y_3 |\} + \max\{| x_2 - x_3 |, | y_2 - y_3 |\} = S$$

$$\Rightarrow \quad \begin{aligned} S \;=\; & \max\{| x_1 - x_3 | + | x_3 - x_2 |, | x_1 - x_3 | + | y_2 - y_3 |, \\ & | y_1 - y_3 | + | x_2 - x_3 |, | y_1 - y_3 | + | y_3 - y_2 |\} \end{aligned}$$

Haciendo uso de la desigualdad del triángulo para cualesquiera $a, b \in \mathbb{R}^2$, $\mid a+b \mid \leq \mid a \mid + \mid b \mid$ se obtiene

$$S \geq \max\{\mid x_1 - x_2 \mid, \mid x_1 - x_3 + y_2 - y_3 \mid, \mid y_1 - y_3 + x_2 - x_3 \mid, \mid y_1 - y_2 \mid\}$$

$$\geq \max\{\mid x_1 - x_2 \mid, \mid y_1 - y_2 \mid\}$$

■

La *norma* y *espacio normado* son conceptos de suma importancia para el desarrollo de este trabajo. Estos conceptos tienen algunas características semejantes a las que poseen la métrica y espacio métrico. La norma se describe de manera informal como la longitud de un vector que tiene la propiedad de ser positivo y satisface la desigualdad del triángulo, como la métrica. Además, al igual que para una métrica se define un espacio métrico, para una norma se define un espacio normado.

Definición 2.1.4 *Sea V un espacio vectorial definido sobre el conjunto de los números reales* $\mathbb{R}$*. Entonces, una* ***norma*** *en V es una función de V en* $\mathbb{R}$ *que posee las propiedades siguientes (se denotará por* $\|x\|$ *a la norma de x):*

1. $\forall\ x \in\ V : \|x\| \geq 0$
2. $\|x\| = 0 \Leftrightarrow x = \theta$*; en donde* θ *es el vector nulo en* V.
3. $\forall\ x \in\ V, \forall\ \lambda \in \mathbb{R}$: $\|\lambda x\| = |\lambda|\ ||x||$
4. $\forall\ x, y \in V$: $\|x+y\| \leq \|x\| + \|y\|$ *(desigualdad triangular de la norma).*

Definición 2.1.5 *Un* ***espacio normado*** *es un par (V,* $\|\|$*) donde el espacio vectorial V está provisto de una norma.*

A partir de un espacio vectorial V se pueden originar diferentes espacios normados con solo definir diversas normas en el espacio, de forma similar que sucede para el espacio métrico.

Observación. Dadas las definiciones anteriores, es importante señalar que un espacio normado es metrizable; es decir, que en un espacio normado se puede defenir una métrica inducida por la norma. Por la importancia de esta afirmación, se demuestra que un espacio normado es metrizable.

Sea la función

$$d : V \times V \to \mathbb{R}$$

tal que

$$\forall\ x,\ \ y \in V\ \ d(x,y) = \| x - y \|$$

Se quiere verificar que d es realmente una métrica en V. Para ello véase que por las propiedades de una norma se tiene $\| x - y \| \geq 0$ y que la igualdad se cumple para el caso en que $x - y = \theta$, donde θ es el vector nulo, entonces $x = y$ con lo que se verifica que la función d cumple con las primeras dos propiedades de una métrica. Por otro lado, obsérvese, por las propiedades de una norma, que:

$$\begin{aligned} d(x,y) &= \| x - y \| \\ &= \| (-1)(y - x) \| \\ &= | -1 | \| y - x \| \\ &= \| y - x \| \\ &= d(y,x) \end{aligned}$$

entonces d cumple con la propiedad de simetría. Finalmente verifíquese que d también satisface la desigualdad triangular. Si $x, y, z \in V$ entonces

$$\begin{aligned} d(x,y) &= \| x - y \| \\ &= \| (x - z) + (z - y) \| \end{aligned}$$

y por las propiedades de una norma se tiene que

$$\begin{aligned} \| (x - z) + (z - y) \| &\leq \| x - z \| + \| z - y \| \\ &= d(x,z) + d(z,y) \end{aligned}$$

finalmente

$$d(x,y) \leq d(x,z) + d(z,y)$$

por lo tanto se concluye que d es una métrica en V inducida por la norma, entonces el espacio normado es metrizable. ■

En esta sección se estudiaron las definiciones, propiedades y algunos ejemplos de métrica, norma, espacio métrico y espacio normado que permitirán

entender y utilizar los algoritmos que nos ocupan en este trabajo y que se presentan en los siguientes capítulos; además, éstos conceptos son de importancia para la siguiente sección en la que se habla de distancia de un punto a un conjunto y entre dos conjuntos.

2.2 Distancia entre conjuntos

Los conceptos que se estudian en esta sección se requieren para el estudio del algoritmo c-means y c-means difuso. Ya que lo que se busca en estos algoritmos es minimizar la distancia (medida de similitud) entre elementos de un agrupamiento y maximizar la distancia entre los agrupamientos (distancia entre conjuntos).

Dado un espacio métrico (E,d), fíjese un punto $x_0 \in E$ y un conjunto no vacío $A \subset E$. Se designa por $\{d(x_0, x)\}_{x \in A}$ al conjunto de las distancias de x_0 a cada uno de los elementos de A. Obsérvese que el conjunto está acotado inferiormente por 0, ya que por definición de métrica $d(x_0, x) \geq 0 \, \forall \, x \in A$, por lo tanto la mínima distancia posible es cero.

Adóptese la notación

$$d(x_0, A) = \inf\{d(x_0, x)\}_{x \in A}$$

Definición 2.2.1 *La distancia de x_0 al conjunto A está dado por el número $d(x_0, A)$.*

Es importante mencionar que si $x_0 \in A$ entonces $d(x_0, A) = 0$ pero el recíproco no siempre se cumple.

Ejemplo 5

Sea A un conjunto de puntos y d una métrica definidos como:

$$A = \{(1, 2), (2, 5), (-1, 0), (-5, 3), (-2, -1), (4, -2)\}$$

$$d((x_1, y_1), (x_2, y_2)) = \mid x_1 - x_2 \mid + \mid y_1 - y_2 \mid$$

Tómese el punto $(x_0, y_0) = (0, 0)$, entonces

$$d((0,0),A) = \inf\{3,7,1,8,3,6\} = 1$$

Donde $3 = d((0,0),(1,2))$, $7 = d((0,0)(2,5))$ y así sucesivamente para cada punto de A. Por lo tanto, la distancia del punto $(0,0)$ al conjunto A es 1.

Definición 2.2.2 *Dados dos conjuntos no vacíos $A, B \subset E$ desígnese por*

$$d(x,y)_{x\in A,\ y\in B}$$

al conjunto de los números reales constituidos por todas las distancias entre un punto de A a todos los puntos de B y todas las distancias entre un punto de B a todos los puntos de A. Entonces, la ***distancia entre el conjunto A y el conjunto B*** *está definido por el número*

$$d(A,B) = inf\{d(x,y)\}_{x\in A,\ y\in B}$$

Véase que existe la cota inferior del conjunto $d(x,y)_{x\in A,\ y\in B}$ y es más, existe el infimo ya que el valor mínimo posible en este conjunto es cero y resulta cuando se tiene al menos un punto en común entre ambos conjuntos.

Ejemplo 6

Sean A y B dos conjuntos definidos como:

$$A = \{(1,1),(2,2),(3,3),(4,4),(5,5)\}$$

$$B = \{(2,4),(5,1),(6,-3),(0,-7),(1,3)\}$$

y d la métrica,

$$d((x_1,y_1),(x_2,y_2)) = \mid x_1 - x_2 \mid + \mid y_1 - y_2 \mid$$

Para calcular la distancia del conjunto A al B lo primero es tomar un punto de A, $(1,1)$, para encontrar las distancias de este punto a cada uno de los puntos de B. Los resultados son:

$$\{4,4,9,9,2\}$$

Donde el primer valor se refiere a la distancia de $(1,1)$ a $(2,4)$, el segundo valor se obtiene de calcular la distancia de $(1,1)$ a $(5,1)$, y así sucesivamente. De manera tal que último valor corresponde a la distancia del punto $(1,1)$

a $(1, 3)$. Ahora tómese el punto $(2, 2)$, que pertenece a A, y calcúlese la distancia de este punto a cada uno de los puntos de B. Los resultados se presentan ordenados con el mismo criterio que para el punto $(1, 1)$.

$$\{2, 4, 9, 11, 2\}$$

Se hace de forma similar para cada punto de A, teniendo para $(3, 3)$ $\{2, 4, 9, 13, 2\}$; para $(4, 4)$, $\{2, 4, 9, 15, 4\}$; y, para $(5, 5)$, $\{4, 4, 9, 17, 6\}$. Por lo tanto, se concluye que:

$$d(A, B) = inf\{d(x, y)\}_{x\in A,\ y\in B} = 2$$

Lo que indica que la distancia del conjunto A al conjunto B es 2.

Lema 2.2.3 *Si A y B son conjuntos no vacíos en un espacio métrico* (E, d) *se tiene:*

$$d(A, B) = inf\{d(x, B)\}_{x\in A} = inf\{d(y, A)\}_{y\in B}$$

Demostración. Sean A y B conjuntos no vacíos en (E, d) y x cualquier elemento en A, entonces por definición de distancia entre conjuntos se tiene

$$\inf\{d(x, y)\}_{x\in A, y\in B} = d(A, B) \leq d(x, y)\ \ \forall y \in B$$

lo que indica que $d(A, B)$ es cota inferior del conjunto $\{d(x, y)\}_{y\in B}$. Luego

$$d(A, B) \leq d(x, B) = \inf\{d(x, y)\}_{y\in B}$$

para un x en A. Y como x es arbitrario entonces se sigue que d(A,B) es cota inferior del conjunto $\{d(x, B)\}_{x\in A}$, así:

$$d(A, B) \leq \inf\{d(x, B)\}_{x\in A}$$

Luego, existen $x \in A$ y $y \in B$ tales que:

$$d(A, B) \leq d(x, y) \leq d(A, B) + \varepsilon$$

con $\varepsilon > 0$ número real, entonces $d(A, B) + \varepsilon$ no es cota inferior del conjunto

$$\{d(x, B)\}_{x\in A}$$

por lo tanto se tiene:

$$\inf\{d(x,B)\}_{x\in A} < d(A,B) + \varepsilon \Rightarrow \inf\{d(x,B)\}_{x\in A} - d(A,B) < \varepsilon$$

Por otro lado:

$$d(A,B) \leq \inf\{d(x,B)\}_{x\in A} \Rightarrow 0 \leq \inf\{d(x,B)\}_{x\in A} - d(A,B)$$

así

$$0 \leq \inf\{d(x,B)\}_{x\in A} - d(A,B) \leq \varepsilon.....(1)$$

Se asegura que

$$\inf\{d(x,B)\}_{x\in A} - d(A,B) = 0$$

Para justificarlo se supone $\inf\{d(x,B)\}_{x\in A} - d(A,B) = \varepsilon_0$ con $\varepsilon_0 > 0$ número real entonces ya que la expresión (1) se debe cumplir para cualquier $\varepsilon > 0$ número real se toma $\varepsilon_0 = \frac{1}{2}\varepsilon$ con lo que se llega a una contradicción.

$$0 \leq \varepsilon_0 \leq \frac{1}{2}\varepsilon_0 \quad \Rightarrow \quad 0 \leq 2 \leq 1$$

Así que se debe cumplir que

$$\inf\{d(x,B)\}_{x\in A} = d(A,B)$$

■

2.3 Conjuntos Difusos

En esta sección se presentan los conceptos básicos de la teoría de conjuntos difusos. Se estudia la definición de un conjunto difuso y de la función de pertenencia. Además contiene ejemplos de conjuntos difusos y su representación gráfica. Así, una vez que se entienden estos conceptos, se es capaz de entender el algoritmo (c-menas difuso) que permitirá la formación de agrupamientos de este tipo.

2.3.1 Introducción

Dado un conjunto A, se define una función que indique que elementos pertenencen a A y cuales no. En el caso de los conjuntos no difusos (*duros*), se define una función $\mu_A(x)$ la cual asigna el valor de uno a aquellos elementos que pertenecen al conjunto A y 0 para aquellos que no.

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & en\ otro\ caso \end{cases}$$

Sin embargo, dicha función puede generalizarse de forma que indique en que grado un elemento pertenece al conjunto dado, a esa función se le llama *función de pertenencia* y al conjunto que la tiene asociada se le conoce como *conjunto difuso*.

Lo más común es utilizar el rango $[0, 1]$ para la función de pertenencia. Si $\mu_A(x) = 0$ significa que el elemento no pertenece al conjunto(o no se "parece" a los elementos de ese conjunto) y si $\mu_A(x) = 1$ indica que el elemento pertenece al conjunto dado unicamente. La notación más utilizada en los textos para indicar la función de pertencia es:

$$\mu_A : X \to [0, 1]$$

donde X es un conjunto difuso. Cada conjunto difuso está definido por una función de pertenencia única. Para entender mejor los conceptos anteriores veáse el siguiente ejemplo.

Nuestro conjunto universo consiste de siete niveles de educación:

0-No-educación

1-estudios elementales

2-estudios de preparatoria

3-dos años de estudios en Universidad

4-Título de Licenciatura

5-Título de Maestría

6-Título Doctoral

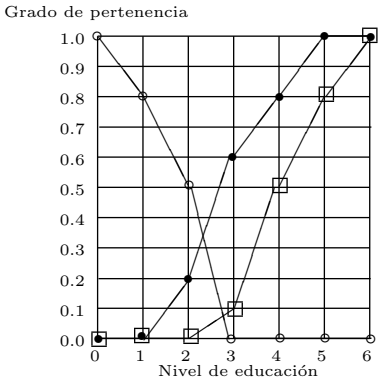
En el conjunto universo se definen tres conjuntos difusos:

A_1 : personas con nivel bajo de educación

A_2 : personas con nivel alto de educación

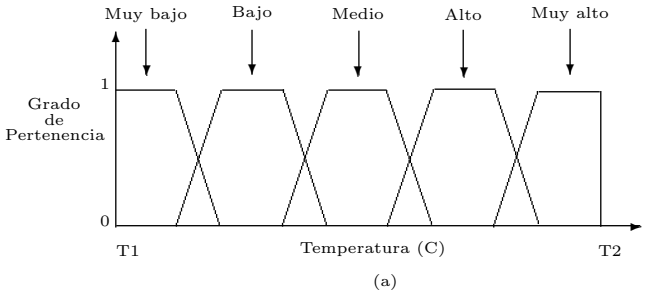
A_3 : personas con nivel muy alto de educación

las funciones de pertenencia correspondientes se definen en la figura y se representan por los símbolos ∘, • y □, respectivamente.



Por ejemplo, una persona que tiene Título de Licenciatura (4) tiene un grado de pertenencia de 0.5 en A_3 (nivel muy alto de educación) y 0.8 de pertenencia a A_2 (Nivel alto de educación). Mientras que una persona con estudios de preparatoria (2) tienen un grado de pertenencia de 0.5 a A_1, 0.2 a A_2 y 0 a A_3. El que tenga 0 grado de pertenencia al conjunto de personas con muy alto nivel de educación indica que no puede considerarse una persona muy preparada academicamente. Sin embargo, muy difícilmente se puede decir que es una persona con un nivel alto de educación. Su mejor clasificación es en el conjunto de personas con nivel bajo de educación.

En muchos conjuntos difusos se utilizan los conceptos de bajo, medio, alto, etc. para definir el estado de las variables. Estas son llamadas *variables difusas*. Observe la siguiente figura:



En la figura se utiliza una variable difusa de temperatura en un rango $[T_1, T_2]$. Los estados de la variable difusa son conjuntos difusos representados por cinco conceptos lingüísticos: muy baja, baja, media, alta y muy alta. Estos conjuntos son definidos por funciones de pertenencia de la forma

$$[T_1, T_2] \rightarrow [0, 1]$$

Se considera que las variable difusas se ajustan mejor a la realidad que las variables no difusas, a pesar de que las primeras se miden con incertidumbre. Las variables difusas proporcionan evidencia más exacta sobre los fenómenos naturales que las no difusas. Esta idea la expresa mejor lo dicho por Albert Einstein en 1921: *So far as law of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.*

Definición 2.3.1 *Dado un conjunto universo, X, un conjunto difuso ordinario se define por una función de la forma*

$$\mu_A : X \rightarrow [0, 1]$$

Los conjuntos difusos requieren que a cada uno de sus elementos le sea asignado un número real particular; sin embargo, en ocaciones no resulta tan simple definir una función de pertenencia de forma exacta y sólo se podrán hacer aproximaciones. Tal vez sólo se pueda establer límites entre los que se encuentra el grado de pertenencia de los elementos del conjunto universo. Los

conjuntos difusos definidos por este tipo de funciones son llamados *conjuntos difusos evaluados en un intervalo.*

Definición 2.3.2 *Dado un conjunto universo X, un conjunto difuso evaluado en un intervalo es definido por una función de pertenencia de la forma:*

$$\mu_A : X \to \epsilon([0,1])$$

donde $\epsilon([0,1])$ *denota la familia de todos los intervalos cerrados de números reales en* $[0,1]$

2.3.2 Tipos básicos

En esta parte de la sección se darán los conceptos más importantes en la teoría de conjuntos difusos.

Definición 2.3.3 *Dado un conjunto difuso A definido en X, conjunto universo, y un número* $\alpha \in [0,1]$ *se dice que un* α*-corte es un conjunto no difuso que está dada por:*

$$^{\alpha}A = \{x \mid A(x) \geq \alpha\}$$

Cuando se tiene la desigualdad estrictamente se dice que se tiene un α*-corte fuerte y se designa por:*

$$^{\alpha+}A = \{x \mid A(x) > \alpha\}$$

De la definición anterior se dice que $^{\alpha}A$ es un conjunto no difuso que contiene a los elementos del universo X que tienen grado de pertenencia en A mayor o igual a α.
Observación. Se dice que el conjunto $^{\alpha}A$ es no difuso ya que los elementos pertenecen o no a éste. Es decir, el conjunto $^{\alpha}A$ puede expresarse como:

$$\mu_{^{\alpha}A} = \begin{cases} 1, & si \ \ A(x) \geq \alpha \\ 0, & en \ \ otro \ \ caso \end{cases}$$

Se tiene un conjunto de personas a las cuales se desea clasificar dentro de un conjunto según la edad de cada una de ellas. Las funciones de pertenencia

correspondiente a los conjuntos difusos son representados por las siguientes funciones:

$$\mu_{\mathrm{A}_1}(x) = \begin{cases} 1 & \text{si } x \leq 20 \\ (35-x)/15 & \text{si } 20 < x < 35 \\ 0 & \text{si } x \geq 35 \end{cases}$$

$$\mu_{\mathrm{A}_2}(x) = \begin{cases} 0 & \text{si } x \leq 20 \text{ o } \geq 60 \\ (x-20)/15 & \text{si } 20 < x < 35 \\ (60-x)/15 & \text{si } 45 < x < 60 \\ 1 & \text{si } 35 \leq x \leq 45 \end{cases}$$

$$\mu_{\mathrm{A}_3}(x) = \begin{cases} 0 & \text{si } x \leq 45 \\ (x-45)/15 & \text{si } 45 < x < 60 \\ 1 & \text{si } x \geq 60 \end{cases}$$

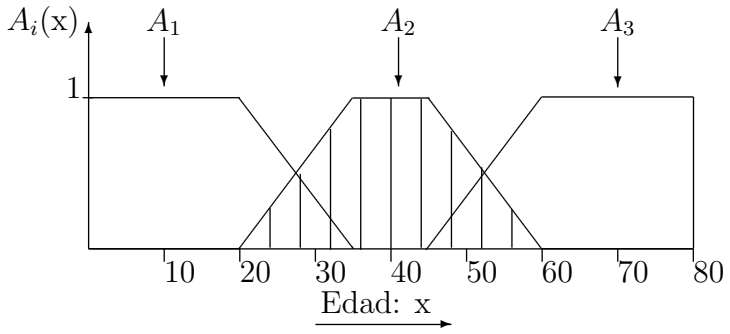
donde:

A_1: Joven

A_2: Mediana edad

A_3: Edad avanzada

Las gráficas correspondientes a las funciones de pertenencia se muestran en la siguiente figura:



Así, personas de entre 20 y 35 años pueden considerarse como personas jóvenes y, también, como personas de mediana edad. En el rango de 45 a 60 años las personas son consideradas de mediana y avanzada edad. Pero en los casos anteriores, el grado de pertenencia de un conjunto al otro es

diferente. Por ejemplo, una persona de 25 años tiene un grado de pertencia al conjunto de personas jóvenes de aproximadamente 0.667 y al conjunto de personas de edad mediana de 0.333. Así que si tuviera que colocarse a esta persona sólo en un conjunto, este sería al de personas jóvenes. En otros casos, como en el de personas de 10 años y el de personas de 70 años de edad, los elementos pertenencen únicamente a un conjunto, en este caso al de jóvenes y al de personas de avanzada edad, respectivamente. No pueden considerarse personas de mediana edad.

Lo que cumplen los elementos (las personas de edad de 0 a 80 años) es que pertenencen al menos a un conjunto. De la misma forma, los conjuntos cumplen que al menos contienen un elemento.

$$^{0}A_1 =^{0} A_2 =^{0} A_3 = X$$
$$^{1+}A_1 =^{1+} A_2 =^{1+} A_3 = \emptyset$$

En la primera línea se tiene que todos los elementos de A_1, A_2 y A_3 tienen grado de pertenencia mayor o igual a 0; es decir, cada elemento pertenece al menos a un conjunto. Y en la segunda línea se expresa que ningún elemento tiene grado de pertenencia mayor que uno. Ahora, para calcular $^{\alpha}A_1$, para cualquier α, se tiene:

$$(35 - x)/15 \geq \alpha \Rightarrow \quad 35 - x \geq 15\alpha \Rightarrow 35 - 15\alpha \geq x$$
$$^{\alpha}A_1 = [0, 35 - 15\alpha]$$

de la misma forma

$$^{\alpha}A_2 = [15\alpha + 20, 60 - 15\alpha]$$
$$^{\alpha}A_3 = [15\alpha + 45, 80]$$

para $\alpha \in (0, 1]$. Y, para $\alpha \in [0, 1)$ se tiene

$$\begin{aligned} ^{\alpha+}A_1 &= (0, 35 - 15\alpha) \\ ^{\alpha+}A_2 &= (15\alpha + 20, 60 - 15\alpha) \\ ^{\alpha+}A_3 &= (15\alpha + 45, 80) \end{aligned}$$

Definición 2.3.4 *Dado un conjunto difuso A, se llama conjunto de nivel de A al conjunto de los diferentes valores de α que representan distintos α-cortes tal que $\alpha \in [0, 1]$ y se escribe como:*

$$\Lambda(A) = \{\alpha | A(x) = \alpha \,\text{para algún}\ \ x \in X\}$$

Para el ejemplo anterior se tiene:

$$\Lambda(A_1) = \Lambda(A_2) = \Lambda(A_3) = [0, 1]$$

lo cual indica que para cada $\alpha \in [0, 1]$ existe al menos un elemento x con grado de pertenecia igual a α en el conjunto A_i, con $i = 1, 2, 3$.

Obsérvese que los conjuntos α-corte y α-corte fuerte cumplen con dos propiedades muy importantes. Para un conjunto difuso A dado, ${}^{\alpha_1}A$ y ${}^{\alpha_2}A$, con α_1, $\alpha_2 \in [0, 1]$ y $\alpha_1 < \alpha_2$ se tiene:

$${}^{\alpha_1}A \supseteq {}^{\alpha_2}A$$

y

$${}^{\alpha_1+}A \supseteq {}^{\alpha_2+}A$$

esto implica

$${}^{\alpha_1}A \bigcap {}^{\alpha_2}A = {}^{\alpha_2}A, \ \ {}^{\alpha_1}A \bigcup {}^{\alpha_2}A = {}^{\alpha_1}A$$

y a su vez:

$${}^{\alpha_1+}A \bigcap {}^{\alpha_2+}A = {}^{\alpha_2+}A, \ \ {}^{\alpha_1+}A \bigcup {}^{\alpha_2+}A = {}^{\alpha_1+}A$$

Véase que lo anterior origina dos familias distintas de conjuntos no difusos anidados.

Ejemplo 7

Considere la función A_1 del ejemplo anterior y tome $\alpha_1 < \alpha_2$, con $\alpha_1 = 0.5$ y $\alpha_2 = 0.8$. Así,

$${}^{\alpha_2}A_1 = [0, 35 - 15(0.8)] = [0, 23]$$
$${}^{\alpha_1}A_1 = [0, 35 - 15(0.5)] = [0, 27.5]$$

Entonces, las personas con edad de 0 a 23 años pertenecen al conjunto ${}^{\alpha_2}A_1$, mientras que las personas de edad de 0 a 27.5 años pertenecen al conjunto ${}^{\alpha_1}A_1$. Así,

$$^{\alpha_1}A_1 \bigcap {}^{\alpha_2}A_1 = {}^{\alpha_2}A_1 = [0, 23]$$
$$^{\alpha_1}A_1 \bigcup {}^{\alpha_2}A_1 = {}^{\alpha_1}A_1 = [0, 27.5]$$

Un caso particular de los conjuntos α-corte fuerte es cuando $\alpha = 0$. El **soporte de un conjunto difuso A** en un conjunto universo X es el conjunto no difuso que contiene todos los elementos de X con grado de pertenencia en A diferente de cero, y se expresa como:

$$S(A) = \sup(A) = {}^{0+}A$$

Al conjunto ${}^{1}A$ se le llama **núcleo de A**. La **altura del conjunto difuso A**, denotado por $h(A)$, es el grado de pertenencia más grande obtenido para cualquier elemento en A:

$$h(A) = \sup_{x \in X} (A(x))$$

La altura de A, *h(A)*, también puede ser vista como el valor máximo de α tal que ${}^{\alpha}A \neq \emptyset$. Si $h(A) = 1$ se dice que el conjunto difuso A es *normal*, mientras que si $h(A) < 1$ se dice que es *subnormal*.

Otra propiedad importante de los conjuntos difusos es la *convexidad*. Esto es una generalización de la convexidad en los conjuntos no difusos. Se requiere que α-corte de un conjunto difuso convexo sea convexo para todo $\alpha \in (0, 1]$. Si todos los α-cortes, $\alpha > 0$, son convexos entonces se dice que el conjunto difuso correspondiente es convexo.

Observe que si un conjunto difuso es convexo no implica que la función de pertenencia, correspondiente al conjunto, sea convexa. De hecho, la función de pertenencia de un conjunto difuso convexo no es cóncava ni convexa.

Definición 2.3.5 *Un conjunto difuso A se dice convexo si para todo* $\alpha \in (0, 1]$, $x_1, x_2 \in {}^{\alpha}A$ *y* $0 \leq \lambda \leq 1$ *se tiene*

$$\lambda x_1 + (1 - \lambda)x_2 \in {}^{\alpha}A$$

Observación. Si A es no difuso entonces:

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

Por lo tanto para cada $\alpha \in (0,1]$, ${}^{\alpha}A = \{x|A(x) \geq \alpha\} = A$ de donde la definición de conjunto convexo difuso coincide con la definición usual de convexo cuando A es no difuso. Así que otra forma de definir un conjunto difuso convexo es la siguiente:

Definición 2.3.6 *Un conjunto difuso A es convexo si* $\forall\alpha \in (0,1]$ ${}^{\alpha}A$ *es convexo.*

Teorema 2.3.7 *Un conjunto difuso* $A \in \mathbb{R}$ *es convexo si y sólo si*

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \textit{mín}[A(x_1), A(x_2)]$$

para todo x_1, $x_2 \in \mathbb{R}$ *y todo* $\lambda \in [0,1]$.

Demostración.

$\Rightarrow$) Sea A un conjunto difuso convexo, x_1, $x_2 \in {}^{\alpha}A$, con $\alpha \in (0,1]$, y $0 \leq \lambda \leq 1$.

Ahora, se define

$$\alpha_0 = \text{máx}\{\alpha \mid x_1, x_2 \in {}^{\alpha}A\}$$

entonces, por definición de convexidad,

$$\lambda x_1 + (1-\lambda)x_2 \in {}^{\alpha_0}A$$

$$\Rightarrow \quad A(\lambda x_1 + (1-\lambda)x_2) \geq \alpha_0 \geq \text{mín}\{A(x_1), A(x_2)\}$$

Luego,

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \text{mín}\{A(x_1), A(x_2)\}$$

$\Leftarrow$) Suponga

$$A(\lambda x_1 + (1-\lambda)x_2) \geq \text{mín}\{A(x_1), A(x_2)\}$$

con x_1, $x_2 \in {}^{\alpha}A$ y $\alpha \in [0,1]$, entonces: $A(x_1) \geq \alpha$ y $A(x_2) \geq \alpha$, así

$$\begin{aligned} &min\{A(x_1), A(x_2)\} \geq \alpha \\ &\Rightarrow A(\lambda x_1 + (1-\lambda)x_2) \geq \alpha \\ &\Rightarrow \lambda x_1 + (1-\lambda)x_2 \in {}^{\alpha}A \end{aligned}$$

■

2.4 Planteamiento formal de un problema de CSA.

En esta sección se estudian los elementos de un clasificación sin aprendizaje.

Dadas las descripciones $I(O_j) = (x_1(O_j), ..., x_n(O_j))$ de los objetos $O_1, ..., O_m$ de un universo dado U. Cada x_i tiene asociado un conjunto de valores admisibles $M_i, i = 1, ..., n$. Sobre M_i se define un criterio de comparación de valores de dicha variable $C_i : M_i \times M_i \rightarrow \Delta$, donde Δ es un conjunto dado que pudiera ser $\{0, 1\}, \{0, 1, ..\}$, un subconjunto de $\mathbb{R}$, etc. En dependencia de lo cual los criterios de comparación recibirán diferentes denominaciones. Entre las descripciones de los objetos se define una función de semejanza

$$\beta : \prod_{i=1}^{n} M_i \times \prod_{i=1}^{n} M_i \longrightarrow \Delta$$

en donde

$$\prod_{i=1}^{n} M_i$$

denota el producto cartesiano de los conjuntos de valores admisibles de los rasgos indicados y Δ es como antes.

El problema de la clasificación sin aprendizaje, desde el punto de vista formal, consiste en determinar un criterio de agrupamiento (agrupacional) $\prod$, tal que se puede obtener la estructura interna del conjunto de objetos U, las relaciones entre los objetos y las agrupaciones de los objetos. Es inmediato que el criterio de agrupamiento buscado $\prod$ tendrá que ser una función de la función de semejanza β. Cuando este criterio $\prod$ y la medida de similaridad entre los objetos β, son obtenidos por medio de un proceso de modelación matemática se tiene una garantía o al menos una mayor certeza de que la estructuración obtenida "es natural". En la medida que esos parámetros del problema reflejen adecuadamente la realidad que pretenden modelar, en esa medida pudiéramos hablar de "una estructuración natural" del universo de objetos sujeto a estudio.

Resumiendo pudiera decirse que en un problema de clasificación sin aprendizaje los tres elementos esenciales lo constituyen:

- El espacio de representación de los objetos (ERI)

- La medida de similaridad (β, función de semejanza, no necesariamente una distancia) y
- El criterio de agrupamiento $\prod$

Es decir, la manera en que será utilizada la similaridad para la solución del problema planteado. En otras palabras, resolver un problema de clasificación sin aprendizaje (no supervisada) consiste en hallar un algoritmo

$$A(U, \beta, \prod)$$

En términos generales, se puede considerar que existen dos tendencias fundamentales en la solución de problemas de clasificación sin aprendizaje: una basada en la función de semejanza, que denominaremos ***clasificatoria***, cuyo objetivo central consiste en encontrar los elementos que, dadas sus relaciones de semejanza, deben estar en un mismo agrupamiento y una segunda, que denominamos ***conjuntual***, cuyo interés básico es hallar la determinación intencional de los conjuntos que formarán la estructuración final.

En el primer caso se tiene un universo de objetos y es necesario agruparlos de modo tal que los objetos que estén en el mismo agrupamiento se parezcan (se asemejen) más entre sí que con objetos de otros agrupamientos. En el segundo caso, tenemos un universo de objetos y necesitamos formar agrupamientos de modo tal que los objetos que estén en el mismo agrupamiento cumplan (satisfagan en cierto grado equivalente entre ellos) la propiedad que caracteriza al agrupamiento como conjunto, en su determinación intencional.

Las técnicas para la solución del problema por una u otra vía son variadas y en esta tesis sólo se abordaran algunas de ellas. Es natural pensar que las mismas estarán en fuerte dependencia de las características del problema, en especial del espacio de representación inicial de los objetos sujetos a estudio.

2.5 El espacio de representación inicial (ERI)

Como se vio en la sección anterior, el espacio de representación inicial es un elemento esencial de la Clasificación sin aprendizaje (CSA). En esta sección se describe con mayor detalle en que consiste y bajo que enfoques se determina el ERI. Éste puede ser

$$(\mathbb{R}^n, 2^n \text{ y } \ E_1 \times ... \times E_n)$$

En el proceso de modelación matemática debe obtenerse la información necesaria para poder determinar las características del ERI. Esto suele violentarse, el área de aplicación de una técnica estará determinada por las características del ERI.

Desde el punto de vista de las herramientas matemáticas que se emplean para la solución de un problema de clasificación sin aprendizaje y de los supuestos que las mismas conllevan, existen diferentes enfoques.

En el enfoque *estadístico* se asume una distribución de los valores, se suponen características sobre el ERI y se aplican las técnicas. Por lo general nos encontraremos ERI's que están definidos sobre $\mathbb{R}^n$, el espacio de los números reales (n-dimensional) o el 2^n, el booleano n-dimensional que consiste en un vector de n entradas, donde cada entrada puede ser 1 ó 0. La entrada i con valor 0 indica que el objeto correspondiente no cumple con la propiedad i-ésima, pero si el valor es 1 entonces cumple con la característica. Ambos proveerán al problema de la posibilidad de definir una métrica sobre el mismo, por lo que las ideas centrales se moverán en el entorno de agrupar a los objetos en la medida de la distancia a la que éstos se encuentran unos de otros, siguiendo la idea básica de que los objetos de un mismo agrupamiento estarán más cerca entre sí que lo que están respecto a otros en agrupamientos diferentes. Esta idea estará presente también en otros enfoques.

En el enfoque *lógico combinatorio*, la filosofía es totalmente a la inversa, no se partirá de asumirle al ERI propiedades que no sean aquellas que han sido formalizadas a partir de un proceso de modelación matemática. Por lo general en este enfoque nos encontraremos con ERI que son simplemente productos cartesianos de los conjuntos de valores admisibles de las variables en términos de las cuales se describen todos los objetos.

Para las técnicas basadas en el enfoque estadístico la forma más obvia de medir la similitud o divergencia entre dos muestras es la distancia entre ambos conjuntos, análogamente entre objetos.

2.6 Medidas de (divergencia) similitud

El segundo elemento para la CSA es la función de similaridad. En el algoritmo el c-means la función busca minimizar la distancia entre elementos del mismo agrupamiento.

Veamos ejemplos y algunas características de las medidas de similaridad más usadas en este enfoque:

Sean dadas las descripciones de dos objetos

$$I(O_i) = (x_1(O_i), ..., x_n(O_i))$$

y

$$I(O_j) = (x_1(O_j), ..., x_n(O_j))$$

Sea D^2 una función que denominaremos ***divergencia***

$$D^2 : \mathbb{R}^n \to \mathbb{R}$$

tal que

i) Será positiva para dos objetos distintos, es decir,

$$D^2(I(O), I(O')) > 0$$

ii) La divergencia de un objeto con sigo mismo es nula, esto es

$$D^2(I(O), I(O)) = 0$$

iii) No debe estar afectada por la denominación de los objetos,

$$D^2(I(O), I(O')) = D^2(I(O'), I(O))$$

iv) La divergencia debe ser aditiva para características independientes, esto es

$$D^2(I(O), I(O')) = \sum_{i=1}^{n} D^2(x_i(O), x_i(O'))$$

suponiendo que los rasgos son independientes.

v) $D^2(I(O), I(O'))$ no debe crecer al aumentar el número de rasgos.

vi) Debe ser invariante ante traslaciones y rotaciones.

vii) Debe tener en cuenta la interdependencia de los rasgos en el agrupamiento en que se hayan ubicado los objetos

viii) Debe ser sensible a las diferentes ponderaciones a introducir en la cuantificación de la divergencia de cada característica, de acuerdo con su mayor o menor importancia informacional en la discriminación de los objetos

1. Distancia Euclideana

$$D^2(I(O), I(O')) = \sum_{i=1}^{n} (x_i(O) - x_i(O')^2$$

Esta D^2 no satisface las propiedades **(vi)** ni **(vii)**

2. Distancia Euclideana normalizada

$$D^2(I(O), I(O')) = \sum_{i=1}^{n} ((x_i(O) - x_i(O'))S_i)^2$$

siendo S_i la desviación típica de x_i.

3. Distancia de Mahalanobis

$$D^2(I(O), I(O')) = \frac{(x_1(O), ..., x_n(O))^t \cdot (x_1(O'), ..., x_n(O'))}{I(O)^t \cdot I(O) + I(O')^t I(O) - I(O)^t I(O')}$$

siendo $I(O) = (x_1(O), ..., x_n(O))$ Supongamos que las variables son booleanas y denotemos por:

$\eta_{ij}^{(1,1)}$: el número de unos coincidentes entre los objetos O_i, O_j.

$\eta_{ij}^{(0,0)}$: el número de ceros coincidentes entre los objetos O_i, O_j.

$\eta_{ij}^{(1,0)}$: el número de unos en O_i coincidentes con ceros de O_j.

$\eta_{ij}^{(0,1)}$: el número de ceros en O_i coincidentes con unos de O_j.

ρ_{ij}: el número de rasgos coincidentes entre O_i y O_j, es decir

$$\rho_{ij} = \eta_{ij}^{(1,1)} + \eta_{ij}^{(0,0)}$$

θ_{ij}: el número de rasgos **no** coincidentes entre O_i y O_j, es decir

$$\theta_{ij} = \eta_{ij}^{(0,1)} + \eta_{ij}^{(1,0)}$$

n : el número total de variables en términos de las cuales se describen los objetos

$\eta_i^{(1)}$, $\eta_j^{(1)}$: el número de unos en O_i (respectivamente en O_j)

$\eta_i^{(0)}$, $\eta_j^{(1)}$: el número de ceros en O_i (respectivamente en O_j)

4. Promedio de rasgos coincidentes:

$$S_{ij} = \frac{\rho_{ij}}{m}$$

siendo $0 \leq S_{ij} \leq 1$. Se aplica cuando el cero y el uno aportan información.

5. Coeficiente de Rao:

$$S_{ij} = \frac{\eta_{ij}^{(1,1)}}{m}$$

siendo $0 \leq S_{ij} \leq 1$

6. Coeficiente Hamman:

$$S_{ij} = \frac{\rho_{ij} - \theta_{ij}}{m}$$

se aplica cuando $\rho ij \geq \theta_{ij}$, esto es, $S_{ij} \geq 0$

7. Coeficiente de Rodgers y Tanimoto:

$$S_{ij} = \frac{\eta_{ij}^{(1,1)}}{\eta_i^{(1)}\eta_j^{(1)} - \eta_{ij}^{(1,1)}}$$

siendo $0 \leq S_{ij} \leq 1$

8. Coeficiente de Gekard:

$$S_{ij} = \frac{\eta_{ij}^{(1,1)}}{\eta_{ij}^{(1,1)} - \theta_{ij}}$$

siendo $0 \leq S_{ij} \leq 1$

9. Coeficiente Dake

$$S_{ij} = \frac{2\eta_{ij}^{(1,1)}}{2\eta_{ij}^{(1,1)} + \theta_{ij}}$$

siendo $0 \leq S_{ij} \leq 1$

10.

$$S_{ij} = \frac{\eta_{ij}^{(1,1)}}{\eta_{ij}^{(1,1)} + 2\theta_{ij}}$$

siendo $0 \leq S_{ij} \leq 1$

11.

$$S_{ij} = \frac{\rho_{ij}}{2m - \theta_{ij}} = \frac{\rho_{ij}}{m + \theta_{ij}}$$

Nota: Todos los coeficientes de semejanza (para variables booleanas) pueden ser extendidos al caso de variables cualesquiera, con la única restricción que todos los criterios de comparación de valores de las variables sean booleanos.

Capítulo 3

El caso estadístico y el algoritmo c-means.

En este capítulo se habla del enfoque estadístico del Espacio de Representación Inicial (ERI). Este enfoque posee características importantes y que resultan atractivas para muchas aplicaciones. A pesar de que los algoritmos que se estudiarán en este texto no tienen un ERI bajo el enfoque estadístico (el ERI de los dos algoritmos es en base al enfoque lógico combinatorio) se estudia éste con mayor profundidad en la primera parte de este capítulo.

En el enfoque estadístico, para el ERI, ante un problema de clasificación sin aprendizaje, en la mayoría de las situaciones, se asume una distribución dada de los datos y posteriormente se aplican las técnicas adecuadas para estas suposiciones. Por ejemplo, si los objetos se distribuyen en el espacio de representación (que se asume normado) según una única distribución normal, entonces *"lo máximo que se puede obtener de los datos está contenido en los parámetros que definen la distribución del vector de los promedios y la matriz de covarianzas de la muestra"* afirma Laureano Escudero [2]. Y continua *"El vector de los promedios localiza el centro de gravedad de la nube de puntos. Puede conside rarse como el patrón prototipo X que mejor representa a todos los patrones de la muestra, en el sentido de minimizar la suma de los cuadrados de las distancias de los demás patrones con respecto a si mismo. La matriz de covarianzas de la muestra indica el grado de representatividad con que el vector de los promedios representa al conjunto de patrones que la componen"*. Aunque es de señalar que el patrón pudiera no existir, es decir, puede no ser un elemento de la muestra dada (el universo U). *"Si los patrones de la nube de objetos siguen una distribución normal, el vector de*

los promedios tiende a encuadrarse en la región en que la muestra está más densamente concentrada. Si la muestra no está normalmente distribuida, promedio y covarianzas pueden dar una errónea distribución de los datos".

Por otro lado debemos de añadir que el vector de los promedios y la matriz de covarianza son insuficientes para describir totalmente un conjunto arbitrario de datos.

Volviendo sobre el concepto de "naturalidad", Laureano Escudero dice *"Para llegar a agrupar los patrones en clases "naturales" el análisis cluster utiliza el criterio de minimizar la desviación interna de los patrones de un mismo grupo y por lo tanto maximizar la distancia entre los diversos grupos"*, lo que es sin lugar a dudas una de las ideas intuitivamente más inmediata ante un problema de clasificación sin aprendizaje.

Es muy frecuente en este enfoque plantear las cosas del siguiente modo:

ERI $= \mathbb{R}^n$; β es igual distancia sobre $\mathbb{R}^n$; y como criterio de agrupamiento condiciones de optimalidad sobre la base de la distancia definida.

En ocasiones ERI $= 2^n$, es decir, el booleano n-dimensional sobre el que se define alguna distancia. Esto bajo el supuesto de que todas las variables que describen a los objetos son booleanas.

Estos planteamientos se sostienen sobre la base de la afirmación "la forma más obvia de medir la similitud o divergencia entre dos muestras es la distancia entre ambos conjuntos."

En la sección siguiente se presentan diversas estrategias del enfoque estadístico para formación de agrupamientos.

3.1 Estrategias de agrupamiento

Con base en lo expuesto en la sección anterior, se puede decir que en el enfoque estadístico se consideran las siguientes estrategias para la formación de agrupamientos:

1. **Métodos de reagrupamiento y jerárquicos** . Un método de reagrupamiento es aquel en el que habiendo determinado el número **c** de agrupamientos a formar, el problema consiste en distribuir los objetos de universo en ciertos **c** agrupamientos de tal forma que se maximice alguna medida de similitud (prefijada) entre los objetos del mismo agrupamiento.

Los métodos jerárquicos tienen por objetivo agrupar a agrupamientos (cada agrupamiento de objetos se trata como si fuera un objeto de un nuevo universo) para formar uno nuevo , aquí se tiende hacia un nivel de mayor generalización o bien separar agrupamientos formando nuevos agrupamientos llamados subagrupamientos, es decir, aquí se tiende hacia un nivel de mayor particularización.

2. **Métodos aglomerativos y divisivos.** Los métodos jerárquicos se subdividen en aglomerativos (en los que se parte de **c** agrupamientos hasta llegar al universo, en cada nivel se procede a fusionar agrupamientos que sean similares (este concepto hay que definirlo en cada caso)).

   Los métodos divisivos son aquellos en los que se parte del universo y se van obteniendo en cada nivel dos agrupamientos repartidos de modo tal que se maximice una medida de divergencia preestablecida.

3. **Método tipológico.** Los métodos tipológicos, aunque también son jerárquicos, se diferencian de éstos en que los jerárquicos contemplan simultáneamente las n características de cada objeto para formar los agrupamientos o separarlos en nuevos agrupamientos, según se observe una mayor similitud o divergencia, respectivamente. En cambio los métodos tipológicos tienen en cuenta una característica en la agrupación o separación de los objetos, de tal forma que si el método es, por ejemplo, tipológico divisivo se comienza a separar los objetos seleccionados según la característica de mayor efecto discriminante (¿cómo se puede lograr esto?). Una vez formados los dos nuevos agrupamientos de objetos se estudia, independientemente a cada uno de ellos, los dos nuevos estratos de objetos que se pueden formar con la influencia de la característica que más discrimina. Éstas se obtienen sobre la base de maximizar la homogeneidad interna (entre los objetos del mismo agrupamiento) y por tanto maximizar la heterogeneidad entre los estratos. Cuando no se puede seguir subdividiendo con la variable más discriminante, se continúa el proceso con la siguiente variable en orden de importancia discriminatorio y así sucesivamente.

   Si atendemos a las relaciones entre los conjuntos que se forman en la estructuración del universo:

4. **Métodos generadores de agrupamientos solapados y ajenos.** En los métodos que generan agrupamientos solapados se admite que

un objeto pueda formar parte simultáneamente de más de un agrupamiento. Por su parte, en los generadores de agrupamientos ajenos (exclusivos) si un objeto pertenece a un agrupamiento no puede pertenecer a otro. Es valioso apuntar aquí, aunque después retomemos esta idea, que en primer caso la propiedad que caracteriza a los agrupamientos (considerados como conjuntos) no excluye a otras, mientras que en el caso de los agrupamientos exclusivos sí ocurre.

Análogamente, si se toma en cuenta la naturaleza de dichos conjuntos, es decir si son conjuntos duros (de la Teoría Clásica De Conjuntos) o difusos (Teoría de los Subconjuntos Difusos), aparecerían nuevas subdivisiones para los métodos generadores de agrupamientos difusos considerando los diferentes conceptos de cubrimientos y particiones difusas.

Considerando la forma en que se obtienen dichos agrupamientos:

5. **Métodos directos e iterativos.** Los métodos directos se caracterizan por utilizar algoritmos que operan de modo tal que una vez que se asigna un objeto a un agrupamiento no lo remueve del mismo. Los iterativos, por su parte, corrigen sus propias asignaciones sobre la base de volver a comprobar en subsiguientes iteraciones si la asignación de la muestra total es óptima. Si no lo fuese estos métodos efectúan un nuevo agrupamiento.

6. **Métodos secuenciales y simultáneos.** Los métodos secuenciales se caracterizan por aplicar la misma serie recursiva de operaciones a cada agrupamiento (por ejemplo los métodos de reagrupamiento). Mientras, los simultáneos son aquellos que de una vez provocan los agrupamientos de una muestra.

7. **Métodos adaptativos y no adaptativos.** Se trata de métodos que "aprenden", en su ejecución, de la conformación de los agrupamientos que están fusionando de manera tal que, según sea la fusión, cambian de medida de similitud (divergencia) o del criterio a optimizar.

   Los métodos no adaptativos (que son mayoría) son aquellos en que el algoritmo se encamina directa o iterativamente a la solución sin variar los parámetros esenciales del método, el cual está predeterminado.

La estructuración de los métodos de agrupamiento antes expuesta no pretende otra cosa que no sea dar una idea global de las formas en que puede

ser abordada la solución del problema que nos ocupa. Solo referiré a una de esas familias y dentro de ellas a un procedimiento específico, que contiene en esencia una de las idea más importantes para la realización de agrupamientos, para la solución del problema de clasificación sin aprendizaje.

3.2 Técnicas de reagrupamiento

Sobre la base de conceptos de similaridad como los ejemplificados a través de las medidas de divergencia anteriores, se realiza el proceso de agrupamiento del universo en cuestión siguiendo los pasos siguientes:

a) Dado un universo U, se conoce el número c de agrupamientos que se desean obtener, en los que U debe estructurarse, de modo tal que se optimice un criterio.

b) Los criterios por lo general estarán dirigidos a optimizar una cierta medida del grado de homogeneidad interna de los agrupamientos o de heterogeneidad entre los mismos.

Ejemplo 8 *Ejemplos de estos criterios pueden ser:*

$$J_c = t_r(S_B) = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \|I(O_j^i) - m_i\|^n$$

Donde m_i es el vector de los n promedios de los valores de las variables en el i-ésimo agrupamiento; n_i la cantidad de objetos en dicho agrupamiento; t_r denota la traza de la matriz, es decir, la suma de los elementos de su diagonal y S_B esta dado por

$$S_B = \sum_{i=1}^{c} n_i(m_i - m)(m_i - m)^t$$

que es una matriz que recoge la suma ponderada de las desviaciones de cada agrupamiento respecto al objeto prototipo, promedio m de la muestra total.

Si bien esta medida es simple, en particular por su cálculo, y por ser invariante ante transformaciones lineales (cambio de escala) siempre y cuando se normalicen las variables, tiene el inconveniente de que no es sensible a la

relación de dependencia de las variables. Lo que sí se tiene en cuenta en el siguiente ejemplo.

$$J_M = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (I(O_j) - m_i)^t S^{(i)^{-1}} (I(O_j) - m_i)$$

que se conoce con el nombre de suma de las desviaciones de Mahalanobis, donde

$$S^{(i)} = \sum_{j=1}^{n_i} (I(O_j^i) - m_i)(I(O_j^i) - m_i)^t$$

es la matriz asociada al agrupamiento i-ésimo que nos da una medida de la variación de las descripciones de los objetos del i-ésimo agrupamiento respecto al vector de los promedios del mismo agrupamiento.

Una vez determinado el criterio a optimizar, se procede de manera iterativa; en el caso que, por ejemplo las variables no tienen mayor relación de dependencia, se trabaja en la optimización del criterio J_c, si por el contrario los objetos estarán correlacionados, entonces se utiliza J_M.

Veamos un ejemplo de como funcionaría este proceso:

Supongamos que se decidió trabajar con el criterio de la traza de la matriz S_B, es decir, con distancia euclideana.

paso 1.- Se considera que hay una solución inicial previa para la estructuración de los objetos del universo U en **c** agrupamientos. Ésta pudo haber sido obtenida de manera aleatoria o por vía del criterio de los expertos del área en cuestión. Se inicializa la variable TOTAL en cero.

paso 2.- Se calcula la diagonal de la matriz de covarianzas S de la muestra U de objetos, misma que viene dada por

$$S = \|S_{pf}\|_{nxn}$$

siendo

$$S_{pf} = \frac{\sum_{i=1}^{m} (x_p(O_i) - \bar{x}_p)(x_f(O_i) - \bar{x}_f)}{m-1}$$

en la cual $\bar{x}_p$ es el promedio de los valores de dicha variable en toda la muestra, m el tamaño de la misma.

paso 3.- Normalización del valor $x_f(O_i)$ de la variable en el objeto en cuestión, para cada una de las n variables y para los m objetos de la muestra por medio de

$$x_f(O_i)/\sqrt{S_f f}$$

paso 4.- Obtención de la suma total del cuadrado del error de las desviaciones de los objetos con respecto a su propio promedio según la ecuación

$$J_c = t_r(S_B) = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \|I(O_j^i) - m_i\|^2$$

paso 5.- Selección del próximo objeto.- En este paso se selecciona al azar el objeto $I(O_j)$ del i-ésimo agrupamiento K_i, sobre el que se va a efectuar el estudio sobre el efecto que causa sobre el criterio a optimizar su traslado del i-ésimo agrupamiento al k-ésimo, con $i \neq j$. Si $n_i = 1$ y ya se ha seleccionado $TOTAL = M - 1$ objetos desde el último traslado, se termina el programa ya que este sería el agrupamiento óptimo. Si $n_i = 1$ y $TOTAL < (m - 1)$, se actualiza el índice de $TOTAL$ y se selecciona un nuevo objeto.

paso 6.- Se calcula el valor de ρ_t asociado a cada agrupamiento de K_t mediante las expresiones

$$\rho_k = \frac{n_k}{n_k + 1} \|I(O_j^i) - m_k\|^2$$

para $k \neq i$

$$\rho_k = \frac{n_k}{n_k - 1} \|I(O_j^i) - m_k\|^2$$

para $k = i$

que recoge para cada $k \neq i$ el incremento de la desviación que supondría si el objeto señalado se traslada del i-ésimo agrupamiento al k-ésimo, para $k = i$ la expresión nos dice el decremento que se producirá en la desviación en el i-ésimo agrupamiento al quitar dicho objeto.

paso 7.- Obtención del ρ_t: menor incremento en las desviaciones asociadas a cada agrupamiento al que fue llevado el objeto que se quitó del i-ésimo agrupamiento, siendo t el índice del agrupamiento en que se produjo el menor incidente.

paso 8.- Si $\rho_i > \rho_k$ se efectúa el traslado del objeto desde el i-ésimo agrupamiento al k-ésimo. Se actualizan las magnitudes de las desviaciones asociadas a cada agrupamiento. Con lo cual se obtiene un nuevo valor para J_c. Así mismo se actualizan los nuevos vectores promedio de cada agrupamiento, los nuevos tamaños de cada uno de ellos, el índice TOTAL se inicializa en cero nuevamente y se regresa al paso 4

paso 9.- Si $\rho_i \leq \rho_k$ no se debe efectuar el traslado del objeto desde el i-ésimo agrupamiento al k-ésimo ya que se incrementaría el valor del funcional J_c. Se actualiza TOTAL incrementándolo en una unidad. Si $TOTAL = m$, el tamaño de la muestra, significará que se han efectuado m iteraciones desde el paso 4 sin que se haya logrado mejoría alguna en el valor del funcional, por tanto se ha alcanzado un óptimo local del funcional con la distribución de los objetos en **c** agrupamientos. Si $TOTAL < m$ todavía no se han seleccionado todos los objetos desde el último traslado de agrupamiento efectuado y por tanto se regresa al paso 4.

3.3 Algoritmo c-means.

El algoritmo c-means es un procedimiento de agrupamiento que requiere definir una métrica en su Espacio de Representación Inicial y la considera como función de similaridad. Es precisamente la distancia de un elemento a un conjunto la que determina la pertenencia de un objeto a un agrupamiento o a otro. En este algoritmo se pretende agrupar los objetos que más se parezcan entre si en un agrupamiento, es decir que la distancia del objeto al agrupamiento (distancia de un elemento a un conjunto) sea mínima, mientras que el agrupamiento tenga una distancia máxima con respecto al resto de los agrupamientos (distancia entre conjuntos).

El algoritmo c-means es uno de los más conocidos y se basa sobre la idea de minimizar el cuadrado de las distancias de todos los elementos del agrupamiento al centro del mismo. Esta es una de las formas de maximizar

la homogeneidad de cada uno de los agrupamientos, minimizando por ende la heterogeneidad entre los mismos. Los pasos generales del algoritmo son:

Paso 1.- Seleccionemos en la primera iteración, **c** centros y denotémoslos por $z_1(1), z_2(1), ..., z_c(1)$ para los agrupamientos a formar, la primera estructuración propuesta, se realiza de manera arbitraria, aleatoriamente, o siguiendo el criterio de los expertos. En los siguientes pasos denótese al vector de las características del objeto i-ésimo, O_i, por $I(O_i)$

Paso 2.- En el k-ésimo paso de la iteración del algoritmo, distribuyamos los elementos de la muestra en los c agrupamientos siguiendo el siguiente criterio:

$$O \in S_j(k) \quad \text{si} \quad \|I(O) - z_j(k)\| < \|I(O) - z_i(k)\| \tag{3.1}$$

para todo i=1,...,c; $i \neq j$, donde $S_j(k)$ denota el agrupamiento del cual $z_j(k)$ es el centro.

Observe que la expresión (3.1) significa que un elemento del universo se ubica en el agrupamiento cuyo centro se encuentra más cercano a dicho elemento.

Paso 3.- A partir de los resultados obtenidos en el paso anterior se calcula los nuevos centros $z_j(k+1)$, j=1,...,c, de manera tal que la suma de los cuadrados de las distancias de cualquier punto de $S_j(k)$ al nuevo centro de ese agrupamiento sea mínima. Esto es, el nuevo agrupamiento con centro $z_j(k+1)$ se determina de modo tal que el parámetro

$$J_j = \sum_{O \in s_j(k)} \|I(O) - z_j(k+1)\|^2 \quad \text{con} \quad j = 1, ..., c \tag{3.2}$$

sea minimizado, El $z_j(k+1)$ que minimiza la expresión (3.2) es el valor medio del conjunto $S_j(k)$. De esta forma el centro del nuevo agrupamiento viene dado por la expresión

$$z_j(k+1) = \frac{1}{N_j} \sum_{O \in S_j(k)} I(O) \tag{3.3}$$

donde $N_j = \|S_j(k)\|$

El nombre del algoritmo, c-means, obedece a la manera en que se van calculando secuencialmente los centros de los nuevos agrupamientos.

Paso 4.- Si $z_j(k+1) = z_j(k)$ para todo j=1,...,**c**, termina el procedimiento con la propuesta de estructuración del universo, de lo contrario se regresa al paso 2.

Antes de hacer algunos comentarios acerca de los supuestos sobre los que se ha elaborado este algoritmo, veamos un ejemplo numérico en el plano.

Sea U un universo dado por la siguiente tabla:

Objeto	x_1, x_2
O_1	0,0
O_2	1,0
O_3	0,1
O_4	1,1
O_5	2,1
O_6	1,2
O_7	2,2
O_8	3,2
O_9	6,6
O_{10}	7,6

Objeto	x_1, x_2
O_{11}	8,6
O_{12}	6,7
O_{13}	7,7
O_{14}	8,7
O_{15}	9,7
O_{16}	7,8
O_{17}	8,8
O_{18}	9,8
O_{19}	8,9
O_{20}	9,9

Paso 1.- Sea c=2. Tomemos como distribución inicial de los centros la siguiente:

$$z_1(1) = I(O_1) = (0,0), \quad z_2(1) = I(O_2) = (1,0) \tag{3.4}$$

Paso 2.- Como

$$\|I(O_1) - z_1(1)\| < \|I(O_1) - z_i(1)\| \tag{3.5}$$

y

$$\|I(O_3) - z_1(1)\| < \|I(O_3) - z_i(1)\| \tag{3.6}$$

para i=2; y para los restantes objetos de la muestra se cumple que

$$\|I(O_p) - z_2(1)\| < \|I(O_p) - z_i(1)\| \tag{3.7}$$

para p=2,4,5,...,20; i=1, tenemos que:

$$S_1(1) = \{O_1, O_3\}, \quad \text{y} S_2(1) = \{O_2, O_4, O_5, ..., O_{20}\}.$$

De esta manera tenemos la primera iteración, es decir la primera propuesta de estructuración para el universo dado.

Paso 3.- Determinemos los nuevos centros para los agrupamientos calculados verificando que éstos satisfagan la expresión a (3.2). Para ello haremos uso de la expresión (3.3):

$$z_1(2) = \frac{1}{N_1} \sum_{O \in S_1(1)} I(O) = \frac{1}{2}(I(O_1), I(O_3)) = (0, 0.5) \tag{3.8}$$

$$z_2(2) = \frac{1}{N_2} \sum_{O \in S_2(1)} I(O) = \frac{1}{18}(I(O_2), I(O_4), I(O_5), ..., I(O_{20}))$$

$$= (5.67, 5.33)$$

Paso 4.- como quiera que $z_j(2) \neq z_j(1)$ para j=1,2, es decir, como los nuevos centros calculados son diferentes a los iniciales, regresamos al paso 2.

Paso 2(A).- A partir de los nuevos centros, haciendo uso de la expresión (3.1), distribuimos los elementos del universo en una nueva estructuración: dado que

$$\|I(O_p) - z_1(2)\| \leq \|I(O_p) - z_2(2)\| \tag{3.9}$$

para p=1,2,...,8 y

$$\|I(O_p) - z_2(2)\| \leq \|I(O_p) - z_1(2)\| \tag{3.10}$$

para p=9,20,...,20 formamos los nuevos agrupamientos:

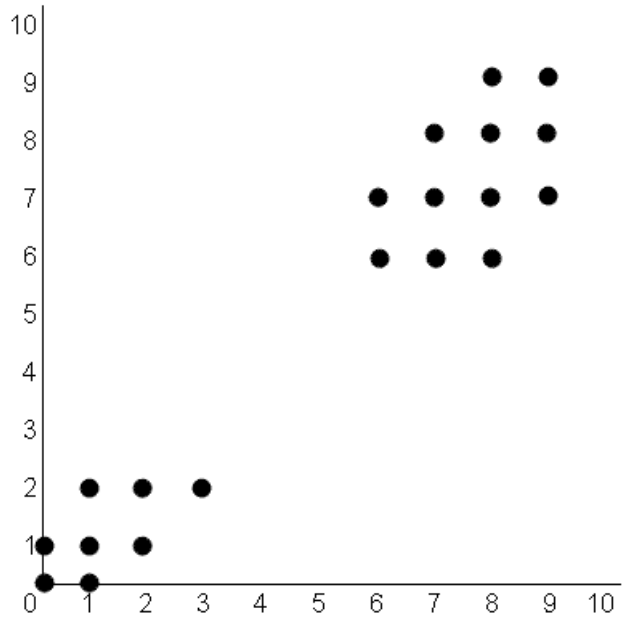
$$S_1(2) = \{O_1, ..., O_8\} \quad \text{y} \quad S_2(2) = \{O_9, ..., O_{20}\}$$

Vamos al paso 3

Paso 3.- De nuevo determinamos los nuevos centros haciendo uso de la expresión (3.3):

$$z_1(3) = \frac{1}{N_1} \sum_{O \in S_1(2)} I(O) = \frac{1}{8}(I(O_1) + I(O_2) + ... + I(O_8)) = (1.25, 1.13) \tag{3.11}$$

$$z_2(3) = \frac{1}{N_2} \sum_{O \in S_2(2)} I(O) = \frac{1}{12}(I(O_9) + ... + I(O_{20})) = (7.67, 7.33) \tag{3.12}$$



Paso 4.- Se nos presenta la misma situación anterior que $z_j(3) \neq z_j(2)$ para j=1,2; por lo que regresaremos al paso 2.

Paso 2.- Al calcular los nuevos agrupamientos resulta que $S_j(4) = S_j(3)$ para j=1,2. Vamos al paso 3

Paso 3.- Como era de esperar aquí también obtenemos los mismos centros que en la iteración anterior.

Paso 4.- Y como $z_j(4) = z_j(3)$ para j=1,2; el algoritmo termina

Si observa la representación geométrica de los puntos dados en el plano, corroborará que la estructuración obtenida es la que razonablemente se esperaría.

Es importante que subrayemos que el comportamiento del algoritmo, su aplicabilidad y efectividad práctica depende de los siguientes supuestos:

✓ Se debe conocer la cantidad **c** de agrupamientos a formar.

✓ Se tienen que seleccionar, entre los objetos del universo a estructurar, los **c centros (semillas)**, siguiendo criterios de expertos, de manera aleatoria o por medio de una heurística.

✓ El ERI tiene que ser al menos métrico. Esto descarta la posibilidad de aplicarlo en problemas donde aparezcan mezcladas variables cuantitativas y cualitativas. En los problemas que aparezcan sólo variables cuantitativas o cualitativas, no puede haber ausencia de información en la descripción de objeto alguno.

✓ La expresión (3.1) garantiza que los agrupamientos elaborados formen una partición, luego sólo se obtendrán estructuraciones de este tipo.

✓ El algoritmo, como otros de este tipo y enfoque, supone que **homogeneidad** (en el sentido del cumplimiento de la(s) propiedad(es) que caracteriza(n) a los objetos de un mismo agrupamiento) es equivalente a **cercanía** (en el sentido de la distancia que se defina sobre el ERI). Eso conlleva a la estrecha vinculación de dependencia entre la selección del ERI, la distancia definida y la calidad real de la solución propuesta.

✓ La expresión (3.3) puede llevar a un centro que no pertenezca a la muestra de estudio. Esto puede tener diferentes lecturas, algúnas en las que no se le de mucha importancia al asunto y otras en las que se llegue a desconfiar totalmente de la estructuración propuesta. Una "solución" a este hecho puede venir dada por la decisión de tomar como centro al objeto de la muestra **más cercano** al **centro virtual** calculado y radio igual al mínimo de las distancias de éste a los objetos de la muestra dada. Esto lleva a estructuraciones diferentes, ¿cuál de ellas es la buscada? Por otro lado está suponiendo que dicho centro es un valor central, ya que lo determina mediante el promedio de los valores, y como ya sabemos esto es cierto siempre y cuando la distribución de datos en la muestra sea al menos simétrica.

✓ No es difícil apreciar que la velocidad con que se alcance la solución estará en dependencia de la geometría de los datos, siendo más factible alcanzar respuesta rápidas y relativamente confiables en los casos en que las descripciones de los objetos queden como empaquetadas en el ERI respecto a la distancia seleccionada.

✓ El algoritmo no establece qué hacer cuando un objeto equidiste de más de uno de los centros en una iteración cualquiera. En términos de la expresión(1) no se puede decidir la ubicación de dicho elemento.

✓ No hay un teorema que garantice en el caso general la convergencia del algoritmo.

Ejemplo 9

En el siguiente ejemplo del algoritmo c-means se hace una clasificación en $\mathbb{R}^2$ con la finalidad de observar el comportamiento del algoritmo de forma gráfica. Las gráficas permitirán ver como se mueven los objetos de un agrupamiento a otro, y como cambia el centro de cada uno de ellos. Con ayuda del programa c-means (que se elaboró como parte del trabajo de esta tesis) se clasifican veinte objetos en dos particiones y se presentan las pantallas que arroga el programa en esta clasificación. En la tabla siguiente se tienen los veinte objetos a clasificar:

Objeto	x_1, x_2
O_1	1,2
O_2	0,5
O_3	5,9
O_4	3,2
O_5	5,2
O_6	6,7
O_7	1,9
O_8	9,3
O_9	4,3
O_{10}	5,6

Objeto	x_1, x_2
O_{11}	2,7
O_{12}	9,8
O_{13}	5,0
O_{14}	7,6
O_{15}	2,4
O_{16}	6,6
O_{17}	8,5
O_{18}	8,3
O_{19}	5,4
O_{20}	3,7

En la figura 3.1 se observan los veinte objetos representados en el plano.

En la figura 3.2 se muestran los datos requeridos por el programa(según el algoritmo presentado anteriormente) para poder llevar a cabo la clasificación. Como centros iniciales se consideraron los objetos O_{18} y O_{20}. Los objetos se almacenaron en un archivo llamado "ejemplo.dat" el cual es llamado por el programa.

En la gráfica "Iteración 0" se muestran los objetos pertenecientes a cada una de las particiones, así como los respectivos centros, según la primera

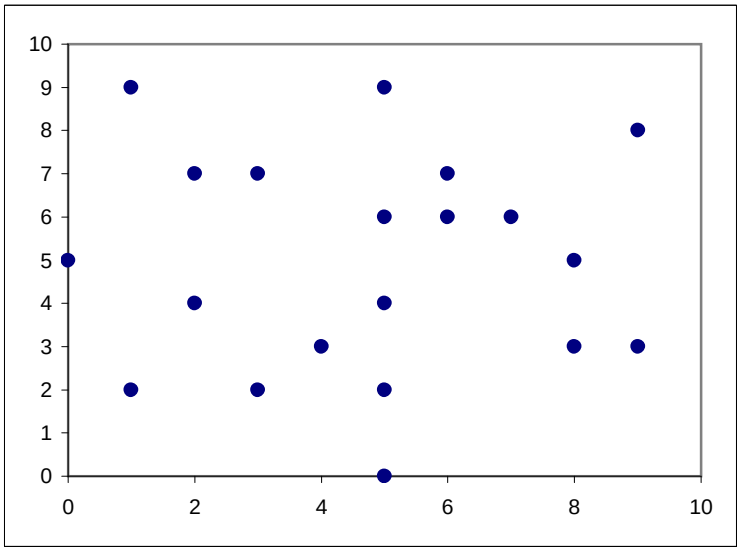


Figura 3.1: Conjunto de datos

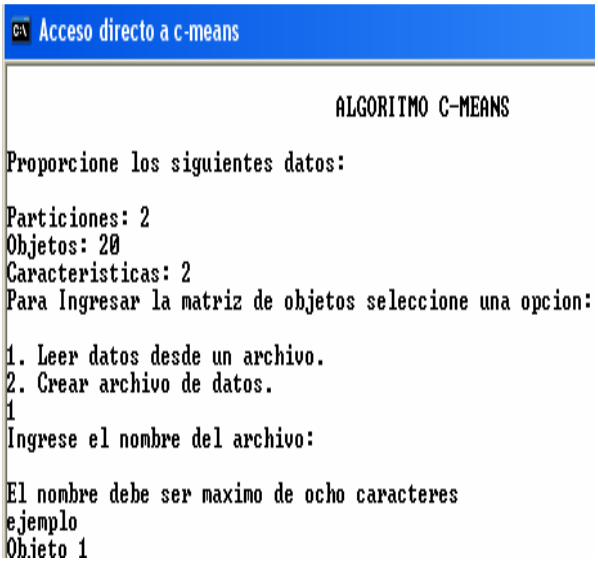


Figura 3.2: Programa c-means

```
Indique los objetos que considerara como los primeros centros


centro 1=objeto 18

centro 2=objeto 20

Objetos que pertenecen a la particion 1
 5,  8,  9,  12,  13,  14,  17,  18,  19,
Objetos que pertenecen a la particion 2
 1,  2,  3,  4,  6,  7,  10,  11,  15,  16,  20,

Presione una tecla para continuar . . .
```

Figura 3.3: primera clasificación

clasificación. Como en esta iteración los centros son elementos del universo éstos fueron encerrados para poder identificarlos del resto de los elementos. Los elementos pertenencientes a la partición 1 son representados por triángulos pequeños, mientras que los de la partición 2 son representados por un cuadrado.

En la gráfica que muestra la clasificación inicial (iteracón 0), se puede ver que en los objetos se agruparon de forma que están más cerca de su respectivo centro que del centro del otro agrupamiento.

Los centros de la iteración uno y de las siguientes se representa, en la gráfica correspondiente, por un circulo negro para diferenciarlos de los elementos, éstos representados por triángulos y cuadrados.

En la iteración 1 ya se tienen centros que no pertenecen al universo de objetos que, como se mencionó anteriormente, pudiera ocasionar conflictos respecto a la partición; sin embargo, es útil para formar nuevos agrupamientos de forma tal que los elementos de cada uno ellos se encuentren más cerca

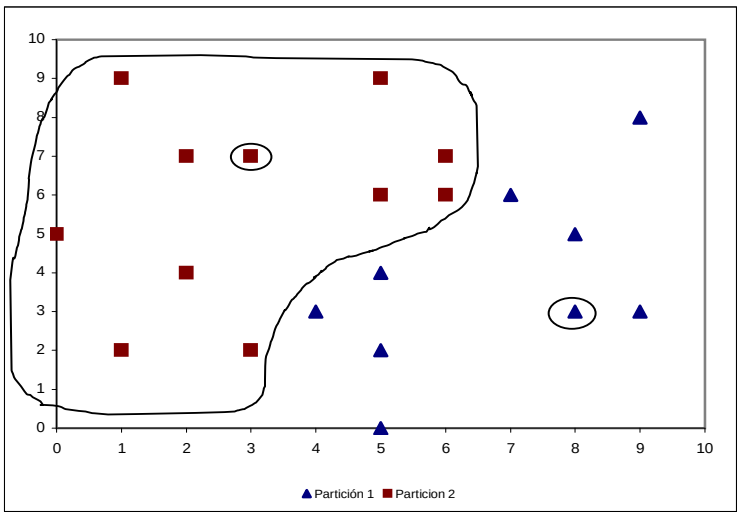


Figura 3.4: Gráfica de iteración 0

de su respectivo centro. El centro se toma como referencia para calcular las distancias. También se puede ver que en esta iteración se tiene un elemento menos en la agrupación dos; es decir, uno de los elementos de ésta ahora está más cerca del nuevo centro del otro agrupamiento.

En las siguientes iteraciones los centros cambian y, aunque tienen una variación casi insignificante, en cada una de ellas un elemento de la partición dos se "movió" a la partición uno.

El programa realizó cinco iteraciones para llegar a la clasificación final de los objetos. Se muestran las imágenes del programa y las gráficas de cada iteración. Observése que los centros encontrados a partir de la iteración 1 no son elementos del universo, característica que se señalo anteriormente. Además, gráficamente resulta más fácil ver que por la simetría de los objetos no se requiere de muchas iteraciones para que el algoritmo converja y los centros no tienen variaciones grandes en cada iteración.

Los centros que se encontraron en la iteración 4 y 5 son los mismos por lo que la gráfica de la clasificación final corresponde a la de la iteración 4.

```
**********Iteracion 1 **********

El centro de la particion 1 es
( 6.66667,  3.77778, )

El centro de la particion 2 es
( 3.09091,  5.81818, )

Objetos que pertenecen a la particion 1
 5,  8,  9,  12,  13,  14,  16,  17,  18,  19
Objetos que pertenecen a la particion 2
 1,  2,  3,  4,  6,  7,  10,  11,  15,  20,

Presione una tecla para continuar . . .
```

Figura 3.5: Iteración 1

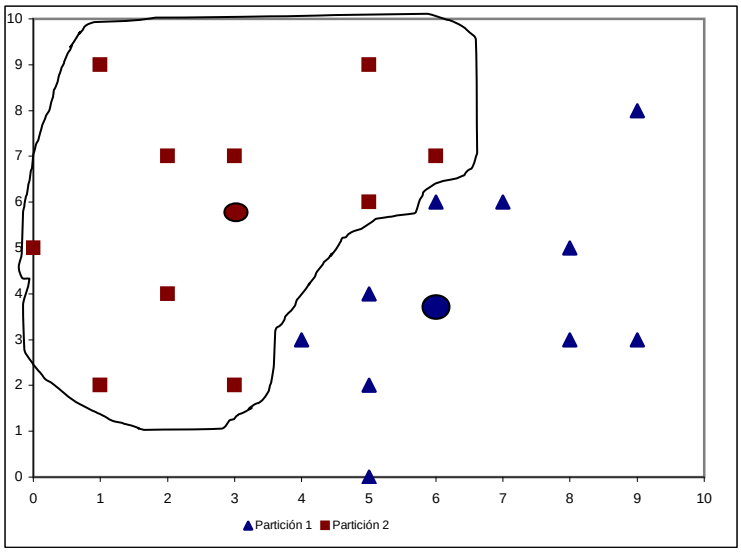


Figura 3.6: Gráfica de iteración 1

```
**********Iteracion 2 **********

El centro de la particion 1 es
( 6.6,  4, )

El centro de la particion 2 es
( 2.8,  5.8, )

Objetos que pertenecen a la particion 1
 5,  6,  8,  9,  12,  13,  14,  16,  17,  18,  19
Objetos que pertenecen a la particion 2
 1,  2,  3,  4,  7,  10,  11,  15,  20,

Presione una tecla para continuar . . .
```

Figura 3.7: Iteración 2

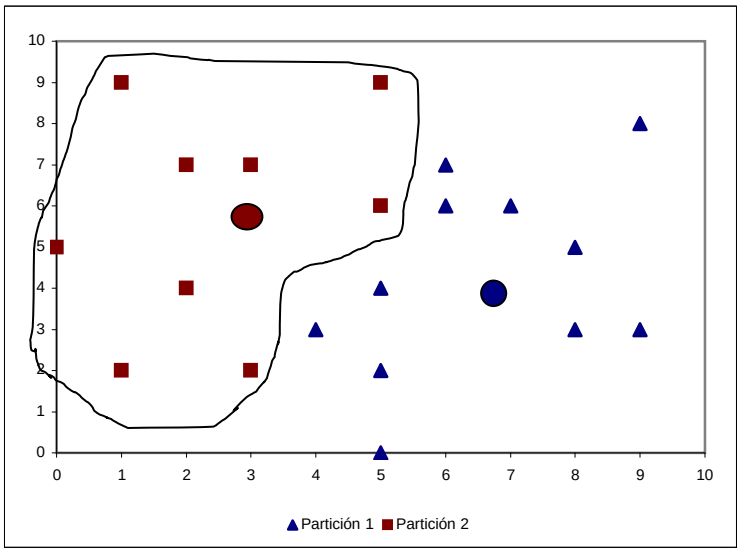


Figura 3.8: Gráfica de iteración 2

```
**********Iteracion 3 **********

El centro de la particion 1 es
( 6.54545,  4.27273, )

El centro de la particion 2 es
( 2.44444,  5.66667, )

Objetos que pertenecen a la particion 1
 5,  6,  8,  9,  10,  12,  13,  14,  16,  17,  18,  19
Objetos que pertenecen a la particion 2
 1,  2,  3,  4,  7,  11,  15,  20,

Presione una tecla para continuar . . .
```

Figura 3.9: Iteración 3

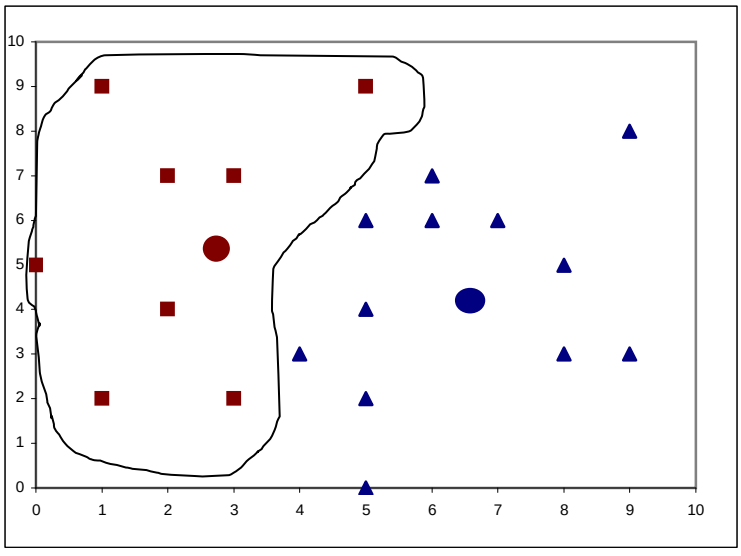


Figura 3.10: Gráfica de iteración 3

```
**********Iteracion 4 **********

El centro de la particion 1 es
( 6.41667,  4.41667, )

El centro de la particion 2 es
( 2.125,  5.625, )

Objetos que pertenecen a la particion 1
 5,  6,  8,  9,  10,  12,  13,  14,  16,  17,  18,  19
Objetos que pertenecen a la particion 2
 1,  2,  3,  4,  7,  11,  15,  20,

Presione una tecla para continuar . . .
```

Figura 3.11: Iteración 4

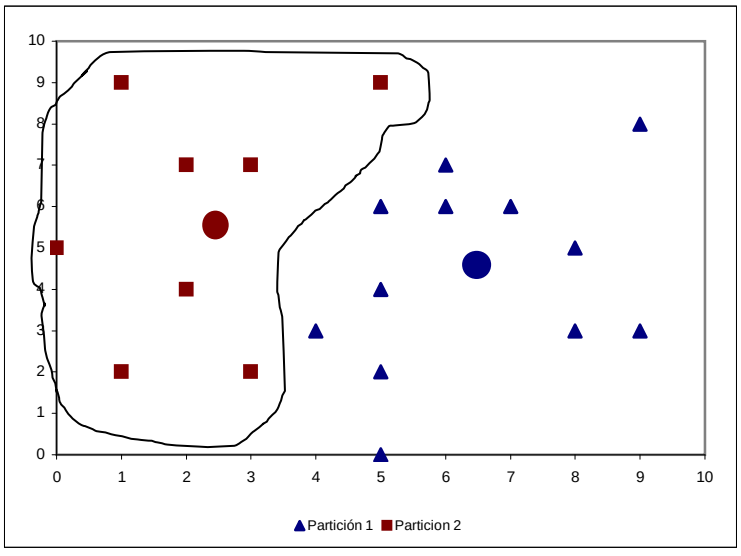


Figura 3.12: Gráfica de iteración 4

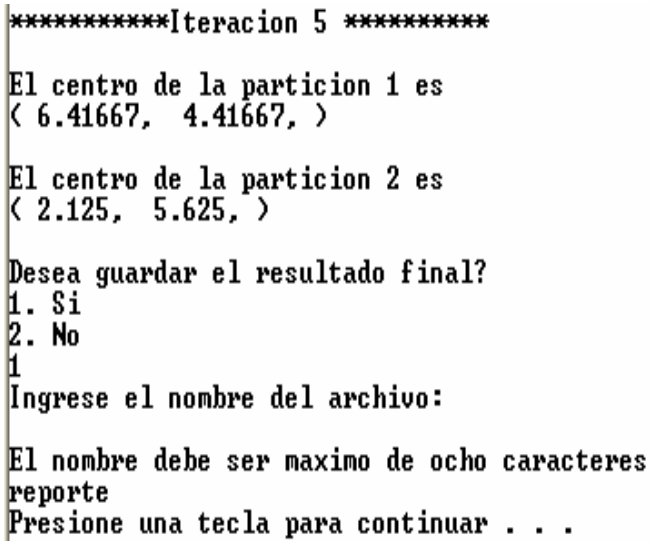


Figura 3.13: Iteración 5

El programa ofrece la opción de guardar un reporte del resultado final. Dicho análisis se guardo en el archivo "reporte.tex". La última figura, Reporte, muestra la parte final del reporte generado por el programa. En este se observan los datos iniciales, la clasificación final y los centros de cada partición.

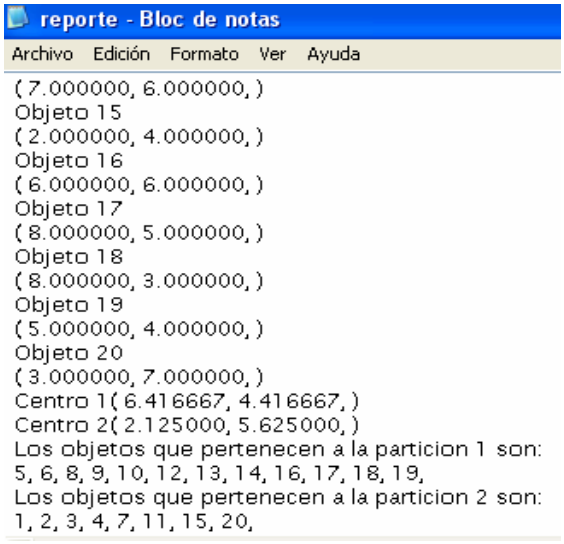


Figura 3.14: Reporte

Capítulo 4

Enfoque difuso

En este capítulo se define una partición difusa y se presenta el algoritmo que permitirá la clasificación difusa bajo en el enfoque lógico combinatorio de su ERI. Es decir, no se supondrá alguna distribución de los objetos para poder clasificarlos, si no que, dadas las características de los objetos se hace la clasificación sin ninguna suposición.

4.1 Particiones difusas

Definición 4.1.1 *c-Partición Dura. Sea X=*$\{x_1, x_2, ..., x_n\}$ *cualquier conjunto finito;* V_{cn} *es el conjunto de las matrices reales* $(c \times n)$*, con c número natural,* $2 \leq c < n$*. El conjunto de la c-partición dura para X es*

$$M_c = \{U \in V_{cn} \mid u_{ik} \in \{0,1\} \forall i,k; \sum_{i=1}^{c} u_{ik} = 1 \forall k; 0 < \sum_{k=1}^{n} u_{ik} < n \forall i\}$$

Ejemplo 10

Considérese el conjunto X definido por:

$$X = \{x_1 = \text{durazno}, x_2 = \text{limón}, x_3 = \text{naranja}\}$$

Se desea clasificar los elementos de X en dos particiones o subconjuntos; es decir, se tiene c=2. Una posible matriz de partición es:

$$x_1 \quad x_2 \quad x_3$$

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

Donde la primer fila corresponde a la partición 1 y la segunda a la partición 2. Así, la matriz anterior nos dice que en la primera partición se agruparán a los elementos x_1 y x_2, mientras que el elemento x_3 pertenecerá a la partición 2. Otras posibles matrices de partición son:

$$\begin{matrix} x_1 & x_2 & x_3 \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\begin{matrix} x_1 & x_2 & x_3 \end{matrix}$$
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

La siguiente matriz no cumple con la definición de c-partición dura ya que la primer fila suma 3 y la segunda cero.

$$\begin{matrix} x_1 & x_2 & x_3 \end{matrix}$$
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Observación. Como se observó en el ejemplo anterior, existen varias matrices de partición diferentes para el mismo conjunto y con el mismo número de particiones.

La siguiente expresión nos permite determinar el número de las distintas c-particiones duras posibles dado un conjunto de elementos y el número de particiones deseadas:

$$|M_c^n| = \frac{1}{c!}\left[\sum_{j=1}^{c} \binom{c}{j} (-1)^{c-j} j^n\right]$$

Demostración. La expresión anterior se verifica de forma intuitiva:

Para $n = 1$ y $c = 1$

$$|M_1^1| = \frac{1}{1!}\left[\sum_{j=1}^{1} \binom{1}{j} (-1)^{1-j} j^1\right] = \binom{1}{1} = 1$$

Esto es fácil de ver si se considera un conjunto de un elemento, entonces solo existe una forma de hacer una participación.

Para $n = 2$ y $c = 1$

$$|M_1^2| = \frac{1}{1!}\left[\sum_{j=1}^{1}\left(\begin{array}{c}1\\ j\end{array}\right)(-1)^{1-j}j^2\right] = \left(\begin{array}{c}1\\ 1\end{array}\right)(-1)^{1-1}1^2 = 1$$

Para $n = 2$ y $c = 2$

$$\begin{aligned}|M_2^2| &= \tfrac{1}{2!}\left[\textstyle\sum_{j=1}^{2}\left(\begin{array}{c}2\\ j\end{array}\right)(-1)^{2-j}j^2\right]\\ &= \tfrac{1}{2!}\left[\left(\begin{array}{c}2\\ 1\end{array}\right)(-1) + \left(\begin{array}{c}2\\ 2\end{array}\right)(-1)^{1-1}2^2\right]\\ &= \left(\tfrac{1}{2}\right)(-2+4)\\ &= 1\end{aligned}$$

Para $n = 3$ y $c = 1$

$$|M_1^3| = \frac{1}{1!}\left[\sum_{j=1}^{1}\left(\begin{array}{c}1\\ j\end{array}\right)(-1)^{1-j}j^3\right] = \left(\begin{array}{c}1\\ 1\end{array}\right)(-1)^{1-1}1^3 = 1$$

Para $n = 3$ y $c = 2$

$$\begin{aligned}|M_2^3| &= \tfrac{1}{2!}\left[\textstyle\sum_{j=1}^{2}\left(\begin{array}{c}2\\ j\end{array}\right)(-1)^{2-j}j^3\right]\\ &= \tfrac{1}{2!}\left[\left(\begin{array}{c}2\\ 1\end{array}\right)(-1) + \left(\begin{array}{c}2\\ 2\end{array}\right)(-1)^{1-1}2^3\right]\\ &= \left(\tfrac{1}{2}\right)(-2+8)\\ &= 3\end{aligned}$$

Para $n = 3$ y $c = 3$

$$\begin{aligned}|M_3^3| &= \tfrac{1}{3!}\left[\textstyle\sum_{j=1}^{3}\left(\begin{array}{c}3\\ j\end{array}\right)(-1)^{3-j}j^3\right]\\ &= \tfrac{1}{3!}\left[\left(\begin{array}{c}3\\ 1\end{array}\right) - \left(\begin{array}{c}3\\ 2\end{array}\right)2^3 + \left(\begin{array}{c}3\\ 3\end{array}\right)3^3\right]\\ &= \left(\tfrac{1}{3!}\right)(3-24+27)\\ &= 1\end{aligned}$$

Suponiendo cierto para n entonces se debe verificar para $n+1$

$$\begin{aligned}
|M_c^{n+1}| &= c|M_c^n| + |M_{c-1}^n| \\
&= c\tfrac{1}{c!}\left[\sum_{j=1}^{c}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j}j^n\right] + \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c-1}\begin{pmatrix} c-1 \\ j \end{pmatrix}(-1)^{c-1-j}j^n\right] \\
&= \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j}j^n + \sum_{j=1}^{c-1}\begin{pmatrix} c-1 \\ j \end{pmatrix}(-1)^{c-1-j}j^n\right] \\
&= \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c-1}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j}j^n + \sum_{j=1}^{c-1}\begin{pmatrix} c-1 \\ j \end{pmatrix}(-1)^{c-1-j}j^n + c^n\right] \\
&= \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c-1}\left(\begin{pmatrix} c \\ j \end{pmatrix} - \begin{pmatrix} c-1 \\ j \end{pmatrix}\right)(-1)^{c-j}j^n + c^n\right]
\end{aligned}$$

Por otro lado

$$\begin{aligned}
\begin{pmatrix} c \\ j \end{pmatrix} - \begin{pmatrix} c-1 \\ j \end{pmatrix} &= \tfrac{c!}{(c-j)!j!} - \tfrac{(c-1)!}{(c-1-j)!j!} \\
&= \tfrac{c!-(c-j)(c-1)!}{(c-j)!j!} \\
&= \tfrac{(c-1)!j}{(c-j)!j!}
\end{aligned}$$

Entonces:

$$\begin{aligned}
|M_c^{n+1}| &= \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c-1}\left(\tfrac{(c-1)!}{(c-j)!j!}j\right)(-1)^{c-j}j^n + c^n\right] \\
&= \tfrac{1}{(c-1)!}\left[\sum_{j=1}^{c-1}\tfrac{(c-1)!}{(c-j)!j!}(-1)^{c-j}j^{n+1} + c^n\right] \\
&= \tfrac{1}{c!}\left[\sum_{j=1}^{c-1}\tfrac{c!}{(c-j)!j!}(-1)^{c-j}j^{n+1} + c^{n+1}\right] \\
&= \tfrac{1}{c!}\left[\sum_{j=1}^{c-1}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j}j^{n+1} + c^{n+1}\right] \\
&= \tfrac{1}{c!}\left[\sum_{j=1}^{c}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j}j^{n+1}\right]
\end{aligned}$$

■

Ejemplo 11

Supógase un conjunto de tres elementos
Para c=1, se tiene

$$\begin{aligned}
|\ M_1\ | &= \tfrac{1}{1!}\left[\sum_{j=1}^{1}\begin{pmatrix} 1 \\ j \end{pmatrix}(-1)^{1-j}j^3\right] \\
&= 1
\end{aligned}$$

$$\begin{matrix} x_1 & x_2 & x_3 \end{matrix}$$

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

Para c=2 se tiene

$$\begin{aligned} \mid M_2 \mid &= \tfrac{1}{2!}\left[\sum_{j=1}^{2} \begin{pmatrix} 2 \\ j \end{pmatrix} (-1)^{2-j} j^3\right] \\ &= \tfrac{1}{2}\left[\begin{pmatrix} 2 \\ 1 \end{pmatrix} (-1)^{2-1} + \begin{pmatrix} 2 \\ 2 \end{pmatrix} (-1)^{2-2} 2^3\right] \\ &= 3 \end{aligned}$$

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

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

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

Para c=3 se tiene

$$\begin{aligned} \mid M_3 \mid &= \tfrac{1}{3!}\left[\sum_{j=1}^{3} \begin{pmatrix} 3 \\ j \end{pmatrix} (-1)^{3-j} j^3\right] \\ &= \tfrac{1}{3!}\left[\begin{pmatrix} 3 \\ 1 \end{pmatrix} (-1)^{3-1} + \begin{pmatrix} 3 \\ 2 \end{pmatrix} (-1)^{3-2} 2^3 + \begin{pmatrix} 3 \\ 3 \end{pmatrix} (-1)^{3-3} 3^3\right] \\ &= 1 \end{aligned}$$

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

Definición 4.1.2 *c-Partición Fuzzy. Sea X cualquier conjunto finito, V_{cn} el conjunto real de matrices $(c \times n)$, c es un entero; $2 \leq c < n$. El conjunto de la c-partición fuzzy para X es*

$$M_{fc} = \{U \in V_{cn} \mid u_{ik} \in [0,1] \forall i,k; \sum_{i=1}^{c} u_{ik} = 1 \forall k; 0 < \sum_{k=1}^{n} < n \forall i\}$$

Ejemplo 12

Considerando los datos del ejemplo anterior, la nueva clasificación puede establecerse como:

$$\begin{bmatrix} 0.91 & 0.02 & 0.04 \\ 0.09 & 0.98 & 0.96 \end{bmatrix}$$

Entonces el elemento x_1 pertenece a la primera partición en 0.91 mientras que los elementos x_2 y $x_3$pertenecen a la misma partición en 0.02 y 0.04, respectivamente.

El algoritmo c-means difuso se basa en el algoritmo c-means (de ahí su nombre), la idea principal de ambos algoritmos es encontrar un centro en cada agrupamiento y clasificar los objetos maximizando una medida de similitud. Se buscan centros para cada agrupamiento y se calcula el grado de pertenencia de cada objeto a cada uno de ellos. En el algoritmo difuso se habla de particiones difusas ya que se supone que los elementos pertenecen a cada agrupamiento pero en cierto grado.

Antes de presentar el algoritmo c-means difuso se tiene el teorema siguiente que permite el desarrollo del algoritmo.

Teorema 4.1.3 *Asume* $\|\cdot\|$ *el producto interno inducido. Fija* $min(1,\infty)$*, permite que* X *tenga al menos* $c < n$ *distintos puntos, y define* $\forall k$ *los conjuntos*

$$I_k = \{i \mid 1 \leq i \leq c; d_{ik} = \| \boldsymbol{x}_k - \boldsymbol{v}_i \| = 0\}$$

$$\widetilde{I}_k = \{1, 2, ..., c\} - I_k$$

entonces $(U, \boldsymbol{v}) \in M_{fc} \times \mathbb{R}^{cp}$ *seria el minimo global para* J_m *solo si*

$$I_k = \emptyset \Rightarrow u_{ik} = 0 \forall i \in \widetilde{I}_k y \sum_{i \in I_k} u_{ik} = 1$$

$$\boldsymbol{v}_i = \sum_{k=1}^{n} (u_{ik})^m \boldsymbol{x} \diagup \sum_{k=1}^{n} (u_{ik})^m \forall i$$

Del algoritmo anterior (y para el algoritmo) se define **c** como el número de agrupamientos a formar; u_{ik} como el elemento de la matriz de pertenencia de la fila i-ésima y la columna j-ésima.

4.2 Algoritmo C-Means difuso

Algoritmo 4.2.1 *Fuzzy c-Means FCM, Bezdek*

a) Fija c, 2≤ c < n, elige cualquier norma métrica de producto interno para $\mathbb{R}^p$; y fija m, 1≤ m < ∞. Inicializa $U^{(0)} \in M_{fc}$. Entonces para la iteración l, l=0, 1, 2,..., y en base al teorema anterior:

b) Calcula los centros cluster c difuso $\{v_i^{(l)}\}$

c)Actualiza $U^{(l)}$

d)Compara $U^{(l)}$ y $U^{(l+1)}$. Si $\| U^{(l+1)} - U^{(l)} \| \leq \varepsilon_L$ parar, de otro modo regresar a a).

La selección de la matriz de pertenencia inicial puede hacerse de manera aleatoria o siguiendo el criterio de los expertos.

Ejemplo 13

A manera de ejemplo del algoritmo c-means difuso obsérvese el conjunto siguiente de datos:

Objeto	x_1,	x_2,	x_3
O_1	0,	0,	0
O_2	205,	197,	250
O_3	150,	94,	102
O_4	246,	55,	17
O_5	250,	201,	22
O_6	118,	199,	192
O_7	210,	176,	34
O_8	140,	64,	102
O_9	233,	2,	167
O_{10}	229,	149,	23
O_{11}	162,	96,	194

Objeto	x_1,	x_2,	x_3
O_{12}	15,	66,	156
O_{13}	28,	47,	11
O_{14}	3,	171,	9
O_{15}	51,	165,	139
O_{16}	99,	133,	149
O_{17}	18,	83,	61
O_{18}	108,	124,	60
O_{19}	203,	243,	174
O_{20}	57,	60,	235
O_{21}	38,	194,	79
O_{22}	250,	250,	250

Con el programa "fuzzy.cpp" (programa elaborado como parte de esta tesis y cuyo código se presenta en la siguiente sección) del algoritmo c-means difuso se corren los datos de la tabla anterior para poder clasificarlos en tres particiones. La figura 4.1 muestra los datos iniciales que solicita el programa.

Se considera la siguiente matriz inicial de pertenencia:

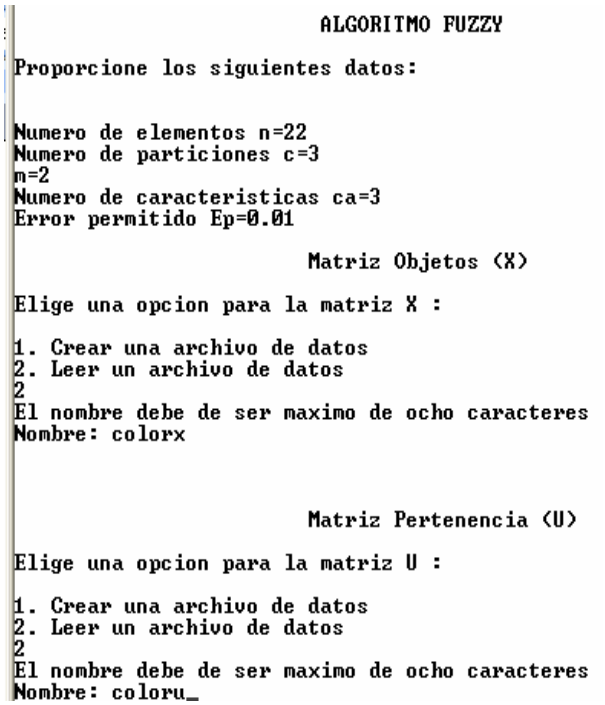


Figura 4.1: Datos iniciales del programa

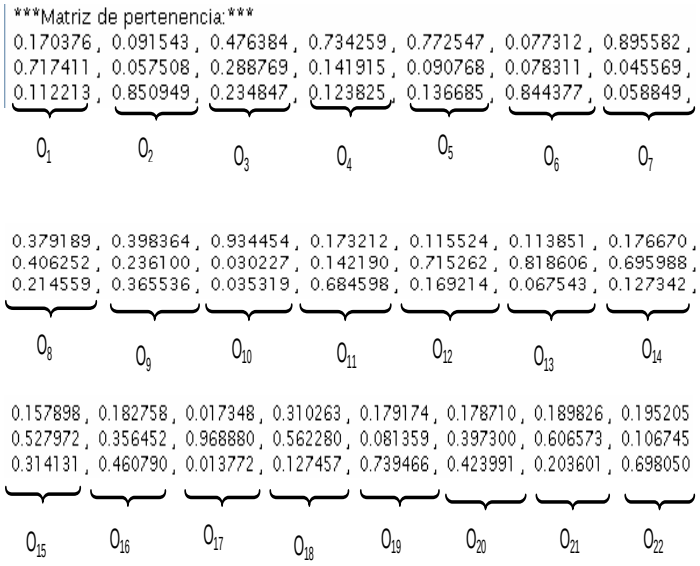


Figura 4.2: Matriz de pertenencia

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

Debe recordarse que la matriz inicial se puede elegir de forma aleatoria o bajo ciertos criterios de los expertos de la materia según los objetos que se estén analizando. En este caso la matriz inicial de pertenencia fue elegida aleatoriamrente.

El programa Fuzzy con los datos anteriores, después de diez iteraciones y con un error de 0.00954412, presenta la matriz de pertenencia que se muestra en la figura 4.2.

La matriz de pertenencia de la última iteración dista de la primera en un 0.00954412. Esto permite ver que no hubo una variación grande en los grados de pertenencia de cada objeto a cada agrupamiento por lo que se puede aceptar a la última matriz de pertenecia para la clasificación.

La primera fila de la matriz de pertenencia indica el grado de pertenecia de cada objeto a la partición uno, la sedunda fila indica el grado de pretencia de cada objeto a el agrupamiento dos y la tercer fila indica la pertenencia al tercer agrupamiento.

Recuerdese que el enfoque difuso, por analizarse bajo la óptica clasificatoria, busca maximizar la similitud de los elementos de cada agrupamiento. Por lo tanto, de la matriz de pertenencia final, se puede concluir que los elementos con grado de pertenencia mayor al agrupamiento uno se parecen más entre sí de lo que se parecen con los otros elementos, utilizando la norma como medida de similitud.

Así se concluye que los elementos 3, 4, 5, 7, 9 y 10 se parecen más entre ellos por tener el grado de pertencia mayor al agrupamiento uno, de la misma forma se tiene que los elementos 1, 8, 12, 13, 14, 15, 17, 18 y 21 tienen mayor similitud entre ellos, y finalmente los mismo sucede con los elementos 2, 6, 11, 16, 19, 20 y 22. Sin embargo, esto no significa que no se tenga cierta similitud con el resto de los elementos, solo que la similitud que se comparte es menos con los otros agrupamientos.

Por la similaridad del algoritmo c-means difuso con el c-means se tienen supuestos y características semejantes.

✓ Se concoce **c**, la cantidad de agrupamientos a formar.

✓ Se seleccionan los **c** centros iniciales entre los objetos del universo.

✓ El ERI tiene que ser métrico.

✓ Se supone que **homogeneidad** es equivalente a **cercanía**.

✓ Se puede obtener un centro que no sea objeto del universo de datos que se analizarán.

✓ El algoritmo no indica qué hacer en el caso que se encuentre un objeto que equidiste de los **c** centros.

4.3 Código del algoritmo

En esta sección se presenta el código fuente de los programas de los algoritmos c-means y c-means difuso.

Programa c-means

```
#include <iostream>
#include <cmath>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
void Verifica_Datos (char H, int tipo, char numero[10])
{
  int i=0, punto, diagonal;
  float division;
  if (tipo==0)
  punto=0;
  else
  punto=1;
  H=getch();
  while(H=='\r')
  {
    cout<<"\nProporcione informacion.\n";
    H=getch();
  }
  while(H!='\r')
  {
    if(H=='\b')
    {
      if ((numero[i-1]=='.')&&(tipo==0))
      punto=0;
      if(i!=0)
      {
        i=i-1;
        cout<<"\b \b";
      }
    }
    if((tipo==0)||(tipo==1))
    {
      if((H>='0')&&(H<='9'))
      {
        numero[i]=H;
        cout<<H;
```

```
            i++;
        }
    }
    else
    {
        if((H=='1')||(H=='2'))
        {
            numero[i]=H;
            cout<<H;
            i++;
            break;
        }
        else
        cout<<"Presione 1 o 2\n";
    }
    if ((H=='.')&&(punto==0))
    {
        numero[i]='.';
        i++;
        punto=1;
        cout<<H;
    }
    H=getch();
  }
  numero[i]='\0';
}
//Función que pide el nombre del archivo de datos
void Nombre(char S[13], int q) {
  int i=0;
  char H;
  H=getch();
  while(H=='\r')
  {
    cout<<"\nProporcione informacion.\n";
    H=getch();
  }
  while(H!='\r')
  {
```

```
      if(((H>='a')&&(H<='z'))||((H>='0')&&(H<='9')))
      {
        cout<<H;
        S[i]=H;
        i++;
      }
      if((i!=0)&&(H=='\b'))
      {
        i=i-1;
        cout<<"\b \b";
      }
      if(i==8)
      break;
      H=getch();
    }
    if(q==0)
    {
      S[i]='.';
      S[i+1]='d';
      S[i+2]='a';
      S[i+3]='t';
      S[i+4]='\0';
    }
    else
    {
      S[i]='.';
      S[i+1]='t';
      S[i+2]='x';
      S[i+3]='t';
      S[i+4]='\0';
    }
}
//Función que crea el archivo de datos
int Crear(char*S,int r, int q, float *t) {
    FILE *in;
    int k, j, i, punto, z;
    char H, numero[10];
    if((in=fopen(S,"w+"))==NULL)
```

```
    {
      fprintf(in, "No se puede crear el archivo");
      return (0);
    }
    fprintf(in, "%d", r);
    fprintf(in, "\n%d", q);
    cout<<"\nIngresa los datos. \n";
    cout<<"\nNota: Ingre los datos por fila. Donde Objeto[i][j]\n";
    cout<<"es la entrada correspondiente al i-esimo objeto y la \n";
    cout<<"j-esima caracteristica. \n\n";
    for(k=0; k<r; k++)
    {
      cout<<"\nObjeto "<<k+1<<"\n";
      for(j=0; j<q; j++)
      {
        Verifica_Datos(H, 0, numero);
        t[k*q+j]=atof(numero);
        fprintf(in, "\n%f", t[k*q+j]);
        cout<<"\n";
      }
    }
    fclose(in);
    return(1);
}
//Función para leer un archivo de datos
int Leer(char*S, float *t){
    FILE *in;
    int i, j, z, q;
    if((in=fopen(S,"r+"))==NULL)
    {
      fprintf(in, "No encontro el archivo");
      return(0);
    }
    else
    {
      fscanf(in, "%d",&z);
      fscanf(in, "%d",&q);
      for(i=0; i<z; i++)
```

```
      for(j=0; j<q; j++)
      fscanf(in, "%f",&t[i*q+j]);
   }
   fclose(in);
   return(1);
}
//Función para la distribución de los elementos
void Distribucion(int n, int c, int ca, float *obj,float*centro,int*particion)
{   int i, k, j, q, indice;
   float dist, *distancia, min;
   for(i=0; i<c; i++)
   for(j=0; j<n; j++)
   particion[i*n+j]=0;
   distancia=new float[c*n];
   for(i=0; i<n; i++)
   {
      q=-1;
      for(k=0; k<c; k++)
      {
         q++;
         dist=0;
         for(j=0; j<ca; j++)
         dist=pow(centro[k*ca+j]-obj[i*ca+j],2)+dist;
         distancia[q]=sqrt(dist);
      }
      min=distancia[0];
      indice=0;
      for(q=0; q<c; q++)
      {
         if(distancia[q]<min)
         {
            min=distancia[q];
            indice=q;
         }
      }
      particion[indice*n+i]=i+1;
   }
   delete(distancia);
```

```
}
//Funcion para calcular los nuevos centros
void Centros(int c, int n, int ca, int*particion, float*ncentro, float*obj)
{
  int q, N, i, j, k, indice[n];
  for(q=0; q<c; q++)
  {
    N=0;
    for(i=0; i<n; i++)
    if(particion[q*n+i]==i+1)
    {
      N=N+1;
      indice[N]=i;
    }
    for(j=0; j<ca; j++)
    {
      ncentro[q*ca+j]=0;
      for(k=0; k<N; k++)
      ncentro[q*ca+j]=obj[indice[k+1]*ca+j]+ncentro[q*ca+j];
      ncentro[q*ca+j]=ncentro[q*ca+j]/N;
    }
  }
}
int Compara(int c, int ca, float *ncentro, float *centro)
{
  int i, j, k=0;
  for(i=0; i<c; i++)
  for(j=0; j<ca; j++)
  if(ncentro[i*ca+j]!=centro[i*ca+j])
  k=k+1;
  return k;
}
void Cambia_centros(int c, int ca, float* centro, float *ncentro)
{
  int i, j;
  for(i=0; i<c; i++)
  for(j=0; j<ca; j++)
  centro[i*ca+j]=ncentro[i*ca+j];
```

```
    }
//Funcion para guardar el resultado final
int Guardar_Resultado(float *centro, int *particion, int c, int n, int ca, char
*S, float *obj) {
        FILE *in;
        int i, j;
        char H, numero[10];
        if((in=fopen(S,"w+"))==NULL)
        {
          fprintf(in, "No se puede crear el archivo");
          return (0);
        }
        //datos
        fprintf(in, "particiones c=%d", c);
        fprintf(in, "\nobjetos n=%d", n);
        fprintf(in, "\ncaracteristicas ca=%d", ca);
        //objetos
        for(i=0; i<n; i++)
        {
          fprintf(in, "\nObjeto %d\n", i+1);
          fprintf(in, "( ");
          for(j=0; j<ca; j++)
          fprintf(in, "%f, ", obj[i*ca+j]);
          fprintf(in, ")");
        }
        //centros
        for(i=0; i<c; i++)
        {
          fprintf(in, "\nCentro %d", i+1);
          fprintf(in, "( ");
          for(j=0; j<ca; j++)
          fprintf(in, "%f, ", centro[i*ca+j]);
          fprintf(in, ")");
        }
        //particiones
        for(i=0; i<c; i++)
        {
          fprintf(in, "\nObjetos que pertenecen a la particion %d:\n", i+1);
```

```
      for(j=0; j<n; j++)
      if(particion[i*n+j]==j+1)
      fprintf(in, "%d, ", j+1);
   }
   fclose(in);
   return(1);
}
int main() {
   int c, n, ca, i, j, k, opcion, Iteracion=0, *particion, Termina=0;
   float *obj, *centro, *ncentro;
   char S[13], H, numero[10];
   cout<<"\n\t\t\t\tALGORITMO C-MEANS\n\n";
   cout<<"Proporcione los siguientes datos:\n\n";
   cout<<"Particiones: ";
   Verifica_Datos(H, 1, numero);
   c=atoi(numero);
   cout<<"\nObjetos: ";
   Verifica_Datos(H, 1, numero);
   n=atoi(numero);
   cout<<"\nCaracteristicas: ";
   Verifica_Datos(H, 1, numero);
   ca=atoi(numero);
   //Matriz de objetos:
   cout<<"\nPara Ingresar la matriz de objetos seleccione una
opcion:\n\n";
   cout<<"1. Leer datos desde un archivo.\n";
   cout<<"2. Crear archivo de datos.\n";
   Verifica_Datos(H, 2, numero);
   opcion=atoi(numero);
   cout<<"\nIngrese el nombre del archivo: ";
   cout<<"\n\nEl nombre debe ser maximo de ocho caracteres\n";
   Nombre(S, 0);
   obj=new float[n*ca];
   switch(opcion)
   {
      case 1:
         Leer(S, obj);
         for(i=0; i<n; i++)
```

```
        {
          cout<<"\nObjeto "<<i+1<<"\n";
          for(j=0; j<ca; j++)
          cout<<" "<<obj[i*ca+j]<<"\n";
        }
      break;
      case 2:
        Crear(S, n, ca, obj);
      break;
    }
    //Pedir los centros iniciales:
    cout<<"\n\nIndique los objetos que considere como los primeros
centros\n\n";
    centro= new float[c*ca];
    for(i=0; i<c; i++)
    {
      cout<<"\ncentro "<<i+1<<"=objeto ";
      cin>>j;
      for(k=0; k<ca; k++)
      centro[i*ca+k]=obj[(j-1)*ca+k];
    }
    particion= new int[c*n];
    ncentro= new float [c*ca];
    Iteracion=0;
    while(Termina==0)
    {
    //Paso 2: Distribucion de los elementos.
    Distribucion(n, c, ca, obj, centro, particion);
    for(i=0; i<c; i++)
    {
      cout<<"\nObjetos que pertenecen a la particion "<<i+1<<
" \n";
      for(k=0; k<n; k++)
      if(particion[i*n+k]==k+1)
      cout<<" "<<k+1<<", ";
    }
    cout<<"\n\n";
    system("pause");
```

```
    Iteracion++;
    //Paso 3: Nuevos centros.
    cout<<"\n\n**********Iteracion "<<Iteracion<<" **********\n";
    Centros(c, n, ca, particion, ncentro, obj);
    for(i=0; i<c; i++)
    {
      cout<<"\nEl centro de la particion "<<i+1<<" es \n(";
      for(j=0; j<ca; j++)
      cout<<" "<<ncentro[i*ca+j]<<", ";
      cout<<")\n";
    }
    //Paso 4: Comparar centros.
    if (Compara(c, ca, ncentro, centro)==0)
    Termina=1;
    //
    Cambia_centros(c, ca, centro, ncentro);
    }
    cout<<"\nDesea guardar el resultado final? \n";
    cout<<"1. Si\n2. No\n";
    Verifica_Datos(H, 2, numero);
    opcion=atoi(numero);
    if(opcion==1)
    {
      cout<<"\nIngrese el nombre del archivo: ";
      cout<<"\n\nEl nombre debe ser maximo de ocho caracteres\n";
      Nombre(S, 1);
      Guardar_Resultado(centro, particion, c, n, ca, S, obj);
    }
    cout<<"\n";
    system("pause");
    delete(obj);
    delete(centro);
    delete(particion);
    delete(ncentro);
  }
```

Programa Fuzzy (c-means difuso)

```
//nombre: Herrera Ocaranza Elizabeth
```

```
#include <iostream>
#include <cmath>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
//Función para verificar los datos que ingresan los usuarios
void Verifica_Datos (char H, int tipo, char numero[10])
{
 int i=0, punto;
if (tipo==0)
punto=0;
else
punto=1;
H=getch();
while(H=='\ r')
{
cout<< "\n Proporcione informacion.\n";
H=getch();
}
while(H!='\r')
{
 if(H=='\b')
 {
  if ((numero[i-1]=='.')&&(tipo==0))
  punto=0;
  if(i!=0)
  {
   i=i-1;
   printf("\b \b");
  }
 }
 if((tipo==0)||(tipo==1))
 {
  if((H>='0')&&(H<='9'))
  {
   numero[i]=H;
   printf("%c",H);
```

```
            i++;
          }
        }
        else
        {
          if((H=='1')||(H=='2'))
          {
            numero[i]=H;
            printf("%c",H);
            i++;
            break;
          }
        }
        if ((H=='.')&&(punto==0))
        {
          numero[i]='.';
          i++;
          punto=1;
          printf("%c",H);
        }
        H=getch();
    }
    numero[i]='\0';
}

//Función para crear el nombre de los archivos de datos
void Nombre(char S[13])
{
    int i=0;
    char H;
    H=getch();
    while(H=='\r')
    {
      cout<<"\n Proporcione informacion.\n";
      H=getch();
    }
    while(H!='\r')
    {
```

```
      if((H>='a')&&(H<='z'))
      {
        cout<<H;
        S[i]=H;
        i++;
      }
      if((i!=0)&&(H=='\b'))
      {
        i=i-1;
        printf("\b \b");
      }
      if(i==8)
      break;
      H=getch();
   }
   S[i]='.';
   S[i+1]='d';
   S[i+2]='a';
   S[i+3]='t';
   S[i+4]='\0';
}
//Función para crear un archivo de datos
int Crear(char*S, int r, int q, float *t)
{
   FILE *in;
   int k, j, i, punto, z;
   char H, numero[10];
   if((in=fopen(S,"w+"))==NULL)
   {
      printf("No pude crear el archivo");
      return (0);
   }
   fprintf(in, "%d", r);
   fprintf(in, "\n%d", q);
   printf("Escribe los valores\n");
   for(k=0; k<r; k++)
   for(j=0; j<q; j++)
   {
```

```
      Verifica_Datos(H, 0, numero);
      t[k*q+j]=atof(numero);
      fprintf(in, "\n%f", t[k*q+j]);
      cout<<"\n";
    }
    fclose(in);
    return(1);
}
//Función para leer un archivo de datos
int Leer(char*S, float *t)
{
    FILE *in;
    int i, j, z, q;
    if((in=fopen(S,"r+"))==NULL)
    {
      fprintf(in, "No encontro el archivo");
      return(0);
    }
    else
    {
      fscanf(in, "%d",&z);
      fscanf(in, "%d",&q);
      for(i=0; i<z; i++)
      for(j=0; j<q; j++)
      fscanf(in, "%f",&t[i*q+j]);
    }
    fclose(in);
    return(1);
}
//Función que verifica la suma de las columnas y filas de la matriz U
void Verifica_U(float *u, int c, int n, int k, int g)
{
    int i, j;
    float suma;
    for(j=0; j<n; j++)
    {
      suma=0;
      for(i=0; i<c; i++)
```

```
      suma=suma+u[i*n+j];
      if(suma!=1)
      {
        cout<<"\n Error!! La columna "<<j+1<<" no suma 1.";
        k=1;
      }
    }
    for(i=0; i<c; i++)
    {
      suma=0;
      for(j=0; j<n; j++)
      suma=suma+u[i*n+j];
      if((suma==0)||(suma==n))
      {
        cout<<"\n Error!! En la fila "<<i+1;
        g=1;
      }
    }
}
//Función que calcula el vector V
void VectorV(float *u, int n, int m, int ca, float *v, int c, float *x)
{
    float *suma, potencia1, *vector;
    int i, j, k;
    suma=new float[c];
    vector=new float[n*ca];
    for(i=0; i<c; i++)
    {
      suma[i]=0;
      for(k=0; k<n; k++)
      {
        potencia1=pow(u[i*n+k],m);
        suma[i]=suma[i]+potencia1;
        for (j=0; j<ca; j++)
        vector[k*ca+j]=potencia1*x[k*ca+j];
      }
      for(k=0; k<ca; k++)
      {
```

```
            v[i*ca+k]=0;
            for(j=0; j<n; j++)
            v[i*ca+k]=v[i*ca+k]+(vector[j*ca+k]/suma[i]);
         }
      }
      delete(suma);
      delete(vector);
}
//Función para calcular los d[i][k]=‖x[i][ ]-v[k][ ]‖
void Indices(int k, int c, float *d, int ca, float *x, float *v, int n)
{
      int i, j;
      for(i=0; i<c; i++)
      {
         d[i*n+k]=0;
         for(j=0; j<ca; j++)
         d[i*n+k]=pow(x[k*ca+j]-v[i*ca+j],2)+d[i*n+k];
         d[i*n+k]=sqrt(d[i*n+k]);
      }
}
//Función para determinar cuantos d[i][k] no son cero
int Suma(int k, int c, float *d, int n)
{
      int suma2=0, i;
      for(i=0; i<c; i++)
      {
         if (d[i*n+k]==0)
         suma2=suma2+1;
         else;
      }
      return suma2;
}
// Función para calcular los u[i][k]
float MatrizU(int i, int k, int suma2, int m, int c, float *d,
float*u, int n)
{
      int j;
      float p=2.0/(m-1), *UInv;
```

```
    UInv=new float[c*n];
    UInv[i*n+k]=0;
    if (suma2==0)
    {
      for(j=0; j<c; j++)
      {
        UInv[i*n+k]=UInv[i*n+k]+pow(d[i*n+k]/d[j*n+k],p);
      }
      return (1/UInv[i*n+k]);
    }
    else
    {
      if(d[i*n+k]!=0)
      return 0;
      else
      return (u[i*n+k]/suma2);
    }
    delete(UInv);
}
//Función que calcula el error
float CalcularError(float *uActualizada, int c, int n, float *u)
{
    int i, k;
    float error=0;
    for(i=0; i<c; i++)
    for(k=0; k<n; k++)
    error=error+pow(uActualizada[i*n+k]-u[i*n+k],2);
    error=sqrt(error);
    return error;
}
//
void UIteracionMas(float *uActualizada, int c, int n, float *u)
{
    int i, k;
    for(i=0; i<c; i++)
    for(k=0; k<n; k++)
    u[i*n+k]=uActualizada[i*n+k];
}
```

```
int main()
{
  int c, m, n, ca, i, j, k, suma2, opX, opU, Iteracion=0, g;
  float *uActualizada, *x, *u, *v, *d, Respuesta, Ep;
  char S[13], H, numero[10];
  cout<<"Proporcione los siguientes datos:\n\n";
  c=1;
  n=0;
  while(c>n)
  {
    cout<<"\nNumero de elementos n=";
    Verifica_Datos(H, 1, numero);
    n=atoi(numero);
    cout<<"\nNumero de particiones c=";
    Verifica_Datos(H, 1, numero);
    c=atoi(numero);
    if(c>n)
    cout<<"\n\nError!\nEl numero de particiones debe ser menor
que el numero de datos.\n";
  }
  cout<<"\nm=";
  Verifica_Datos(H, 1, numero);
  m=atoi(numero);
  cout<<"\nNumero de caracteristicas ca=";
  Verifica_Datos(H, 1, numero);
  ca=atoi(numero);
  cout<<"\nError permitido Ep=";
  Verifica_Datos(H, 0, numero);
  Ep=atof(numero);
//x[i][k] entrada de la matriz X que indica el valor de la
k-ésima característica del elemento i-ésimo.
  cout<<"\n\n\t\t\tDatos Matriz X \n\n";
  cout<<"Elige una opcion para la matriz X : \n\n";
  cout<<"1. Crear una archivo de datos. \n" ;
  cout<<"2. Leer un archivo de datos \n";
  Verifica_Datos(H, 2, numero);
  opX=atoi(numero);
  cout<<"\nEl nombre debe de ser maximo de ocho caracteres";
```

```
    cout<<"\nNombre: ";
    Nombre(S);
    cout<<"\n\n";
    x=new float[n*ca];
    switch(opX)
    {
      case 1:
      Crear(S, n, ca, x);
      break;
      case 2:
      Leer(S, x);
      break;
    }
  //u[i][k] indica el valor del grado de pertenencia del elemento k-ésimo
  a la particin i-ésima
    cout<<"\n\n\t\t\tDatos Matriz U \n\n";
    cout<<"Elige una opcion para la matriz U : \n\n";
    cout<<"1. Crear una archivo de datos \n" ;
    cout<<"2. Leer un archivo de datos \n";
    Verifica_Datos(H, 2, numero);
    opU=atoi(numero);
    cout<<"\nEl nombre debe de ser maximo de ocho caracteres";
    cout<<"\nNombre: ";
    Nombre(S);
    cout<<"\n\n";
    u=new float[c*n];
    switch(opU)
    {
      case 1:
      Crear(S, c, n, u);
      break;
      case 2:
      Leer(S, u);
      break;
    }
    cout<<"\n\n\t\t\t\tMatriz X\n";
    for(i=0; i<n; i++)
    {
```

```
      cout<<"\n\t\t";
      for(j=0; j<ca; j++)
      cout<<" "<<x[i*ca+j];
    }
    cout<<"\n\n\t\t\t\tMatriz U\n";
    for(i=0; i<c; i++)
    {
      cout<<"\n\t\t";
      for(j=0; j<n; j++)
      cout<<" "<<u[i*n+j];
    }
    Verifica_U(u, c, n, k, g);
    if((k==0)&&(g=0))
    {
//Iteraciones
    Respuesta=Ep+1;
    while(Respuesta>Ep)
    {
      Iteracion++;
//Paso b
      v=new float[c*ca];
      VectorV(u, n, m, ca, v, c, x);
//Comienza paso c
      d=new float[c*n];
      uActualizada=new float[c*n];
      for(k=0; k<n; k++)
      {
        Indices(k, c, d, ca, x, v, n);
        suma2=Suma(k, c, d, n);
        for(i=0; i<c; i++)
        uActualizada[i*n+k]=MatrizU(i, k, suma2, m, c, d, u, n);
      }
      cout<<"\n\n\t\t\t\tMatriz Û("<<Iteracion<<")\n\n";
      for(i=0; i<c; i++)
      {
        cout<<"\n\t\t";
        for(k=0; k<n; k++)
```

```
        cout<<" "<<uActualizada[i*n+k];
      }
//Inicia paso d
      Respuesta=CalcularError(uActualizada, c, n, u);
      if(Respuesta>Ep)
      cout<<"\n\n\n\t\t El error es="<<Respuesta<<">"<<Ep<<
"\n\n";
      else
      cout<<"\n\n\n\t\t El error es="<<Respuesta<<"<"<<Ep<<
"\n\n";
      UIteracionMas(uActualizada, c, n, u);
      system("pause");
   }
   }
   system("pause");
   delete(x);
   delete(u);
   delete(v);
   delete(d);
   delete(uActualizada);
}
```

Conclusiones

Los algoritmos que se analizaron son el **c-means** y el **c-means difuso**. Estos algoritmos presentan ciertas ventajas o desventajas según sea la velocidad de convergencia del algoritmo, las características que deba tener el ERI para poder utilizarlos, la cantidad de datos que se desean manejar, entre otras.

Según el número de objetos que se quieran clasificar, los algoritmos pueden converger con velocidad o no. Para considerar la velocidad de convergencia se ha calculado el tiempo de cada algoritmo.

Para el algoritmo c-means el tiempo por cada iteración está dado por la siguiente expresión:

$$c(2 + n(2k + 1) + k) + 2(k + n + c) + 1$$

donde **c** es el número de agrupamientos a formar, **n** es el número de objetos y **k** es el número de características o descripciones de cada objeto.

En el caso del algoritmo c-means difuso el tiempo se determina por la expresión:

$$c\{n[6k + 5 + 2(\ulcorner log_2 m + 1 \urcorner)] + 2 - k - \ulcorner log_2(m - 1) \urcorner\} + 1$$

Por la similaridad de ambos algoritmos se puede agrupar las ventajas y desventajas de utilizarlos.

Ventajas

✓ La ejecución de los algoritmos es ya una ventaja por el manejo de conceptos básicos de matemáticas y de las operaciones que requieren.

✓ La velocidad con que se alcance la solución del problema, en el caso del algoritmo c-means, dependerá de la geometría de los datos, ya que entre más dispersos se encuentren los objetos el algoritmo tendrá una menor velocidad de convergencia.

✓ Ambos algoritmos garantizan la obtención de particiones y la inclusión de todos los objetos en algún agrupamiento. En el caso c-means se garantiza un cubrimiento de los objetos en base a la expresión (3.1) del algoritmo. Cuando se presenta la igualdad de la expresión nos encontramos ante un problema y desventaja que se comenta más adelante. Para el caso del algoritmo difuso es el teorema 4.1.8 el que garantiza lo anterior.

✓ El algoritmo c-means difuso presenta la ventaja de poder clasificar objetos bajo información incierta, es decir, bajo la lógica difusa.

Desventajas

✓ El ERI debe ser métrico. Esto no permite el manejo de características cuantitativas y cualitativas de los objetos. Solo pueden analizarse aquellos objetos en los que las variables son cualitativas o cuantitativas. Esto es por basarse en el enfoque estadístico.

✓ Existe la posibilidad de obtener centros que no pertenecen al universo de objetos que se analizan. Lo que ocasiona desconfianza en la estructura obtenida con los algoritmos. Esto puede observarse en la expresión (3.3) del algoritmo c-means y en el teorema 4.1.8 en el caso difuso.

✓ El algoritmo c-means no indica que hacer cuando se encuentre un objeto que equidista de los **c** centros. Así que se pueden originar diferentes particiones según se decida la pertenencia de dicho objeto a un agrupamiento o a otro.

✓ No hay un teorema que garantice en el caso general la convergencia del algoritmo. Aunque la convergencia depende de la simetría de los objetos, el número de objetos y el error admisible.

✓ Por ser logarítmica la expresión que determina el tiempo del algoritmo c-means difuso es una desventaja de éste.

Ambos algoritmos resultan sumamente útilles para la clasificación de objetos y son actualmente aplicados en diversos campos de la ciencia y tecnología. Las desventajas de los algoritmos no deben desanimar el uso de ellos. Esto depende de las características de los agrupamientos que se deseen obtener. Por ejemplo, el no utilizar variables cualitativas y cuantitativas evitan la ausencia de información. Al analizar variables solo cualitativas o solo cuantitativas permite tener "completa" la información requerida para el análisis. Así la desventaja de no poder utilizar variables cualitativas y cuantitativas puede resultar ser una ventaja según como se desee formar los agrupamientos.

Por otro lado, el uso de las herramientas matemáticas que se requieren en ambos algoritmos resultar ser ventajoso por la sencillez de éstas.

Como punto principal de ambos algoritmos, aunque resulta aun mayor la importancia en el c-means difuso, se puede mencionar la amplia aplicación en el desarrollo de tecnología moderna.

En general se agruparon la ventajas y desventajas de los algoritmos c-means y cmeans difuso; sin embargo, existen características de ellos que pueden traducirse en ventajas o desventasa de un algoritmo sobre el otro. Primero, y tal vez, la ventaja más importante del c-means difuso sobre el c-means es que en el primero se forman agrupamientos en base a la teoría de conjuntos difusos siendo estos los que mejor se ajustan a la realidad.

Por lo anterior, otra desventaja del algoritmo c-means es que éste no indica que hacer cuando un objeto equidista de dos o más centros. Sin embargo, este problema no se presenta en el difuso ya que un elemento puede pertenencer a más de un agrupamiento con el mismo grado de pertenencia.

Con respecto al tiempo de convergencia, el algoritmo c-means tiene ventaja sobre el difuso por ser la expresión del tiempo de éste último del tipo logarítmico.

Según las características de cada algoritmo y las propiedades que se desee de los agrupamientos a formar es como se decidirá que algoritmo conviene utilizar.

Esta tesis planteó dos enfoques actualmente utilizados en diversos campos de la ciencia y tecnolgía. Se presentaron ejemplos y se programaron ambos algoritmos que permitan comprenderlos mejor esperando que el lector pueda hacer uso de ellos y le sea de utilidad como una forma de introducción al estudio de la teoría de conjuntos difusos.

Bibliografía

[1] George J. Klir and Bo Yuan, Fuzzy sets and fuzzy logic (theory and applications),1995, Prentice Hall, USA.

[2] James C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms.

[3] G. Florg, Ejercicios de topología y de análisis, Tomo 1 Topología, Editorial Serie reverté de problemas.

[4] Warren Alvarado y Roberto Chavarria, Introducción al Control mediante lógica difusa, Facultad de Ingeniería, Universidad de Costa Rica.

[5] Jesús Antonio Acosta Sarmiento, Tesis Doctral: Aprendizaje de particiones difusas para razonamiento inductivo, 2006, Universidad Politécnica de Catalunya, Barcelona.

[6] Ignacio L. Iribarren, Topología de espacios métricos, 1973, Limusa, primera edición.

[7] Jantzen, J. "Tutorial on fuzzy logic", http://www.iau.dfu.dk/-jj/index.html

[8] Chevuie, F. Guely, F. "Fuzzy logic", squared.com

[9] Morales Luna, G., "Introducción a la lógica difusa", http://delta.cs.cinvestav.mx/ gmorales/ldifll/ldifll.html