Algoritmos Conceptuales Restringidos basados en Semillas

por

Irene Olaya Ayaquica Martínez

Tesis sometida como requisito parcial
para obtener el grado de

DOCTOR EN CIENCIAS EN EL ÁREA DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
Julio 2007
Tonantzintla, Puebla

Supervisada por:

Dr. José Francisco Martínez Trinidad
Coordinación de Ciencias Computacionales, INAOE

Dr. Jesús Ariel Carrasco Ochoa
Coordinación de Ciencias Computacionales, INAOE

Dedicatoria

A mis padres Jesús e Irene

Con amor, respeto, admiración y gratitud.
Por enseñarme a luchar por mis ideales.
Pero sobre todo por darme las bases para ser quien soy.

A mis hermanos Jesús, María Esther y Ricardo

Por el cariño, la comprensión y el apoyo
que me han brindado a lo largo de mi vida.

Agradecimiento Especial

A mis asesores el Dr. José Francisco Martínez Trinidad y el Dr. Jesús Ariel Carrasco Ochoa a quienes debo el desarrollo y culminación del presente trabajo, por su apoyo incondicional en todo momento, por creer en mí y por los conocimientos transmitidos.

Al Dr. José Ruiz Shulcloper, quien, con sus observaciones ha contribuido de manera fundamental en el desarrollo del presente trabajo, por el apoyo que me ha brindado a lo largo de mi vida profesional, gracias por su confianza.

A los Doctores Angélica Muñoz Meléndez, Gustavo Rodríguez Gómez, Carlos Alberto Reyes García y Jesús Antonio González Bernal, por sus valiosos comentarios que contribuyeron al mejoramiento de esta tesis.

Agradecimientos

A los Doctores que conforman la academia del área de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica, por darme la oportunidad de realizar mis estudios de Doctorado en esta Institución.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), por las facilidades y apoyos brindados durante mi estancia en esta Institución.

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

A mis amigos y compañeros, por los gratos momentos que tuve la dicha de compartir con ustedes, por su amistad y apoyo incondicional.

Resumen

El estudio de la clasificación no supervisada ha sido enfocado principalmente a desarrollar métodos que determinen agrupamientos tales que objetos en un mismo agrupamiento sean muy similares entre ellos, mientras que objetos de agrupamientos diferentes sean poco similares. Sin embargo, para algunos problemas prácticos se requiere, además de determinar los agrupamientos, conocer las propiedades que describan cómo son dichos agrupamientos. A este problema se le conoce como agrupamiento conceptual.

Existen diversos algoritmos que permiten resolver el problema de agrupamiento conceptual entre los que se encuentra el algoritmo k-means conceptual, el cual es una versión conceptual del algoritmo k-means; uno de los algoritmos más estudiados y utilizados para resolver el problema de clasificación no supervisada restringida (cuando se especifica *a priori* el número de agrupamientos). La principal característica del algoritmo k-means conceptual es que requiere retículos de generalización para la construcción de los conceptos. El retículo de generalización para los atributos cualitativos debe ser dado por el especialista y el retículo de generalización para los atributos cuantitativos se construye a partir de una codificación de los valores de estos atributos. En esta tesis, se propone una mejora del algoritmo k-means conceptual, la cual usa una estrategia diferente para construir los agrupamientos y en la fase de caracterización, un retículo de generalización diferente para los atributos cuantitativos.

Un inconveniente al usar retículos de generalización es que, en general, es difícil determinar los retículos de generalización. Además, no se tienen métodos automáticos para construir los retículos, por lo que esta tarea se deja al especialista. Por esta razón, en esta tesis, se propone también un algoritmo k-means conceptual que no depende de retículos de generalización para la construcción de los conceptos.

Finalmente, en esta tesis, se proponen dos algoritmos conceptuales difusos, los cuales son versiones difusas de los algoritmos conceptuales duros propuestos.

Abstract

The non-supervised classification algorithms determine clusters such that objects in the same cluster are very similar among them, while objects in different clusters are not similar. However, there are some problems where it is required, besides determining the clusters, to know the properties that characterize them. This problem is known as conceptual clustering.

There are different methods that allow to solve the conceptual clustering problem, one of them is the conceptual k-means algorithm, which is a conceptual version of the k-means algorithm; one of the most studied and used algorithms for solving the restricted non-supervised classification problem (when the number of clusters is specified *a priori*). The main characteristic of the conceptual k-means algorithm is that it requires generalization lattices for the construction of the concepts. The generalization lattices for the qualitative features must be given and the generalization lattices for the quantitative features are built starting from a codification of their values. In this thesis, an improvement of the conceptual k-means algorithm, which uses a different strategy for building the clusters and in the characterization phase, a different generalization lattice for the quantitative features, is proposed.

The inconvenience of using generalization lattices is that, in general, it is difficult to determine the generalization lattices. Also, there are not automatic methods to build the generalization lattices; therefore, this task must be done by the user. For this reason, in this thesis, a conceptual k-means algorithm that does not depend on generalization lattices for building the concepts is proposed.

Finally, in this thesis, two fuzzy conceptual clustering algorithms, which are fuzzy versions of the proposed hard conceptual clustering algorithms, are proposed.

Contenido

Lista de Tablas

Lista de Figuras

Notación

O_j	un objeto de la muestra
x_s	un atributo
n	número de objetos
m	número de atributos
k	número de agrupamientos
U	k-partición
μ_{ij}	grado de pertenencia de un objeto a un agrupamiento
L	conjunto totalmente ordenado
A_i	un agrupamiento
C_i	un concepto
P	un predicado duro
(P, μ_P)	un predicado difuso
R	conjunto de atributos
T	conjunto de objetos
D_s	conjunto de valores admisibles para un atributo
FC_s	función de comparación para comparar valores de atributos
Γ	función de similaridad para comparar objetos
Ψ	función de disimilaridad para comparar objetos

Acrónimos

KMSF	k-means con funciones de similaridad
FKMSF	k-means difuso con funciones de similaridad
CKM	k-means conceptual
CKMSF	k-means conceptual con funciones de similaridad
CKMCF	k-means conceptual con rasgos complejos
FCKMSF	k-means conceptual difuso con funciones de similaridad
FCKMCF	k-means conceptual difuso con rasgos complejos

Introducción

El objetivo principal de esta tesis es proponer algoritmos basados en semillas, duros y difusos, para resolver el problema de agrupamiento conceptual restringido cuando se tienen objetos[1] descritos por atributos cualitativos y cuantitativos mezclados (datos mezclados), así como ausencia de información[2] (datos incompletos).

Por problemas de clasificación no supervisada restringida (agrupamiento restringido) entenderemos todos aquellos problemas de agrupamiento en los cuales se conoce *a priori* el número de agrupamientos que se desean formar.

El problema de clasificación no supervisada puede resolverse desde dos enfoques: un *enfoque clasificatorio*, cuyo objetivo es encontrar los objetos que, dadas sus relaciones de semejanza, deben estar en un mismo agrupamiento; o un *enfoque conceptual*, cuyo objetivo es encontrar los agrupamientos que formarán la estructuración, así como las propiedades que caracterizan a dichos agrupamientos.

El estudio de la clasificación no supervisada ha sido orientado principalmente al enfoque clasificatorio en el cual, los métodos que se han propuesto determinan agrupamientos tales que objetos en el mismo agrupamiento sean muy similares entre ellos, mientras que objetos de diferentes agrupamientos sean poco similares. Sin embargo, estos métodos tienen como inconveniente el hecho de dejar al especialista la tarea de interpretar los agrupamientos. Es decir, construyen agrupamientos para los cuales no se da una interpretación. Esta desventaja es significativa en aquellos problemas donde el especialista requiere, además de determinar los agrupamientos, conocer las propiedades que los caracterizan. Para resolver este problema, se introdujo el agrupamiento conceptual.

[1] Un objeto puede ser: una persona, un concepto o una idea, en general un ente.
[2] Con el término ausencia de información nos referimos a los valores faltantes de un atributo (missing data).

El agrupamiento conceptual surge en la década de los 80's a partir de los trabajos de Michalski (Michalski, 1980; Michalski and Diday, 1981; Michalski, 1983; Michalski and Stepp, 1983; Michalski, 1986; Stepp and Michalski, 1986). En este enfoque se propone encontrar, a partir de un conjunto de datos, no sólo los agrupamientos en los que éstos se estructuran sino además una caracterización de los mismos.

Los algoritmos conceptuales pueden dividirse en dos tipos: restringidos y no restringidos. En los algoritmos conceptuales restringidos se da *a priori* el número de agrupamientos, mientras que en los algoritmos conceptuales no restringidos no se conoce el número de agrupamientos. Los algoritmos restringidos usualmente trabajan con base en semillas[3], donde el número de semillas es el mismo que el número de agrupamientos que se formarán.

Los algoritmos conceptuales restringidos utilizan una función de distancia para evaluar la semejanza entre objetos. Para comparar objetos descritos por atributos cualitativos y cuantitativos mezclados, los datos usualmente son transformados para poder aplicar una función de distancia y el problema se transforma en uno nuevo, lo cual ocasiona que los resultados no puedan interpretarse en términos del problema original. Por esta razón, en esta tesis proponemos algoritmos conceptuales que utilizan funciones de similaridad que no requieren de transformaciones de los atributos, y además pueden tomar en consideración los criterios de analogía usados por el especialista del área bajo estudio.

Los algoritmos conceptuales restringidos basados en semillas que se han desarrollado hasta el momento resuelven el problema de agrupamiento conceptual cuando se desea obtener conceptos que describan agrupamientos duros, es decir, cuando se evalúa si los objetos que pertenecen a un agrupamiento cumplen la propiedad de dicho agrupamiento; a este tipo de agrupamiento conceptual se le denomina agrupamiento conceptual duro. Sin embargo, en algunos problemas prácticos los especialistas están interesados en evaluar el grado de pertenencia de un objeto a un agrupamiento más que en decidir si un objeto

[3] Una semilla es el centroide inicial de un agrupamiento, el cual generalmente se selecciona de manera aleatoria.

pertenece o no a dicho agrupamiento, es decir, requieren de agrupamientos difusos. Además, desean obtener una interpretación conceptual de dichos agrupamientos difusos. A este problema lo denominaremos agrupamiento conceptual difuso.

El problema de agrupamiento conceptual restringido difuso ha sido poco estudiado, por lo que en esta tesis introducimos una definición formal de este problema, así como algoritmos que lo resuelven.

Este trabajo está organizado de la siguiente manera:

En el Capítulo 1 se da una introducción al Reconocimiento de Patrones, se mencionan los problemas que se estudian dentro de esta línea de investigación, se ubica el problema de agrupamiento conceptual y se presentan los conceptos y definiciones básicas necesarias para entender los algoritmos que se proponen en esta tesis.

En el Capítulo 2 se hace un estudio de los principales algoritmos de clasificación no supervisada restringida basados en semillas, tanto en el enfoque clasificatorio como en el enfoque conceptual. Estos algoritmos conforman el estado del arte relacionado con el problema de agrupamiento conceptual restringido.

En el Capítulo 3 se da un planteamiento formal del problema de agrupamiento conceptual restringido duro, se propone una función para evaluar la calidad de los conceptos y se introducen nuevos algoritmos de agrupamiento conceptual duro. Estos algoritmos constan de dos fases: una fase de agrupamiento, en la que se forman los agrupamientos en que se estructura la muestra de datos y una fase de caracterización, en la que se generan los conceptos o propiedades que caracterizan a los agrupamientos. Los algoritmos propuestos utilizan, en la fase de agrupamiento, el algoritmo k-means con funciones de similaridad. Mientras que en la fase de caracterización utilizan diferentes estrategias para generar los conceptos.

En el Capítulo 4 se da un planteamiento formal del problema de agrupamiento conceptual restringido difuso, se propone una función para evaluar la calidad de los conceptos difusos y se introduce una versión difusa de los algoritmos propuestos en el Capítulo 3. Estos algoritmos, al igual que los algoritmos duros, constan de dos fases: una fase de agrupamiento en la que se forman los agrupamientos difusos, y una fase de caracterización en la que se generan los conceptos que caracterizan a los agrupamientos difusos.

Finalmente se exponen nuestras conclusiones, se enlistan las aportaciones de esta tesis y se proponen algunas direcciones para el trabajo futuro.

Capítulo 1

Conceptos Preliminares

En este capítulo se presenta una introducción al Reconocimiento de Patrones, se ubica el problema de agrupamiento conceptual, y se dan los conceptos y definiciones básicas necesarias para poder entender los algoritmos que se presentan en esta tesis.

1.1. Introducción

El Reconocimiento de Patrones proporciona herramientas importantes para el análisis de datos, la toma de decisiones, la clasificación y el pronóstico. El Reconocimiento de Patrones también puede ser descrito como un proceso de reducción, mapeo o etiquetación de la información (Schalkoff, 1992).

Por problemas de Reconocimiento de Patrones entenderemos todos aquellos relacionados con la clasificación de objetos y fenómenos, y la determinación de los factores que inciden en ellos.

Existen diferentes enfoques para resolver los problemas de Reconocimiento de Patrones; los más estudiados son: el Estadístico (Escudero, 1977; Fukunaga, 1990; Schalkoff, 1992), el Sintáctico Estructural (Fu, 1974; Schalkoff, 1992) y el Lógico Combinatorio (Ruiz-Shulcloper et al., 1999; Martínez-Trinidad and Guzmán-Arenas 2001), aunque también se reportan trabajos en esta dirección utilizando otras técnicas como Redes Neuronales (Schalkoff, 1992; Pal and Mitra, 1999) y Algoritmos Genéticos (Pal and Wang, 1996). El presente trabajo se enmarca en el Reconocimiento Lógico Combinatorio de Patrones.

1.1.1. Reconocimiento Lógico Combinatorio de Patrones

La Lógica Matemática, la Teoría de Testores (Alba-Cabrera, 1997; Lazo-Cortés et al., 2001), la Teoría Clásica de Conjuntos, la Teoría de Subconjuntos Difusos, la Teoría Combinatoria y la Matemática Discreta en general, constituyen la base teórico-matemática sobre la que se desarrolla el denominado Reconocimiento Lógico Combinatorio de Patrones.

La representación de objetos usualmente es considerada como una secuencia de valores exclusivamente numéricos o exclusivamente categóricos (Booleanos, nominales, en general valores discretos). Sin embargo, cuando observamos los problemas de Reconocimiento de Patrones en situaciones prácticas, encontramos que existen problemas en los que, en la descripción de los objetos, aparecen de manera simultánea ambos tipos de valores; además, se pueden tener descripciones de objetos incompletas. Esto es, estaríamos tratando con descripciones de objetos con información mezclada e incompleta. Este tipo de información aparece frecuentemente en disciplinas, tales como: Medicina, Geociencias, Sociología, Psicología, Biología, Ciencias Políticas, Economía, Criminalística, entre otras.

Un ejemplo interesante aparece en Medicina. La representación del conocimiento involucrado en el proceso de diagnóstico médico o pronósticos de salud no es un problema trivial. Normalmente, un médico evalúa signos y síntomas del paciente para establecer un diagnóstico o un pronóstico. Los signos como: la temperatura, la presión sanguínea, edad, número de hijos, entre otros, son usualmente valores numéricos. Sin embargo, hay signos y síntomas que no pueden ponerse en correspondencia con un número, pero sí con un código, por ejemplo: la palidez, transpiración, dolor, entre otros. Cuando se presentan valores numéricos y no numéricos mezclados, los objetos se representan en un producto cartesiano sin alguna propiedad algebraica, lógica o topológica asumida sobre el espacio de representación. ¿Cómo seleccionar entonces los atributos más informativos o relevantes en términos de la información que proporcionan? ¿cómo clasificar un nuevo objeto? o ¿cómo encontrar las relaciones entre todos los objetos basados en cierta medida de similaridad? Estos son algunos de los problemas abordados por el Reconocimiento Lógico Combinatorio de Patrones. Este enfoque es una respuesta metodológica al hecho de que en problemas de

Reconocimiento de Patrones las descripciones de los objetos contienen información mezclada o con ausencia de información.

En el Reconocimiento Lógico Combinatorio de Patrones se busca modelar los problemas prácticos de una manera más cercana al problema específico que se desea resolver. Este enfoque se caracteriza por lograr una mejor adaptación de sus modelos a las características de los problemas prácticos. Por esta razón, en el presente trabajo se utilizará el Reconocimiento Lógico Combinatorio de Patrones para abordar el problema de Reconocimiento de Patrones con información mezclada e incompleta.

1.2. Problemas de Reconocimiento de Patrones

En el área de Reconocimiento de Patrones se estudian las siguientes familias de problemas, aunque no son los únicos:

1. Selección de atributos.
2. Clasificación supervisada.
3. Clasificación parcialmente supervisada.
4. Clasificación no supervisada.

El problema de la **selección de atributos** consiste en seleccionar del conjunto completo de atributos, aquellos que son relevantes. Tiene dos propósitos fundamentales:

a) Reducir el número de atributos en términos de los cuales se deben describir los objetos para la clasificación (Fukunaga, 1990).
b) Encontrar los atributos que inciden en el problema de manera determinante y en general la forma en que cada uno de estos atributos incide en el problema (Ruiz-Shulcloper et al., 1999).

En la **clasificación supervisada** se conoce que un universo de objetos se agrupa en un número dado de clases, de cada una de las cuales se tiene una muestra de objetos que se sabe pertenecen a ella. El problema consiste en que, dado un nuevo objeto, establecer su relación con cada una de dichas clases.

El problema de la **clasificación parcialmente supervisada** es análogo al de la clasificación supervisada, excepto en que hay una o varias clases de objetos de las que no se tiene muestra; pero el problema sigue siendo el mismo: dado un nuevo objeto, relacionarlo con las clases.

En un problema de **clasificación no supervisada** no se conoce cómo se agrupan los objetos, siendo éste justamente el objetivo que se persigue.

En esta tesis se aborda el problema de clasificación no supervisada. Por lo que, en la siguiente sección, se describe de manera más detallada en qué consiste este problema.

1.3. Clasificación no Supervisada

El problema de clasificación no supervisada puede verse como un proceso de dividir o estructurar un conjunto de objetos en grupos que mantengan alguna relación entre sí. Por tanto, el problema consiste en obtener los agrupamientos en que se estructuran los objetos de una muestra dada. En este tipo de problemas se tienen dos variantes:

- *Clasificación no supervisada libre*: Cuando no se especifica el número de agrupamientos a formar.

- *Clasificación no supervisada restringida*: Cuando se especifica *a priori* el número de agrupamientos a formar.

Con base en los trabajos que se han propuesto para resolver el problema de clasificación no supervisada pueden considerarse tres paradigmas de solución:

- *Paradigma del conjunto cociente.* Consiste en la formación de una partición del conjunto de objetos, bajo el presupuesto de que los mismos serán conjuntos en el sentido clásico de la Teoría de Conjuntos.

- *Paradigma del solapamiento.* Consiste en generar un cubrimiento del conjunto de objetos por subconjuntos no necesariamente disjuntos, es decir, permite que las agrupaciones tengan elementos comunes.

- *Paradigma difuso.* Parte del supuesto de que no se puede afirmar categóricamente que los objetos pertenecen o no a un conjunto dado, sino sólo se puede hablar de grados de pertenencia. Por lo tanto, consiste en construir una partición o un cubrimiento difuso del conjunto de objetos.

Los paradigmas que más se han utilizado para resolver el problema de clasificación no supervisada son el paradigma del conjunto cociente y el paradigma difuso. Esta tesis se ubica en estos paradigmas.

Por otro lado, podemos encontrar diversas clasificaciones (Escudero, 1977; Aldenderfer and Blashfield, 1984) de los métodos de agrupamiento. Algunas de estas clasificaciones son las siguientes:

- *Métodos de reagrupamiento y jerárquicos*. Se considera que un método de clasificación no supervisada es de reagrupamiento, cuando se parte de agrupamientos iniciales los cuales se van refinando de manera iterativa, de tal forma que se maximice alguna medida de similitud entre objetos que pertenecen al mismo agrupamiento. Los métodos jerárquicos tienen por objetivo aglomerar agrupamientos para formar uno nuevo, o bien separar agrupamientos formando nuevos subagrupamientos. En los métodos jerárquicos, los objetos se van agrupando formando subfamilias a partir de las cuales se pueden estudiar las cualidades comunes a los patrones que se agrupan en un determinado nivel. Los métodos jerárquicos se pueden subdividir a su vez en *métodos aglomerativos* y *métodos divisivos*. Los métodos aglomerativos son aquellos en los que se parte de agrupamientos formados por objetos individuales, procediendo en cada nivel a fusionar aquellos dos agrupamientos que sean más similares, hasta llegar a tener la muestra completa en un solo agrupamiento. Los métodos divisivos son aquellos en los que se parte de la muestra completa como un solo agrupamiento y se van obteniendo en cada

nivel dos nuevos agrupamientos, de modo tal que se maximice una medida de divergencia preestablecida.

- *Métodos tipológicos*. Los métodos tipológicos, aunque también son jerárquicos, se diferencian de éstos en que tienen en cuenta una característica a la vez, tanto en la agrupación como en la separación de objetos. Mientras que los métodos jerárquicos contemplan simultáneamente todos los atributos, de cada objeto, para unir o separar agrupamientos según la mayor similitud o divergencia de los mismos.

- *Métodos con agrupamiento solapado y exclusivo*. En los métodos con agrupamiento solapado se admite que un objeto pueda formar parte simultáneamente de más de un agrupamiento. En cambio, en los métodos con agrupamiento exclusivo, si un objeto pertenece a un agrupamiento no puede pertenecer a otro.

- *Métodos directos y métodos iterativos*. Los métodos directos se caracterizan por utilizar algoritmos que operan de modo tal, que una vez que asignan un objeto a un agrupamiento, no lo remueven del mismo. Por el contrario, los métodos iterativos corrigen sus propias asignaciones, comprobando en cada iteración si la asignación de la muestra total es óptima. Si no lo es, realizan un nuevo agrupamiento.

- *Métodos adaptivos y no adaptivos*. Los métodos no adaptivos son aquellos en los que el algoritmo se encamina directa o iterativamente a la solución, el método de agrupamiento es fijo y está predeterminado de antemano. En cambio, los métodos adaptivos contemplan en su ejecución un cambio de medida de similitud o de criterio a optimizar dependiendo de los resultados parciales obtenidos.

- *Métodos basados en semillas y métodos que no requieren semillas*. Los métodos de agrupamiento basados en semillas son métodos que requieren de semillas iniciales para formar los agrupamientos, las cuales se van ajustando de manera iterativa hasta obtener las mejores semillas para los agrupamientos. En los métodos que no requieren semillas, se van tomando uno a uno los objetos y se colocan en el mismo agrupamiento si

cumplen con cierta función de similitud, en caso contrario, estarán en agrupamientos diferentes. Este procedimiento se realiza hasta agrupar a todos los objetos.

Estas clasificaciones de los métodos de agrupamiento, nos permiten tener una idea global de las formas en que se puede obtener la solución del problema de clasificación no supervisada.

Los algoritmos que se proponen en esta tesis se clasifican como métodos de reagrupamiento, iterativos, basados en semillas.

Para resolver el problema de clasificación no supervisada, además de los paradigmas de agrupamiento y la clasificación de los métodos de agrupamiento, se toman en consideración dos enfoques:

- *Enfoque clasificatorio*, se tiene un universo de objetos y se desea agruparlos de tal manera que los objetos que estén en el mismo agrupamiento se parezcan más entre sí que a objetos de otros agrupamientos. El objetivo es encontrar los objetos que, dadas sus relaciones de semejanza, deben estar en un mismo agrupamiento.

- *Enfoque conceptual*, a partir de un universo de objetos se desea conocer no sólo los agrupamientos en que se estructuran los objetos, sino además las propiedades que caracterizan a dichos agrupamientos.

El estudio de la clasificación no supervisada ha sido enfocado principalmente al enfoque clasificatorio, es decir, a determinar diferentes medidas de proximidad (Fukunaga, 1990; Bezdek, 1992) o medidas de similaridad entre objetos (Ruiz-Shulcloper et al., 1999; Martínez-Trinidad and Guzmán-Arenas, 2001) y a desarrollar herramientas que las utilicen. Los métodos de clasificación no supervisada, consisten en determinar agrupamientos tales que objetos en el mismo agrupamiento sean muy similares entre ellos, mientras que objetos de diferentes agrupamientos sean poco similares. En estos métodos no se da una descripción de los agrupamientos. Sin embargo, para algunos problemas prácticos se

requiere, además de determinar los agrupamientos, conocer las propiedades que describan cómo son dichos agrupamientos. Para resolver este problema, se introdujo el agrupamiento conceptual, el cual se describe a continuación.

1.4. Agrupamiento Conceptual

Como se mencionó anteriormente, el problema de agrupamiento conceptual consiste en determinar no sólo los agrupamientos en que se estructuran los objetos de una muestra dada, sino además los conceptos o propiedades que caracterizan a dichos agrupamientos.

Una de las problemáticas a la que hay que enfrentarse al desarrollar algoritmos de agrupamiento conceptual es el manejo de información mezclada. Dentro de las soluciones que se han propuesto a este problema se tienen las siguientes variantes:

- Codificar valores de atributos nominales como valores enteros numéricos, y aplicar las medidas de distancia que se usan para los atributos numéricos. La codificación de datos nominales mediante números no tendría mayores dificultades si no se realizaran sobre ellos operaciones aritméticas las cuales no tienen sentido, ya que éstas no están definidas para este tipo de valores. En los casos en que este hecho es violado, los valores de similaridad no son interpretables.

- Discretizar atributos numéricos y aplicar algoritmos que manejen sólo información cualitativa. El proceso de discretización generalmente causa pérdida de información importante, especialmente la referente a la diferencia relativa entre valores para los atributos numéricos. Además de que el problema original se modifica o cambia de espacio de representación, lo que puede en algunas ocasiones complicar la interpretación de los resultados.

- Generalizar funciones de comparación para manejar atributos numéricos y no numéricos. Las funciones usadas para atributos numéricos están basadas en distancias que no pueden usarse directamente para manejar atributos cualitativos por estar éstos en un espacio diferente. Varios intentos o propuestas violan este hecho al calcular la

distancia total como la suma de las distancias de los atributos numéricos más la distancia de los atributos cualitativos (considerando los códigos 0 y 1 como números) y asumir que el resultado tiene interpretación en el espacio original. Además, el cálculo de promedios en este espacio pierde sentido.

Los primeros algoritmos conceptuales fueron propuestos por Michalski y a partir de éstos se han desarrollado diversos algoritmos, los cuales pueden dividirse en dos tipos: restringidos y no restringidos. Los algoritmos conceptuales restringidos son aquellos en los que se especifica *a priori* el número de agrupamientos y conceptos a formar y usualmente requieren de semillas para su funcionamiento, mientras que los algoritmos conceptuales no restringidos son aquellos en los que no se especifica el número de agrupamientos y conceptos.

Los principales algoritmos conceptuales restringidos basados en semillas son: CLUSTER/PAF (Michalski and Diday, 1981), CLUSTER/2 (Michalski and Stepp, 1983), CLUSTER/S (Stepp and Michalski, 1986), CLUSTER/3 (Seeman and Michalski, 2006), WITT (Hanson, 1990) y k-means conceptual (Ralambondrainy, 1995).

Dentro de los algoritmos conceptuales no restringidos se presenta una subdivisión:

i) algoritmos incrementales, entre los que se encuentran: EPAM (Feigenbaum, 1963), UNIMEM (Lebowitz, 1986), COBWEB (Fisher, 1990; Gennari et al., 1990), CLASSIT (Gennari et al., 1990), COBWEB/3 (McKusick and Thompson, 1990) y LINNEO$^+$ (Béjar and Cortés, 1992).

ii) algoritmos no incrementales, entre los que se encuentran los algoritmos LC-conceptual (Martínez-Trinidad and Ruiz-Shulcloper, 1999; Martínez-Trinidad and Sánchez-Díaz, 2001) y RGC (Pons-Porrata, 1999; Pons-Porrata et al., 2002). Así como otros trabajos relacionados con agrupamiento conceptual (Briscoe and Caelli, 1996; Stumme et al., 2001; Jonyer et al., 2001; Osinski and Weiss, 2004; Mishra et al., 2004; Jänichen and Perner, 2005) y con la Teoría de Conceptos Formales (Stumme, 2002; Valtchev et al., 2004; Cimiano et al., 2004). También se tienen otros algoritmos que no son

conceptuales pero que construyen ciertas propiedades de los agrupamientos (Gowda and Diday, 1991; Gowda and Diday, 1992).

Todos estos algoritmos construyen agrupamientos duros, es decir, determinan si un objeto está o no en un agrupamiento. Sin embargo, en algunos problemas práctios los especialistas están interesados en determinar en qué grado un objeto pertenece a un agrupamiento, más que en decidir si un objeto pertenece o no a dicho agrupamiento. Además, los especialistas requieren de una interpretación conceptual de dichos agrupamientos difusos.

Existen algunos trabajos que resuelven el problema de agrupamiento conceptual difuso, entre los que se encuentran: LC-conceptual difuso (Martínez-Trinidad and Ruiz-Shulcloper, 1998), un modelo de estructuración conceptual difuso propuesto por Martínez-Trinidad (2000) y trabajos relacionados con la Teoría de Conceptos Formales Difusos (Quan et al., 2004a; Quan et al., 2004b). Estos trabajos resuelven el problema de agrupamiento conceptual difuso cuando no se conoce el número de agrupamientos. Sin embargo, hasta el momento no se ha estudiado el problema de agrupamiento conceptual restringido difuso. Por esta razón, en esta tesis introducimos una formalización del agrupamiento conceptual restringido difuso, que se expone en la siguiente sección.

1.5. Agrupamiento Conceptual Difuso

En la vida diaria está presente la información difusa. Todos asimilamos y utilizamos datos difusos, reglas vagas e información imprecisa. Por consiguiente, se desea que los sistemas puedan reconocer, representar, manipular, interpretar y utilizar incertidumbre estadística y difusa. Los modelos estadísticos se ocupan de acontecimientos y de resultados al azar; mientras que los modelos difusos procuran capturar y cuantificar la imprecisión no aleatoria.

Es por esto que uno de los problemas que se desean resolver dentro del Reconocimiento de Patrones es el de estimar, a partir de los datos observados, el grado de pertenencia de un objeto a un agrupamiento difuso, más que el de decidir si un objeto dado

pertenece o no a un agrupamiento duro. Ésta es una de las razones por las que la Teoría de Conjuntos Difusos ha sido utilizada para resolver problemas de Reconocimiento de Patrones.

La Teoría de Conjuntos Difusos fue introducida por Lofti A. Zadeh (1965) como una nueva forma de representar la imprecisión en la vida diaria. Esta teoría es una generalización de la Teoría Clásica de Conjuntos.

El uso de conjuntos difusos en clasificación no supervisada fue sugerido primero por Bellman, Kalaba y Zadeh (1966). Posteriormente, Negoita (1973) usó un teorema de separación de conjuntos difusos para describir un sistema de recuperación de información basado en agrupamiento. Por otro lado, Ruspini (1969, 1973) introdujo una noción de partición difusa para describir la estructura del agrupamiento de un conjunto de datos y sugirió un algoritmo para computar la partición difusa óptima. Dunn (1974) generalizó el procedimiento de agrupamiento mínima-variación a una técnica de agrupamiento ISODATA Difuso. Finalmente, Bezdek (1973, 1981) generalizó el enfoque de Dunn para obtener una familia infinita de algoritmos conocida como algoritmos c-means difusos (FCM).

Los algoritmos antes mencionados permiten resolver el problema de agrupamiento no supervisado difuso cuando los objetos están descritos en términos de atributos numéricos, y se cuenta con métricas para la comparación entre los objetos. Sin embargo, como se mencionó anteriormente, en algunas disciplinas se trabaja con descripciones de objetos con datos mezclados e incompletos. Para este tipo de descripciones aplicar una función de distancia para medir el parecido entre ellos no siempre es lo mejor. Recientemente se han propuesto diversas extensiones del algoritmo c-means difuso para trabajar con este tipo de descripciones de objetos. Estos algoritmos permiten resolver el problema de agrupamiento no supervisado difuso cuando se tienen datos mezclados e incompletos; sin embargo, no dan una interpretación conceptual de los agrupamientos difusos.

Hasta el momento, el problema de agrupamiento conceptual difuso ha sido poco estudiado (Martínez-Trinidad and Ruiz-Shulcloper, 1998; Martínez-Trinidad, 2000; Quan et al., 2004a; Quan et al., 2004b). El estudio de este problema se ha orientado a resolver problemas para los cuales no se conoce el número de agrupamientos. Sin embargo, existen problemas para los cuales se da *a priori* el número de agrupamientos que se desean formar; este tipo de problemas no se han estudiado hasta el momento. Por esta razón, en el Capítulo 4 se propone una formalización del problema de agrupamiento conceptual restringido difuso, se introduce una función para evaluar la calidad de los conceptos difusos y se proponen algoritmos para resolverlo.

A continuación se dan algunos conceptos y definiciones básicas.

1.6. Conceptos y Definiciones

Para entender el funcionamiento de los algoritmos que se proponen en esta tesis es necesario dar algunos conceptos y definiciones básicas.

Consideremos un conjunto $T = \{O_1,...,O_n\}$ de n objetos. Cada objeto descrito por un conjunto $R = \{x_1,...,x_m\}$ de m atributos. Cada atributo x_s toma valores en un conjunto de valores admisibles D_s, $x_s(O_j) \in D_s$, $s = 1,...,m$ y $j = 1,\ldots,n$. Además, asumiremos que en D_s existe un símbolo "?" para denotar ausencia de información, por lo que pueden ser consideradas descripciones de objetos incompletas. El tratamiento que se da, en esta tesis, a la ausencia de información es el siguiente: cuando el valor del atributo x_s es ausencia ("?") entonces el valor es diferente de cualquier otro, incluso de otra ausencia.

Para representar a los objetos se define un *espacio de representación* (ER) el cual no es más que el producto cartesiano de los conjuntos D_s:

$$O_j = (x_1(O_j),...,x_m(O_j)) \in (D_1 \times ... \times D_m)$$

siendo $x_s(O_j)$ el valor del atributo x_s en el objeto O_j.

Usualmente, esta información acerca de los objetos (sus descripciones) está dada en forma de una tabla o matriz $MA = \left| x_s(O_j) \right|_{n \times m}$, con *n* renglones (descripciones de objetos) y *m* columnas (valores de cada atributo en esos objetos).

Para cada atributo se define una función de comparación (Ruiz-Shulcloper et al., 1999), la cual está definida de la siguiente manera.

Definición 1.1: Sea L_s un conjunto totalmente ordenado (Grimaldi, 1998). Sobre D_s, el conjunto de valores admisibles del atributo x_s, definimos una función $FC_s : D_s \times D_s \rightarrow L_s$, que denominaremos ***función de comparación de valores de x_s***, tal que:

i) $FC_s(x_s(O_j), x_s(O_j)) = \min_{O_p \in X} \{FC_s(x_s(O_j), x_s(O_p))\}$ si FC_s es una función de comparación de disimilaridad entre valores de x_s, para $s = 1,\ldots,m$.

ii) $FC_s(x_s(O_j), x_s(O_j)) = \max_{O_p \in X} \{FC_s(x_s(O_j), x_s(O_p))\}$ si FC_s es una función de comparación de similaridad entre valores de x_s, para $s = 1,...,m$.

La función FC_s nos ofrece el grado de similaridad entre dos valores cualesquiera del atributo x_s, para $s = 1,...,m$.

Algunos tipos de funciones de comparación de valores de un atributo atendiendo a L_s son:

- Si $L_s = \{0,1\}$, FC_s es una función de comparación Booleana.
- Si $L_s = \{0,1,...,k\text{-}1\}$, FC_s es una función de comparación *k*-valente.
- Si $L_s \subseteq \Re$, el conjunto de los números reales, FC_s es una función de comparación real.

Ejemplos de funciones de comparación, para valores de un atributo:

1. $FC_s\left(x_s\left(O_j\right),x_s\left(O_p\right)\right)=\begin{cases}0 & si\ \ x_s\left(O_j\right)\neq x_s\left(O_p\right)\vee x_s\left(O_j\right)=?\vee x_s\left(O_p\right)=?\\ 1 & en\ otro\ caso\end{cases}$

   0 significa que los valores son diferentes y 1 que son coincidentes. Ésta es una función de comparación Booleana.

2. $FC_s\left(x_s\left(O_j\right),x_s\left(O_p\right)\right)=1-\left|\,x_s\left(O_j\right)-x_s\left(O_p\right)\right|$

   Para usar esta función de comparación es necesario que los valores de los atributos sean numéricos. Ésta es una función de comparación real.

3. $FC_s\left(x_s\left(O_j\right),x_s\left(O_p\right)\right)=\begin{cases}1 & si\ \ x_s\left(O_j\right),\ x_s\left(O_p\right)\in B_{s_1}\\ 2 & si\ \ x_s\left(O_j\right),\ x_s\left(O_p\right)\in B_{s_2}\\ \vdots & \\ k-1 & si\ \ x_s\left(O_j\right),x_s\left(O_p\right)\in B_{s_{k-1}}\\ 0 & en\ otro\ caso\end{cases}$

   donde $B_{s_s}\subset D_s$; el valor de x_s no necesariamente es numérico. Ésta es una función de comparación *k*-valente.

También es necesario definir una función que permita comparar descripciones de objetos. Es de esperar que cuando comparamos dos descripciones de objetos, la similaridad entre los mismos sea una función de lo que ocurra entre los atributos que los describen. Esto es, entre las funciones de comparación de valores de los atributos y las funciones de similaridad existirá una relación, en la que las primeras influyan sobre las segundas. Una función de similaridad (Ruiz-Shulcloper et al., 1999) se define de la siguiente manera:

Definición 1.2: Sobre $D_{s_1}\times\ldots\times D_{s_s}$ definimos una función $\Gamma:\left(D_{s_1}\times\ldots\times D_{s_q}\right)^2\rightarrow L$, donde $\left(D_{s_1}\times\ldots\times D_{s_q}\right)^2=\left(D_{s_1}\times\ldots\times D_{s_q}\right)\times\left(D_{s_1}\times\ldots\times D_{s_q}\right)$ y L es un conjunto totalmente ordenado. Γ es una ***función de similaridad*** con denominaciones análogas a FC_s en dependencia de L, la cual cumple $\Gamma\left(O_j,O_j\right)=\max\limits_{O_j,O_p\in X}\left\{\Gamma\left(O_j,O_p\right)\right\}$.

Sea $\Gamma(O_j, O_p)$ la similaridad entre O_j y O_p, en este trabajo se asume que $\forall O_j, O_p$:

1. $\Gamma(O_j, O_p) \in [0,1]$ *para* $1 \le j \le n, 1 \le p \le n$
2. $\Gamma(O_j, O_j) = 1$ *para* $1 \le j \le n$
3. $\Gamma(O_j, O_p) = \Gamma(O_p, O_j)$ *para* $1 \le j \le n, 1 \le p \le n$

Ejemplos de funciones de similaridad:

1. $$\Gamma(O_j, O_p) = \begin{cases} 1 & si \quad \left| \sum_{s=1}^{m} \{x_s \mid FC_s(x_s(O_j), x_s(O_p)) = 0\} \right| \le \varepsilon \\ 0 & en\ otro\ caso \end{cases}$$

siendo FC_s una función de comparación Booleana y ε un umbral dado por el usuario. Aquí las descripciones de dos objetos son semejantes si el número de atributos diferentes no excede un umbral determinado (ε).

2. $$\Gamma(O_j, O_p) = \begin{cases} 1 & si \quad \left| \sum_{s=1}^{m} \{x_s \mid FC_s(x_s(O_j), x_s(O_p)) = 0\} \right| \le \varepsilon_2 \\ & \quad \wedge \quad \left| \sum_{s=1}^{m} \{x_s \mid FC_s(x_s(O_j), x_s(O_p)) = 1\} \right| \ge \varepsilon_1 \\ 0 & en\ otro\ caso \end{cases}$$

siendo FC_s una función de comparación Booleana, ε_1 y ε_2 parámetros que regulan respectivamente la cantidad máxima admisible de atributos diferentes y la cantidad mínima de atributos coincidentes (semejantes), es decir, dos objetos serán semejantes si la cantidad de atributos coincidentes es superior que ε_1 y la cantidad de atributos diferentes no excede ε_2.

3. $$\Gamma(O_j, O_p) = \begin{cases} 1 & si\ un\ \%\ de\ rasgos\ coincidentes\ es \ge \lambda\% \\ 0 & en\ otro\ caso \end{cases}$$

siendo λ un parámetro de umbral dado por el usuario.

4. $\Gamma(O_j, O_p) = 1 - \dfrac{\sum_{x_s \in R} FC_s\left(x_s(O_j), x_s(O_p)\right)}{|R|}$

Aquí, la medida de similaridad está dada por la suma de la semejanza de los atributos, normalizada por el número de atributos en consideración.

Para representar a los agrupamientos se utiliza una matriz de dimensión $k \times n$, donde k representa el número de agrupamientos y n el número de objetos. Esta matriz será una k-partición difusa si cumple las propiedades de la Definición 1.3.

Definición 1.3*:** Una matriz $U_{k\times n} = \lfloor u_{ij} \rfloor$ representa una ***k-partición difusa (Ruspini, 1969) de T cuando y sólo cuando:

i) $u_{ij} \in [0,1]$ *para* $1 \le i \le k, \quad 1 \le j \le n$

ii) $\sum_{i=1}^{k} u_{ij} = 1$ *para* $1 \le j \le n$

iii) $0 < \sum_{j=1}^{n} u_{ij}$ *para* $1 \le i \le k$

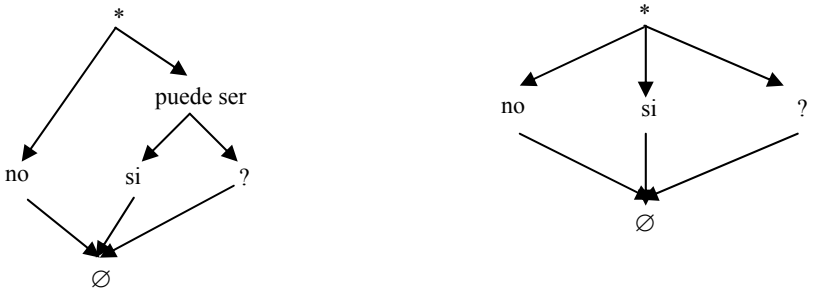
donde u_{ij} representa el grado de pertenencia del objeto O_j al agrupamiento A_i. La primera condición nos dice que el grado de pertenencia del objeto O_j al agrupamiento A_i es un valor entre 0 y 1. Cuando $u_{ij} \in \{0,1\}$, la matriz $U_{k\times n} = \lfloor u_{ij} \rfloor$ representa una ***k-partición dura*** (Ruspini, 1969). La segunda condición nos dice que la suma de los grados de pertenencia de un objeto a los agrupamientos debe ser 1 y la última condición nos garantiza que los agrupamientos sean no vacíos.

Los retículos de generalización (Ralambondrainy, 1995) han sido utilizados para construir los conceptos o propiedades que caracterizan a los agrupamientos. Estos retículos son definidos para cada uno de los atributos, de la siguiente manera.

Definición 1.4*:** Un ***retículo de generalización (Ralambondrainy, 1995) es una estructura $L = (E, \le, \vee, \wedge, *, \varnothing)$, donde E es un conjunto de elementos llamado *espacio de búsqueda*, $\le$ es una relación de orden parcial "*es menos general que*", la cual redefine la

relación de inclusión como sigue: $\forall e, f \in E, e \leq f \Rightarrow e \subseteq f$, el símbolo * denota el mayor elemento de E y es interpretado como "*todos los valores son posibles*" y $\varnothing$ denota el elemento mínimo de E y es interpretado como "*ningún valor es posible*". Cada par (e,f) tiene un mínimo límite superior que es denotado por $e \vee f$ llamado también la generalización de e y f, y un máximo límite inferior de e y f denotado por $e \wedge f$.

Ejemplos de retículos de generalización:



1.7. Sumario

En este capítulo se dio una introducción al Reconocimiento de Patrones, se describieron los tipos de problemas que se abordan dentro de esta línea de investigación y se mencionaron los distintos enfoques que se han utilizado para resolver este tipo de problemas. Se describió en qué consisten los problemas de clasificación no supervisada, de agrupamiento conceptual y agrupamiento conceptual difuso, mostrando las principales problemáticas a las que hay que enfrentarse al resolver este tipo de problemas. Finalmente, se introdujeron algunos conceptos y definiciones básicas necesarias para entender los algoritmos, tanto duros como difusos, que se proponen en esta tesis.

Capítulo 2

Estado del Arte

En esta tesis se desea resolver, desde un enfoque conceptual, el problema de clasificación no supervisada restringida, es decir, cuando se da *a priori* el número de agrupamientos a formar. En este capítulo se presentan los trabajos relacionados con esta línea de investigación.

Como se mencionó anteriormente, el problema de clasificación no supervisada restringida ha sido estudiado principalmente desde un enfoque clasificatorio; por otra parte, los trabajos que se han propuesto en el enfoque conceptual restringido toman como base la filosofía de los algoritmos de clasificación no supervisada. Por lo cual, se considera importante estudiar primero el enfoque clasificatorio.

2.1. Enfoque Clasificatorio

Una de las herramientas más estudiadas y utilizadas para resolver el problema de clasificación no supervisada restringida es el algoritmo k-means (Bezdek, 1981). El objetivo de este algoritmo es construir agrupamientos en los cuales objetos muy cercanos pertenezcan a un mismo agrupamiento, mientras que objetos lejanos pertenezcan a agrupamientos diferentes. Las descripciones de los objetos están dadas en términos de atributos numéricos, no se permite ausencia de información en los datos y se utiliza una función de distancia para comparar el parecido entre objetos. Ésta es una restricción fuerte cuando se tienen descripciones de objetos en términos de atributos numéricos y no numéricos mezclados, ya que en diversas ocasiones no se tiene una métrica para evaluar el parecido entre los objetos, por lo cual es necesario definir una función de similaridad.

A partir de esto se han propuesto diversas generalizaciones de las funciones de comparación que permiten trabajar con descripciones que contengan atributos numéricos y no numéricos mezclados (Gowda and Diday, 1991; Gowda and Diday, 1992; Hathaway et al., 1996; El-Sonbaty and Ismail, 1998; Ravi and Gowda, 1999; García-Serrano and Martínez-Trinidad, 1999; Ayaquica-Martínez and Martínez-Trinidad, 2001; Miin-Shen et al., 2004).

En este capítulo se presentan únicamente los algoritmos que se utilizarán como base para proponer nuevos algoritmos conceptuales restringidos basados en semillas.

2.1.1. Algoritmo k-means con funciones de similaridad (KMSF)

En este trabajo se propone utilizar funciones de similaridad para evaluar el parecido entre objetos. Los objetos de estudio pueden estar descritos por atributos cuantitativos y cualitativos mezclados, además se permite ausencia de información. Las funciones de similaridad se definen tomando en cuenta el tipo de atributo y la forma en que se comparan los valores de ese atributo dependiendo del contexto del problema a resolver.

El objetivo del algoritmo KMSF (García-Serrano and Martínez-Trinidad, 1999) es obtener agrupamientos tales que objetos muy similares pertenezcan al mismo agrupamiento, mientras que objetos poco similares estén en agrupamientos diferentes, es decir, se desea obtener una *k*-partición dura (U) optimizando la función objetivo

$$J_m(U,\vartheta) = \sum_{i=1}^{k}\sum_{p=1}^{n} u_{ip}\Gamma\left(O_i^r, O_p\right)$$

donde $\Gamma\left(O_i^r, O_p\right)$ es la similaridad entre el centroide O_i^r del agrupamiento A_i y el objeto O_p; y u_{ip} es 1 si el objeto pertenece al agrupamiento A_i y 0 si no pertenece. En este caso el centroide es un objeto de la muestra, al cual se le denomina objeto representativo, a diferencia del k-means original que utiliza la media.

Para seleccionar el objeto representativo O_i^r del agrupamiento A_i, se introduce la siguiente expresión:

$$r_{\substack{A_i \\ O_j \in A_i}}(O_j) = \frac{\beta_{A_i}(O_j)}{\left(\alpha_{A_i}(O_j) + \left(1 - \beta_{A_i}(O_j)\right)\right)} + \eta_{A_i}(O_j) \tag{2.1}$$

A continuación se describe cada una de las partes de la expresión (2.1).

La expresión $\beta_{A_i}(O_j)$ evalúa el promedio de similaridad entre el objeto O_j y los objetos del agrupamiento A_i y se calcula usando (2.2).

$$\beta_{A_i}(O_j) = \frac{1}{|A_i| - 1} \sum_{\substack{O_j, O_q \in A_i \\ O_j \neq O_q}} \Gamma(O_j, O_q) \tag{2.2}$$

con $|A_i| \neq 1$. Cuando $|A_i| = 1$, el centroide de A_i es el objeto contenido en el agrupamiento.

También se introduce la expresión $\alpha_{A_i}(O_j)$.

$$\alpha_{A_i}(O_j) = \frac{1}{|A_i| - 1} \sum_{\substack{O_j, O_q \in A_i \\ O_j \neq O_q}} \left|\beta_{A_i}(O_j) - \Gamma(O_j, O_q)\right| \tag{2.3}$$

esta expresión evalúa la diferencia entre el promedio de similaridad (2.2) y la similaridad entre el objeto O_j y los objetos del agrupamiento A_i. Entonces, cuando (2.3) decrece, los valores de (2.1) crecen.

La expresión $\left(1 - \beta_{A_i}(O_j)\right)$ representa el promedio de disimilaridad de O_j con respecto a los objetos del agrupamiento A_i.

Finalmente, la función (2.4) evalúa la disimilaridad entre el objeto O_j y los objetos representativos de los otros agrupamientos. Esta función se usa para disminuir los casos donde existen dos objetos con el mismo valor en (2.1).

$$\eta_{A_i}(O_j) = \sum_{\substack{q=1 \\ q \neq i}}^{k} \left(1 - \Gamma(O_q^r, O_j)\right) \tag{2.4}$$

Por consiguiente, es razonable que el objeto representativo para el agrupamiento A_i esté definido como el objeto $O \in A_i$ que alcance el máximo $r_{A_i}(O)$, es decir:

$$O_i^r = \max_{O_s \in A_i} \left\{ r_{A_i}(O_s) \right\} \tag{2.5}$$

Cuando el objeto que alcanza el máximo en la expresión (2.5) no es único, entonces se toma el primer objeto que se encuentra.

Por otro lado, la función objetivo $J_m(U,\vartheta)$ se maximiza cuando se obtiene una k-partición U tal que los objetos que pertenecen al mismo agrupamiento son objetos muy similares y objetos que pertenecen a diferentes agrupamientos son poco similares. El valor de u_{ip} se determina como:

$$u_{ip} = \begin{cases} 1 & si\ \Gamma(O_i^r, O_p) = \max\limits_{1 \le q \le c} \left\{ \Gamma(O_q^r, O_p) \right\} \\ 0 & en\ otro\ caso \end{cases} \tag{2.6}$$

Es decir, un objeto O_p será asignado al agrupamiento A_i, si O_p es más similar con el objeto representativo de A_i, que con los objetos representativos de los otros agrupamientos.

2.1.2. Algoritmo k-means difuso con funciones de disimilaridad (FKMDF)

Este algoritmo es una versión difusa del algoritmo k-means con funciones de similaridad, descrito en la sección anterior. Utiliza funciones de disimilaridad que son duales a las funciones de similaridad definidas para el algoritmo KMSF.

El objetivo del algoritmo FKMDF (Ayaquica-Martínez and Martínez-Trinidad, 2001; Ayaquica-Martínez, 2002) es obtener agrupamientos tales que objetos muy similares pertenezcan con alto grado al mismo agrupamiento, mientras que objetos poco similares estén con alto grado en agrupamientos diferentes, es decir, se desea obtener una k-partición difusa optimizando la función objetivo $J_m(U,\vartheta) = \sum_{i=1}^{k} \sum_{p=1}^{n} u_{ip} \Psi(O_i^*, O_p)$, donde $\Psi(O_i^*, O_p)$ es la disimilaridad entre el centroide O_i^* del agrupamiento A_i y el objeto O_p; y u_{ip} es el

grado de pertenencia del objeto O_p al agrupamiento A_i. En este caso, al igual que en el algoritmo KMSF, el centroide es un objeto de la muestra al que se denomina objeto representativo.

El objeto representativo para el agrupamiento A_i será aquel objeto que en promedio sea menos disimilar con los objetos que alcanzan su máximo grado de pertenencia en el agrupamiento A_i.

Para seleccionar el objeto representativo O_i^* del agrupamiento A_i, se introduce la siguiente expresión:

$$O_i^* = \min_{q \in K\alpha_i} \left\{ \sum_{p=1}^{r} u_{ip} \Psi(O_p, O_q) \right\} \qquad (2.7)$$

donde $K\alpha_i = \left\{ O_p \,\middle|\, u_{ip} = \max_{j=1,\ldots,k} \{u_{jp}\} \right\}$ y $r = |K\alpha_i|$ para $i = 1,\ldots,k$.

Para elegir los nuevos objetos representativos O_i^* para cada agrupamiento A_i, $i = 1,\ldots,k$, sólo tomaremos en consideración a los objetos que alcanzan su máximo grado de pertenencia en el agrupamiento A_i (ver $K\alpha_i$). La elección se hace de esta manera para reducir el número de casos en los que un mismo objeto sea seleccionado como objeto representativo de más de un agrupamiento, es decir, el mismo objeto alcance el mínimo en la expresión (2.7) para agrupamientos diferentes. Cuando el objeto que alcanza el mínimo en la expresión (2.7) no es único, entonces se toma el primer objeto que se encuentra.

Por otro lado, la función objetivo $J_m(U,\vartheta)$ se maximiza cuando se obtiene una k-partición difusa tal que los objetos que pertenecen con alto grado al mismo agrupamiento son objetos muy similares y objetos que pertenecen con alto grado a agrupamientos diferentes son poco similares. El valor de u_{ip} se calcula de la siguiente manera:

Para cada objeto O_p se definen los siguientes conjuntos:

$$I_p = \left\{ i \,\middle|\, 1 \le i \le k; \quad \Psi\left(O_p, O_i^*\right) = 0 \right\}$$

$$\bar{I}_p = \{1,...,k\} - I_p$$

Entonces $(U,\vartheta) \in M_{fc} \times \left(M_1 \times ... \times M_m\right)$ será un mínimo para J_m si:

$$I_p = \varnothing \quad \Rightarrow \quad u_{ip} = \frac{1}{\sum_{j=1}^{k} \left(\frac{\Psi\left(O_p, O_i^*\right)}{\Psi\left(O_p, O_j^*\right)} \right)^2}$$

o

$$I_p \neq \varnothing \quad \Rightarrow \quad u_{ip} = 0, \forall i \in \bar{I}_p \quad y \quad \sum_{i \in I_p} u_{ip} = 1 \;\; con \; u_{ip} = \frac{1}{\left| I_p \right|}$$

Es decir, un objeto O_p obtendrá alto grado de pertenencia al agrupamiento A_i, si O_p es menos disimilar con el objeto representativo de A_i, que con los objetos representativos de los otros agrupamientos.

2.1.3. Discusión

En los algoritmos KMSF (García-Serrano and Martínez-Trinidad, 1999) y FKMDF (Ayaquica-Martínez and Martínez-Trinidad, 2001; Ayaquica-Martínez, 2002), se utilizan funciones de similaridad/disimilaridad, las cuales se definen tomando en cuenta el tipo de atributo y la forma en que se comparan los valores de dicho atributo dependiendo del contexto del problema a resolver. Los atributos pueden ser de cualquier naturaleza, cualitativos, cuantitativos, difusos o estructurados. Además, se permite trabajar con ausencia de información en las descripciones de los objetos.

Los algoritmos KMSF y FKMDF son herramientas más flexibles ya que permiten definir diferentes maneras para comparar tanto a los atributos como a los objetos, además es posible adaptar las funciones de similaridad/disimilaridad utilizadas por otros algoritmos de agrupamiento.

En esta tesis se aborda el problema de la caracterización de los agrupamientos obtenidos por los algoritmos de clasificación no supervisada restringida.

2.2. Enfoque Conceptual

A partir de los trabajos propuestos por Michalski (Michalski, 1980; Michalski and Diday, 1981; Michalski, 1983; Michalski and Stepp, 1983; Michalski, 1986; Stepp and Michalski, 1986) se introdujo un conjunto de ideas que dieron origen al concepto de agrupamiento conceptual y una primera propuesta de algoritmos que constituye el punto de partida para esta línea de investigación.

Algunos de los algoritmos conceptuales restringidos basados en semillas que se han propuesto son los siguientes:

2.2.1. Algoritmos CLUSTER

Los algoritmos CLUSTER/PAF (Michalski and Diday, 1981), CLUSTER/2 (Michalski and Stepp, 1983), CLUSTER/S (Stepp and Michalski, 1986) y CLUSTER/3 (Seeman and Michalski, 2006) forman la familia de algoritmos CLUSTER. Estos algoritmos pueden ser aplicados en problemas donde los objetos estén descritos por atributos numéricos, cualitativos y estructurados simultáneamente. No se permite ausencia de información en los datos.

Esta familia está basada en la metodología AQ estrella propuesta por Michalski, la cual construye descripciones de los agrupamientos en forma normal disyuntiva. Para construir las estrellas se procede de la siguiente manera: dado un conjunto de objetos no clasificados, se seleccionan k objetos como semillas de k agrupamientos. Se generan descripciones de cada semilla que sean maximalmente generales y que no cubran ninguna otra semilla. Estas descripciones se usan para determinar los centroides de los nuevos agrupamientos formados. Los centroides se utilizan como nuevas semillas para la siguiente iteración. Como función de distancia entre dos objetos se utiliza el número de atributos que tienen valores diferentes en los objetos. A esta función se le llama distancia sintáctica.

En el algoritmo CLUSTER/2 (Michalski and Stepp, 1983) se proponen mejoras al algoritmo CLUSTER/PAF (Michalski and Diday, 1981). En CLUSTER/2, el agrupamiento

se hace descubriendo condiciones necesarias y suficientes que deben satisfacer los objetos para pertenecer a los agrupamientos. Las descripciones obtenidas son simples y los atributos importantes de los objetos de un agrupamiento son determinados directamente por la definición del agrupamiento. CLUSTER/2 incluye procedimientos para reducir la generalidad de las estrellas y para limitar la complejidad de las descripciones (es decir, limitar el número de sentencias relacionales en un complejo tal que la descripción sea más simple). Una función de evaluación lexicográfica se usa para evaluar, ordenar y seleccionar descripciones alternativas de agrupamientos.

CLUSTER/S (Stepp and Michalski, 1986) extiende CLUSTER/2 de varias maneras. El lenguaje de representación de los objetos y los agrupamientos se mejoró para permitir el cálculo de predicados anotados, los cuales aceptan atributos estructurados (atributos que tienen ligada una estructura jerárquica). El papel del conocimiento en la construcción de clasificaciones fue mejorado para incluir objetivos generales de clasificación, reglas de inferencia y heurísticas para derivar nuevos descriptores, definiciones de dominios y tipos de atributos, y diferentes criterios de clasificación. Se usa una red de objetivos para guiar la búsqueda de descriptores relevantes y reglas de inferencia. Esta red liga objetivos, subobjetivos y atributos relevantes y se recorre para encontrar las interacciones entre los objetivos y los descriptores potenciales.

CLUSTER/S fue diseñado para permitir agrupar observaciones que requieren descripciones en términos de atributos estructurados. Permite insertar predicados adicionales dentro de las descripciones. Además, permite que el conocimiento sea expresado como un conjunto de reglas de implicación, en las cuales el antecedente y el consecuente son conjunciones de selectores, donde un selector es una proposición relacional $[x_i \# R_i]$ con R_i un subconjunto de valores admisibles del atributo x_i, y # estandariza los operadores relacionales $>,\geq,<,\leq,=,\neq$.

CLUSTER/3 (Seeman and Michalski, 2006) integra un método de selección de atributos llamado "*view-relevant attribute subsetting*" (VAS). La selección de atributos es un método comúnmente usado para reducir la dimensionalidad de un conjunto de datos,

usando técnicas de búsqueda y evaluación para reducir el espacio de atributos, seleccionando sólo aquellos atributos que son relevantes. El procedimiento VAS utiliza esta idea para considerar subconjuntos de atributos con base en su correlación.

2.2.2. Algoritmo WITT

En este algoritmo (Hanson, 1990), los objetos deben estar descritos por atributos Booleanos o multivaluados y no se permite ausencia de información en las descripciones de los objetos. Además, se requieren las correlaciones entre pares de atributos en forma de tablas de contingencia.

La forma en que se construyen los conceptos es formando hipótesis y probando los conceptos resultantes al aplicar una medida de cohesividad. Si una hipótesis falla, el algoritmo puede probar otras hipótesis, lo cual involucra crear nuevos conceptos o mezclar conceptos creados previamente.

El algoritmo consta de dos fases, en la primera fase se genera el conjunto inicial de agrupamientos llamados protosemillas. Para medir la distancia entre protosemillas se utiliza una medida que compara el contenido de la información en términos de los atributos de los objetos cuando están separados, con el contenido de la información cuando son combinados, tomando en cuenta la correlación entre pares de valores de los atributos. Si la pérdida de información es relativamente pequeña, los objetos son asignados a una nueva protosemilla. La segunda fase es de refinamiento, durante esta fase, para cada ciclo se prueba si es posible agregar miembros a cada agrupamiento sin afectar la calidad de cada concepto. La calidad de cada concepto se determina con base en la coherencia de los valores de los atributos que soporta, y al mismo tiempo, por la diferenciación de este concepto con otros conceptos existentes.

2.2.3. Algoritmo k-means conceptual

Este algoritmo (Ralambondrainy, 1995) consta de dos pasos: un paso de agregación, en el que se forman los agrupamientos y otro de caracterización, en el que se construyen las propiedades o conceptos que satisfacen los objetos de cada agrupamiento. Permite trabajar

con objetos descritos por atributos cuantitativos y cualitativos mezclados y no permite ausencia de información en las descripciones de los objetos.

En el paso de agregación se define una distancia para poder medir el parecido entre objetos, considerando que éstos están descritos por atributos cuantitativos y cualitativos. La función de distancia está definida como la suma de la distancia Euclideana para los atributos cuantitativos y la distancia Chi-cuadrado para los atributos cualitativos. Para aplicar la distancia Chi-cuadrado se codifican los atributos cualitativos en valores Booleanos. La codificación que se realiza de atributos cualitativos a numéricos no transforma el problema a un espacio de representación de los objetos real normado, donde tiene sentido encontrar centroides para la formación de los agrupamientos, ya que los 1's y 0's asociados a los nuevos atributos son códigos y no números, por lo tanto los centroides que calcula el algoritmo no tienen una interpretación en el espacio n-dimensional.

En el paso de caracterización, cada atributo tiene asociado un retículo de generalización (ver definición 1.4). Para los atributos cualitativos el retículo de generalización está dado de antemano por el especialista. Mientras que, para los atributos cuantitativos se realiza una codificación, es decir, una transformación en atributos cualitativos mediante una partición del dominio de valores. A partir de esta codificación se construye un retículo de generalización para los atributos cuantitativos.

La codificación que se realiza para cada atributo cuantitativo es la siguiente:

$$c_r = \begin{cases} inf & si \quad r < \mu_x - \sigma_x \\ typical & si \quad \mu_x - \sigma_x \le r \le \mu_x + \sigma_x \\ sup & si \quad \mu_x + \sigma_x < r \end{cases} \tag{2.8}$$

donde r es un valor del atributo x; μ_x es la *media* del atributo x en el agrupamiento A y σ_x es la *desviación estándar* de x en el agrupamiento A.

Una vez codificados los valores de los atributos cuantitativos, se construye el retículo de generalización, el cual está asociado al espacio de búsqueda $E = \{inf, typical, sup\}$.

El retículo de generalización para los atributos numéricos es el siguiente:

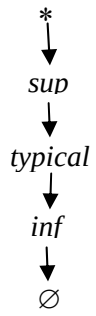


Figura 1. Retículo de generalización para los atributos cuantitativos

Para generar los conceptos, este algoritmo construye predicados *P*, los cuales deben satisfacer las siguientes condiciones:

1. Ser α-discriminante, es decir, el número de contraejemplos (número de objetos en el complemento de *A* cubiertos por *P*) debe ser menor o igual que α.

2. Ser β-caracterizante, es decir, el número de ejemplos (número de objetos en *A* cubiertos por *P*) debe ser mayor o igual que β.

2.2.4. Discusión

Los algoritmos que se discutieron en esta sección utilizan como base la idea de crear agrupamientos para los cuales los objetos más similares pertenezcan al mismo agrupamiento, mientras que objetos poco similares pertenezcan a agrupamientos diferentes, utilizando distintas estrategias para generar los conceptos.

Los algoritmos CLUSTER permiten trabajar con atributos cualitativos y cuantitativos simultáneamente, sin realizar ninguna transformación de los atributos para poder manejarlos. Además, la formación de los agrupamientos y de los conceptos se realiza en cada iteración del algoritmo. Por otro lado, en estos algoritmos las propiedades que se construyen deben ser disjuntas, por lo que se tiene un procedimiento que permite transformar un conjunto de propiedades no disjuntas en disjuntas. Sin embargo, esto no siempre se puede hacer, en cuyo caso se eliminan de la muestra los objetos que causan la intersección y son considerados como objetos especiales.

El algoritmo WITT requiere que los atributos sean Booleanos o multivaluados, por lo que es necesario transformar los atributos cuantitativos y nominales a este tipo de atributos. Este algoritmo trabaja en dos fases: en la primera fase se generan agrupamientos iniciales (protosemillas) tomando en cuenta la correlación entre pares de valores de los atributos y en la segunda fase se refinan estos agrupamientos. Los conceptos que se obtienen son disjuntos.

El algoritmo k-means conceptual trabaja en dos fases: en la primera fase agrupa los objetos y en la segunda fase genera los conceptos de dichos agrupamientos. En la primera fase de este algoritmo los atributos cualitativos se trasforman en atributos Booleanos para su manejo. Para la segunda fase es necesario asignar a cada atributo un retículo de generalización. Sin embargo, para algunas aplicaciones es difícil determinar cuál es el mejor retículo de generalización, ya que dependiendo del contexto del problema será la interpretación que se le dará a dicho retículo; además no se tienen métodos automáticos para construirlos, por lo que esta tarea se deja al especialista.

Los algoritmos conceptuales restringidos basados en semillas propuestos en la literatura presentan algunas limitantes. Entre las que se encuentran las siguientes:

i) Algunos de estos algoritmos, para trabajar con atributos mezclados, realizan transformaciones de los atributos cualitativos en cuantitativos, o viceversa.

ii) Es necesario que las descripciones de los objetos sean completas, es decir no se permite ausencia de información.

iii) Los objetos pertenecen a los agrupamientos siempre en el mismo grado (agrupamientos duros).

Por lo anterior consideramos necesario realizar mejoras a los procedimientos utilizados para generar los conceptos, y proponer nuevas formas de realizar la caracterización de los agrupamientos.

2.3. Sumario

En este capítulo se presentaron trabajos relacionados con la línea de investigación de agrupamiento conceptual restringido. Estos trabajos constituyen el estado del arte de esta línea de investigación.

Inicialmente se presentaron los algoritmos de agrupamiento que utilizaremos como base para proponer nuevos algoritmos de agrupamiento conceptual. Los trabajos analizados utilizan funciones de similaridad/disimilaridad las cuales permiten trabajar con atributos cualitativos y cuantitativos mezclados.

Adicionalmente se mencionaron algunos aspectos sobre los cuales sería importante continuar trabajando dentro del enfoque clasificatorio. Uno de estos problemas es la interpretación de los agrupamientos obtenidos por los algoritmos de clasificación no supervisada restringida, el cual será abordado en esta tesis.

Posteriormente se presentaron los trabajos relacionados con el enfoque conceptual. Los algoritmos que se discutieron en este enfoque utilizan como base la idea de crear agrupamientos para los cuales los objetos más similares pertenezcan al mismo agrupamiento, mientras que objetos poco similares pertenezcan a agrupamientos diferentes, utilizando distintas estrategias para generar los conceptos.

Por otro lado, estos algoritmos requieren que las descripciones de los objetos sean completas, es decir, sin ausencia de información y pueden ser aplicados a problemas donde los objetos estén descritos por atributos numéricos, cualitativos y estructurados simultáneamente.

Finalmente, se mencionaron algunos inconvenientes que se tienen con los algoritmos conceptuales restringidos descritos en este capítulo, los cuales sirven como base para proponer mejoras a los procedimientos actuales y proponer nuevas estrategias para realizar la caracterización de los agrupamientos.

En la Tabla 1 se muestra una comparación entre los algoritmos descritos en este capítulo, con base en sus características principales.

Características	Algoritmos restringidos basados en semillas				
	García-Serrano (1999)	Ayaquica-Martínez (2001)	Cluster (1981) (2006)	WITT (1990)	k-means conceptual (1995)
Algoritmo conceptual	×	×	✓	✓	✓
Atributos mezclados	✓	✓	✓	×	✓
Ausencia de información	✓	✓	×	×	×
Transformación de atributos	×	×	×	✓	✓
Conceptos disjuntos	N/A	N/A	✓	✓	✓
Agrupamientos difusos	×	✓	×	×	×

Tabla 1. Comparación entre algoritmos con base en sus características.

En esta tabla podemos observar que los algoritmos conceptuales descritos en este capítulo no permiten trabajar con ausencia de información y algunos requieren transformaciones en los atributos para poder trabajar con ellos. Además, construyen conceptos disjuntos y no permiten que los agrupamientos sean difusos. Por lo que se considera importante desarrollar algoritmos conceptuales que permitan trabajar con atributos mezclados, sin realizar transformaciones en los atributos, permitiendo ausencia de información en las descripciones de los objetos. Conjuntamente, debido a que en diversos problemas prácticos los objetos pueden pertenecer en distinto grado a los agrupamientos, es necesario desarrollar algoritmos conceptuales difusos, es decir, que generen conceptos a partir de agrupamientos difusos.

Capítulo 3

Algoritmos Conceptuales Duros

En este capítulo se da un planteamiento formal del problema de agrupamiento conceptual restringido duro, una función para evaluar la calidad de los conceptos duros y dos mejoras al algoritmo k-means conceptual (CKM); el algoritmo k-means conceptual con funciones de similaridad (CKMSF) y el algoritmo k-means conceptual con rasgos complejos (CKMCF).

3.1. Planteamiento Formal del Problema

Consideremos un conjunto $X = \{O_1,...,O_n\}$ de *n* objetos. Cada objeto descrito por un conjunto $R = \{x_1,...,x_m\}$ de *m* atributos. Cada atributo x_s toma valores de un conjunto de valores admisibles D_s, $x_s(O_j) \in D_s, s = 1,...,m$. Los atributos pueden ser de cualquier naturaleza (cualitativos: Booleano, *k*-valente, nominal o cuantitativos: entero, real, etc.). Además, asumiremos que en D_s existe un símbolo "?" para denotar ausencia de información, por lo que pueden ser consideradas descripciones de objetos incompletas.

Para cada atributo se define una función de comparación $FC_s : D_s \times D_s \to L_s$ *s*=1,2,...,*m*, donde L_s es un conjunto totalmente ordenado. La función FC_s es una evaluación del grado de similaridad entre dos valores del atributo x_s. Además, sea $\Gamma : (D_1 \times ... \times D_m)^2 \to [0,1]$ una función de similaridad, la cual permite evaluar el grado de similaridad entre dos descripciones de objetos.

El problema de agrupamiento conceptual restringido duro consiste en encontrar k agrupamientos duros $\{A_1,...,A_k\}$, $k>1$ del conjunto de objetos T, así como las propiedades o conceptos, C_i, que caracterizan a los agrupamientos A_i, $i = 1,\ldots,k$.

Un concepto C_i para el agrupamiento A_i debe satisfacer que si el objeto O pertenece al agrupamiento A_i entonces debería ser cubierto por el concepto C_i y si el objeto O no pertenece al agrupamiento A_i entonces no debería ser cubierto por el concepto C_i.

3.2. Función de Calidad

Consideramos que es importante tener una manera de evaluar la calidad de los conceptos. Algunas consideraciones que podrían tomarse en cuenta para evaluar la calidad de los conceptos son: las diferencias entre agrupamientos, la dimensionalidad, o la simplicidad de las representaciones de los agrupamientos.

En 1995, Ralambondrainy propuso tomar como medida de calidad el porcentaje de objetos en el agrupamiento que son cubiertos por el concepto. Sin embargo, consideramos que también es necesario tomar en cuenta los objetos fuera del agrupamiento que son cubiertos por el concepto; ya que esto nos permite evaluar no sólo qué tan bien caracterizan los conceptos a los agrupamientos, sino además qué tanto diferencian a los objetos de un agrupamiento de los objetos de otros agrupamientos.

La función de calidad que se propone en esta tesis toma en cuenta tanto el número de objetos del agrupamiento que son cubiertos por el concepto (ejemplos) como el número de objetos fuera del agrupamiento que son cubiertos por el concepto (contraejemplos). Un concepto será de mejor calidad en la medida en que reconozca mayor número de ejemplos y menor número de contraejemplos. El caso ideal es cuando un concepto cubre a todos los objetos del agrupamiento y ningún objeto fuera.

La función de calidad que se propone es la siguiente:

$$calidad(C_1,...,C_k) = \frac{1}{k}\sum_{i=1}^{k}\frac{ejemplos(C_i)}{|A_i| + contraejemplos(C_i)} \qquad (3.1)$$

donde:

- k es el número de agrupamientos.
- C_i es el concepto asociado al agrupamiento A_i, $i = 1,...,k$.
- $ejemplos(C_i)$ es el número de objetos en el agrupamiento A_i que son cubiertos por el concepto C_i.
- $contraejemplos(C_i)$ es el número de objetos fuera del agrupamiento A_i que son cubiertos por el concepto C_i.
- $|A_i|$ es el número de objetos del agrupamiento A_i.

Esta función obtiene valores altos (cercanos a uno) si el número de ejemplos crece y el número de contraejemplos decrece. La función obtiene 1.0 cuando el concepto cubre todos los objetos del agrupamiento A_i y no cubre objetos fuera de A_i.

En la función de calidad propuesta se evalúa la calidad de los conceptos generados a partir de los agrupamientos sin tomar en cuenta la calidad de los mismos. Existen índices de validación para los agrupamientos (Bezdek, 1974; Xie and Beni, 1991; Dave, 1991; Pal and Bezdek, 1995; Bezdek et al., 1997; Bezdek and Pal, 1998; Kwon, 1998; Rezaee et al., 1998; Flores-Sintas et al., 2000; Hathaway and Bezdek, 2003; Mali and Mitra, 2003; Dae-Won et al., 2003); sin embargo, estos índices están basados en la función de distancia utilizada por el algoritmo de agrupamiento. Tomar en cuenta uno de estos índices de validación en la función de calidad la haría dependiente de una función de distancia. Por lo tanto no podría usarse para comparar la calidad de los conceptos obtenidos por algoritmos que utilicen funciones de distancia diferentes ya que esta comparación no sería justa; además, no podrían aplicarse cuando el algoritmo de agrupamiento utiliza funciones de similaridad.

La función propuesta se utilizará para evaluar la calidad de los conceptos obtenidos por los algoritmos de agrupamiento conceptual restringido duro.

A continuación se introducen dos mejoras al algoritmo k-means conceptual. La primera de las cuales utiliza la estrategia del algoritmo k-means conceptual pero se usa un retículo de generalización diferente para los atributos cuantitativos. En la segunda mejora se usa una estrategia diferente para generar los conceptos, basada en rasgos complejos.

Los algoritmos que se proponen en este capítulo constan de dos fases: una fase de agrupamiento, en la cual se forman los agrupamientos en que se estructuran los objetos de la muestra y una fase de caracterización en la cual se generan los conceptos o propiedades que caracterizan a los agrupamientos.

3.3. Algoritmo K-means Conceptual con Funciones de Similaridad

El algoritmo k-means conceptual con funciones de similaridad (CKMSF) es una modificación al algoritmo k-means conceptual (CKM) (Ralambondrainy, 1995).

Debido a que los centroides obtenidos por el algoritmo CKM son elementos que no pueden representarse en el mismo espacio en que los objetos de la muestra se representan, proponemos usar, en la fase de agrupamiento, un algoritmo que seleccione objetos de la muestra como centroides, el cual se presenta en la Sección 3.3.1.

Por otra parte, consideramos que el retículo definido por Ralambondrainy, para los atributos cuantitativos (Figura 1) es incorrecto ya que no satisface la definición de retículo de generalización (definición 1.4). Por lo tanto, usaremos un retículo de generalización diferente, el cual permite obtener conceptos de mejor calidad. Esta modificación se presenta en la Sección 3.3.2.

3.3.1. Fase de Agrupamiento

En esta fase proponemos utilizar el algoritmo k-means con funciones de similaridad (KMSF) (descrito en la Sección 2.1.1. del Capítulo 2) para construir los agrupamientos en que se estructuran los objetos de la muestra.

Este algoritmo, a diferencia del algoritmo CKM, permite usar cualquier función de comparación entre valores de atributos y cualquier función de similaridad dependido del problema que se desea resolver.

En esta tesis usamos la siguiente función de similaridad:

$$\Gamma(O_i, O_j) = \frac{\sum_{x_s \in R} FC_s\left(x_s(O_i), x_s(O_j)\right)}{|R|}$$

donde $FC_s\left(x_s(O_i), x_s(O_j)\right)$ es la función de comparación usada para comparar dos valores del atributo x_s.

3.3.2. Fase de Caracterización

Para esta fase se requiere un retículo de generalización para cada atributo. El retículo de generalización para los atributos cualitativos está dado de antemano por el especialista mientras que para los atributos cuantitativos usamos el retículo de generalización propuesto por Pons-Porrata (1999) [4]:

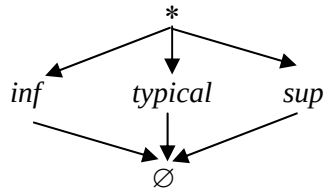


Figura 2. Retículo de generalización para los atributos cuantitativos

Este retículo satisface la condición $\forall e, f \in E, e \leq f \Rightarrow e \subseteq f$, ya que $inf \leq * \Rightarrow inf \subseteq *$, $typical \leq * \Rightarrow typical \subseteq *$ y $sup \leq * \Rightarrow sup \subseteq *$, por lo cual es posible trabajar de una manera más apropiada con los atributos cuantitativos que con el retículo propuesto por Ralambondrainy.

[4] A partir de este momento se denominará retículo original al retículo de generalización propuesto por Ralambondrainy y retículo nuevo al retículo propuesto por Pons-Porrata.

Por otra parte, los valores de μ_x y σ_x utilizados en la función de codificación (ver (2.8) de la Sección 2.2.3) dependen del agrupamiento A_i. Por esta razón, no es apropiado tomar sólo las etiquetas *inf*, *typical* y *sup*, sino además es necesario verificar si el valor del atributo *x*, para el objeto que se está analizando, está dentro del rango de valores para la etiqueta del atributo *x* en el agrupamiento A_i.

Una vez realizada la codificación de los atributos cuantitativos y definidos los retículos de generalización, se forma un predicado inicial *P* para cada objeto O_j, $j = 1,\ldots,n$ del agrupamiento de la siguiente manera: a cada atributo x_s, $s = 1,\ldots,m$ se asocia un par (x_s, a_s), donde x_s es el *s*-ésimo atributo y a_s es el valor de x_s en el objeto O_j; el predicado *P* está formado por $(x_1, a_1) \wedge \ldots \wedge (x_m, a_m)$ siendo $\wedge$ el operador lógico "y".

Ejemplo 3.1: Supongamos que después de aplicar la fase de agrupamiento para los objetos de la Tabla 2, se obtienen: $A_1 = \{O_1, O_2, O_3, O_4, O_6\}$ y $A_2 = \{O_5, O_7, O_8, O_9\}$. Para estos agrupamientos, los predicados iniciales son los siguientes:

Para A_1:

P_1: $(C, rojo) \wedge (T, chico) \wedge (P, typical) \wedge (F, redondo)$

P_2: $(C, rojo) \wedge (T, mediano) \wedge (P, typical) \wedge (F, redondo)$

P_3: $(C, azul) \wedge (T, chico) \wedge (P, sup) \wedge (F, redondo)$

P_4: $(C, azul) \wedge (T, mediano) \wedge (P, sup) \wedge (F, cuadrado)$

P_5: $(C, verde) \wedge (T, chico) \wedge (P, typical) \wedge (F, redondo)$

Para A_2:

P_1: $(C, verde) \wedge (T, grande) \wedge (P, typical) \wedge (F, triangular)$

P_2: $(C, amarillo) \wedge (T, grande) \wedge (P, typical) \wedge (F, triangular)$

P_3: $(C, amarillo) \wedge (T, mediano) \wedge (P, typical) \wedge (F, triangular)$

P_4: $(C, verde) \wedge (T, grande) \wedge (P, typical) \wedge (F, redondo)$

Objetos	Atributos			
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)
O_1	rojo	chico	20	redondo
O_2	rojo	mediano	20	redondo
O_3	azul	chico	25	redondo
O_4	azul	mediano	25	cuadrado
O_5	verde	grande	30	triangular
O_6	verde	chico	20	redondo
O_7	amarillo	grande	30	triangular
O_8	amarillo	mediano	35	triangular
O_9	verde	grande	35	redondo

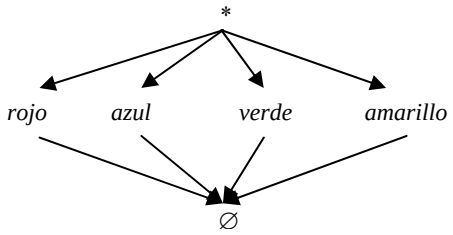
Tabla 2. Muestra con 9 objetos descritos por 4 atributos.

A partir de estos predicados y con base en los retículos de generalización se generan nuevos predicados. Dos predicados $P = (x_1, a_1) \wedge ... \wedge (x_m, a_m)$ y $P' = (x_1, b_1) \wedge ... \wedge (x_m, b_m)$ se generalizan si para todos los atributos x_s, $s = 1,\ldots,m$ de los predicados P y P', los valores a_s y b_s son iguales o pueden generalizarse con base en el retículo de generalización definido para cada atributo x_s. Si los valores a_s y b_s para algún atributo x_s no pueden generalizarse, entonces los predicados P y P' no se generalizan.

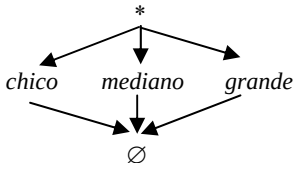
La generalización de los valores a_s y b_s de cada atributo x_s se realiza de la siguiente manera: si los valores a_s y b_s son iguales, entonces el nuevo predicado toma ese valor para el atributo x_s; si no son iguales entonces el valor para el atributo x_s del nuevo predicado será la generalización de a_s y b_s dada por el retículo. Si un valor es más general que otro entonces el valor para el atributo x_s será el valor más general.

Por ejemplo: supongamos que para la muestra de la Tabla 2, los retículos de generalización son los siguientes:

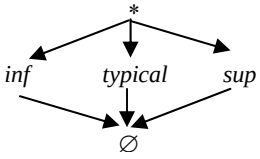
Para el atributo Color (C)



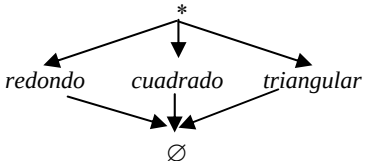
Para el atributo Tamaño (T)



Para el atributo Peso (P)



Para el atributo Forma (F)



El nuevo predicado que se forma a partir, por ejemplo, de los predicados P_1 y P_3 del agrupamiento A_1 (de nuestro ejemplo) será: $(C,*) \wedge (T,chico) \wedge (P,*) \wedge (F,redondo)$, ya que los valores *rojo* y *azul* se generalizan con *, así como los valores *inf* y *typical*; por otra parte para los atributos Tamaño y Forma ambos predicados toman el mismo valor (*chico* y *redondo*, respectivamente).

Posteriormente se verifica la condición de ser α-discriminante (el número de contraejemplos es menor o igual que α). Si un nuevo predicado es α-discriminante se almacena, en caso contrario se elimina. Si un nuevo predicado fue eliminado entonces los predicados iniciales, a partir de los cuales se generó dicho predicado, se almacenan. Este proceso se repite hasta que ya no sea posible generar nuevos predicados α-discriminantes.

Para el ejemplo 3.1, los predicados α-discriminantes, tomando α=1, son los siguientes:

Para A_1:

P'$_1$: $(C,rojo) \wedge (T,*) \wedge (P,typical) \wedge (F,redondo)$

P'$_2$: $(C,*) \wedge (T,*) \wedge (P,*) \wedge (F,redondo)$

P'$_3$: $(C,azul) \wedge (T,*) \wedge (P,sup) \wedge (F,*)$

P'$_4$: $(C,azul) \wedge (T,mediano) \wedge (P,sup) \wedge (F,cuadrado)$

Para A_2:

P'$_1$: $(C,*) \wedge (T,grande) \wedge (P,typical) \wedge (F,triangular)$

P'$_2$: $(C,amarillo) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular)$

P'$_3$: $(C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,*)$

P'$_4$: $(C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular)$

Una vez generados todos los predicados α-discriminantes, se eliminan aquellos que no son β-caracterizantes (el número de ejemplos es mayor o igual que β).

Para el ejemplo 3.1, los predicados α-discriminantes que son β-caracterizantes, tomando β=3, son los siguientes:

Para A_1:

P'$_1$: $(C,*) \wedge (T,*) \wedge (P,*) \wedge (F,redondo)$

Para A_2:

P'$_1$: $(C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,*)$

P'$_2$: $(C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular)$

El conjunto de predicados obtenido puede contener dos o más predicados que reconozcan los mismos objetos. Por lo tanto, este conjunto puede reducirse eliminando aquellos predicados que reconozcan los mismos objetos que algún otro predicado. Esta reducción se hace utilizando la estrategia propuesta por Ralambondrainy (1995) que trabaja como sigue: los predicados se ordenan en forma descendente de acuerdo al número de objetos que cada uno cubre. El primer predicado formará parte del concepto. Para los predicados restantes, si un predicado cubre algún objeto no cubierto por los predicados almacenados entonces se agrega al concepto; si no, se elimina. Finalmente, el concepto estará formado por la disyunción de los predicados almacenados.

En un retículo de generalización el símbolo * significa que el atributo puede tomar cualquier valor; por lo tanto, para simplificar se pueden eliminar de los predicados aquellos atributos que contengan *; los cuales no son relevantes para el concepto.

Para el ejemplo, los conceptos que se obtienen después de aplicar el proceso de reducción descrito anteriormente son:

Para A_1:

$$\boldsymbol{C_1}\textbf{:}\quad (Forma, Redondo)$$

es decir, los objetos que serán cubiertos por el concepto C_1 son aquellos objetos que tengan forma redonda o aquellos objetos que sean de color azul y tengan peso mayor a 24.7386.

Para A_2:

$$\boldsymbol{C_2}\textbf{:}\quad (29.6132 < Peso < 35.3867)$$

esto es, los objetos que serán cubiertos por el concepto C_2 son aquellos objetos que tengan un peso entre 29.6132 y 35.3867.

La calidad obtenida con estos conceptos es de 0.83, ya que el objeto O_4 de la Tabla 2, con la descripción (azul, mediano, 25, cuadrado), no es cubierto por el concepto del agrupamiento A_1 y el objeto O_9, con la descripción (verde, grande, 35, redondo), es cubierto por ambos conceptos.

El algoritmo CKMSF es el siguiente:

3.3.3. Algoritmo CKMSF

Entrada: Un conjunto T de objetos a ser agrupados.

Un número k de agrupamientos deseados.

Un par de umbrales, α y β, para la fase de caracterización.

Salida: Una partición $\{A_1,\ldots,A_k\}$ en k agrupamientos de T y el concepto C_i que caracteriza a cada agrupamiento A_i, $i = 1,\ldots,k$.

Fase de agrupamiento

Paso 1: Aplicar el algoritmo k-means con funciones de similaridad, para generar los agrupamientos A_i, $i = 1,\ldots,k$.

Fase de caracterización

Paso 1: Para cada A_i, $i = 1,\ldots,k$ hacer

Paso 2: Construir el predicado inicial para cada objeto $O_j \in A_i$.

Paso 3: Generar nuevos predicados a partir de la generalización de dos predicados.

Paso 4: De los predicados obtenidos en el paso 2, almacenar aquellos que sean α-discriminantes.

Paso 5: Si se almacenaron nuevos predicados entonces

Ir al paso 3.

Si no,

Ir al paso 6.

Paso 6: Eliminar todos los predicados que no sean β-caracterizantes.

Paso 7: Reducir el número de predicados utilizando el procedimiento de Ralambondrainy.

Paso 8: Construir el concepto C_i como la disyunción de los predicados obtenidos en el paso 7.

En la siguiente sección se presentan los resultados experimentales obtenidos con el algoritmo k-means conceptual con funciones de similaridad (CKMSF).

3.3.4. Resultados Experimentales

Con el objetivo de mostrar el desempeño del algoritmo CKMSF con base en la función de calidad propuesta, en esta sección se presentan los resultados obtenidos al aplicar el algoritmo CKMSF sobre diferentes bases de datos. Estas bases de datos son supervisadas y

fueron tomadas del repositorio de bases de datos de la UCI (Blake et al., 1998). Para los experimentos se ignoraron las etiquetas de las clases.

En la Tabla 3 se exponen las bases de datos utilizadas para la experimentación. De cada base de datos se muestra el número de agrupamientos a formar, el número de objetos que contiene cada base de datos, el número de atributos, el tipo de atributos que describen a los objetos y si se observa ausencia de información.

Para realizar las pruebas sobre las bases de datos que presentan ausencia de información se remplazaron los valores ausentes. Estos valores fueron sustituidos por la media de los valores de ese atributo, cuando el atributo es cuantitativo y por la moda de los valores de ese atributo, cuando el atributo es cualitativo; esto debido a que para trabajar con los retículos de generalización es necesario tener descripciones de objetos "completas".

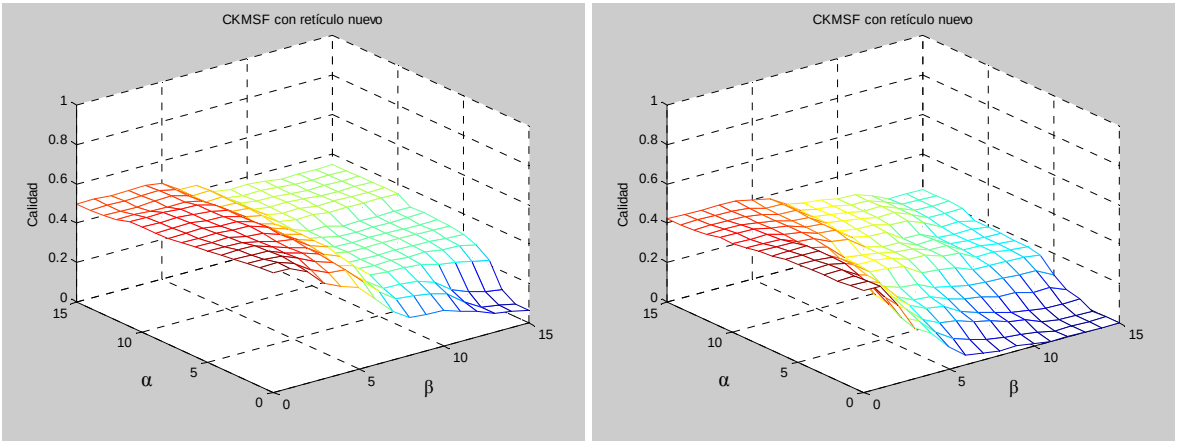
Base de Datos	No. de agrupamientos	No. de objetos	No. de atributos	Tipo de atributos	Ausencia de información?
Diabetes	2	768	8	Cuantitativas	no
Glass	6	214	9	Cuantitativas	no
Iris	3	150	4	Cuantitativas	no
Wine	3	178	13	Cuantitativas	no
Hayes	3	132	4	Cualitativas	no
Lenses	3	24	4	Cualitativas	no
Zoo	7	101	16	Cualitativas	no
Auto-mpg	3	398	7	Mezcladas	si
Bridges	7	108	11	Mezcladas	si
Echocardiogram	3	132	11	Mezcladas	si
Hepatitis	2	155	19	Mezcladas	si
Import85	6	205	25	Mezcladas	si
Tae	3	151	5	Mezcladas	no

Tabla 3. Bases de datos utilizadas para la experimentación.

Dado que el algoritmo CKMSF puede manejar la ausencia de información en la fase de agrupamiento, se realizaron pruebas completando los datos faltantes antes de realizar el agrupamiento y completando los datos después de agrupar los objetos.

Cabe recordar que los valores de los parámetros α y β son dados por el usuario, por lo cual se realizaron pruebas con diferentes valores para α y β con la finalidad de observar que tanto influyen los valores de estos parámetros en la calidad de los conceptos. Los resultados obtenidos se ilustran en las Figuras 3-7.

En las gráficas de superficie mostradas a continuación las zonas en rojo obscuro son regiones donde la función de calidad obtiene los valores más altos y las zonas en azul obscuro son regiones donde la función de calidad obtiene los valores más bajos.

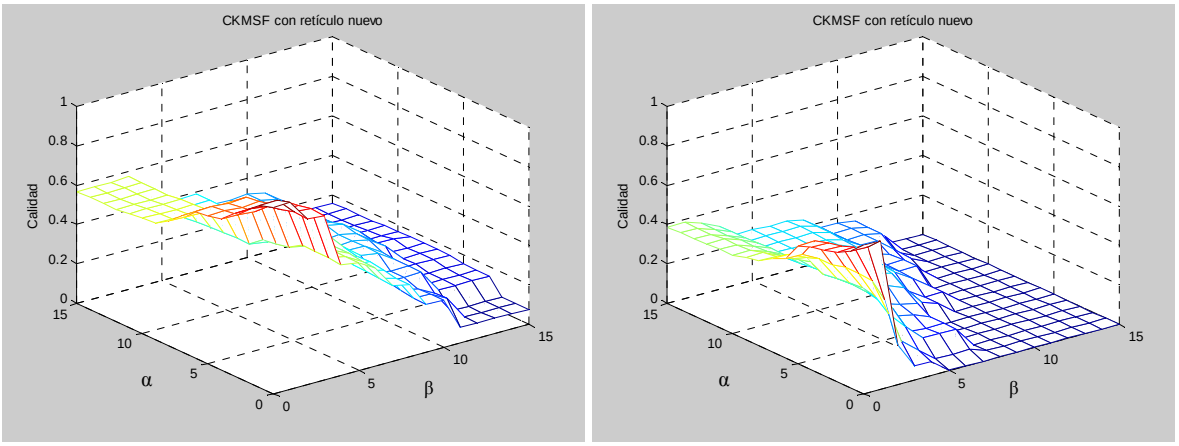


a) **b)**

Figura 3. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Auto-mpg, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

En las gráficas de la Figura 3 observamos que para la base de datos Auto-mpg, las mejores calidades de los conceptos se obtienen para valores de α y β cercanos a cero. Sin embargo, en la gráfica de la Figura 3 a) se observa que la calidad de los conceptos es similar para cualquier valor de α, cuando β toma valores entre 0 y 5. Mientras que, en la gráfica de la Figura 3 b) la calidad de los conceptos es similar sólo para valores de α entre 0 y 10, con β entre 0 y 2. Es decir, la calidad de los conceptos depende más del parámetro β que del parámetro α, esto se debe a que conforme crece el valor de β la condición β-caracterizante es más restrictiva (cada predicado que forma el concepto debe reconocer

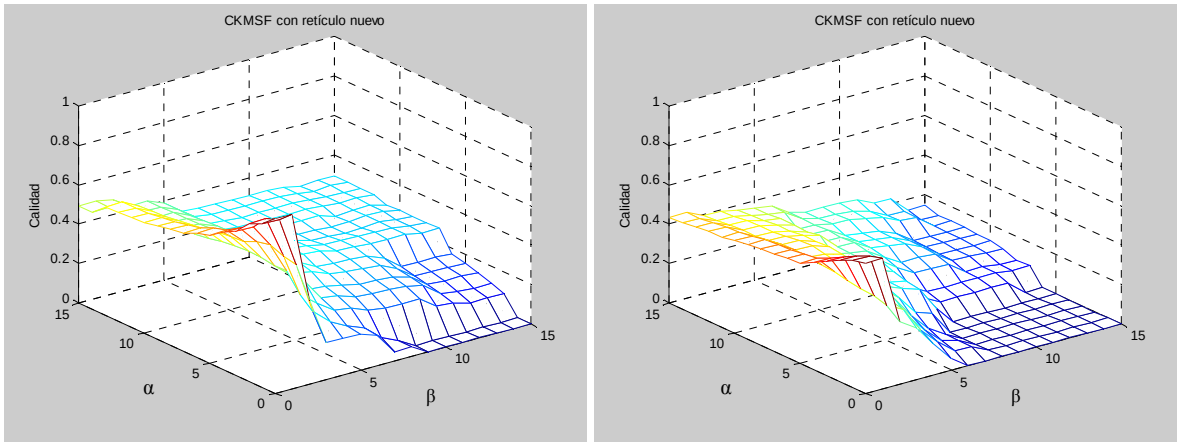
mayor número de objetos del agrupamiento); mientras que, cuando crece el valor de α la condición α-discriminante es menos restrictiva (se permite que mayor número de objetos que están fuera del agrupamiento sean cubiertos por cada uno de los predicados que forman los conceptos).



a) **b)**

Figura 4. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Bridges, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

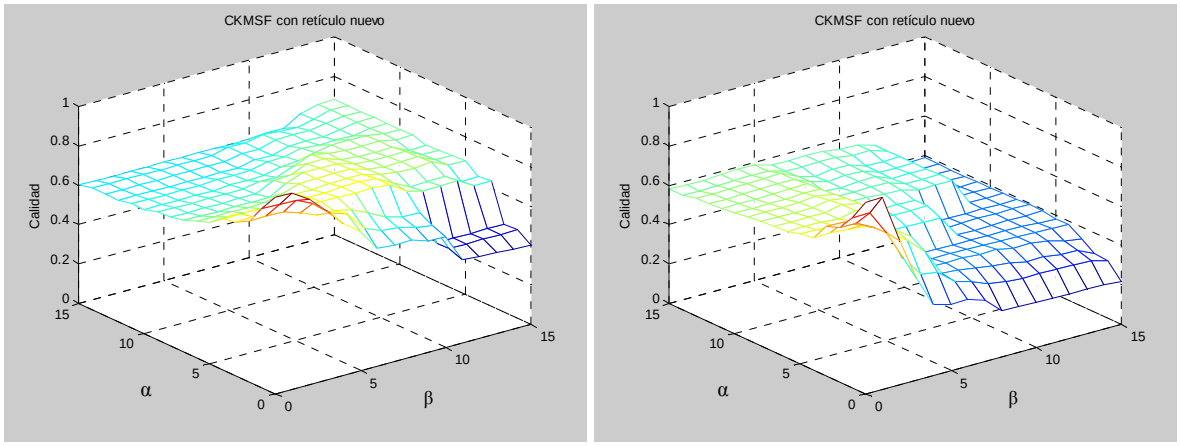
En la gráfica de la Figura 4 a) observamos que, para la base de datos Bridges, las mejores calidades de los conceptos se obtienen cuando el parámetro α toma valores entre 0 y 10, con β entre 0 y 5. Mientras que, en la gráfica de la Figura 4 b) las mejores calidades de los conceptos se obtienen cuando α toma valores entre 0 y 5, y β toma como valores 0 o 1.



a) **b)**

Figura 5. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Echocardiogram, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

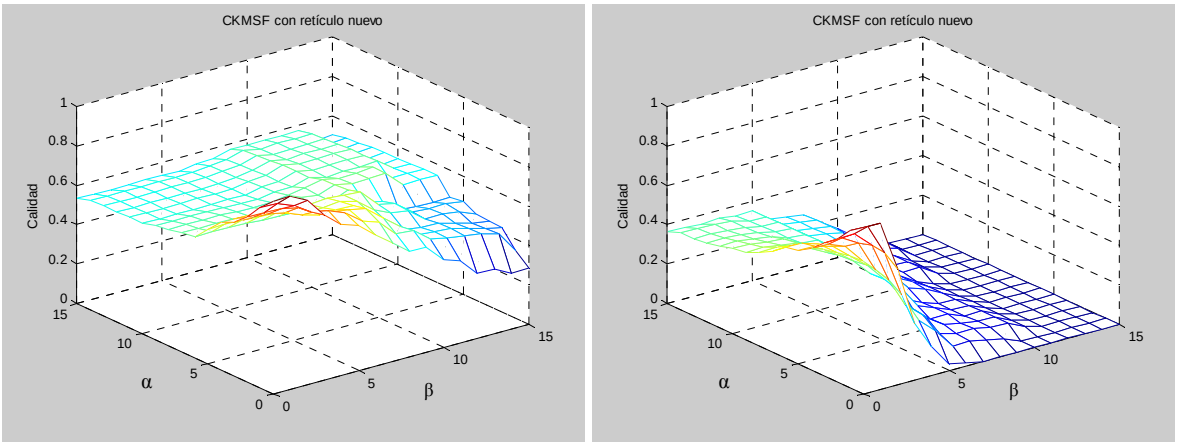
En las gráficas de la Figura 5 observamos que al completar la información después de agrupar (Figura 5 b)), la calidad de los conceptos es similar para cualquier valor de α, cuando β vale 0 o 1; mientras que, al completar la información antes de agrupar, sólo para valores de α entre 0 y 5 y cuando β vale 0 o 1 se obtienen conceptos con buena calidad (Figura 5 a)).



a) **b)**

Figura 6. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Hepatitis, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

En las gráficas de la Figura 6 observamos que, para la base de datos Hepatitis, la calidad de los conceptos depende menos del valor de los parámetros α y β cuando se completa la información antes de agrupar (Figura 6 a)), que cuando se completa la información después de agrupar (Figura 6 b)).



a) **b)**

Figura 7. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Import85, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

En las gráficas de la Figura 7 podemos observar que, para la base de datos Import85, las calidades de los conceptos dependen menos de los parámetros α y β cuando se completa la información antes de agrupar los objetos que cuando se completa la información después de realizar el agrupamiento.

En las gráficas de las Figuras 3-7 observamos que las mejores calidades de los conceptos se obtienen para valores de α y β cercanos a cero.

En la Tabla 4 se muestran los mejores resultados obtenidos con el algoritmo CKMSF utilizando el retículo nuevo, completando la información antes y después de agrupar los objetos.

Algoritmo CKMSF				
Base de Datos	Completando la información antes de agrupar		Completando la información después de agrupar	
	No. de predicados	Calidad	No. de predicados	Calidad
Auto-mpg	136	0.61	202	0.51
Bridges	33	0.95	84	0.75
Echocardiogram	89	0.88	94	0.66
Hepatitis	46	0.99	72	0.97
Import85	60	0.98	124	0.84
Promedio	**73**	**0.88**	**115**	**0.75**

Tabla 4. Resultados obtenidos por el algoritmo CKMSF completando la información antes y después de agrupar.

En la Tabla 4 observamos que para todas las bases de datos se obtienen conceptos de mejor calidad cuando se completan los valores ausentes antes de agrupar los objetos. Además que el número de predicados obtenidos completando la información antes de agrupar es menor que el número de predicados obtenidos completando la información después de agrupar. Por lo tanto, los mejores resultados se obtienen cuando se completan los valores ausentes antes de agrupar los objetos.

En las Figuras 8 y 9 se muestran gráficamente los resultados de la Tabla 4. En la Figura 8 se muestran las calidades obtenidas por el algoritmo CKMSF completando la información antes y después de agrupar y en la Figura 9 se muestra el número de predicados obtenidos por el algoritmo CKMSF completando la información antes y después de agrupar.

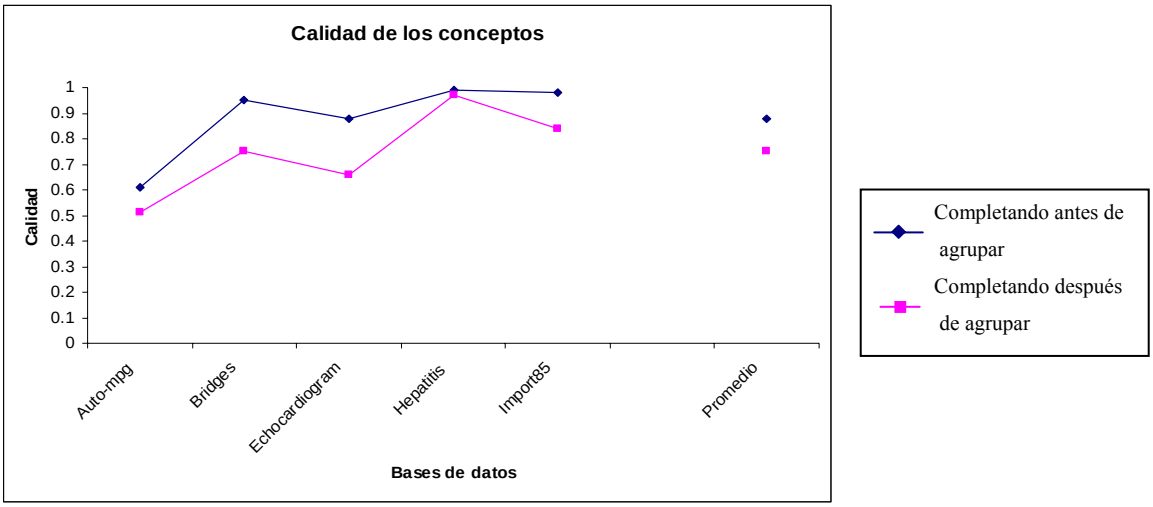


Figura 8. Calidades de los conceptos obtenidos por el algoritmo CKMSF completando la información antes de agrupar y completando la información después de agrupar.

En la gráfica de la Figura 8 observamos que se obtienen mejores resultados cuando se completan las descripciones de los objetos antes de realizar la fase de agrupamiento que cuando se completan las descripciones de los objetos después de realizar la fase de agrupamiento.

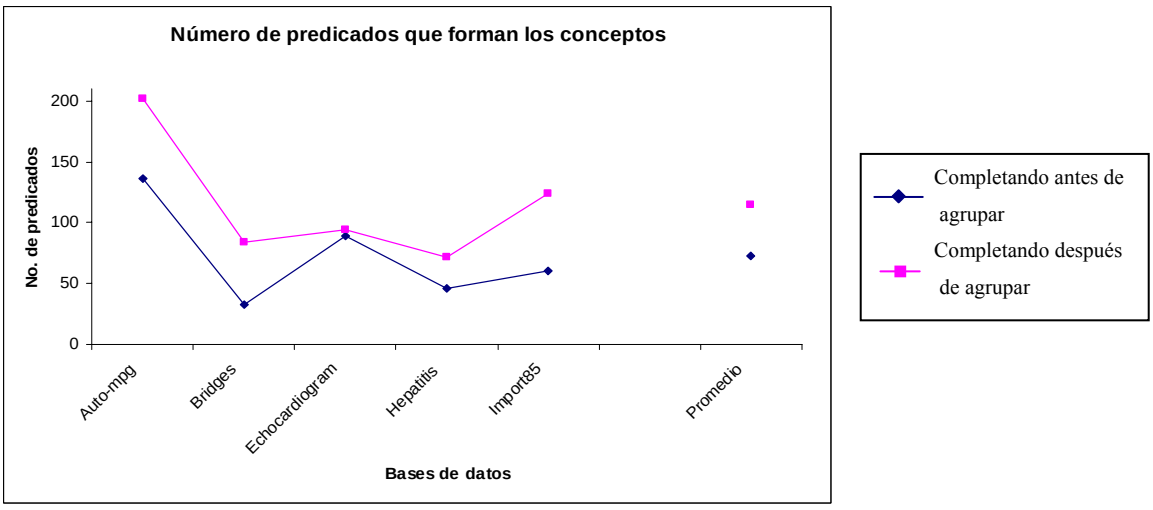


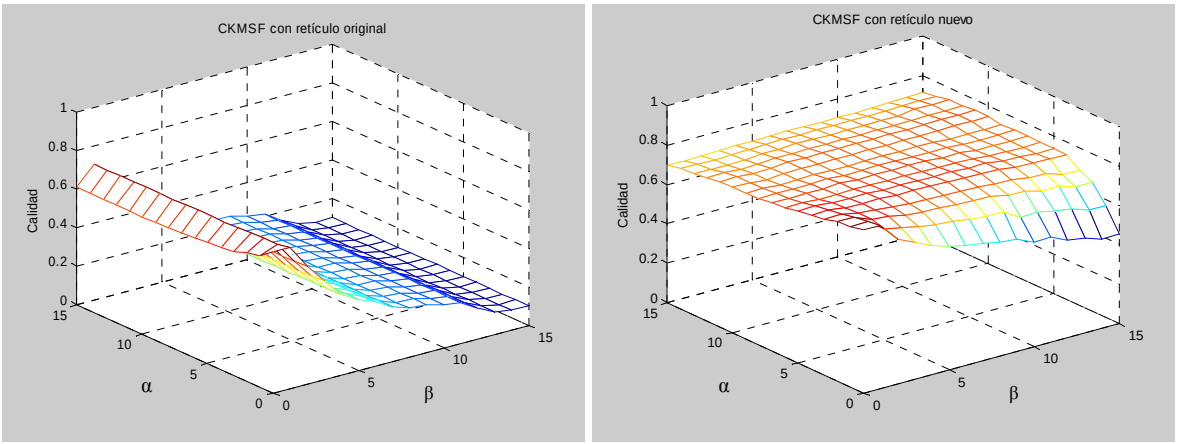
Figura 9. Número de predicados que forman los conceptos obtenidos por el algoritmo CKMSF completando la información antes de agrupar y completando la información después de agrupar.

En la gráfica de la Figura 9 se observa que el número de predicados obtenidos por el algoritmo CKMSF cuando se completan las descripciones de los objetos antes de realizar la fase de agrupamiento es menor que el número de predicados obtenidos cuando se completan las descripciones de los objetos después de realizar la fase de agrupamiento.

Posteriormente, se realizaron pruebas con todas las bases de datos de la Tabla 3, tomando diferentes valores de α y β. Los resultados obtenidos se muestran en las Figuras 10-19.

Para las bases de datos con ausencia de información se muestran los resultados obtenidos al completar la información antes de agrupar los objetos.

Para las bases de datos que contienen información cuantitativa, las pruebas se realizaron utilizando el retículo original (propuesto por Ralambondrainy, 1995) y el retículo nuevo (propuesto por Pons-Porrata, 1999),

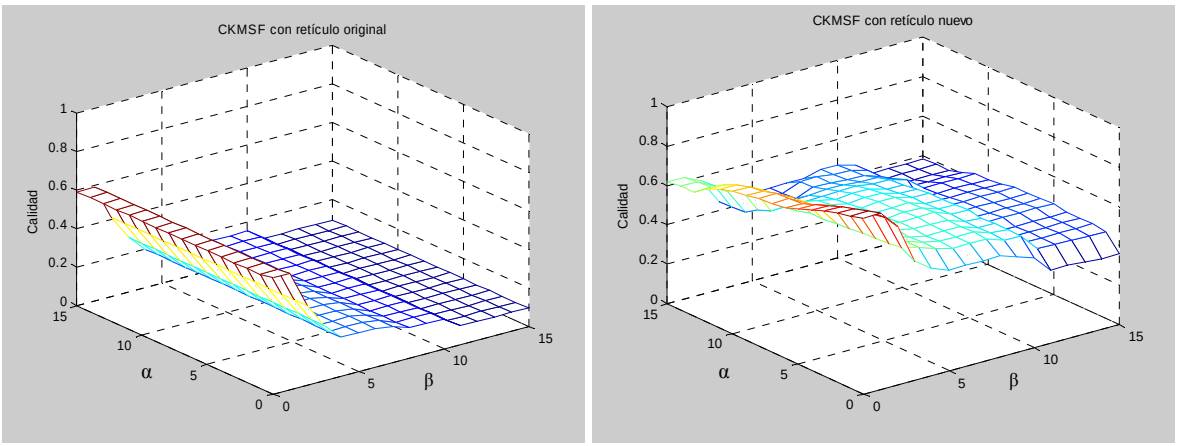


a) **b)**

Figura 10. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Diabetes, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

En las gráficas de la Figura 10 observamos que las calidades de los conceptos obtenidas utilizando el retículo nuevo no dependen tanto de los parámetros α y β, ya que para cualquier valor que tomen estos parámetros se obtienen conceptos con buena calidad. Mientras que, utilizando el retículo original sólo para valores de β entre 0 y 2 se obtienen

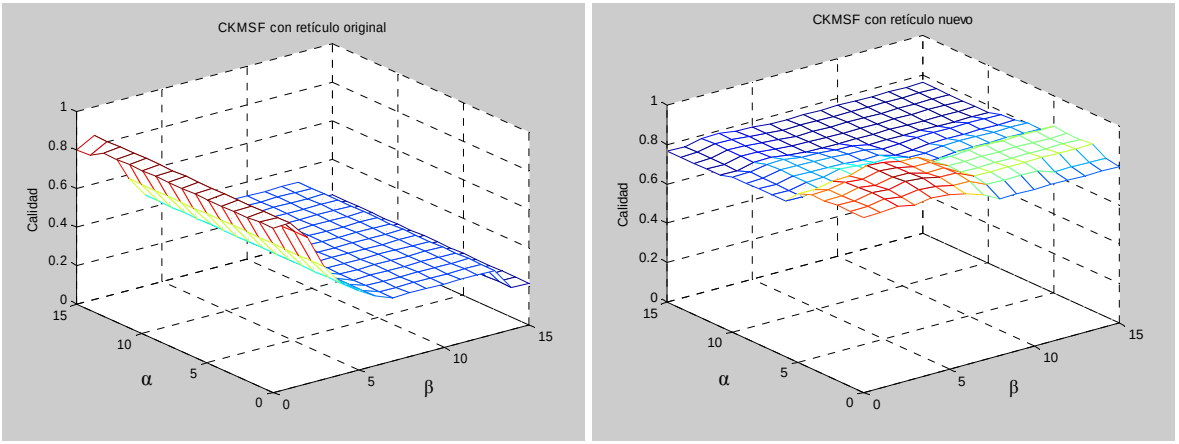
conceptos con buena calidad. Es decir, cuando se utiliza el retículo original es necesario hacer una buena selección de los parámetros α y β.



a) **b)**

Figura 11. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Glass, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

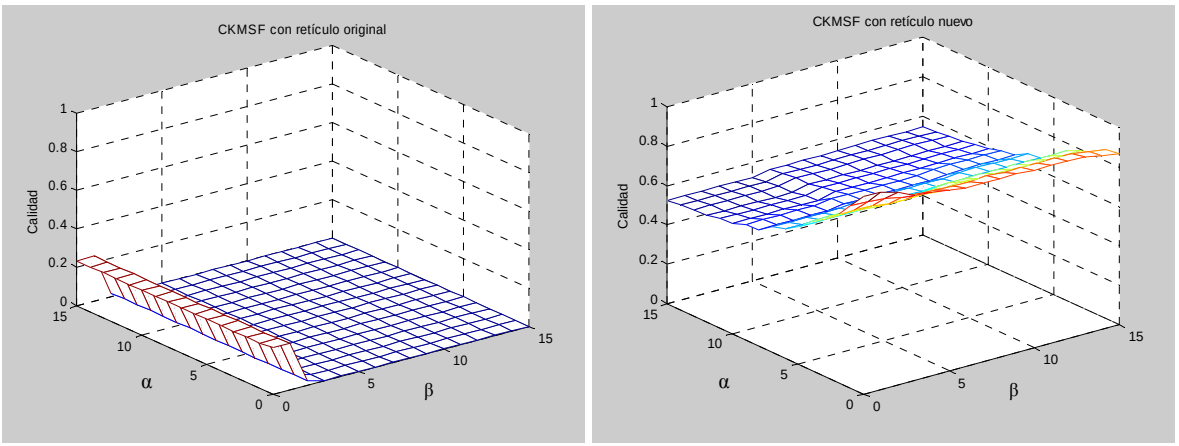
En la Figura 11 observamos que, al igual que para la base de datos Diabetes (Figura 10), las calidades de los conceptos no dependen tanto de los parámetros α y β al utilizar el retículo nuevo; mientras que al utilizar el retículo original es necesario hacer una buena selección de estos parámetros.



a) **b)**

Figura 12. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Iris, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

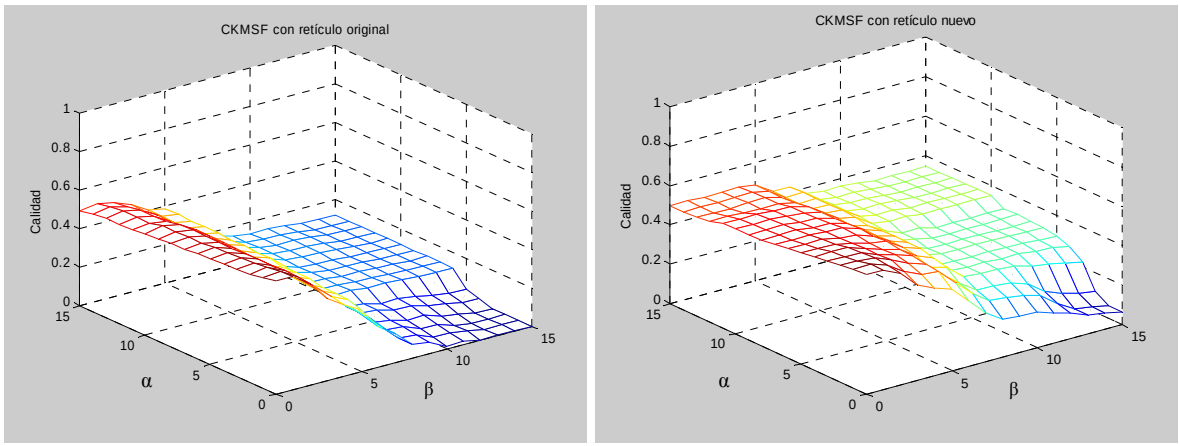
En la Figura 12 observamos que el comportamiento del algoritmo CKMSF, aplicado sobre la base de datos Iris, es similar al obtenido con las bases de datos Diabetes (Figura 10) y Glass (Figura 11).



a) **b)**

Figura 13. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Wine, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

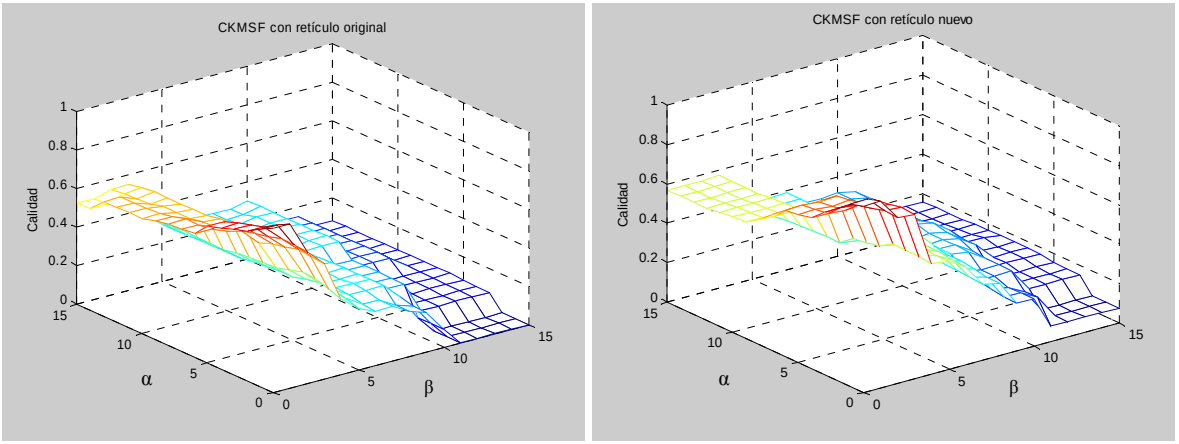
En la gráfica de la Figura 13 b) observamos que, cuando se utiliza el retículo nuevo, las mejores calidades se obtienen para cualquier valor β con α entre 0 y 2; mientras que al utilizar el retículo original las mejores calidades se obtienen para cualquier valor α con β entre 0 y 2 (Figura 13 a)).



a) **b)**

Figura 14. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Auto-mpg, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

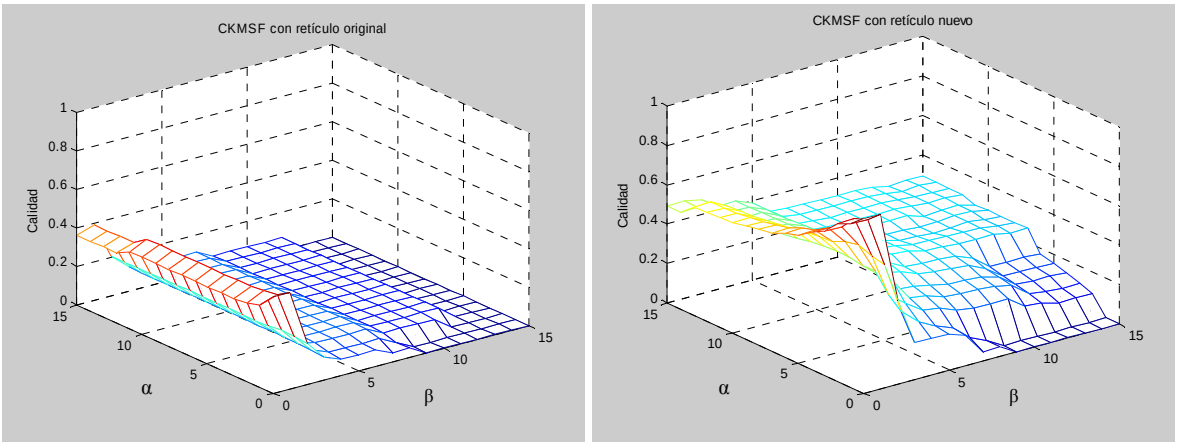
En las gráficas de la Figura 14 notamos que, aún cuando las calidades de los conceptos dependen menos de α y β cuando se utiliza el retículo nuevo que utilizando el retículo original, esta diferencia no es tan significativa como en el caso de las bases de datos que contienen únicamente información cuantitativa (Diabetes, Glass, Iris, Wine).



a) **b)**

Figura 15. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Bridges, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

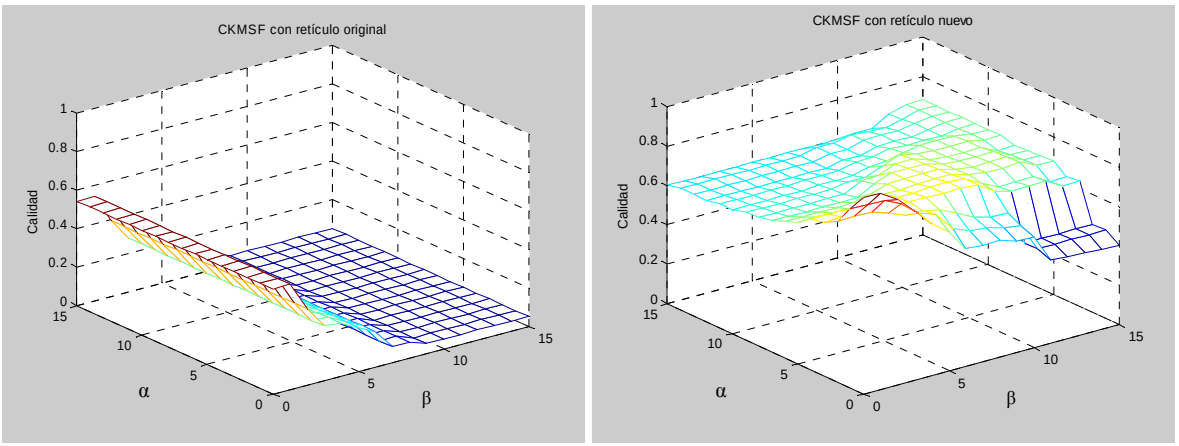
En la Figura 15 se observa que las calidades de los conceptos obtenidas utilizando el retículo nuevo son muy similares a las calidades de los conceptos obtenidas cuando se utiliza el retículo original. Esto se debe a que esta base de datos contiene únicamente un atributo cuantitativo.



a) **b)**

Figura 16. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Echocardiogram, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

En las gráficas de la Figura 16 notamos que, aún cuando las calidades de los conceptos dependen menos de α y β cuando se utiliza el retículo nuevo que utilizando el retículo original, esta diferencia no es tan significativa como en el caso de las bases de datos que contienen únicamente información cuantitativa (Diabetes, Glass, Iris, Wine).



a) **b)**

Figura 17. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Hepatitis, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

En las gráficas de la Figura 17 observamos que las calidades de los conceptos obtenidas utilizando el retículo nuevo no dependen tanto de los parámetros α y β, ya que para cualquier valor que tomen estos parámetros se obtienen conceptos con buena calidad. Mientras que, utilizando el retículo original sólo para valores de β entre 0 y 2 se obtienen conceptos con buena calidad.

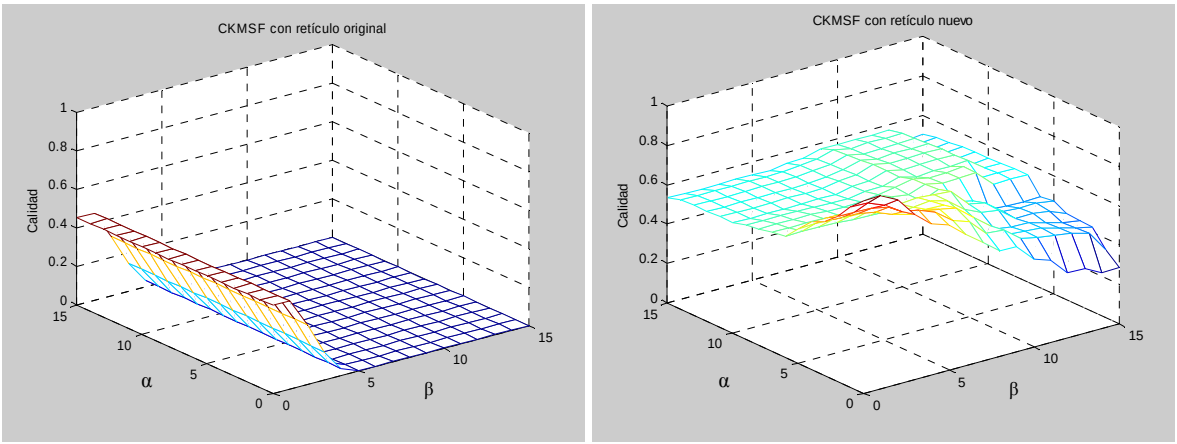


Figura 18. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Import85, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

En la Figura 18 observamos que, al igual que para la base de datos Hepatitis (Figura 17), las calidades de los conceptos no dependen tanto de los parámetros α y β al utilizar el retículo nuevo; mientras que al utilizar el retículo original es necesario hacer una buena selección de estos parámetros.

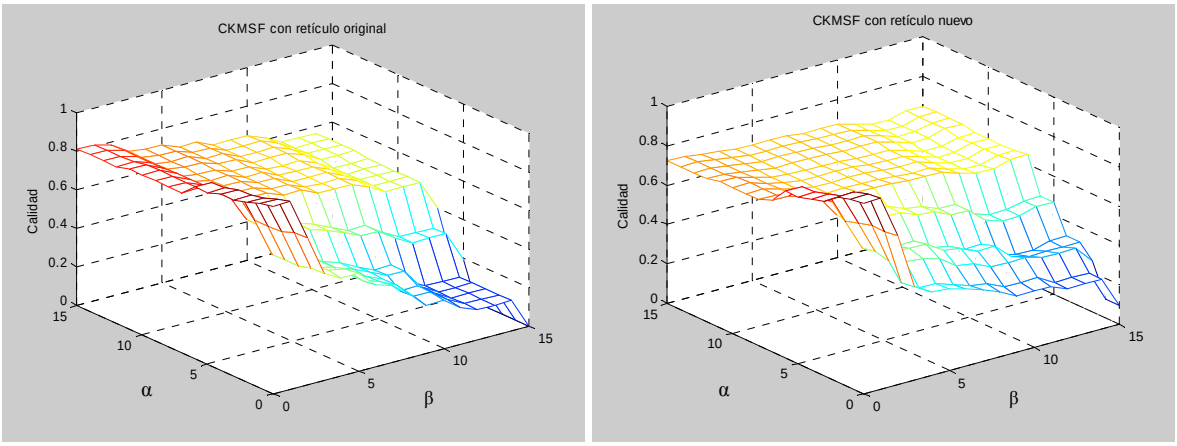


Figura 19. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Tae, para valores de α y β entre 0 y 15; a) utilizando el retículo original y b) utilizando el retículo nuevo.

En la Figura 19 se observa que comportamiento del algoritmo CKMSF es muy similar al obtenido con la base de datos Bridges (Figura 15). La base de datos Tae, al igual que la base de datos Bridges, sólo contiene un atributo cuantitativo.

En las Figuras 10-19 se observó que al utilizar el retículo nuevo la calidad de los conceptos no depende tanto de los parámetros α y β, como ocurre cuando se utiliza el retículo original; en este caso es necesario hacer una buena selección de los parámetros α y β, para obtener conceptos con buena calidad.

En la Tabla 5 se muestran los mejores resultados obtenidos con el algoritmo CKMSF utilizando ambos retículos de generalización (original y nuevo) para las bases de datos cuantitativas y mezcladas que contienen información numérica.

Base de Datos	Algoritmo CKMSF con el retículo original		Algoritmo CKMSF con el retículo nuevo	
	No. de predicados	Calidad	No. de predicados	Calidad
Diabetes	263	0.73	218	0.83
Glass	78	0.60	83	0.89
Iris	50	0.85	10	0.92
Wine	41	0.24	40	1.00
Auto-mpg	153	0.59	136	0.61
Bridges	35	0.85	33	0.95
Echocardiogram	51	0.50	89	0.88
Hepatitis	42	0.51	46	0.99
Import85	63	0.46	60	0.98
Tae	67	0.97	71	0.97
Promedio	**84**	**0.63**	**79**	**0.90**

Tabla 5. Resultados obtenidos por el algoritmo CKMSF utilizando el retículo original y el retículo nuevo en las bases de datos con información numérica.

En la Tabla 5 podemos observar que se obtienen conceptos de mejor calidad cuando se utiliza el retículo nuevo que utilizando el retículo original y para algunas bases de datos el número de predicados que se obtienen es menor utilizando el retículo nuevo que cuando se utiliza el retículo original.

En las Figuras 20 y 21 se muestran gráficamente los resultados de la Tabla 5. En la Figura 20 se muestran gráficamente las mejores calidades de los conceptos obtenidos utilizando ambos retículos (original y nuevo), en las bases de datos que contienen

información numérica y en la Figura 21 se muestra el número de predicados obtenidos por el algoritmo CKMSF usando el retículo original y el número de predicados obtenidos utilizando el retículo nuevo.

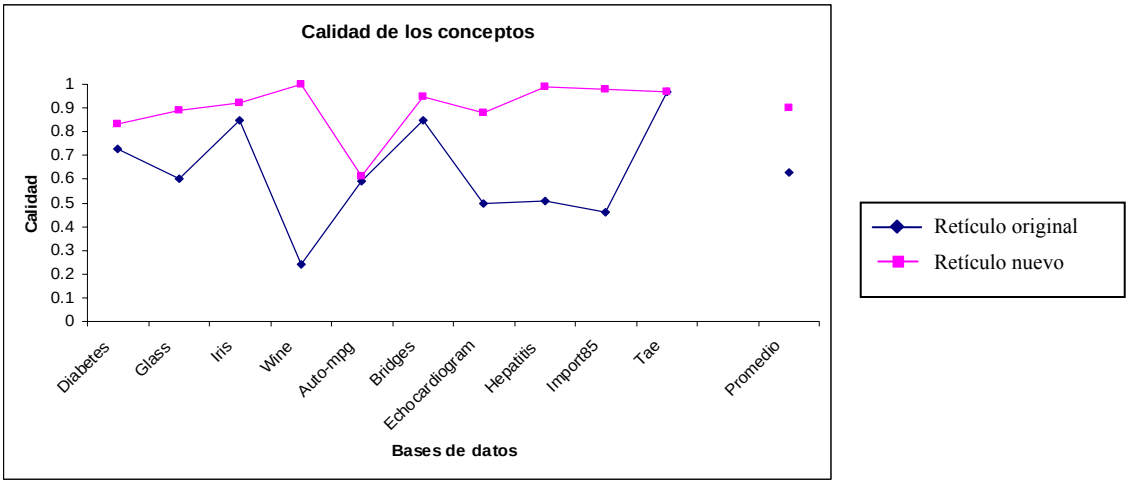


Figura 20. Calidades de los conceptos utilizando el retículo original y el retículo nuevo.

En la gráfica de la Figura 20 podemos observar que en general se obtienen mejores resultados al utilizar el retículo de generalización nuevo que utilizando el retículo original.

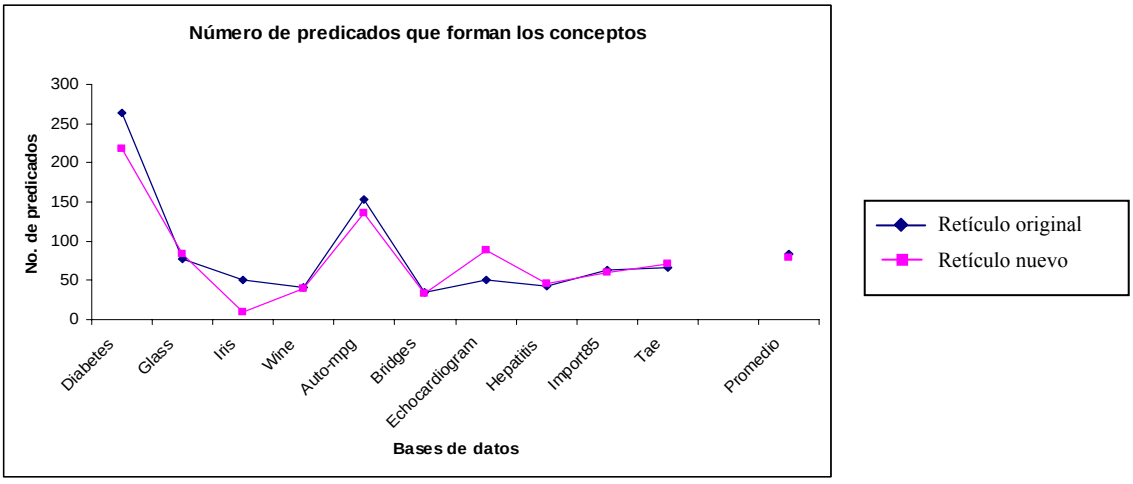


Figura 21. Número de predicados que forman los conceptos obtenidos utilizando el retículo original y el retículo nuevo.

En la gráfica de la Figura 21 podemos observar que en promedio el número de predicados obtenidos utilizando el retículo nuevo es menor que el número de predicados

utilizando el retículo original; aunque para algunas bases de datos como Echocardiogram se obtuvo menor número de predicados cuando se utiliza el retículo original que utilizando el retículo nuevo.

Las bases de datos Hayes, Lenses y Zoo contienen únicamente información cualitativa, las calidades de los conceptos obtenidas por el algoritmo CKMSF usando estas bases de datos se muestran en las Figuras 22-24.

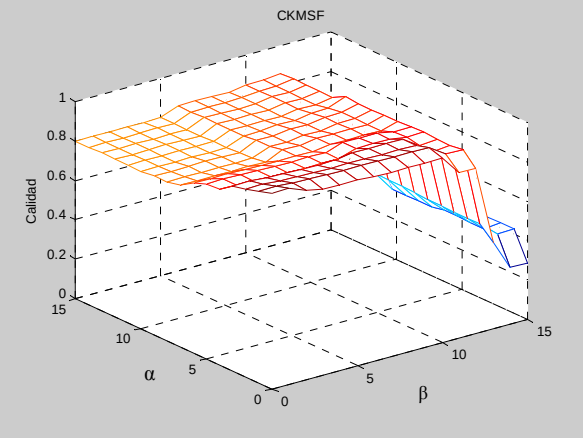


Figura 22. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Hayes, para valores de α y β entre 0 y 15.

En la Figura 22 se observa que la calidad de los conceptos obtenida usando la base de datos Hayes no depende, en la mayoria de los casos, de los parámetros α y β; sólo para valores de β cercanos a 15 se obtienen conceptos con baja calidad.

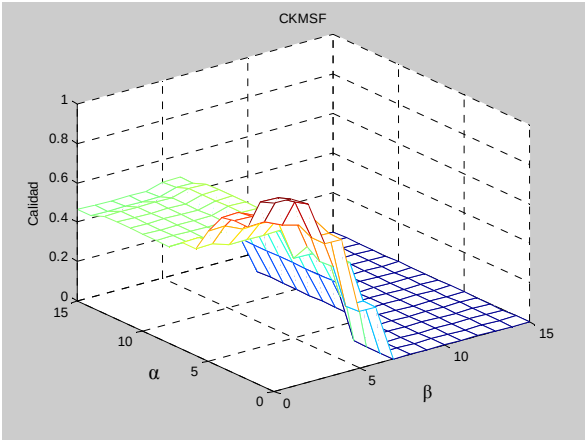


Figura 23. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Lenses, para valores de α y β entre 0 y 15.

En la Figura 23 observamos que, para la base de datos Lenses, es necesario hacer una buena selección de los parámetros α y β; ya que sólo para valores cercanos a 0 se obtienen conceptos con calidad alta.

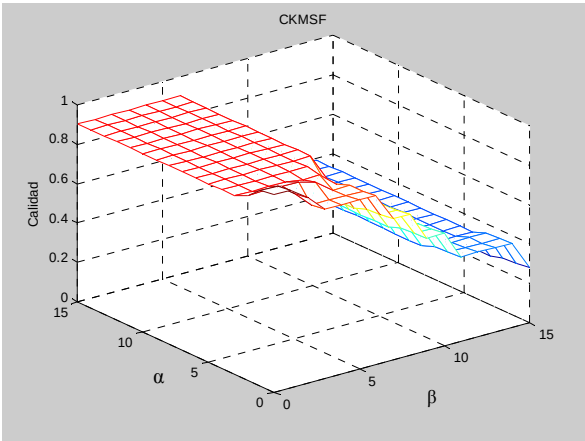


Figura 24. Resultados obtenidos por el algoritmo CKMSF aplicado sobre la base de datos Zoo, para valores de α y β entre 0 y 15.

En la Figura 24 se observa que para la base de datos Zoo las mejores calidades se obtienen para cualquier valor de α, cuando β toma valores entre 0 y 5.

En las gráficas de las Figuras 22-24, podemos observar que para las bases de datos con información cualitativa las mejores calidades de los conceptos se obtienen para valores de α y β cercanos a cero. Sin embargo, las bases de datos Hayes y Zoo no dependen tanto de α y

β; mientras que para la base de datos Lenses es necesario hacer una buena selección de estos parámetros.

3.3.6. Discusión

El algoritmo k-means conceptual con funciones de similaridad permite usar, en la fase de agrupamiento, funciones de similaridad que no requieran de transformaciones de atributos. Además, estas funciones pueden estar definidas en términos de funciones de comparación, las cuales permiten expresar la forma en que los valores de los atributos son comparados dependiendo del contexto del problema a resolver.

Por otro lado, en la fase de agrupamiento se utilizó el retículo de generalización propuesto en (Pons-Porrata, 1999), el cual permite trabajar de manera más adecuada con los atributos cuantitativos. Se realizaron pruebas con ambos retículos, original y nuevo, tomando diferentes valores para los parámetros α y β.

Con base en los experimentos realizados se observó que, para las bases de datos que contienen ausencia de información, se obtienen mejores resultados cuando se completa la información antes de agrupar que cuando se completa la información después de agrupar.

Además, al utilizar el retículo nuevo se obtienen conceptos de mejor calidad que los obtenidos utilizando el retículo original de Ralambondrainy, independientemente de los valores de α y β. Cuando se utiliza el retículo original es necesario hacer una buena selección de α y β para obtener conceptos con buena calidad.

Un inconveniente al usar retículos de generalización es que, para algunas aplicaciones, es difícil determinar cuál es el mejor retículo de generalización; dependiendo del contexto del problema será la interpretación que se le dará a dicho retículo. Además, no se tienen métodos automáticos para construir los retículos, por lo que esta tarea se deja al especialista.

Por esta razón, se propone una mejora al algoritmo k-means conceptual que no dependa de retículos de generalización para la construcción de los conceptos, la cual se presenta en la siguiente sección.

3.4. Algoritmo K-means Conceptual con Rasgos Complejos

Siguiendo la idea de Ralambondrainy (1995), en esta sección se propone una mejora al algoritmo k-means conceptual basada en rasgos complejos; el algoritmo k-means conceptual con rasgos complejos (CKMCF), que consta de dos fases: una fase de agrupamiento y una fase de caracterización.

En la fase de agrupamiento se utiliza el mismo algoritmo de agrupamiento utilizado en el algoritmo CKMSF. Mientras que en la fase de caracterización, proponemos utilizar los rasgos complejos para la construcción de los conceptos que caracterizan a los agrupamientos.

3.4.1. Fase de Agrupamiento

En esta fase, al igual que en el algoritmo CKMSF, se utiliza el algoritmo k-means con funciones de similaridad para formar los agrupamientos (descrito en la Sección 2.1.1. del Capítulo 2).

3.4.2. Fase de Caracterización

En los algoritmos de agrupamiento conceptual se espera que el concepto asociado a cada agrupamiento A_i, caracterice el mayor número posible de objetos de A_i y el menor número posible de objetos fuera de A_i.

Los rasgos complejos (De-la-Vega-Doria, 1994) son combinaciones de valores para un subconjunto de atributos tales que estos valores aparecen frecuentemente en los objetos del agrupamiento A_i y, al mismo tiempo, no aparecen en objetos de otros agrupamientos. Lo cual nos permite caracterizar a los objetos del agrupamiento A_i y al mismo tiempo no

caracterizar a objetos fuera de A_i. Por lo tanto, los rasgos complejos pueden ser utilizados para generar conceptos con la característica antes mencionada.

Para obtener los rasgos complejos es necesario seleccionar conjuntos de apoyo (Ω), los cuales son subconjuntos de atributos que indican las partes de los objetos a ser analizadas; donde Ω*O* es la subdescripción del objeto *O* tomando en cuenta únicamente los atributos de Ω. Los conjuntos de apoyo que se utilizarán para el algoritmo CKMCF son los siguientes:

1) Conjuntos de apoyo Γ-diferenciantes (Alba-Cabrera, 1997); los cuales son subconjuntos de atributos con los que la diferencia entre objetos de distintos agrupamientos es mayor que considerando todos los atributos de *R* y se definen como sigue:

   Definición 3.1*: $\Omega \subseteq R$** es un ***conjunto de apoyo* Γ-*diferenciante si todos los pares de objetos (O_j, O_p) de diferentes agrupamientos satisfacen $\Gamma(\Omega O_j, \Omega O_p) \leq \Gamma(O_j, O_p)$, es decir, Ω es un conjunto de apoyo Γ-diferenciante si objetos de diferentes agrupamientos no tienen mayor similaridad en Ω que en *R*.

2) Conjuntos de apoyo Γ-caracterizantes (Alba-Cabrera, 1997); los cuales son subconjuntos de atributos con los que la semejanza entre objetos de un mismo agrupamiento es mayor que considerando todos los atributos de *R* y se definen como sigue:

   Definición 3.2:* $\Omega \subseteq R$** es un ***conjunto de apoyo* Γ-*caracterizante si todos los pares de objetos (O_j, O_p) en el mismo agrupamiento satisfacen $\Gamma(O_j, O_p) \leq \Gamma(\Omega O_j, \Omega O_p)$, es decir, Ω es un conjunto de apoyo Γ-caracterizante si objetos del mismo agrupamiento no tienen menor similaridad en Ω que en *R*.

3) Conjuntos de apoyo Γ-testores; los cuales son subconjuntos de atributos que satisfacen la propiedad de ser Γ-diferenciantes y Γ-caracterizantes al mismo tiempo.

Para calcular los conjuntos de apoyo se evalúa el grado en que cada subconjunto Ω satisface la propiedad de ser Γ-diferenciante, Γ-caracterizante, o Γ-testor, respectivamente. Es decir,

1) Para conjuntos de apoyo Γ-diferenciantes, el grado en que el subconjunto Ω satisface la definición de conjunto de apoyo Γ-diferenciante se evalúa midiendo el número de pares de objetos en diferentes agrupamientos tal que su similaridad en Ω es menor o igual que su similaridad en *R*.

2) Para conjuntos de apoyo Γ-caracterizantes, el grado en que el subconjunto Ω satisface la definición de conjunto de apoyo Γ-caracterizante se evalúa midiendo el número de pares de objetos del mismo agrupamiento tal que su similaridad en Ω es mayor o igual que su similaridad en *R*.

3) Para conjuntos de apoyo Γ-testores, se evalúa el grado en que el subconjunto Ω satisface ambas definiciones.

Para seleccionar los conjuntos de apoyo aplicamos un algoritmo genético (Martínez-Trinidad et al., 2002; Guevara-Cruz, 2004) donde cada individuo representa un subconjunto de atributos ($\Omega \subseteq R$) formado por *m* genes y cada gen representa un atributo. Un gen vale 1 si el atributo se toma en cuenta y 0 si no se toma en cuenta. Este algoritmo utiliza el operador de cruza en un punto, es decir, se selecciona un punto de cruza y a partir de ahí se intercambia la información de los individuos a cruzar, para aplicar el operador de cruza se seleccionan el individuo más apto y el individuo menos apto; y la mutación uniforme, la cual consiste en seleccionar aleatoriamente un gen de un individuo y cambiar su valor (0 por 1 o 1 por 0). Como función de aptitud, se usa el grado en que un subconjunto Ω satisface la definición de conjunto de apoyo Γ-discriminante, conjunto de apoyo Γ-caracterizante o conjunto de apoyo Γ-testor. La población para la siguiente generación se obtiene seleccionando los individuos con mejor aptitud de entre los obtenidos por medio de

la cruza y la mutación, unidos con la población original. Los conjuntos de apoyo serán los mejores individuos de la última generación.

Ejemplo 3.2: Supongamos que después de aplicar la fase de agrupamiento para los objetos de la Tabla 6, se obtienen: $A_1 = \{O_1, O_2, O_3, O_4, O_6\}$ y $A_2 = \{O_5, O_7, O_8, O_9\}$. Para estos agrupamientos, los conjuntos de apoyo que se obtienen con el algoritmo genético descrito anteriormente son los que se muestran en la Tabla 7.

Objetos	Atributos			
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)
O_1	rojo	chico	20	redondo
O_2	rojo	mediano	20	redondo
O_3	azul	chico	25	redondo
O_4	azul	mediano	25	cuadrado
O_5	verde	grande	30	triangular
O_6	verde	chico	20	redondo
O_7	amarillo	grande	30	triangular
O_8	amarillo	mediano	35	triangular
O_9	verde	grande	35	redondo

Tabla 6. Muestra con 9 objetos descritos por 4 atributos.

Conjuntos de apoyo		
Γd	**Γc**	**Γt**
{C,T,P,F}	{P}	{C,T,P,F}
{C,T}	{P,F}	{T,P,F}
{C,T,P}	{T,P,F}	{C,T,P}
{C,T,F}	{T,P}	{C,P,F}
{T,F}	{C,P}	{T,P}
{T,P,F}	{C,P,F}	{C,P}
{C,P,F}	{C,T,P}	{P}
{C,F}	{T,F}	{T,F}
{P,F}	{F}	{C,T,F}
{C}	{C,F}	{C,T}
{T}	{T}	{C,F}
{F}	{C,T}	{F}
{P}	{C,T,F}	{T}
	{C}	{C}

Tabla 7. Conjuntos de apoyo obtenidos por el algoritmo genético para la muestra de la Tabla 6 con $A_1 = \{O_1, O_2, O_3, O_4, O_6\}$ y $A_2 = \{O_5, O_7, O_8, O_9\}$.

Adicionalmente, para formar los conceptos es necesario asociar, a los conjuntos de apoyo, valores que aparezcan suficientemente en los objetos de un agrupamiento y, al mismo tiempo, no aparezcan en objetos de otros agrupamientos. Para esto, utilizamos los rasgos complejos, los cuales se definen de la siguiente manera:

Definición 3.3*:** Sea $\Omega = \{x_{s_1}, ..., x_{s_p}\}$ un conjunto de apoyo y $(a_1, ..., a_p)$ valores asociados a los atributos $x_{s_1}, ..., x_{s_p}$ tomados de un objeto de la muestra, entonces $\{x_{s_1}, ..., x_{s_p}\}$ y $(a_1, ..., a_p)$ forman un ***rasgo complejo (De-la-Vega-Doria, 1994) del agrupamiento A_i, si y sólo si:

1) $\sum_{O_j \in A_i} \Gamma(\Omega O_j, (a_1, ..., a_p)) \geq \beta_i$

2) $\sum_{O_j \notin A_j} \Gamma(\Omega O_j, (a_1, ..., a_p)) < \lambda_i$

donde β_i es la mínima similaridad que los objetos de un agrupamiento A_i deben tener con la subdescripción $(a_1, ..., a_p)$ y λ_i es la máxima similaridad que los objetos fuera del

agrupamiento A_i deben tener con $(a_1,...,a_p)$. En esta tesis usamos $\lambda_i = 1$ ya que estamos interesados en obtener conceptos que no reconozcan objetos de otros agrupamientos, y para β_i se realiza el siguiente procedimiento, con el objetivo de encontrar el mayor número de rasgos complejos: inicialmente se toma β_i igual al número de objetos en el agrupamiento (máximo valor que alcanza la sumatoria definida en 1) de la definición 3.3 al comparar las subdescripciones de los objetos del agrupamiento A_i con $(a_1,...,a_p)$, puesto que $\Gamma(\Omega O_j,(a_1,...,a_p) \in [0,1])$; posteriormente se sigue un procedimiento iterativo en el cual β_i se decrementa tomando como siguiente valor la máxima suma obtenida en 1) de la definición 3.3 para las subdescripciones que no satisfacen la definición de rasgo complejo, ya que con valores mayores no se obtendrían nuevos rasgos complejos. Entonces, se calculan los rasgos complejos con el nuevo β_i. Este procedimiento se repite mientras $\beta_i > 0$.

En esta fase la función de similaridad que se utiliza para calcular los rasgos complejos es la misma que se utilizó en la fase de agrupamiento. Una ventaja de utilizar la misma función de similaridad en ambas fases del algoritmo es que, la forma en que se generan los conceptos mantiene una estrecha relación con la construcción de los agrupamientos. Esto último no ocurre en los algoritmos k-means conceptual y k-means conceptual con funciones de similaridad ya que estos algoritmos no usan una función de similaridad para construir los conceptos.

Los rasgos complejos que se obtuvieron con los conjuntos de apoyo de la Tabla 7 para el ejemplo 3.2 se muestran en la Tabla 8.

Agrupamiento	Rasgos Complejos para cada conjunto de apoyo		
	Γd	Γc	Γt
1	{P} – (20)	{P}- (20)	{P} – (20)
2	{P} – (35)	{P}- (35)	{P} – (35)

Tabla 8. Rasgos Complejos para el ejemplo de la Tabla 6 y los conjuntos de apoyo de la Tabla 7.

Una vez que se obtuvieron los rasgos complejos, para obtener los conceptos, a cada rasgo complejo se le asocia un predicado *P*, el cual se construye de la siguiente manera: a cada atributo $x_s \in R$ que aparece en el rasgo complejo se le asigna el valor a_s asociado a ese atributo y para los atributos $x_s \in R$ que no aparecen en el rasgo complejo se asigna el símbolo * que significa "cualquier valor es posible".

Para el ejemplo 3.2, los predicados formados, para cada agrupamiento, a partir de los rasgos complejos de la Tabla 8, usando los conjuntos de apoyo Γ-diferenciantes (Γd), Γ-caracterizantes (Γc) o conjuntos de apoyo que son Γ-testores (Γt) de la Tabla 7, se muestran en la Tabla 9.

Predicados obtenidos a partir de los rasgos complejos		
Conjuntos de apoyo	**Agrupamiento A_1**	**Agrupamiento A_2**
Γd	P: $(C,*) \wedge (T,*) \wedge (P,20) \wedge (F,*)$	P: $(C,*) \wedge (T,*) \wedge (P,35) \wedge (F,*)$
Γc	P: $(C,*) \wedge (T,*) \wedge (P,20) \wedge (F,*)$	P: $(C,*) \wedge (T,*) \wedge (P,35) \wedge (F,*)$
Γt	P: $(C,*) \wedge (T,*) \wedge (P,20) \wedge (F,*)$	P: $(C,*) \wedge (T,*) \wedge (P,35) \wedge (F,*)$

Tabla 9. Predicados obtenidos a partir de rasgos complejos usando tres tipos diferentes de conjuntos de apoyo.

El conjunto de predicados obtenido a partir de los rasgos complejos puede contener dos o más predicados que reconozcan los mismos objetos. Por lo tanto, este conjunto de predicados puede reducirse eliminando predicados que reconozcan los mismos objetos que algún otro predicado. Esta reducción se hace utilizando la misma estrategia que la utilizada para el algoritmo CKMSF (ver Sección 3.3.2).

Para el ejemplo 3.2, los conceptos obtenidos después de aplicar el proceso de reducción y eliminando los atributos que contengan el valor * (es decir, atributos que pueden tomar cualquier valor), son los que se muestran en la Tabla 10.

Conceptos generados con rasgos complejos usando conjuntos de apoyo Γ-diferenciantes (Γd), Γ-caracterizantes (Γc) y Γ-testores (Γt)			
Agrupamiento	**Γd**	**Γc**	**Γt**
A_1	C_1: $Peso = 20$	C_1: $Peso = 20$	C_1: $Peso = 20$
A_2	C_2: $Peso = 35$	C_2: $Peso = 35$	C_2: $Peso = 35$

Tabla 10. Conceptos obtenidos para el ejemplo.

El algoritmo CKMCF es el siguiente:

3.4.3. Algoritmo CKMCF

Entrada: Un conjunto T de objetos a ser agrupados.
Un número k de agrupamientos deseados.

Salida: Una partición $\{A_1,\ldots,A_k\}$ en k agrupamientos de T y el concepto C_i que caracteriza a cada agrupamiento A_i, $i = 1,\ldots,k$.

Fase de agrupamiento

Paso 1: Aplicar el algoritmo k-means con funciones de similaridad, para generar los agrupamientos A_i, $i = 1,\ldots,k$.

Fase de caracterización

Paso 1: Para cada agrupamiento A_i, $i = 1,\ldots,m$ hacer

Paso 2: Calcular los conjuntos de apoyo (en nuestro caso, conjuntos Γ- diferenciantes, conjuntos Γ-caracterizantes o conjuntos Γ-testores) para el agrupamiento A_i.

Paso 3: Calcular los rasgos complejos para el agrupamiento A_i.

Paso 4: Asociar a cada rasgo complejo un predicado P.

Paso 5: Reducir el número de predicados utilizando el procedimiento de Ralambondrainy.

Paso 6: Construir el concepto C_i como la disyunción de los predicados obtenidos en el paso 5.

3.4.4. Resultados Experimentales

Para mostrar el desempeño del algoritmo CKMCF se realizaron pruebas usando las mismas bases de datos utilizadas para el algoritmo CKMSF (ver Tabla 3).

Para seleccionar los conjuntos de apoyo se realizaron pruebas usando diferente número de iteraciones: 10, 20 y 30, y diferente número de individuos en la población inicial: 5, 10, 20, 50, 100 y 500. Los mejores resultados se obtuvieron con 10 iteraciones y 500 individuos. Los resultados que se muestran son los obtenidos al aplicar el algoritmo genético con 10 iteraciones y 500 individuos en la población inicial, usando los tres tipos de conjuntos de apoyo, conjuntos Γ-diferenciantes, conjuntos Γ-caracterizantes y conjuntos Γ-testores.

De la misma manera que en el algoritmo CKMSF, para las bases de datos que contienen ausencia de información se realizaron pruebas completando los datos faltantes antes de realizar el agrupamiento y completando los datos después de agruparlos. En las Tablas 11 y 12 se muestran los resultados obtenidos sin completar la información, completando la información antes de agrupar y completando la información después de agrupar.

En la Tabla 11 se muestran las calidades obtenidas por el algoritmo CKMCF usando los tres tipos diferentes de conjuntos de apoyo, sin completar la información, completando la información antes de agrupar y completando la información después de agrupar. En la Tabla 12 se muestra el número de predicados obtenidos por el algoritmo CKMCF con los distintos tipos de conjuntos de apoyo cuando no se completa la información, se completa antes de agrupar y cuando se completa después de agrupar.

Algoritmo CKMCF									
Bases de Datos	**Sin completar la información**			**Completando la información antes de agrupar**			**Completando la información después de agrupar**		
	Γd	**Γc**	**Γt**	**Γd**	**Γc**	**Γt**	**Γd**	**Γc**	**Γt**
Auto-mpg	0.66	0.66	0.66	0.62	0.62	0.62	0.19	0.19	0.19
Bridges	1.00	0.09	1.00	1.00	0.00	1.00	0.98	0.00	0.98
Echocardiogram	0.90	0.93	0.92	0.94	0.94	0.95	0.82	0.76	0.84
Hepatitis	0.52	0.95	0.74	0.98	0.92	0.99	1.00	0.80	0.94
Import85	0.60	0.87	0.93	0.98	0.86	0.98	0.75	0.64	0.75
Promedio	**0.74**	**0.72**	**0.85**	**0.90**	**0.67**	**0.91**	**0.75**	**0.48**	**0.74**

Tabla 11. Calidades de los conceptos obtenidos con el algoritmo CKMCF sin completar la información y completando la información antes y después de agrupar los objetos.

En la Tabla 11 podemos observar que cuando se seleccionan los conjuntos Γ-caracterizantes como conjuntos de apoyo, en la mayoría de los casos, el algoritmo obtiene conceptos de mejor calidad cuando se utilizan las bases de datos sin completar la información, mientras que cuando se seleccionan como conjuntos de apoyo los conjuntos Γ-testores o los conjuntos Γ-diferenciantes se obtienen conceptos de mejor calidad completando la información antes de aplicar la fase de agrupamiento.

Algoritmo CKMCF									
Bases de Datos	**Sin completar la información**			**Completando la información antes de agrupar**			**Completando la información después de agrupar**		
	Γd	**Γc**	**Γt**	**Γd**	**Γc**	**Γt**	**Γd**	**Γc**	**Γt**
Auto-mpg	39	39	39	43	43	43	25	25	25
Bridges	57	3	49	48	0	42	84	0	83
Echocardiogram	40	33	41	48	40	53	56	49	56
Hepatitis	44	33	76	68	39	97	74	41	57
Import85	91	52	117	113	46	114	121	91	123
Promedio	**54**	**32**	**64**	**64**	**34**	**70**	**72**	**41**	**69**

Tabla 12. Número de predicados obtenidos con el algoritmo CKMCF sin completar la información y completando la información antes y después de agrupar los objetos.

En la Tabla 12 podemos observar que, en promedio, el algoritmo obtiene menor número de predicados cuando se utilizan las bases de datos sin completar la información, para los tres tipos de conjuntos de apoyo.

En las Figuras 25 y 26 se muestran gráficamente los resultados de las Tablas 11 y 12.

En la Figura 25 se muestra gráficamente los resultados obtenidos por cada uno de los tipos de conjunto de apoyo completando la información antes de agrupar, después de agrupar y sin completar la información.

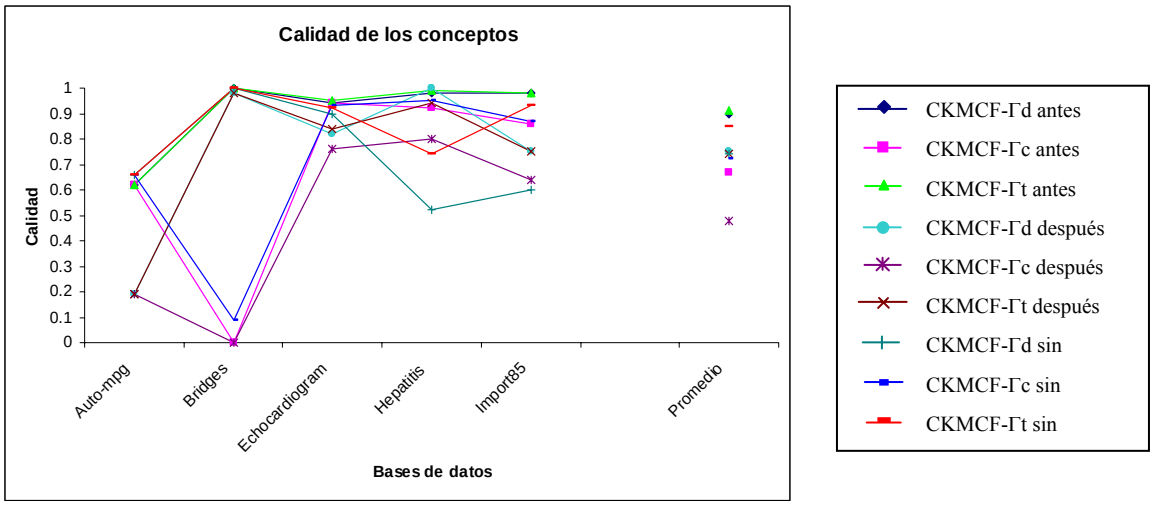


Figura 25. Calidades obtenidas por el algoritmo CKMCF utilizando diferentes conjuntos de apoyo y completando la información antes de agrupar, después de agrupar y sin completar.

En la gráfica de la Figura 25 podemos observar que los mejores resultados se obtienen cuando se seleccionan los conjuntos Γ-diferenciantes o conjuntos Γ-testores como conjuntos de apoyo y se completa la información antes de agrupar.

En la Figura 26 se muestra el número de predicados obtenidos con cada uno de los conjuntos de apoyo completando la información antes de agrupar, después de agrupar y sin completar la información.

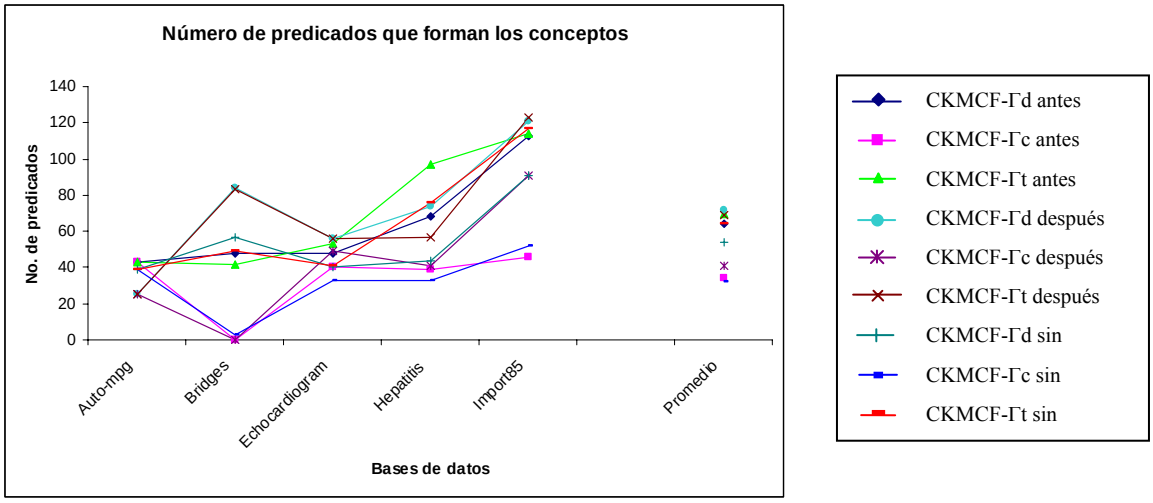


Figura 26. Número de predicados que forman los conceptos obtenidos con los diferentes conjuntos de apoyo para el algoritmo CKMCF completando la información antes de agrupar, después de agrupar y sin completar.

En la gráfica de la Figura 26 podemos observar que se obtiene menor número de predicados cuando se seleccionan los conjuntos Γ-caracterizantes como conjuntos de apoyo. Sin embargo, para la base de datos Bridges no fue posible generar conceptos, para todos los agrupamientos, con este tipo de conjuntos de apoyo.

En la Tabla 13 se muestran los resultados obtenidos aplicando el algoritmo genético con 10 iteraciones y 500 individuos, usando los tres tipos de conjuntos de apoyo, conjuntos Γ-diferenciantes, conjuntos Γ-caracterizantes y conjuntos Γ-testores. Para las bases de datos en las que se observa ausencia de información se muestran los resultados obtenidos completando la información antes de agrupar.

En la Tabla 13, podemos observar que las calidades de los conceptos y el número de predicados que forman dichos conceptos obtenidos con los diferentes tipos de conjuntos de apoyo, en la mayoría de los casos, son muy similares. En promedio se obtiene mejor calidad utilizando conjuntos de apoyo que son Γ-testores. Sin embargo, con este tipo de conjuntos se requiere, en promedio, mayor número de predicados para formar los conceptos.

Algoritmo Conceptual con Rasgos Complejos						
Bases de Datos	Γd		Γc		Γt	
	No. predicados	Calidad	No. predicados	Calidad	No. predicados	Calidad
Diabetes	44	0.89	44	0.89	44	0.89
Glass	21	0.66	20	0.66	21	0.66
Iris	3	0.88	3	0.88	3	0.88
Wine	30	1.00	27	1.00	29	1.00
Hayes	13	1.00	13	1.00	13	1.00
Lenses	8	1.00	8	1.00	8	1.00
Zoo	21	1.00	17	1.00	19	1.00
Auto-mpg	39	0.66	39	0.66	39	0.66
Bridges	57	1.00	3	0.09	49	1.00
Echocardiogram	40	0.90	33	0.93	41	0.92
Hepatitis	44	0.52	33	0.95	76	0.74
Import85	91	0.60	52	0.87	117	0.93
Tae	40	0.95	40	0.95	40	0.95
Promedio	**35**	**0.85**	**26**	**0.84**	**38**	**0.89**

Tabla 13. Resultados obtenidos con el algoritmo CKMCF usando tres tipos de conjuntos de apoyo.

En la Figura 27 se muestra de manera gráfica las calidades de los conceptos obtenidos con cada uno de los tipos de conjunto de apoyo.

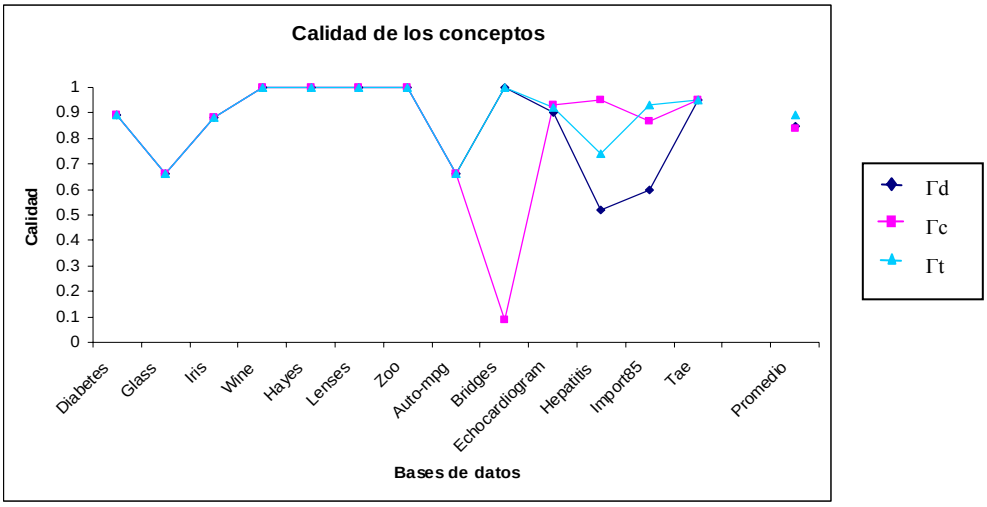


Figura 27. Calidades obtenidas por el algoritmo CKMCF usando diferentes tipos de conjuntos de apoyo.

En la gráfica de la Figura 27 podemos observar que en la mayoría de los casos se obtienen resultados similares con los tres tipos de conjuntos de apoyo. Sin embargo, para la base de datos Bridges, se obtuvo una mala calidad utilizando los conjuntos Γ-caracterizantes, esto es debido a que, con este tipo de conjuntos de apoyo, para esta base de datos no fue posible generar rasgos complejos para algunos agrupamientos. Por otro lado, para la base de datos Hepatitis se obtienen conceptos de mejor calidad cuando se utilizan conjuntos de apoyo Γ-caracterizantes, mientras que para la base de datos Import85 los conceptos de mejor calidad se obtienen con los conjuntos que son Γ-testores.

En la Figura 28 se muestra una comparación entre el número de predicados obtenidos con cada uno de los conjuntos de apoyo.

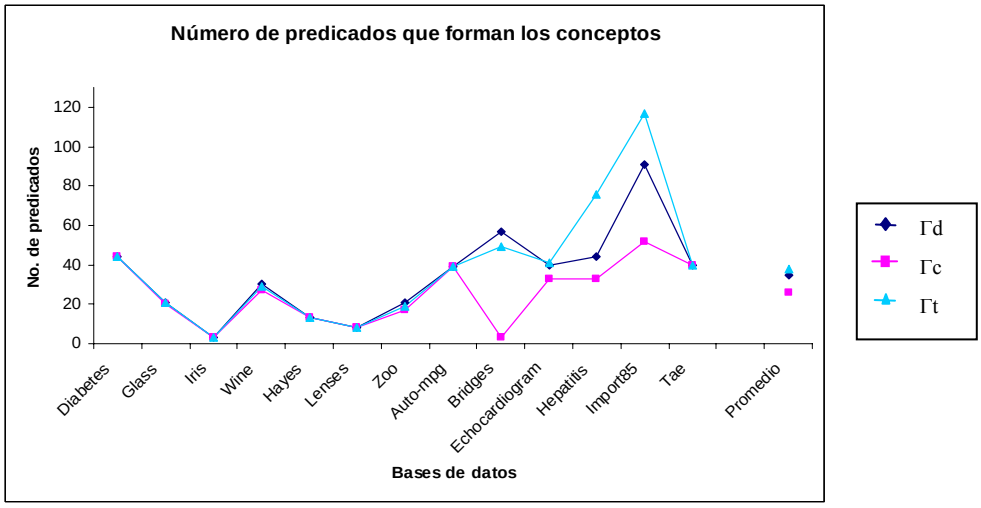


Figura 28. Número de predicados que forman los conceptos obtenidos utilizando los diferentes tipos de conjuntos de apoyo para el algoritmo CKMCF.

En la gráfica de la Figura 28 podemos observar que en algunos casos se obtiene el mismo número de predicados. En promedio los conjuntos Γ-caracterizantes obtuvieron un menor número de predicados. Sin embargo, para la base de datos Bridges no fue posible generar conceptos, para algunos agrupamientos, con este tipo de conjuntos de apoyo.

3.4.5. Discusión

El algoritmo k-means conceptual con rasgos complejos usa, en la fase de agrupamiento, una función de similaridad dada en términos de funciones de comparación, las cuales permiten expresar la forma en que los valores de los atributos son comparados dependiendo del contexto del problema a resolver. Las funciones de similaridad permiten trabajar con datos mezclados e incompletos sin transformar los atributos.

En la fase de caracterización se utilizó el concepto de rasgo complejo para generar los conceptos. Los rasgos complejos son subdescripciones de objetos que permiten diferenciar a objetos de distintos agrupamientos y al mismo tiempo permiten caracterizar a objetos de un mismo agrupamiento. Para calcular los rasgos complejos se utilizaron tres tipos diferentes de conjuntos de apoyo: conjuntos Γ-diferenciantes, conjuntos Γ-caracterizantes y conjuntos Γ-testores.

Con base en los resultados experimentales observamos que, para las bases de datos con ausencia de información, si se seleccionan como conjuntos de apoyo los conjuntos Γ-caracterizantes se obtienen mejores resultados cuando se trabaja sin completar la información; mientras que para los conjuntos Γ-diferenciantes y Γ-testores se obtienen mejores resultados cuando se completa la información antes de aplicar la fase de agrupamiento.

Por otra parte, el algoritmo k-means conceptual con rasgos complejos obtiene resultados similares con los tres tipos de conjuntos de apoyo, para la mayoría de las bases de datos.

En la siguiente sección se presenta una comparación entre los algoritmos k-means conceptual (CKM), k-means conceptual con funciones de similaridad (CKMSF) y k-means conceptual con rasgos complejos (CKMCF).

3.5. Comparación entre los Algoritmos Propuestos

Con el objetivo de comparar el desempeño de los algoritmos propuestos y del algoritmo k-means conceptual, se aplicaron los algoritmos CKM, CKMSF y CKMCF sobre las bases de datos de la Tabla 3.

Para las bases de datos que contienen ausencia de información se realizó una comparación entre los resultados obtenidos por los algoritmos CKM, CKMSF y CKMCF al completar la información antes de agrupar, completando la información después de agrupar (para CKMSF y CKMCF) y sin completar la información (solamente para CKMCF). Esta comparación se muestra en las Tablas 14 y 15.

En la Tabla 14 se muestran los resultados obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar y en la Tabla 15 se muestran los resultados obtenidos por los algoritmos CKMSF y CKMCF completando la información después de agrupar, así como los resultados obtenidos por el algoritmo CKMCF sin completar la información. Para el algoritmo CKM sólo se muestran los resultados obtenidos completando la información antes de agrupar ya que este algoritmo, no permite trabajar con valores ausentes.

Base de datos	Completando la información antes de agrupar				
	Algoritmo CKM	Algoritmo CKMSF	Algoritmo CKMCF		
			Γd	Γc	Γt
Auto-mpg	0.75	0.61	0.62	0.62	0.62
Bridges	0.78	0.95	1.00	0.00	1.00
Echocardiogram	0.40	0.88	0.94	0.94	0.95
Hepatitis	0.53	0.99	0.98	0.92	0.99
Import85	0.47	0.98	0.98	0.86	0.98
Promedio	**0.59**	**0.88**	**0.90**	**0.67**	**0.91**

Tabla 14. Calidades de los conceptos obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar.

Base de datos	Completando la información después de agrupar				Sin completar la información		
	Algoritmo CKMSF	Algoritmo CKMCF			Algoritmo CKMCF		
		Γd	Γc	Γt	Γd	Γc	Γt
Auto-mpg	0.51	0.19	0.19	0.19	0.66	0.66	0.66
Bridges	0.75	0.98	0.00	0.98	1.00	0.09	1.00
Echocardiogram	0.66	0.82	0.76	0.84	0.90	0.93	0.92
Hepatitis	0.97	1.00	0.80	0.94	0.52	0.95	0.74
Import85	0.84	0.75	0.64	0.75	0.60	0.87	0.93
Promedio	**0.75**	**0.75**	**0.48**	**0.74**	**0.74**	**0.72**	**0.85**

Tabla 15. Calidades de los conceptos obtenidos con el algoritmo CKMCF, completando la información antes de agrupar, completando la información después de agrupar y sin completar la información.

En las Tablas 14 y 15 podemos observar que los mejores resultados, en promedio, se obtienen con el algoritmo CKMCF usando conjuntos de apoyo Γ-testores y completando la información antes de agrupar; mientras que los peores resultados, en promedio, se obtienen con el algoritmo CKMCF usando conjuntos de apoyo Γ-caracterizantes y completando la información después de agrupar.

En la Figura 29 se muestra de manera gráfica los resultados mostrados en las Tablas 14 y 15.

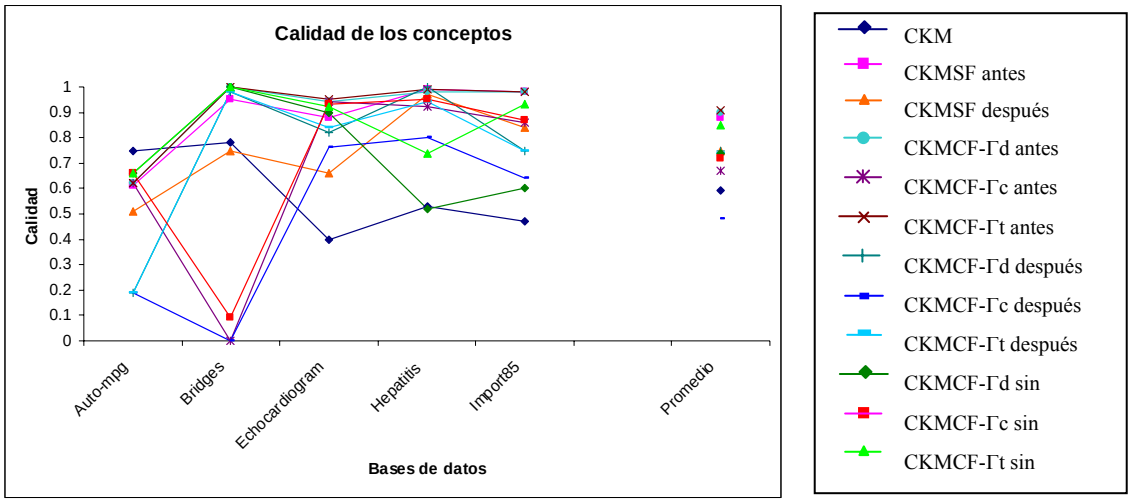


Figura 29. Calidades obtenidas por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar, después de agrupar y sin completar.

En la gráfica de la Figura 29 se puede ver que los mejores resultados, en promedio, se obtienen con el algoritmo CKMSF completando la información antes de agrupar (0.88); con el algoritmo CKMCF usando conjuntos de apoyo Γ-diferenciantes, completando la información antes de agrupar (0.90); con el algoritmo CKMCF usando conjuntos que son Γ-testores, completando la información antes de agrupar (0.91) y con el algoritmo CKMCF usando conjuntos Γ-testores, sin completar la información (0.85).

En las Tablas 16 y 17 se muestra el número de predicados obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar, después de agrupar y sin completar la información.

En la Tabla 16 se muestra el número de predicados obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar y en la Tabla 17 se muestra el número de predicados obtenidos por los algoritmos CKMSF y CKMCF completando la información después de agrupar, así como el número de predicados obtenidos por el algoritmo CKMCF sin completar la información.

Base de datos	Completando la información antes de agrupar				
	Algoritmo CKM	Algoritmo CKMSF	Algoritmo CKMCF		
			Γd	Γc	Γt
Auto-mpg	164	136	43	43	43
Bridges	30	33	48	0	42
Echocardiogram	43	89	48	40	53
Hepatitis	50	46	68	39	97
Import85	57	60	113	46	114
Promedio	**69**	**73**	**64**	**34**	**70**

Tabla 16. Número de predicados obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar.

Base de datos	Completando la información después de agrupar				Sin completar la información		
	Algoritmo CKMSF	Algoritmo CKMCF			Algoritmo CKMCF		
		Γd	Γc	Γt	Γd	Γc	Γt
Auto-mpg	202	25	25	25	39	39	39
Bridges	84	84	0	83	57	3	49
Echocardiogram	94	56	49	56	40	33	41
Hepatitis	72	74	41	57	44	33	76
Import85	124	121	91	123	91	52	117
Promedio	**115**	**72**	**41**	**69**	**54**	**32**	**64**

Tabla 17. Número de predicados obtenidos con el algoritmo CKMCF, completando la información antes de agrupar, completando la información después de agrupar y sin completar la información.

En las Tablas 16 y 17 podemos observar que se obtiene menor número de predicados con el algoritmo CKMCF usando conjuntos Γ-caracterizantes cuando no se completa la información. Sin embargo, para la base de datos Bridges no se generaron conceptos para todos los agrupamientos.

En la Figura 30 se muestra de manera gráfica los resultados mostrados en las Tablas 18 y 19.

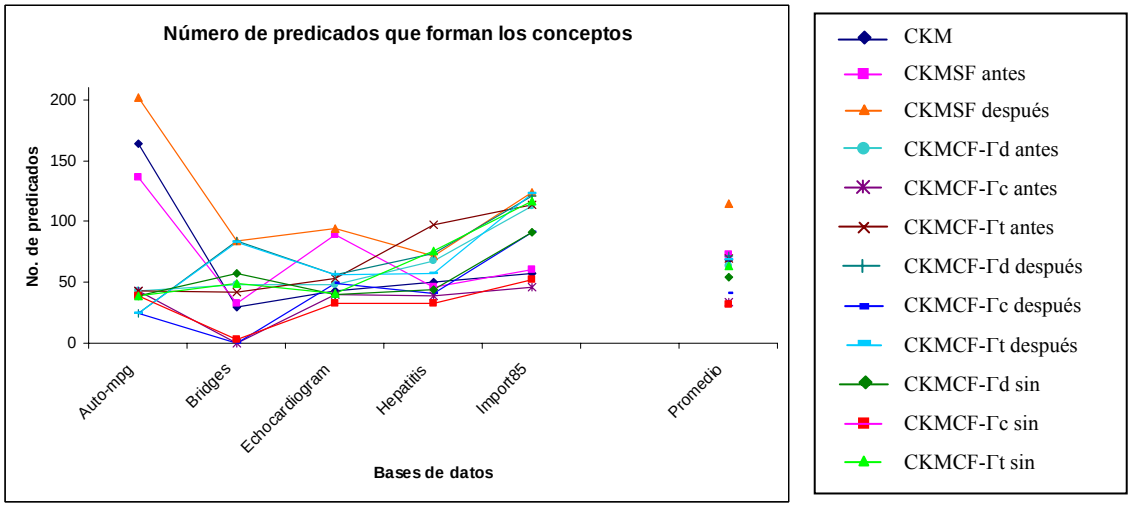


Figura 30. Número de predicados que forman los conceptos obtenidos por los algoritmos CKM, CKMSF y CKMCF completando la información antes de agrupar, después de agrupar y sin completar.

En la gráfica de la Figura 30 se puede ver que, en promedio, se obtienen menor número de predicados con el algoritmo CKMCF cuando se seleccionan conjuntos Γ-caracterizantes como conjuntos de apoyo, completando la información antes de agrupar; completando la información después de agrupar y sin completar la información. Sin embargo, para la base de datos Bridges no fue posible generar conceptos, para todos los agrupamientos, con este tipo de conjuntos de apoyo.

Finalmente, en las Tablas 18 y 19 se muestran los resultados obtenidos con los algoritmos CKM, CKMSF y CKMCF. Para las bases de datos con ausencia de información se muestran los resultados obtenidos cuando se completa la información antes de agrupar. En la Tabla 18 se muestra la calidad obtenida por cada uno de los algoritmos y en la Tabla 19 se muestra el número de predicados que forman los conceptos obtenidos por cada uno de los algoritmos.

Base de datos	Algoritmo CKM	Algoritmo CKMSF	Algoritmo CKMCF		
			Γd	Γc	Γt
Diabetes	0.53	0.83	0.89	0.89	0.89
Glass	0.54	0.89	0.66	0.66	0.66
Iris	0.85	0.92	0.88	0.88	0.88
Wine	0.29	1.00	1.00	1.00	1.00
Hayes	1.00	0.99	1.00	1.00	1.00
Lenses	1.00	0.95	1.00	1.00	1.00
Zoo	1.00	1.00	1.00	1.00	1.00
Auto-mpg	0.75	0.61	0.62	0.62	0.62
Bridges	0.78	0.95	1.00	0.00	1.00
Echocardiogram	0.40	0.88	0.94	0.94	0.95
Hepatitis	0.53	0.99	0.98	0.92	0.99
Import85	0.47	0.98	0.98	0.86	0.98
Tae	0.89	0.97	0.95	0.95	0.95
Promedio	**0.69**	**0.92**	**0.92**	**0.82**	**0.92**

Tabla 18. Calidades de los conceptos obtenidos por los algoritmos CKM, CKMSF y CKMCF.

En la Tabla 18 podemos observar que en promedio los algoritmos CKMSF y CKMCF con conjuntos Γ-diferenciantes y Γ-testores, tienen un desempeño similar mientras que el

algoritmo CKM obtiene calidades más bajas; sólo para la base de datos Auto-mpg el algoritmo CKM fue mejor que los algoritmos CKMSF y CKMCF.

Base de datos	Algoritmo CKM	Algoritmo CKMSF	Algoritmo CKMCF		
			Γd	Γc	Γt
Diabetes	261	218	44	44	44
Glass	67	83	21	20	21
Iris	52	10	3	3	3
Wine	47	40	30	27	29
Hayes	18	17	13	13	13
Lenses	5	8	8	8	8
Zoo	9	14	21	17	19
Auto-mpg	164	136	43	43	43
Bridges	30	33	48	0	42
Echocardiogram	43	89	48	40	53
Hepatitis	50	46	68	39	97
Import85	57	63	113	46	114
Tae	30	71	40	40	40
Promedio	**64**	**64**	**38**	**26**	**40**

Tabla 19. Número de predicados obtenidos por los algoritmos CKM, CKMSF y CKMCF.

En la Tabla 19 observamos que el algoritmo CKMCF genera, en promedio, menor número de predicados que los algoritmos CKM y CKMSF. Cuando se seleccionan, para el algoritmo CKMCF, los conjuntos Γ-caracterizantes como conjuntos de apoyo se obtienen menor número de predicados que los obtenidos por los algoritmos CKMSF y CKM; excepto para las bases de datos Lenses, Zoo y Tae donde el algoritmo CKM obtuvo menor número de predicados que los algoritmos CKMSF y CKMCF.

En las Figuras 31 y 32 se muestran gráficamente los resultados obtenidos por los algoritmos CKM, CKMSF y CKMCF. En la Figura 31 se muestra la calidad obtenida por cada uno de los algoritmos y en la Figura 32 se muestra el número de predicados que forman los conceptos obtenidos.

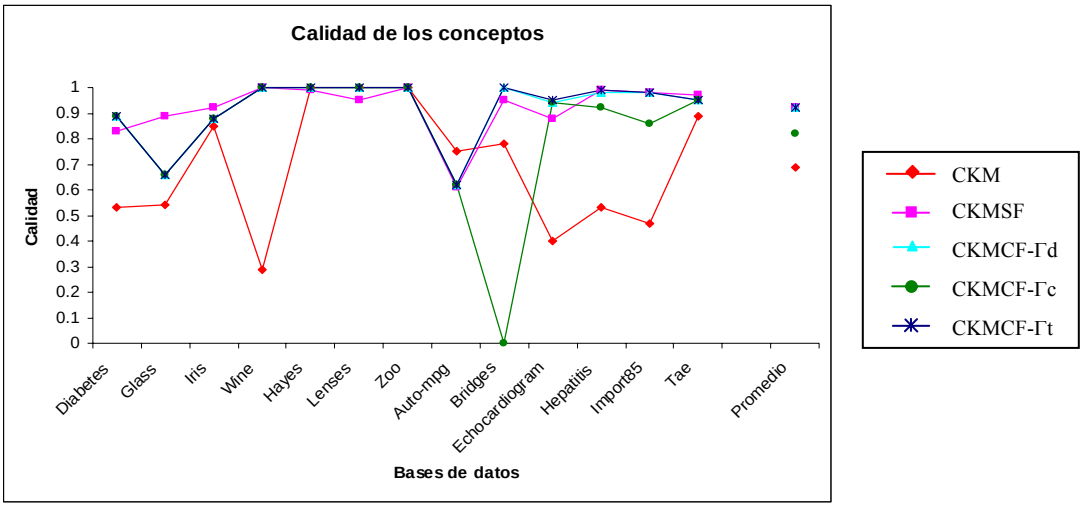


Figura 31. Calidades de los conceptos obtenidos por los algoritmos CKM, CKMSF y CKMCF.

En la gráfica de la Figura 31 podemos observar que los mejores resultados se obtienen, en la mayoría de los casos, con los algoritmos CKMSF y CKMCF. Sólo para la base de datos Auto-mpg el algoritmo CKM obtuvo mejores resultados. Para las bases de datos Glass, Iris y Tae se obtienen mejores resultados con el algoritmo CKMSF; mientras que para las bases de datos Diabetes y Echocardiogram se obtienen mejores resultados con el algoritmo CKMCF. Por otro lado, para la base de datos Bridges no fue posible generar conceptos, para todos los agrupamientos, cuando se utilizan conjuntos Γ-caracterizantes.

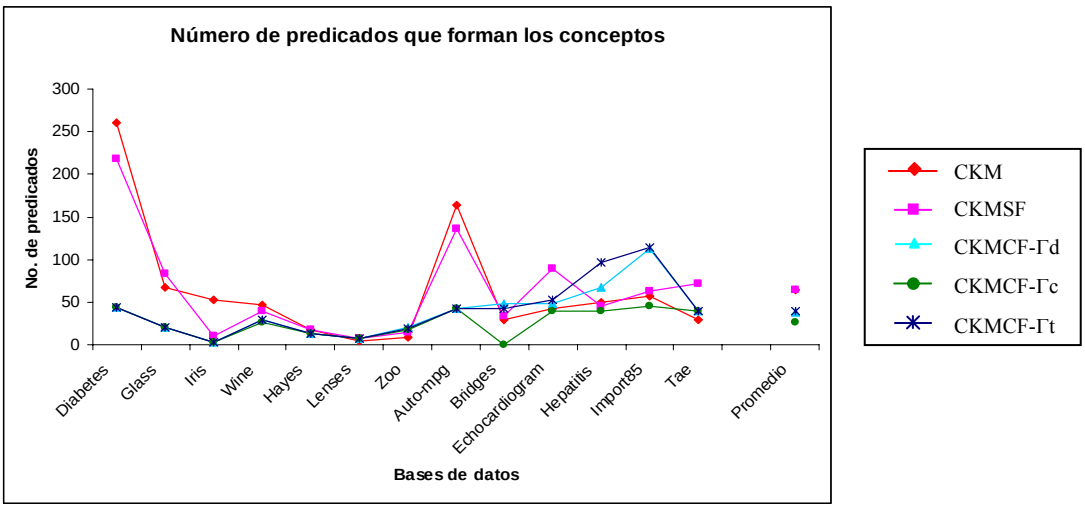


Figura 32. Número de predicados que forman los conceptos obtenidos por los algoritmos CKM, CKMSF y CKMCF.

En la gráfica de la Figura 32 podemos observar que el algoritmo CKMCF usando conjuntos Γ-caracterizantes obtiene menor número de predicados. Sin embargo, para la base de datos Bridges no fue posible generar conceptos, para todos los agrupamientos, cuando se utilizan este tipo de conjuntos de apoyo.

3.6. Análisis de Complejidad de los Algoritmos Conceptuales Duros

En esta sección se hace un análisis de la complejidad de los algoritmos CKMSF, CKMCF y CKM.

Sea n el número de objetos, k el número de agrupamientos, n_i el número de objetos en el agrupamiento A_i, m el número de atributos.

Fase de agrupamiento

Los algoritmos CKMSF y CKMCF utilizan en su fase de agrupamiento el algoritmo k-means con funciones de similaridad (KMSF). El algoritmo KMSF depende del número de iteraciones que el algoritmo requiera para que los centroides se estabilicen. Para cada iteración, n objetos son comparados con k centroides de los agrupamientos, para lo cual se requieren $n \times k$ llamadas a la función de similaridad. Para la ejecución completa del algoritmo KMSF, el tiempo requerido es $O(lnk)$, donde l es el número de iteraciones, en nuestros experimentos fijamos 10 como número máximo de iteraciones, pero en la mayoría de los casos se detenía en 4 o 5 iteraciones. La complejidad en espacio es $O(kn)$, que es el espacio requerido para almacenar los objetos.

Fase de caracterización

Algoritmo CKMSF

Inicialmente se generan predicados a partir de los objetos. A cada objeto de A_i se asigna un predicado y esto se hace para cada agrupamiento, esto es $\sum_{i=1}^{k} n_i = n$. Por lo tanto, el tiempo requerido para generar los predicados iniciales es $O(n)$.

Posteriormente se generan nuevos predicados que deben satisfacer la condición α-discriminante.
Sea q_s el número de valores posibles para x_s, s=1,..,m, entonces se tiene que el máximo número posible de predicados es $\prod_{s=1}^{m} q_s$. Si q es el máximo de los q_s, s=1,..,m, entonces la complejidad en el peor caso es $O(q^m)$.

Para cada uno de los q^m predicados generados se verifica la condición α-discriminante, lo cual requiere comparar contra los n-n_i objetos que no pertenecen al agrupamiento, en los m atributos, lo cual es $O(nm)$, y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para generar los predicados α-discriminantes es, en el peor caso, $O(knmq^m)$.

El siguiente paso consiste en verificar la condición β-caracterizante para cada predicado α-discriminante, lo cual requiere comparar contra los n_i objetos que están en el agrupamiento, en los m atributos, lo cual es $O(nm)$ y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para verificar la condición β-caracterizante es $O(knmq^m)$.

Finalmente se aplica el proceso de reducción de predicados.

Primero se ordenan los predicados de forma descendente con base en el número de objetos que cubre, lo cual requiere de un tiempo $O(q^m \log_2 q^m) = O(mq^m)$. Después se van almacenando aquellos predicados que cubren objetos que no han sido cubiertos lo cual

implica recorrer una lista en la cual se encuentran a lo más n objetos y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para reducir los predicados es $O(knmq^m)$

Por todo lo anterior el tiempo total requerido por el algoritmo CKMSF, en el peor caso, es $O(knmq^m)$.

El espacio requerido, en el peor caso, es $O(kmq^m)$, que es el espacio necesario para almacenar los predicados.

Algoritmo CKMCF

Inicialmente se calculan los conjuntos de apoyo utilizando un algoritmo genético. Para el algoritmo genético se fijaron el número de iteraciones y el tamaño de la población. Sea p el tamaño de la población y r el número de iteraciones. La complejidad de este paso es $O(prm)$. En nuestro caso p y r son constantes, por lo tanto, el tiempo requerido para calcular los conjuntos de apoyo es $O(m)$.

Posteriormente se calculan los rasgos complejos. El tiempo requerido para calcular los rasgos complejos (De-la-Vega-Doria, 1994) es $O(akn^3m)$, donde a es el número de conjuntos de apoyo.

El número máximo de rasgos complejos (De-la-Vega-Doria, 1994), que en nuestro caso es fijo, es $O(kamn)$ y a cada rasgo complejo se asocia un predicado lo cual requiere un tiempo de $O(kam^2n)$.

Finalmente se aplica el proceso de reducción de predicados.

Primero se ordenan los predicados de forma descendente con base en el número de objetos que cubre, lo cual requiere un tiempo de $(kam^2n)\ log_2\ (kam^2n)$. Después se van almacenando aquellos predicados que cubren objetos que no han sido cubiertos lo cual

implica recorrer una lista en la cual se encuentran a lo más n objetos y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para reducir los predicados es $O(kamn^2)$.

Por todo lo anterior, el tiempo total requerido por el algoritmo CKMSF es $O(akn^3m)$.

El espacio requerido es $O(kamn)$, que es el espacio necesario para almacenar los predicados.

Algoritmo CKM

El algoritmo CKM realiza un procedimiento similar al algoritmo CKMSF. Por lo tanto, es del mismo orden de complejidad.

En la Tabla 20 se muestra el tiempo y el espacio requeridos por cada uno de los algoritmos.

Complejidad	CKMSF	CKM	CKMCF
Tiempo	$O(knq^m m)$	$O(knq^m m)$	$O(akn^3 m)$
Espacio	$O(kmq^m)$	$O(kmq^m)$	$O(amn)$

Tabla 20. Complejidad en tiempo y espacio de los algoritmos duros.

3.7. Sumario

En este capítulo se dio un planteamiento formal del problema de agrupamiento conceptual restringido duro. Además se introdujeron el algoritmo k-means conceptual con funciones de similaridad (CKMSF) y el algoritmo k-means conceptual con rasgos complejos (CKMCF). Estos algoritmos permiten resolver problemas de agrupamiento conceptual restringido duro basado en semillas. Ambos algoritmos constan de dos fases:

una fase de agrupamiento, en la que se construyen los agrupamientos; y una fase de caracterización, en la que se generan los conceptos.

El algoritmo CKMSF, es una modificación del algoritmo k-means conceptual. En la fase de agrupamiento se utilizó el algoritmo k-means con funciones de similaridad (KMSF) para construir los agrupamientos. El algoritmo KMSF permite trabajar con datos mezclados sin realizar transformaciones de los atributos. Además, permite utilizar funciones de comparación que pueden ser definidas por el especialista, dependiendo del problema que se esté resolviendo. Por otra parte, en la fase de caracterización se uso un nuevo retículo de generalización para los atributos cuantitativos, el cual permite obtener conceptos de mejor calidad.

Un inconveniente al usar retículos de generalización es que, para algunas aplicaciones es difícil determinar cuál es el mejor retículo de generalización. Dependiendo del contexto del problema será la interpretación que se le dará a dicho retículo. Además, no se tienen métodos automáticos para construir los retículos, por lo que esta tarea se deja al especialista.

Por esta razón, se propuso el algoritmo CKMCF, el cual no depende de retículos de generalización para la generación de los conceptos de los agrupamientos. En este algoritmo, al igual que en el algoritmo CKMSF, se utilizó el algoritmo KMSF para construir los agrupamientos.

En la fase de caracterización se utilizaron los rasgos complejos para generar los conceptos de los agrupamientos. Para calcular los rasgos complejos se utilizaron tres tipos diferentes de conjuntos de apoyo, conjuntos Γ-diferenciantes, conjuntos Γ-caracterizantes y conjuntos Γ-testores.

Para evaluar la calidad de los conceptos obtenidos por los algoritmos de agrupamiento conceptual duro se propuso una función de calidad que toma en cuenta número de objetos

en el agrupamiento que son cubiertos por el concepto así como el número de objetos que están fuera del agrupamiento y que son cubiertos por el concepto.

Adicionalmente, se mostraron resultados experimentales obtenidos con los algoritmos CKMSF y CKMCF, así como una comparación entre los algoritmos CKM, CKMSF y CKMCF. En esta comparación se pudo observar que los algoritmos CKMSF y CKMCF tienen un desempeño similar, mientras que el algoritmo CKM obtiene conceptos de menor calidad. Por otra parte, los algoritmos CKM y CKMSF generan conceptos con mayor número de predicados, en la mayoría de los casos, que el algoritmo CKMCF.

A partir de los resultados experimentales, se concluye que los algoritmos CKMSF y CKMCF son una buena alternativa para la solución de problemas de agrupamiento conceptual restringido duro cuando los objetos están descritos por atributos cualitativos y cuantitativos mezclados y hay ausencia de información en las descripciones de los objetos.

Capítulo 4

Algoritmos Conceptuales Difusos

En este capítulo se introduce una formalización del problema de agrupamiento conceptual difuso, una función para evaluar la calidad de los conceptos difusos y una versión difusa de los algoritmos k-means conceptual con funciones de similaridad y k-means conceptual con rasgos complejos.

4.1. Introducción

El problema de agrupamiento conceptual difuso es importante, ya que en algunos problemas prácticos los especialistas están interesados en determinar en qué grado los objetos pertenecen a los agrupamientos más que en decidir si un objeto pertenece o no a un agrupamiento; así como determinar en qué medida estos objetos cumplen la propiedad o concepto de cada agrupamiento. Por lo tanto, se desea obtener conceptos difusos que nos proporcionen una descripción de cómo son los objetos que pertenecen con diferente grado a los agrupamientos.

El problema de agrupamiento conceptual difuso ha sido poco estudiado en la literatura (Martínez-Trinidad and Ruiz-Shulcloper, 1998; Martínez-Trinidad, 2000; Quan et al., 2004a; Quan et al., 2004b). Los trabajos que se han desarrollado hasta el momento resuelven el problema de agrupamiento conceptual cuando no se conoce el número de agrupamientos. Sin embargo, el problema de agrupamiento conceptual restringido difuso, es decir cuando se conoce *a priori* el número de agrupamientos, no ha sido abordado. Por esta razón, en esta tesis introducimos una formalización de este problema.

4.2. Definición del Problema de Agrupamiento Conceptual Difuso

El problema de agrupamiento conceptual difuso consiste en, dado un conjunto de objetos, construir no sólo una estructuración difusa de estos objetos, sino además una conceptualización de los agrupamientos formados, es decir, una propiedad que nos permita decir cómo es el agrupamiento. Por lo cual, se buscará expresarla en términos de las características que deben cumplir los objetos para pertenecer con cierto grado al agrupamiento difuso. Lo que se desea obtener es un concepto difuso para cada agrupamiento de tal manera que si un objeto pertenece con cierto grado a un agrupamiento, entonces ese objeto sea cubierto con ese grado por el concepto del agrupamiento.

En la siguiente sección introducimos una formalización del problema de agrupamiento conceptual difuso.

4.3. Planteamiento Formal del Problema

Consideremos un conjunto $X = \{O_1,...,O_n\}$ de *n* objetos. Cada objeto descrito por un conjunto $R = \{x_1,...,x_m\}$ de *m* atributos. Cada atributo x_s toma valores en un conjunto de valores admisibles D_s, $x_s(O_j) \in D_s, s = 1,...,m$. Los atributos pueden ser de cualquier naturaleza (cualitativo: Booleano, *k*-valente, nominal; o cuantitativo: entero, real, etc.). Además, asumiremos que en D_s existe un símbolo "?" para denotar ausencia de información, por lo que pueden ser consideradas descripciones incompletas.

Para cada atributo se define una función de comparación $FC_s : D_s \times D_s \to L_s$ *s*=1,2,...,*m*, donde L_s es un conjunto totalmente ordenado. La función FC_s es una evaluación del grado de similaridad entre dos valores del atributo x_s. Además, sea $\Gamma : (D_1 \times ... \times D_m)^2 \to [0,1]$ una función de similaridad, la cual permite evaluar el grado de similaridad entre dos descripciones de objetos.

El problema de agrupamiento conceptual restringido difuso consiste en encontrar k agrupamientos difusos $\{A_1,...,A_k\}$, $k>1$ del conjunto de objetos T, así como las propiedades o conceptos difusos, C_i, que caracterizan a los agrupamientos difusos A_i, $i = 1,\ldots,k$.

Un concepto difuso C_i para el agrupamiento difuso A_i debe satisfacer que si el objeto O pertenece con alto grado al agrupamiento difuso A_i entonces debería ser cubierto con alto grado por el concepto C_i y si el objeto O pertenece con grado bajo al agrupamiento difuso A_i entonces debería ser cubierto con grado bajo por el concepto C_i. Para simplificar el problema de definir conceptos difusos, en esta tesis se propone una manera de describirlos tomando como base la idea de los conceptos duros.

Un concepto difuso C_i estará formado por un conjunto de pares (P,μ_P), con P un predicado duro que describe cómo son ciertos objetos y μ_P un valor que se asociará a los objetos que son cubiertos por el predicado P, es decir, P describe cómo son los objetos que pertenecen al agrupamiento A_i y μ_P es el grado en que son cubiertos los objetos descritos por el predicado P. A cada par (P,μ_P) lo denominaremos predicado difuso.

El concepto difuso C_i estará formado por una disyunción de predicados difusos, con la cual se asigna a cada objeto el grado en que es cubierto por el concepto, de la siguiente manera: Si un objeto O_j es cubierto por un solo predicado difuso P, entonces el grado en que el objeto O_j es cubierto por el concepto difuso C_i será el valor μ_P asociado al predicado P. Si el objeto O_j es cubierto por varios predicados difusos de C_i, el grado en que el objeto O_j es cubierto por el concepto difuso C_i será el máximo de los μ_P asociados a los predicados que lo cubren; por otro lado, si ningún predicado cubre al objeto O_j, entonces diremos que el concepto difuso C_i cubre al objeto O_j con grado 0.

Ejemplo 4.1: Supongamos que se tiene la muestra de la Tabla 21. Esta muestra contiene cuatro objetos descritos por tres atributos y supongamos que se construyeron los dos agrupamientos difusos mostrados en las columnas A_1 y A_2 de la Tabla 21. Los objetos O_1 y

O_2 obtienen su máximo grado de pertenencia hacia el agrupamiento difuso A_1 y los objetos O_3 y O_4 obtienen su máximo grado de pertenencia hacia el agrupamiento difuso A_2.

Objetos	Atributos			Agrupamientos Difusos	
	Color (C)	Peso (P)	Forma (F)	A_1	A_2
O_1	Rojo	20	Redondo	0.80	0.20
O_2	Rojo	25	Redondo	0.70	0.30
O_3	Amarillo	30	Triangular	0.30	0.70
O_4	Amarillo	35	Triangular	0.20	0.80

Tabla 21. Muestra con 4 objetos descritos por 3 atributos.

Algunos predicados que pudieran estar asociados a los objetos de la muestra de la Tabla 21 son los siguientes:

P_1: $\left(\left(Color, Rojo\right) \wedge \left(Peso, 20\right) \wedge \left(Forma, Redondo\right), 0.80\right)$

El predicado P_1 cubre únicamente al objeto O_1 ya que toma los mismos valores en cada uno de los atributos y el grado en que el predicado P_1 cubre al objeto O_1 es 0.80. El grado en que los objetos O_2, O_3 y O_4 son cubiertos por el predicado P_1 es 0, ya que para algunos atributos el predicado P_1 toma valores diferentes. Intuitivamente, este predicado debería estar asociado al agrupamiento A_1.

P_2: $\left(\left(Color, Rojo\right), 0.75\right)$

El predicado P_2 cubre a los objetos O_1 y O_2 con un grado de 0.75. El grado en que los objetos O_3 y O_4 son cubiertos por el predicado P_2 es 0.

P_3: $\left(\left(Forma, Triangular\right), 0.75\right)$

El predicado P_3 cubre a los objetos O_3 y O_4 con un grado de 0.75.

Si el agrupamiento difuso A_1 tuviera asociada la disyunción entre los predicados P_1 y P_2, el concepto C_1 que caracteriza a A_1 quedaría de la siguiente manera:

$$\boldsymbol{C_1}: \ ((Color, Rojo) \wedge (Peso, 20) \wedge (Forma, Redondo), 0.8) \ \vee \ ((Color, Rojo), 0.75)$$

el cual cubre a los objetos O_1 y O_2 con los siguientes grados: $\mu_P(O_1) = 0.8$ y $\mu_P(O_2) = 0.75$ y cubre a los objetos O_3 y O_4 con grado 0.

Si el agrupamiento difuso A_2 tuviera asociado sólo un predicado (P_3), el concepto C_2 de A_2 sería:

$$\boldsymbol{C_2}: \ ((Forma, Triangular), 0.75)$$

el cual cubre a los objetos O_3 y O_4 con los siguientes grados: $\mu(O_3) = 0.75$ y $\mu(O_4) = 0.75$ y cubre a los objetos O_1 y O_2 con grado 0.

En la Tabla 22 se muestran los grados de pertenencia de los objetos a los agrupamientos difusos, así como los grados en que el concepto cubre a los objetos de la Tabla 21.

Objetos	Atributos			Agrupamientos Difusos		Grado en que el concepto cubre a los objetos	
	Color (C)	Peso (P)	Forma (F)	A_1	A_2	C_1	C_2
O_1	Rojo	20	Redondo	0.80	0.20	0.80	0.00
O_2	Rojo	25	Redondo	0.70	0.30	0.75	0.00
O_3	Amarillo	30	Triangular	0.30	0.70	0.00	0.75
O_4	Amarillo	35	Triangular	0.20	0.80	0.00	0.75

Tabla 22. Grados de pertenencia de los objetos a los agrupamientos y grados en que el concepto cubre a los objetos.

Como se mencionó anteriormente, un concepto difuso está formado por predicados que describen cómo son los objetos que pertenecen con cierto grado al agrupamiento. Por lo tanto, un concepto difuso nos da una descripción de un agrupamiento difuso expresada en términos de las características que deben cumplir los objetos para pertenecer con cierto grado al agrupamiento difuso. Por consiguiente, estos conceptos podrían utilizarse para asignar, a nuevos objetos, grados de pertenencia al agrupamiento difuso; para lo cual se asociará como grado de pertenencia, del nuevo objeto al agrupamiento difuso, el grado con que el objeto es cubierto por el concepto difuso correspondiente.

Para medir la calidad de los conceptos difusos es necesario definir una función de calidad, la cual se presenta en la siguiente sección.

4.4. Función de Calidad

En el problema de agrupamiento conceptual duro, se desea que los conceptos cubran a los objetos que pertenecen al agrupamiento y no cubran objetos que pertenecen a otros agrupamientos. Por esta razón, se propuso una función de calidad que toma en cuenta el número de objetos que pertenecen al agrupamiento que son cubiertos por el concepto así como el número de objetos que no pertenecen al agrupamiento pero que son cubiertos por el concepto.

Sin embargo, en el problema de agrupamiento conceptual difuso se desea que los objetos que pertenecen con alto grado al agrupamiento sean cubiertos con alto grado (cercano a 1) por el concepto, y que objetos que pertenecen con bajo grado al agrupamiento sean cubiertos con un grado bajo (cercano a 0) por el concepto; además, que la diferencia entre el grado en que un objeto es cubierto por el concepto y el grado de pertenencia de ese objeto al agrupamiento sea mínima.

Por lo tanto, para evaluar la calidad de los conceptos obtenidos por los algoritmos conceptuales difusos, proponemos una generalización de la función de calidad propuesta para el caso duro, la cual tome en cuenta, para aquellos objetos que pertenecen con alto grado al agrupamiento, la cercanía entre el grado en que un objeto es cubierto por el concepto y el grado de pertenencia del objeto al agrupamiento, y al mismo tiempo, para aquellos objetos que pertenecen con grado bajo al agrupamiento, la lejanía entre el grado en que un objeto es cubierto por el concepto y el grado de pertenencia del objeto al agrupamiento.

La función que se propone es la siguiente:

$$calidad(C_1,...,C_k) = \frac{1}{k}\sum_{i=1}^{k}\frac{\sum_{O\in A_{C_i}}\left(1-\left|\mu_{A_i}(O_j)-\mu_{C_i}(O_j)\right|\right)}{\left|A_{C_i}\right|+\sum_{O\notin A_{C_i}}\left|\mu_{A_i}(O_j)-\mu_{C_i}(O_j)\right|} \qquad (4.1)$$

donde:

$A_{c_i} = \left\{O_j \,\middle|\, \max_{p=1,...,k}\left\{\mu_{A_p}(O_j)\right\} = \mu_{A_i}(O_j)\right\}$ es el conjunto de objetos que obtienen su máximo grado al agrupamiento A_i , $i = 1,\ldots,k$.

k es el número de agrupamientos.

C_i es el concepto asociado al agrupamiento A_i.

$\mu_{A_i}(O_j)$ es la pertenencia del objeto O_j al agrupamiento A_i.

$\mu_{C_i}(O_j)$ es el grado en que el objeto O_j es cubierto por el concepto C_i.

$\left|A_{C_i}\right|$ es el número de objetos que alcanzan su máxima pertenencia hacia el agrupamiento A_i.

Esta función obtiene valores altos si la diferencia entre la pertenencia de los objetos al agrupamiento y el grado en que los objetos son cubiertos por los conceptos es pequeña (cercana a 0). La función obtiene 1.0 (que es el caso ideal) cuando los conceptos cubren a todos los objetos en el mismo grado en que estos pertenecen a los agrupamientos.

Si los grados de pertenencia fueran duros (0's y 1's) entonces esta función sería la misma que la función de calidad definida para el problema de agrupamiento conceptual duro (ver expresión (3.2)), ya que los objetos que alcanzarían su máximo grado de pertenencia hacia el agrupamiento A_i serían los mismos objetos que pertenecen al agrupamiento A_i, es decir, A_{C_i} sería igual a A_i. Por otro lado, como los grados de pertenencia de los objetos a los agrupamientos son 0 o 1, en la expresión $\sum_{O_j\in A_{C_i}}\left(1-\left|\mu_{A_i}(O_j)-\mu_{C_i}(O_j)\right|\right)$ se sumaría un 1 sólo cuando $\mu_{C_i}(O_j)=1$ (puesto que $\mu_{A_i}(O_j)=1$ para todo $O_j \in A_{C_i}$), es decir, cuando un objeto que pertenece al agrupamiento A_i es cubierto por el concepto C_i. Por lo tanto se toma en cuenta el número de

objetos que pertenecen al agrupamiento A_i que son cubiertos por el concepto C_i (*ejemplos*(C_i)); y en la expresión $\sum_{O_j \notin A_{C_i}} \left| \mu_{A_i}(O_j) - \mu_{C_i}(O_j) \right|$ se sumaría un 1 sólo cuando $\mu_{C_i}(O_j) = 1$ (puesto que $\mu_{A_i}(O_j) = 0$ para todo $O_j \notin A_{C_i}$), es decir, cuando un objeto fuera del agrupamiento A_i es cubierto por el concepto C_i, esto es, se toma en cuenta el número de objetos fuera del agrupamiento A_i que son cubiertos por el concepto C_i (*contraejemplos*(C_i)).

A continuación se introducen dos algoritmos de agrupamiento conceptual difuso, los cuales son una versión difusa de los algoritmos propuestos en el Capítulo 3. El primer algoritmo es una versión difusa del algoritmo k-means conceptual con funciones de similaridad y el segundo una versión difusa del algoritmo k-means conceptual con rasgos complejos.

Los algoritmos que se proponen en este capítulo constan de dos fases: una fase de agrupamiento, en la cual se forman los agrupamientos difusos y una fase de caracterización en la cual se generan los conceptos o propiedades difusas que caracterizan a los agrupamientos.

4.5. Algoritmo K-means Conceptual Difuso con Funciones de Similaridad

El algoritmo k-means conceptual difuso con funciones de similaridad (FCKMSF) es una extensión del algoritmo k-means conceptual con funciones de similaridad (CKMSF).

El algoritmo que se propone consta de dos fases: una fase de agrupamiento y una fase de caracterización. Este algoritmo utiliza, en la fase de agrupamiento, el algoritmo k-means difuso con funciones de similaridad (Ayaquica-Martínez and Martínez-Trinidad, 2001; Ayaquica-Martínez, 2002), el cual permite trabajar con atributos cualitativos y cuantitativos mezclados. En la fase de caracterización se propone una manera de utilizar los retículos de generalización para construir conceptos difusos a partir de los agrupamientos difusos obtenidos en la fase de agrupamiento.

4.5.1. Fase de Agrupamiento

En esta fase se utiliza el algoritmo k-means difuso con funciones de similaridad (FKMSF) (Ayaquica-Martínez and Martínez-Trinidad, 2001; Ayaquica-Martínez, 2002) para construir agrupamientos difusos a partir de los objetos de la muestra dada. El algoritmo FKMSF es una versión difusa del algoritmo KMSF (García-Serrano and Martínez-Trinidad, 1999) utilizado en la fase de agrupamiento de los algoritmos duros, propuestos en el Capítulo 3.

El objetivo del algoritmo FKMSF es obtener agrupamientos difusos con la característica de que objetos con alto grado de pertenencia hacia un mismo agrupamiento sean similares entre sí, y objetos con alto grado de pertenencia hacia agrupamientos diferentes sean poco similares.

4.5.2. Fase de Caracterización

En esta fase se propone una generalización de la fase de caracterización del algoritmo CKMSF que nos permitirá generar conceptos difusos a partir de los agrupamientos difusos.

Cabe recordar que para aplicar el algoritmo CKMSF cada atributo debe tener asociado un retículo de generalización. El retículo de generalización para los atributos cualitativos está dado de antemano por el especialista, mientras que para los atributos cuantitativos el retículo de generalización se obtiene a partir de una función de codificación (ver expresión (2.8) de la Sección 2.2.3).

Una vez realizada la codificación de los atributos cuantitativos y definidos los retículos de generalización, se forman los predicados iniciales. Para formar los predicados iniciales tomamos en cuenta sólo los objetos que pertenecen con máximo grado al agrupamiento A_i. A cada objeto O_j se asocia un predicado (P, μ_P), donde P se forma de la misma manera que en el caso duro (ver Sección 3.3.2) y μ_P toma como valor el grado de pertenencia del objeto O_j al agrupamiento A_i.

Ejemplo 4.2: Supongamos que después de aplicar la fase de agrupamiento para los objetos de la Tabla 23 se obtienen los agrupamientos difusos mostrados en las columnas A_1 y A_2 de la Tabla 23.

Objetos	Atributos				Agrupamientos Difusos	
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)	A_1	A_2
O_1	rojo	chico	20	redondo	1.00	0.00
O_2	rojo	mediano	20	redondo	0.94	0.06
O_3	azul	chico	25	redondo	0.90	0.10
O_4	azul	mediano	25	cuadrado	0.50	0.50
O_5	verde	grande	30	triangular	0.06	0.94
O_6	verde	chico	20	redondo	0.94	0.06
O_7	amarillo	grande	30	triangular	0.00	1.00
O_8	amarillo	mediano	35	triangular	0.06	0.94
O_9	verde	grande	35	redondo	0.31	0.69

Tabla 23. Muestra con 9 objetos descritos por 4 atributos.

Para estos agrupamientos, los predicados iniciales son los siguientes:

Para A_1:

P_1: $((C, rojo) \wedge (T, chico) \wedge (P, typical) \wedge (F, redondo), 1.00)$

P_2: $((C, rojo) \wedge (T, mediano) \wedge (P, typical) \wedge (F, redondo), 0.94)$

P_3: $((C, azul) \wedge (T, chico) \wedge (P, sup) \wedge (F, redondo), 0.90)$

P_4: $((C, azul) \wedge (T, mediano) \wedge (P, inf) \wedge (F, cuadrado), 0.50)$

P_5: $((C, verde) \wedge (T, chico) \wedge (P, typical) \wedge (F, redondo), 0.94)$

Para A_2:

P_1: $((C, azul) \wedge (T, mediano) \wedge (P, inf) \wedge (F, cuadrado), 0.50)$

P_2: $((C, verde) \wedge (T, grande) \wedge (P, typical) \wedge (F, triangular), 0.94)$

P_3: $((C, amarillo) \wedge (T, grande) \wedge (P, typical) \wedge (F, triangular), 1.00)$

P_4: $((C, amarillo) \wedge (T, mediano) \wedge (P, typical) \wedge (F, triangular), 0.94)$

P_5: $((C, verde) \wedge (T, grande) \wedge (P, typical) \wedge (F, redondo), 0.69)$

A partir de estos predicados y con base en los retículos de generalización se generan nuevos predicados difusos (P,μ_P). Para generalizar dos predicados difusos (P_1,μ_{P_1}) y (P_2,μ_{P_2}) se verifica si P_1 y P_2 pueden generalizarse y además si se cumple que $|\mu_{P_1}-\mu_{P_2}|<\varepsilon$, con $\varepsilon\in[0,1]$. Es decir, se generalizan sólo aquellos predicados difusos que tienen un valor de μ_P similar, con el objetivo de obtener predicados difusos más generales a partir de predicados difusos que asocien grados parecidos a los objetos que cubren. El predicado *P* del nuevo predicado difuso será la generalización de P_1 y P_2 y el valor de μ_P será el promedio de los valores de μ_{P_1} y μ_{P_2}.

Es importante hacer notar que si el valor de ε es cercano a 1, entonces obtendremos conceptos difusos con baja calidad, ya que se permite generalizar predicados difusos que representan objetos con pertenencias poco similares; mientras que si el valor de ε es cercano a cero, los conceptos difusos que se obtengan serán de mejor calidad, ya que en este caso sólo se permite generalizar predicados difusos que representan objetos con pertenencias muy similares, pudiendo llegar a ser demasiado específicos, si ε=0 los predicados difusos finales serán todos los predicados difusos iniciales y cada predicado difuso cubrirá sólo a un objeto de la muestra original.

Posteriormente, se verifica la condición de ser α-discriminante (el número de contraejemplos es menor o igual que α). Si un nuevo predicado difuso es α-discriminante se almacena, en caso contrario se elimina. Si un nuevo predicado difuso fue eliminado entonces los predicados difusos iniciales, a partir de los cuales se generó dicho predicado difuso, se almacenan. Este proceso se repite hasta que ya no sea posible generar nuevos predicados difusos α-discriminantes.

Para el ejemplo 4.2, los predicados difusos α-discriminantes, con α=1 y ε=0.2, son los siguientes:

Para A_1:

P'1: $((C,rojo) \wedge (T,*) \wedge (P,typical) \wedge (F,redondo), 0.97)$

P'2: $((C,*) \wedge (T,*) \wedge (P,*) \wedge (F,redondo), 0.94)$

Para A_2:

P'1: $((C,*) \wedge (T,grande) \wedge (P,typical) \wedge (F,triangular), 0.97)$

P'2: $((C,amarillo) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular), 0.97)$

P'3: $((C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular), 0.97)$

Una vez generados todos los predicados difusos α-discriminantes, se eliminan aquellos predicados que no cumplen la propiedad de ser β-caracterizante (el número de ejemplos cubiertos es mayor o igual que β).

Para el ejemplo 4.2, los predicados difusos α-discriminantes que son β-caracterizantes, tomando β=3, son:

Para A_1:

P'1: $((C,*) \wedge (T,*) \wedge (P,*) \wedge (F,redondo), 0.94)$

Para A_2:

P'5: $((C,*) \wedge (T,*) \wedge (P,typical) \wedge (F,triangular), 0.97)$

El conjunto de predicados difusos obtenido puede contener predicados que no contribuyan a mejorar la calidad del concepto; por lo tanto, este conjunto de predicados puede reducirse. Esta reducción se hace utilizando una modificación a la estrategia propuesta por Ralambondrainy (1995) que trabaja como sigue: los predicados difusos se ordenan en forma descendente de acuerdo al valor de μ_P. El primer predicado formará parte del concepto. Para los predicados restantes se verifica si al agregar un predicado mejora la calidad del concepto medida con la expresión (4.1), de ser así, ese predicado se agrega al

concepto; si no, se elimina. Finalmente, el concepto estará formado por la disyunción de los predicados difusos almacenados.

Después de aplicar el proceso antes descrito, sobre los predicados difusos del ejemplo y eliminando los atributos que pueden tomar cualquier valor, es decir, los atributos que tienen * como valor; los conceptos difusos obtenidos son los siguientes:

Para A_1:

C_1: $\left(\left(Forma, redondo\right), 0.94\right)$

Para A_2:

C_2: $\left(\left(26.8167 < Peso < 35.1833\right) \wedge \left(Forma, triangular\right), 0.97\right)$

El grado en que un concepto difuso cubre a un objeto se obtiene de la siguiente manera: si un objeto *O* es cubierto por un solo predicado difuso *P*, entonces el grado en que el objeto *O* es cubierto por el concepto difuso C_i será el valor de μ_P asociado al predicado *P*. Si el objeto *O* es cubierto por varios predicados difusos, el grado en que el objeto *O* es cubierto por el concepto difuso C_i será el máximo de los μ_P asociados a los predicados que lo cubren; por otro lado, si ningún predicado cubre al objeto *O*, entonces el concepto difuso C_i cubre al objeto *O* con grado 0.

Para el ejemplo 4.2, los grados en que el concepto cubre a cada uno de los objetos, se muestran en las columnas C_1 y C_2 de la Tabla 24.

Objetos	Atributos				Agrupamientos Difusos		Grado en que el concepto cubre a los objetos	
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)	A_1	A_2	C_1	C_2
O_1	rojo	chico	20	redondo	1.00	0.00	0.94	0.00
O_2	rojo	mediano	20	redondo	0.94	0.06	0.94	0.00
O_3	azul	chico	25	redondo	0.90	0.10	0.94	0.00
O_4	azul	mediano	25	cuadrado	0.50	0.50	0.00	0.00
O_5	verde	grande	30	triangular	0.06	0.94	0.00	0.97
O_6	verde	chico	20	redondo	0.94	0.06	0.94	0.00
O_7	amarillo	grande	30	triangular	0.00	1.00	0.00	0.97
O_8	amarillo	mediano	35	triangular	0.06	0.94	0.00	0.97
O_9	verde	grande	35	redondo	0.31	0.69	0.94	0.00

Tabla 24. Grados de pertenencia de los objetos a los agrupamientos y grados en que el concepto cubre a los objetos.

Finalmente, se evalúa la calidad de los conceptos utilizando la expresión (4.1). La calidad obtenida, para el ejemplo, con la función de calidad propuesta es: 0.74.

El algoritmo FCKMSF es el siguiente:

4.5.3. Algoritmo FCKMSF

Entrada: Un conjunto T de objetos a ser agrupados.

Un número k de agrupamientos deseados.

Los umbrales, α, β y ε para la fase de caracterización.

Salida: Una partición difusa $\{A_1,\ldots,A_k\}$ en k agrupamientos difusos de T y el concepto difuso C_i que caracteriza a cada agrupamiento difuso A_i, $i = 1,\ldots,k$.

Fase de agrupamiento

Paso 1: Aplicar el algoritmo k-means difuso con funciones de similaridad, para generar los agrupamientos difusos A_i, $i = 1,\ldots,k$.

Fase de caracterización

Paso 1: Para cada A_i, $i = 1,\ldots,k$ hacer

Paso 2: Construir el predicado inicial para cada objeto $O \in A_i$.

Paso 3: Generar nuevos predicados difusos a partir de la generalización de dos predicados y verificando que $|\mu_P - \mu_{P'}| < \varepsilon$.

Paso 4: De los predicados obtenidos en el paso 2, almacenar aquellos que sean α-discriminantes.

Paso 5: Si se almacenaron nuevos predicados entonces

Ir al paso 3.

Si no,

Ir al paso 6.

Paso 6: Eliminar todos los predicados que no sean β-caracterizantes.

Paso 7: Reducir el número de predicados utilizando el procedimiento propuesto.

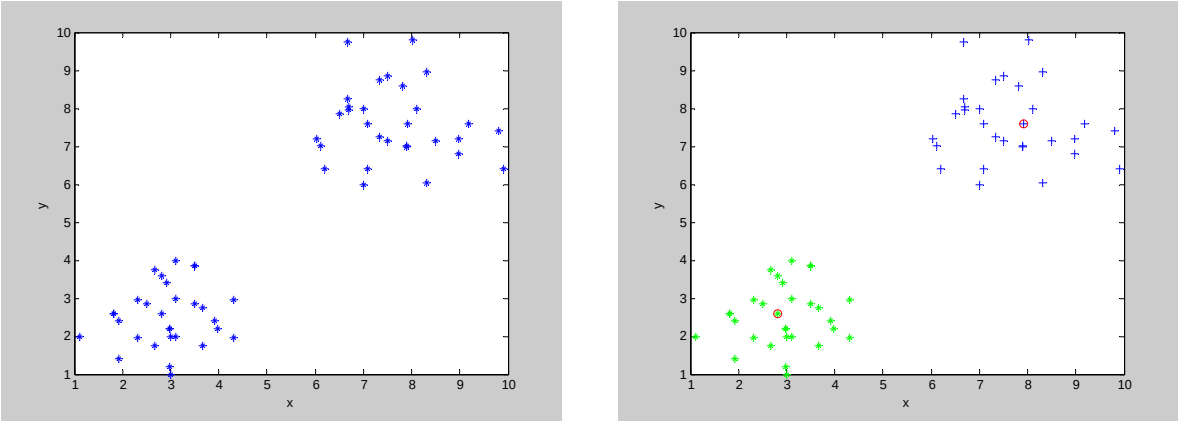
Paso 8: Construir el concepto C_i como la disyunción de los predicados difusos obtenidos en el paso 7.

En la siguiente sección se muestran los resultados experimentales obtenidos con el algoritmo k-means conceptual difuso con funciones de similaridad (FCKMSF).

4.5.4. Resultados Experimentales

Inicialmente, para ilustrar el comportamiento del algoritmo k-means conceptual difuso con funciones de similaridad (FCKMSF) se utilizaron bases de datos sintéticas que contienen objetos descritos por atributos numéricos. Posteriormente, para mostrar el desempeño del algoritmo, con base en la función de calidad propuesta en la sección 4.4, se aplicó el algoritmo FCKMSF sobre las mismas bases de datos utilizadas para los algoritmos duros.

***Ejemplo 4.3*:** En este ejemplo se muestra el comportamiento del algoritmo FCKMSF cuando los agrupamientos que se desean obtener están bien definidos. Para este ejemplo se tomaron 60 objetos descritos por 2 atributos numéricos, como se muestran en la Figura 33 a).



a) **b)**

Figura 33. a) Muestra de datos, **b)** Agrupamientos obtenidos en la fase de agrupamiento del algoritmo FCKMSF.

En la Figura 33 b) se muestran los agrupamientos obtenidos después de aplicar la fase de agrupamiento del algoritmo FCKMSF. Los objetos en verde obtuvieron un mayor grado de pertenencia hacia el agrupamiento A_1 y los objetos en azul obtuvieron un mayor grado de pertenencia hacia el agrupamiento A_2. Los objetos encerrados en un círculo rojo son los centroides finales de los agrupamientos obtenidos por el algoritmo FKMSF.

Sobre los agrupamientos obtenidos en la fase de agrupamiento (ver Figura 33 b)) se aplicó la fase de caracterización y los conceptos obtenidos fueron los siguientes:

Para el agrupamiento A_1:

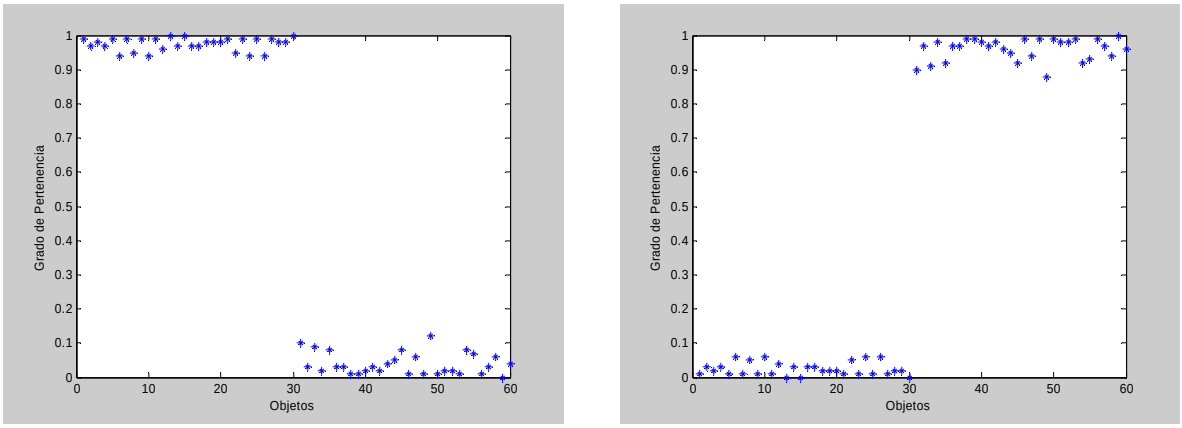
C_1: $((2.1706 < x < 3.7052), 1.00)$
$\vee$ $((1.7383 < y < 3.3305), 1.00)$
$\vee$ $((x < 2.1706), 0.98)$

Para el agrupamiento A_2:

$$\boldsymbol{C_2}: \quad ((6.5859 < x < 8.6807) \wedge (6.6157 < y < 8.5843), 1.00)$$
$$\vee \ ((6.5859 < x < 8.6807), 1.00)$$
$$\vee \ ((6.6157 < y < 8.5843), 1.00)$$
$$\vee \ ((8.6807 < x), 0.96)$$

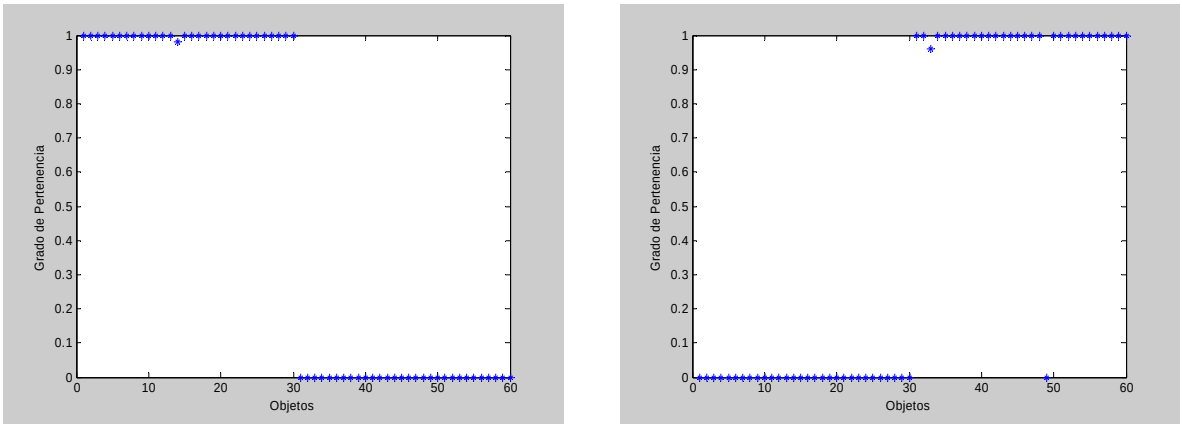
La calidad de estos conceptos es: 0.93.

En la Figura 34 se muestran los grados de pertenencia de cada objeto de la muestra de datos de la Figura 33 a) a cada uno de los agrupamientos, asignados por el algoritmo FKMSF; y en la Figura 35 se muestran los grados en que los objetos de la Figura 33 a) son cubiertos por el concepto.



a) **b)**

Figura 34. a) Grados de pertenencia de los objetos al agrupamiento A_1, **b)** Grados de pertenencia de los objetos al agrupamiento A_2.



a) **b)**

Figura 35. a) Grados en que los objetos son cubiertos por el concepto C_1, **b)** Grados en que los objetos son cubiertos por el concepto C_2.

En la Figura 35 b) observamos que el concepto del agrupamiento A_2 no cubre al objeto O_{49}, con coordenadas (6.191,6.412); por lo tanto le asigna pertenencia 0, aun cuando es un objeto que pertenece con alto grado al agrupamiento. Para el resto de los objetos los grados de pertenencia asignados por el concepto son similares a los grados de pertenencia de estos objetos al agrupamiento A_2 en el sentido de que a objetos con pertenencias altas, el concepto les asigna pertenencias altas; mientras que a objetos con pertenencias bajas se les asignan pertenencias bajas.

En la Figura 36 se muestran los grados de pertenencia asignados por el concepto.

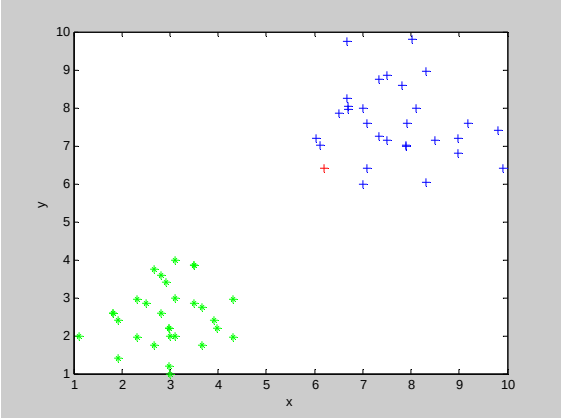
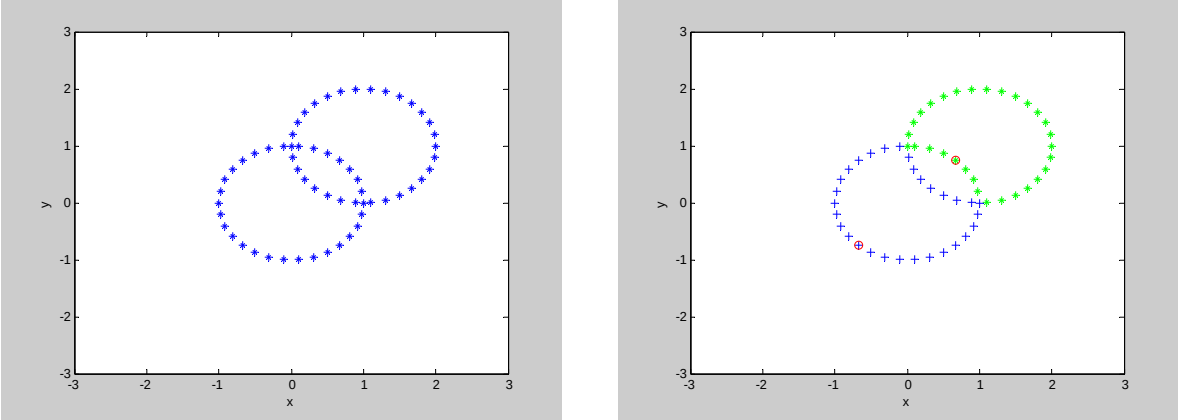


Figura 36. Grados en que los conceptos son cubiertos por el concepto.

En la Figura 36, el objeto en rojo (O_{49}) es cubierto por ambos conceptos con grado cero.

En este ejemplo, podemos observar que los grados en que los objetos son cubiertos por los conceptos (ver Figura 36) son similares a los grados de pertenencia de los objetos a los agrupamientos (Figura 33 b)), excepto para el objeto O_{49}.

***Ejemplo 4.4*:** En este ejemplo se muestra el comportamiento del algoritmo cuando los agrupamientos que se desean obtener se traslapan. Para este ejemplo se tomaron 60 objetos descritos por 4 atributos numéricos (x, y, x+y, x-y). Estos datos representan dos circunferencias de radio 1, una centrada en (0,0) y la otra en (1,1). Estas circunferencias se muestran en la Figura 37 a).



a) **b)**

Figura 37. a) Muestra de datos, **b)** Agrupamientos obtenidos en la fase de agrupamiento del algoritmo FKMSF.

En la Figura 37 b) se muestran los agrupamientos obtenidos después de aplicar la fase de agrupamiento del algoritmo FCKMSF. Para graficar los objetos se toman en cuenta sólo los dos primeros atributos, que son las coordenadas x y y. Los objetos en azul obtuvieron su mayor grado de pertenencia hacia el agrupamiento A_1 y los objetos en verde hacia el agrupamiento A_2. Los objetos encerrados en un círculo rojo son los centroides finales de los agrupamientos obtenidos por el algoritmo FKMSF.

Sobre los agrupamientos obtenidos en la fase de agrupamiento (ver Figura 37 b)) se aplicó la fase de caracterización y los conceptos obtenidos fueron los siguientes:

Para el agrupamiento A_1:

$\boldsymbol{C_1}$: $((x < -0.7308) \wedge (-0.7345 < y < 0.5668), 0.89)$
$\vee$ $((x + y < -0.9926), 0.89)$
$\vee$ $((-0.7308 < x < 0.6263) \wedge (y < -0.7345), 0.89)$
$\vee$ $((x < -0.7308), 0.89)$
$\vee$ $((y < -0.7345), 0.89)$
$\vee$ $((-0.9926 < x + y < 0.7206), 0.85)$
$\vee$ $((0.6263 < x) \wedge (-0.7345 < y < 0.5668) \wedge (0.7206 < x + y) \wedge (-0.9849 < x - y < 1.0481), 0.68)$
$\vee$ $((0.6263 < x) \wedge (-0.7345 < y < 0.5668) \wedge (0.7206 < x + y) \wedge (1.0481 < x - y), 0.61)$
$\vee$ $((-0.7308 < x < 0.6263) \wedge (0.5668 < y) \wedge (0.7206 < x + y) \wedge (-0.9849 < x - y < 1.0481), 0.59)$
$\vee$ $((-0.7308 < x < 0.6263) \wedge (0.5668 < y) \wedge (0.7206 < x + y) \wedge (x - y < -0.9849), 0.53)$

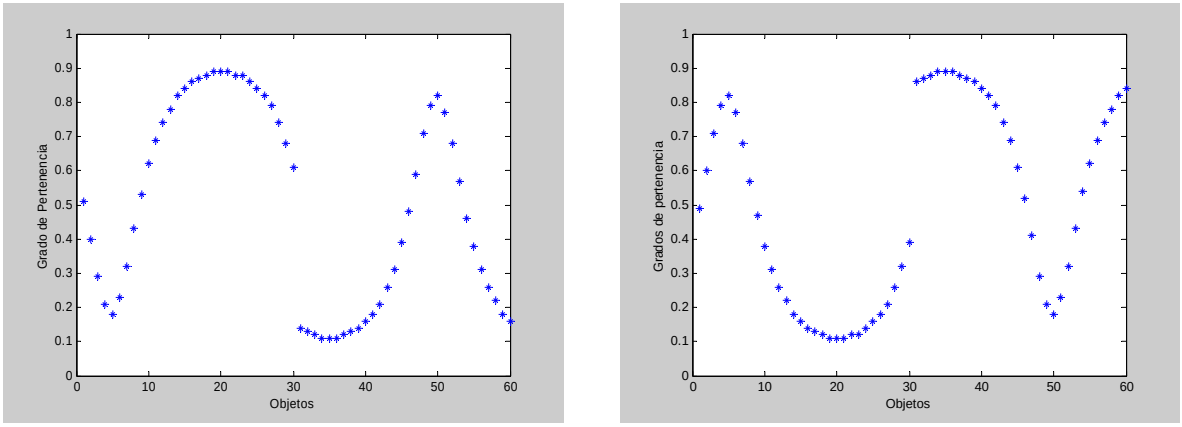
Para el agrupamiento A_2:

$\boldsymbol{C_2}$: $((1.7308 < x) \wedge (0.4332 < y < 1.7345) \wedge (2.9926 < x + y), 0.89)$
$\vee$ $((2.9926 < x + y), 0.89)$
$\vee$ $((1.7308 < x) \wedge (0.4332 < y < 1.7345), 0.89)$
$\vee$ $((0.3737 < x < 1.7308) \wedge (1.7345 < y), 0.89)$
$\vee$ $((1.7308 < x), 0.89)$
$\vee$ $((1.7345 < y), 0.89)$
$\vee$ $((1.2794 < x + y < 2.9926), 0.85)$
$\vee$ $((x < 0.3737) \wedge (0.4332 < y < 1.7345) \wedge (x + y < 1.2794) \wedge (-1.0481 < x - y < 0.9849), 0.68)$
$\vee$ $((x < 0.3737) \wedge (0.4332 < y < 1.7345) \wedge (x + y < 1.2794) \wedge (x - y < -1.0481), 0.61)$
$\vee$ $((0.3737 < x < 1.7308) \wedge (y < 0.4332) \wedge (x + y < 1.2794) \wedge (-1.0481 < x - y < 0.9849), 0.60)$
$\vee$ $((0.3737 < x < 1.7308) \wedge (y < 0.4332) \wedge (x + y < 1.2794) \wedge (0.9849 < x - y), 0.54)$

La calidad de estos conceptos es: 0.77.

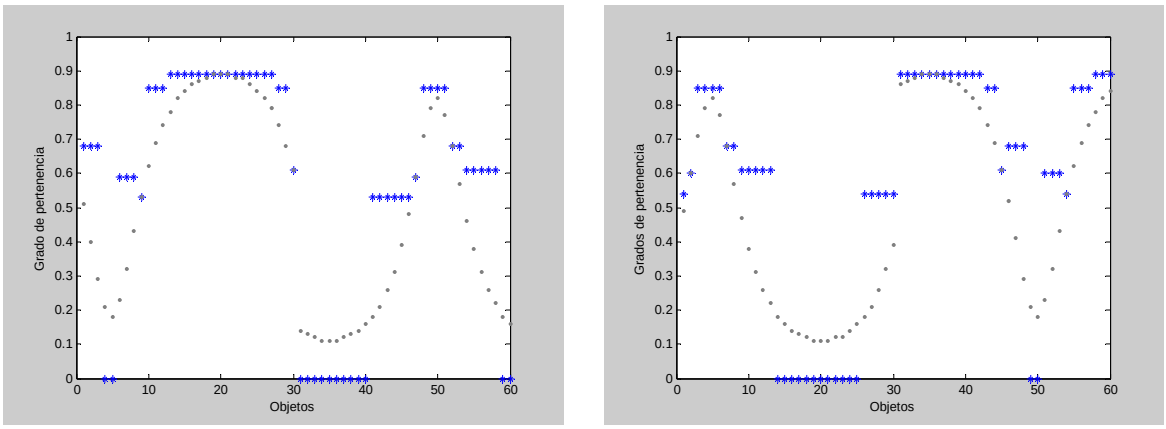
En la Figura 38 se muestran los grados de pertenencia de los objetos de la muestra de datos de la Figura 37 a) a cada uno de los agrupamientos, asignados por el algoritmo

FKMSF; y en la Figura 39 se sobreponen los grados en que los objetos de la Figura 37 a) son cubiertos por los conceptos.



a) **b)**

Figura 38. a) Grados de pertenencia de los objetos al agrupamiento A_1, **b)** Grados de pertenencia de los objetos al agrupamiento A_2.



a) **b)**

Figura 39. a) Grados en que los objetos son cubiertos por el concepto C_1, **b)** Grados en que los objetos son cubiertos por el concepto C_2.

En la Figura 39, podemos observar que cuando los agrupamientos se traslapan, la diferencia entre los grados en que los objetos son cubiertos por el concepto y los grados de pertenencia de los objetos a los agrupamientos crece.

Los grados en que los objetos de la Figura 37 a) son cubiertos por los conceptos se muestran en la Figura 40.

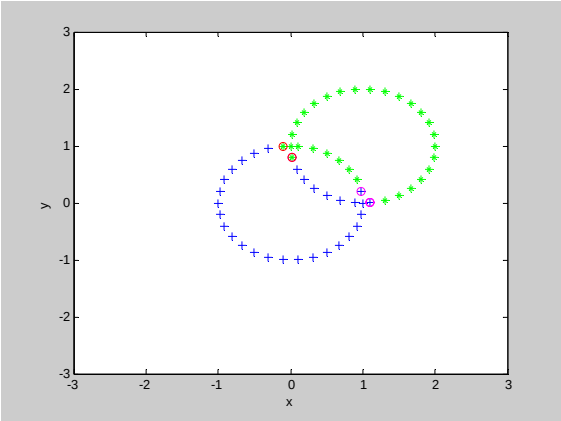


Figura 40. Grados en que los objetos son cubiertos por los conceptos.

En la gráfica de la Figura 40, los objetos encerrados en un círculo rojo son objetos que pertenecen con alto grado al agrupamiento A_1 (ver Figura 37 b)) pero que son cubiertos con alto grado por el concepto del agrupamiento A_2 (ver Figura 40) y los objetos encerrados en un círculo magenta son objetos que pertenecen con alto grado al agrupamiento A_2 pero que son cubiertos con alto grado por el concepto del agrupamiento A_1.

Los ejemplos anteriores nos muestran el comportamiento del algoritmo FCKMSF cuando los agrupamientos que se desean obtener están bien definidos y cuando se traslapan. En estos ejemplos podemos observar que cuando los agrupamientos están bien definidos los grados en que los objetos son cubiertos por los conceptos son más similares a los grados de pertenencia de los objetos a los agrupamientos, que en el caso en que los agrupamientos se traslapan.

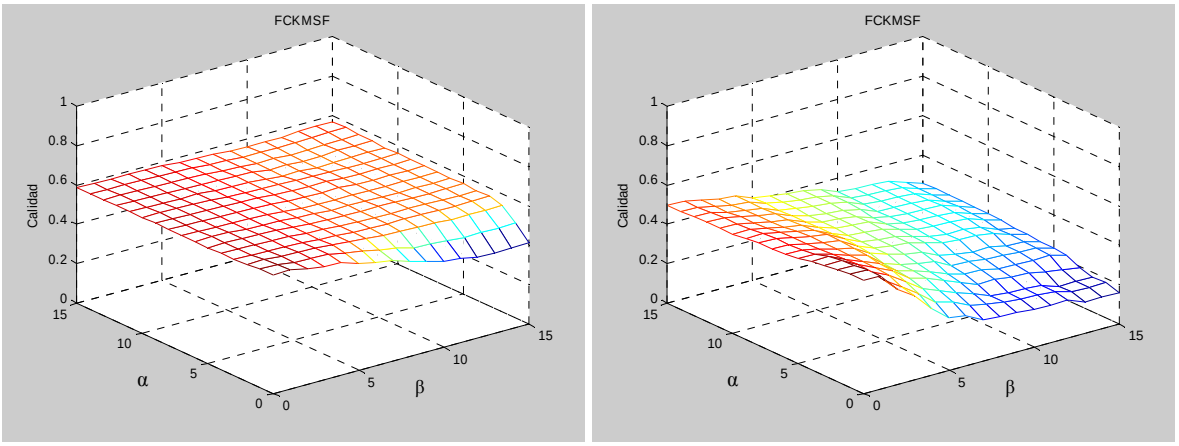
A continuación se muestran los resultados obtenidos al aplicar el algoritmo FCKMSF sobre bases de datos tomadas del repositorio de la UCI (Blake et al., 1998), estas bases de datos se muestran en la Tabla 25.

Base de Datos	No. de agrupamientos	No. de objetos	No. de atributos	Tipo de atributos	Ausencia de infornación?
Diabetes	2	768	8	Cuantitativas	no
Glass	6	214	9	Cuantitativas	no
Iris	3	150	4	Cuantitativas	no
Wine	3	178	13	Cuantitativas	no
Hayes	3	132	4	Cualitativas	no
Lenses	3	24	4	Cualitativas	no
Zoo	7	101	16	Cualitativas	no
Auto-mpg	3	398	7	Mezcladas	si
Bridges	7	108	11	Mezcladas	si
Echocardiogram	3	132	11	Mezcladas	si
Hepatitis	2	155	19	Mezcladas	si
Import85	6	205	25	Mezcladas	si
Tae	3	151	5	Mezcladas	no

Tabla 25. Bases de datos utilizadas para la experimentación.

Para realizar las pruebas sobre las bases de datos que presentan ausencia de información fue necesario completar los datos faltantes; ya que para trabajar con los retículos de generalización las descripciones de los objetos deben estar completas. Estos valores fueron sustituidos por la media, cuando el atributo es cuantitativo y por la moda, cuando el atributo es cualitativo.

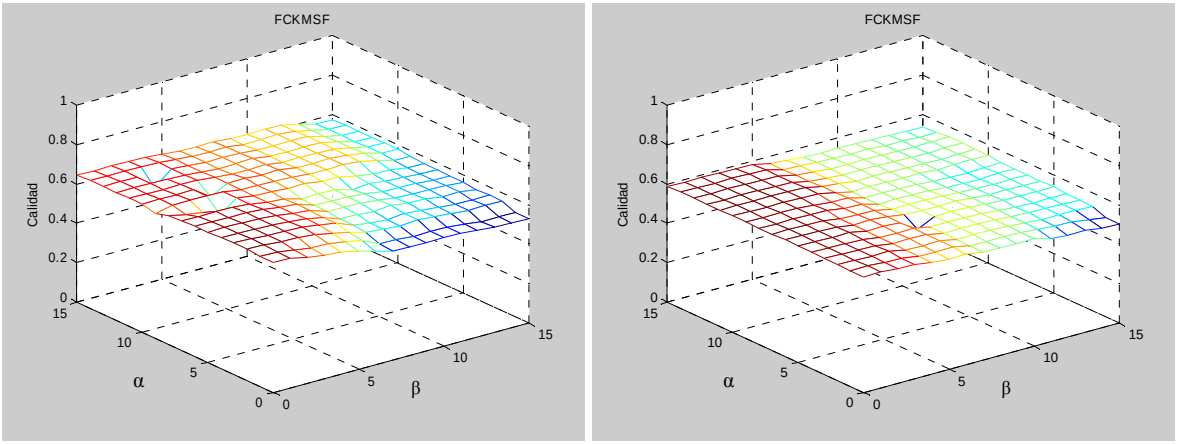
Al igual que en el caso duro, para las bases de datos en las que se observa ausencia de información, se realizaron pruebas completando los datos faltantes antes de realizar la fase de agrupamiento y completando los datos después de agrupar los objetos. Los resultados obtenidos se muestran en las Figuras 41-45.



a) **b)**

Figura 41. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Auto-mpg, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

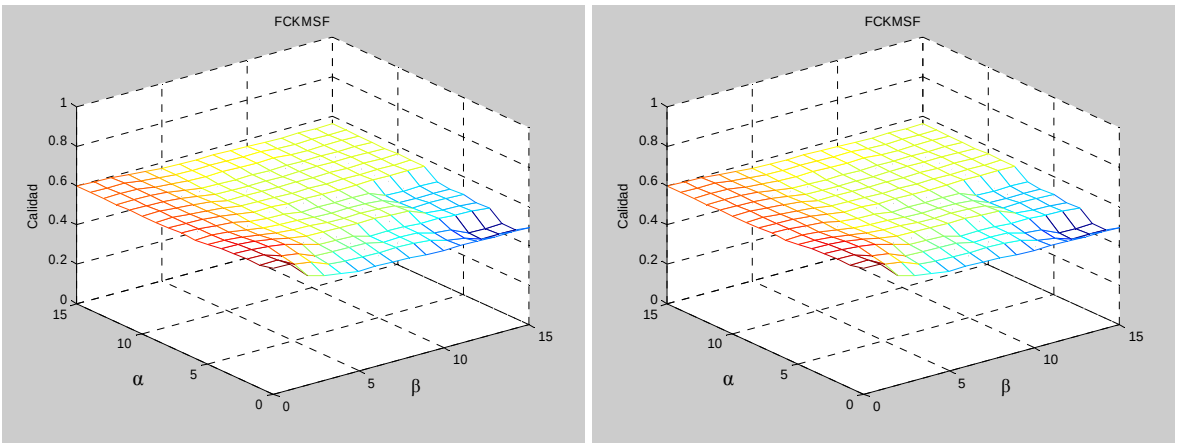
En la Figura 41 observamos que, usando la base de datos Auto-mpg, cuando se completa la información antes de agrupar (Figura 41 a)) se obtienen buenas calidades para cualesquiera valores de α y β, mientras que cuando se completa después de agrupar (Figura 41 b)) se obtienen conceptos con menor calidad.



a) **b)**

Figura 42. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Bridges, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

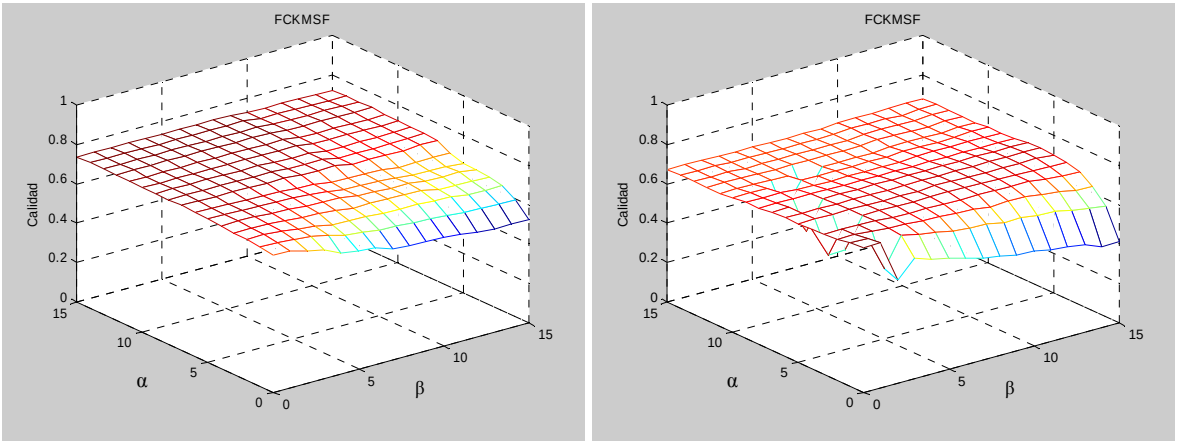
En la Figura 42 se observa que, para la base de datos Bridges, se obtienen calidades ligeramente mejores cuando se completa la información antes de agrupar (Figura 42 a)) que cuando se completa la información después de agrupar (Figura 42 b)).



a) **b)**

Figura 43. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Echocardiogram, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

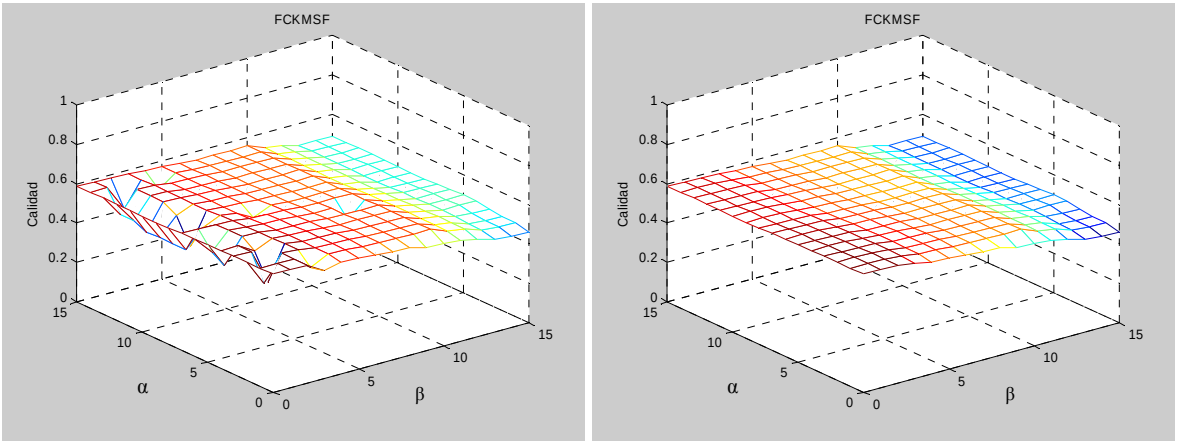
En la Figura 43 observamos que las calidades obtenidas para la base de datos Echocardiogram cuando se completa la información antes de agrupar y cuando se completa la información después de agrupar son similares.



a) **b)**

Figura 44. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Hepatitis, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

En la Figura 44 se observa que, para la base de datos Hepatitis, se obtienen conceptos con mejor calidad cuando se completa la información antes de agrupar que cuando se completa la información después de agrupar. Además, cuando se completa la información después de agrupar, para algunos valores de α y β, se obtiene conceptos con menor calidad, esto se debe a que los conceptos obtenidos cubren menor número de objetos de los agrupamientos y cubren mayor número de objetos fuera de los agrupamientos.



a) **b)**

Figura 45. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Import85, para valores de α y β entre 0 y 15; a) completando la información antes de agrupar y b) completando la información después de agrupar.

En la Figura 45 observamos que las calidades obtenidas para la base de datos Import85 cuando se completa la información antes de agrupar y cuando se completa la información después de agrupar son similares. sin embargo, cuando se completa la información antes de agrupar, para algunos valores de α y β se obtienen conceptos con menor calidad.

En las gráficas de las Figuras 41-45 podemos observar que se obtienen mejores resultados, en la mayoría de los casos, cuando se completa la información antes de aplicar la fase de agrupamiento que cuando se completa la información después de agrupar. Sin embargo, para la base de datos Import85 se obtienen mejores resultados cuando se completa la información después de agrupar.

En la Tabla 26 se muestran las mejores calidades obtenidas (variando α y β) por el algoritmo FCKMSF después de aplicarlo sobre las distintas bases de datos completando la información antes de agrupar y completando la información después de agrupar, así como el número de predicados que forman los conceptos obtenidos por el algoritmo FCKMSF.

En la Tabla 26 podemos ver que se obtienen mejores resultados cuando se completa la información antes de agrupar que cuando se completa la información después de la fase de agrupamiento. Por otra parte, en la mayoría de los casos el número de predicados obtenidos por el algoritmo FCKMSF completando la información después de agrupar es mayor o igual que cuando se completa la información antes de agrupar; sólo para la base de datos Bridges se obtuvo menor número de predicados cuando se completa la información antes de agrupar que completando la información después de agrupar.

Base de Datos	Algoritmo FCKMSF			
	Completando la información antes de agrupar		Completando la información después de agrupar	
	No. de predicados	Calidad	No. de predicados	Calidad
Auto-mpg	96	0.60	190	0.57
Bridges	92	0.67	59	0.59
Echocardiogram	84	0.64	84	0.64
Hepatitis	43	0.74	133	0.73
Import85	112	0.60	112	0.60
Promedio	**83**	**0.68**	**116**	**0.63**

Tabla 26. Resultados obtenidos por el algoritmo FCKMSF antes y después de completar la información.

En las Figuras 46 y 47 se muestran de manera gráfica los resultados de la Tabla 26. En la Figura 46 se grafican las mejores calidades obtenidas (variando α y β) por el algoritmo FCKMSF cuando se completa la información antes y después de agrupar. En la Figura 47 se grafica el número de predicados obtenidos por el algoritmo FCKMSF completando la información antes y después de agrupar.

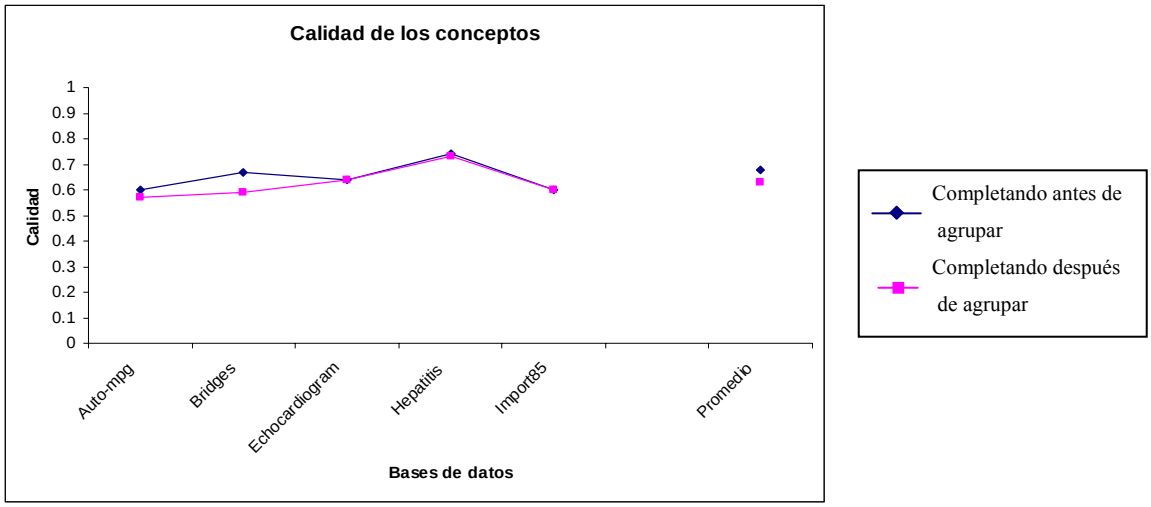


Figura 46. Resultados obtenidos por el algoritmo FCKMSF tomando la mejor calidad obtenida para cada base de datos completando la información antes y después de agrupar.

En la gráfica de la Figura 46 observamos que, en promedio, se obtienen mejores resultados cuando se completa la información antes de agrupar que completando la información después de agrupar.

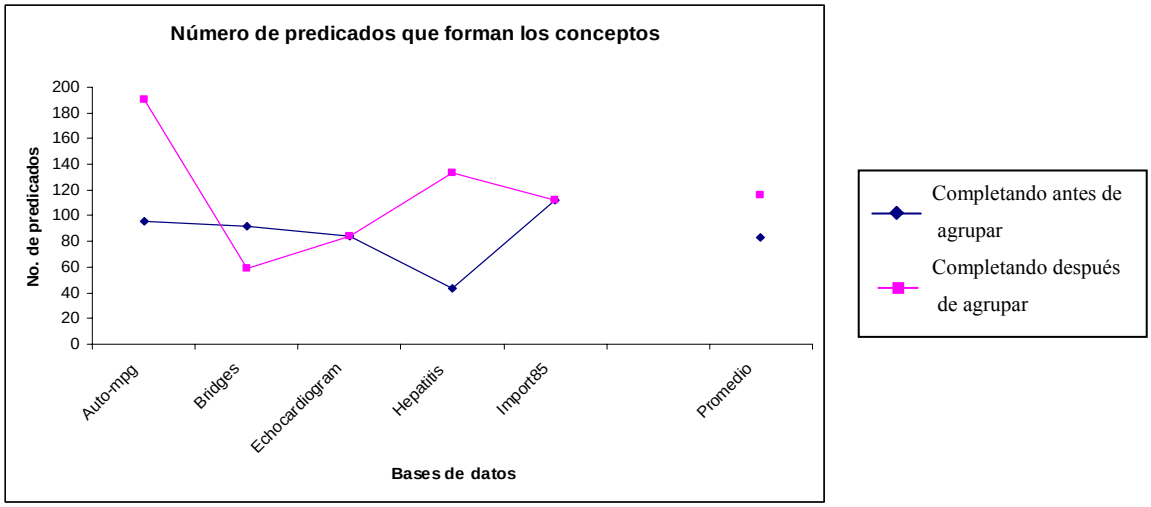


Figura 47. Número de predicados obtenidos por el algoritmo FCKMSF tomando la mejor calidad obtenida para cada base de datos completando la información antes y después de agrupar.

En la gráfica de la Figura 47 podemos ver que, en promedio, se obtiene menor número de predicados cuando se completa la información antes de agrupar que completando la información después de agrupar.

Posteriormente, se realizaron pruebas sobre todas las bases de datos de la Tabla 25. Las pruebas se realizaron usando el retículo de generalización propuesto (ver Figura 2) y para valores de α y β entre 0 y 15, con ε=0.2. Los resultados obtenidos se muestran en la Figuras 48-60.

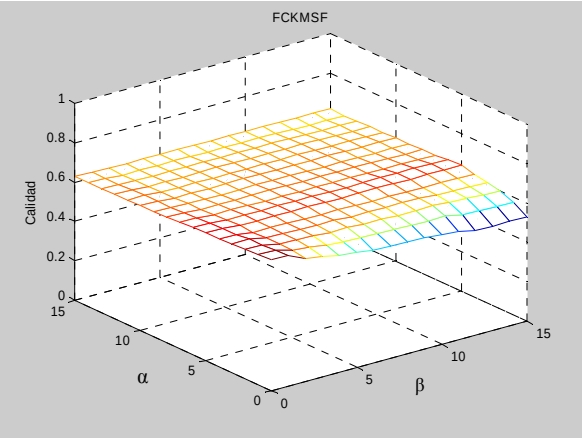


Figura 48. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Diabetes, para valores de α y β entre 0 y 15.

En la Figura 48 observamos que la calidad de los conceptos obtenidos usando la base de datos Diabetes no depende tanto de los valores de α y β, ya que para cualquier valor de estos parámetros se obtienen conceptos con calidad similar.

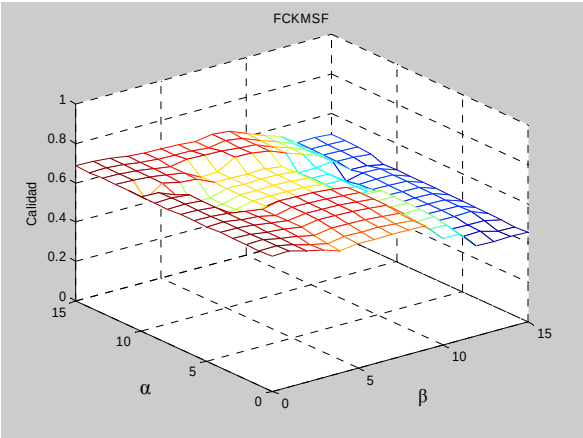


Figura 49. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Glass, para valores de α y β entre 0 y 15.

En la Figura 49 se observa que las mejores calidades se obtienen para cualquier valor de α cuando β toma valores entre 0 y 2; sin embargo, para algunos valores la calidad es ligeramente menor. Esto se debe a que para esos valores de α y β, los conceptos cubren mayor número de objetos que estan fuera del agrupamiento y menor número de objetos del agrupamiento.

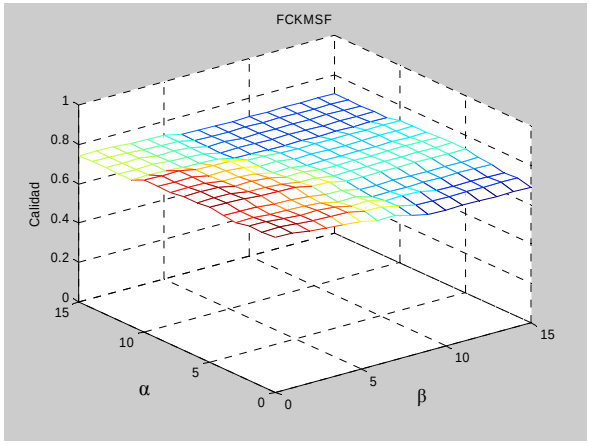


Figura 50. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Iris, para valores de α y β entre 0 y 15.

En la Figura 50 observamos que, al igual que para la base de datos Diabetes, la calidad de los conceptos no depende tanto de los valores de α y β, ya que para cualquier valor de estos parámetros se obtienen conceptos con calidad similar.

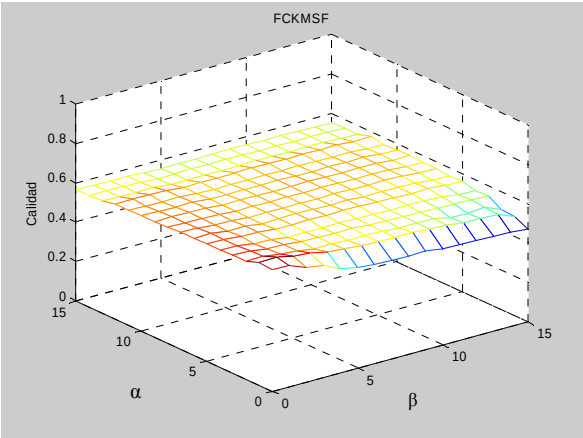


Figura 51. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Wine, para valores de α y β entre 0 y 15.

En la Figura 51 se obseva que el comportamiento del algoritmo FCKMSF aplicado sobre la base de datos Wine es similar al obtenido usando las bases de datos Diabetes e Iris.

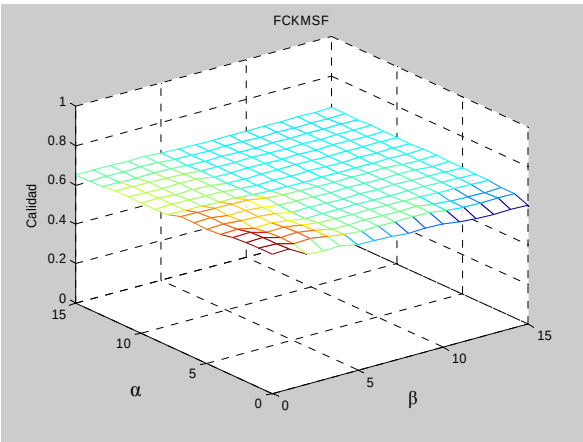


Figura 52. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Hayes, para valores de α y β entre 0 y 15.

En la Figura 52 podemos ver que el comportamiento del algoritmo FCKMSF aplicado sobre la base de datos Hayes es similar al obtenido usando las bases de datos Diabetes, Iris y Wine.

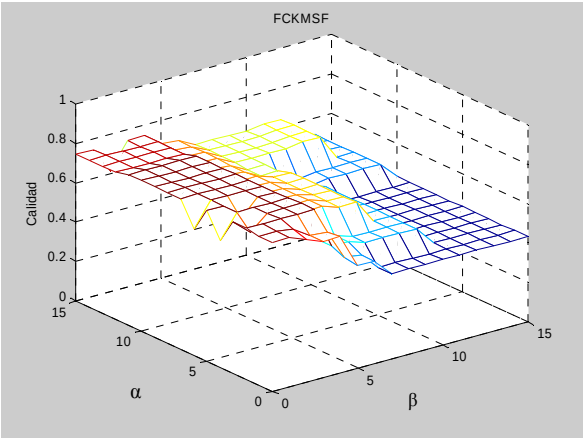


Figura 53. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Lenses, para valores de α y β entre 0 y 15.

En la Figura 53 observamos que, al igual que para la base de datos Glass, para algunos valores de α y β cercanos a 0, se obtienen calidades ligeramente menores.

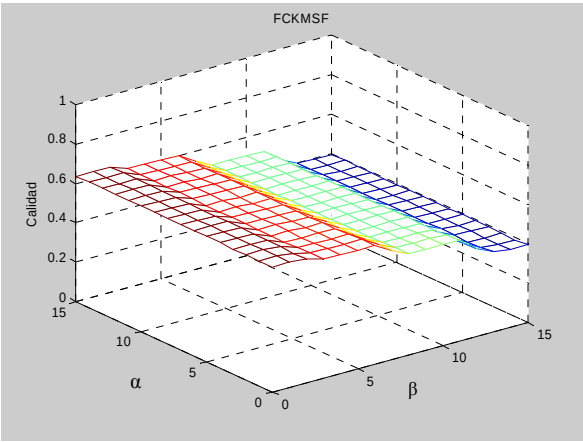


Figura 54. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Zoo, para valores de α y β entre 0 y 15.

En la Figura 54 se observa que las calidades obtenidas dependen únicamente del parámetro β; conforme crece el valor de β la calidad decrece ya que los conceptos cubren menor número de objetos del agrupamiento.

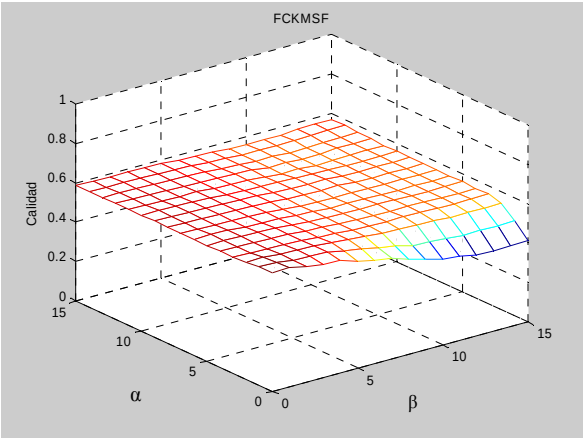


Figura 55. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Auto-mpg, para valores de α y β entre 0 y 15.

En la Figura 55 podemos ver que el comportamiento del algoritmo FCKMSF aplicado sobre la base de datos Auto-mpg es similar al obtenido usando las bases de datos Diabetes, Iris, Wine y Hayes.

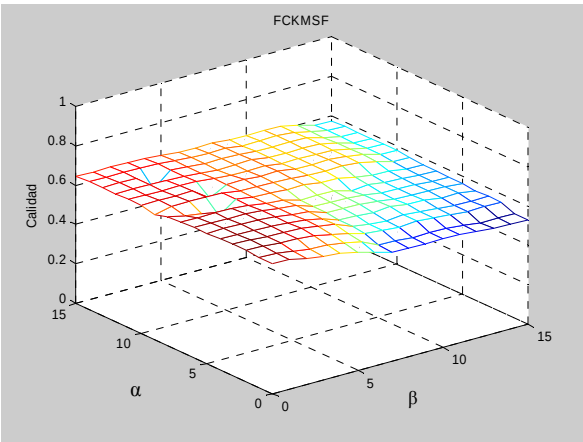


Figura 56. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Bridges, para valores de α y β entre 0 y 15.

En la Figura 56 observamos que, al igual que para las bases de datos Glass y Lenses, para algunos valores de α y β, se obtienen calidades ligeramente menores aún para valores cercanos a 0.

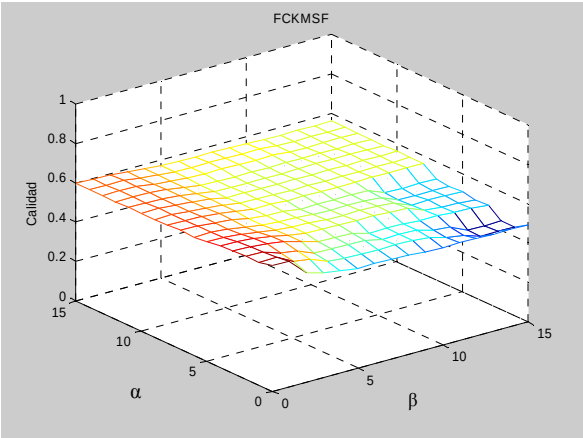


Figura 57. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Echocardiogram, para valores de α y β entre 0 y 15.

En la Figura 57 se observa que para la base de datos Echocardiogram, al igual que en la mayoria de las bases de datos, la calidad de los conceptos no depende tanto de los parámetros α y β.

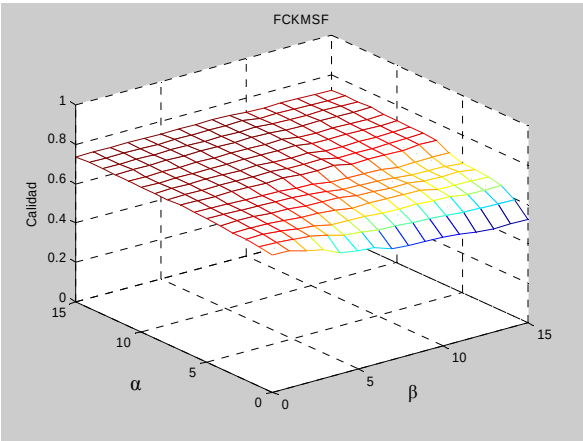


Figura 58. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Hepatitis, para valores de α y β entre 0 y 15.

En la Figura 58 podemos ver que el comportamiento del algoritmo FCKMSF aplicado sobre la base de datos Hepatitis es similar al obtenido para la mayoria de las bases de datos.

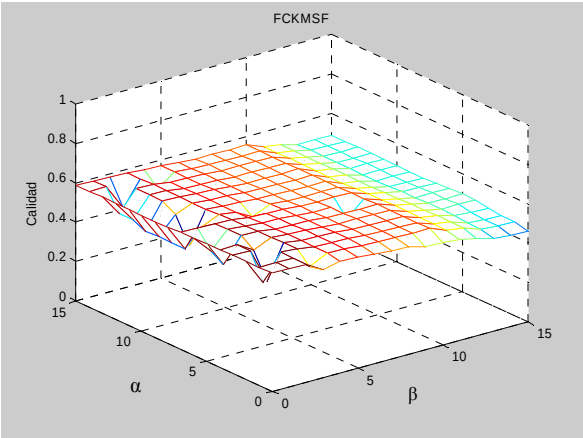


Figura 59. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Import85, para valores de α y β entre 0 y 15.

En la Figura 59 observamos que, al igual que para las bases de datos Glass, Lenses y Bridges, para algunos valores de α y β cercanos a 0, se obtienen calidades ligeramente menores.

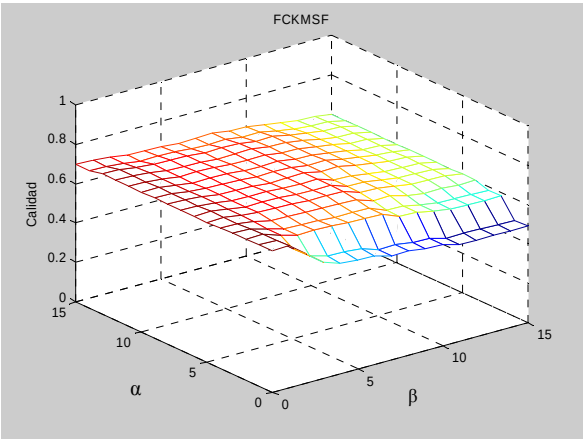


Figura 60. Resultados obtenidos por el algoritmo FCKMSF aplicado sobre la base de datos Tae, para valores de α y β entre 0 y 15.

En la Figura 60 se observa que para la base de datos Tae la calidad de los conceptos no depende tanto de los parámetros α y β, como ocurre en la mayoria de las bases de datos.

En las gráficas de las Figuras 48-60 se puede observar que la calidad de los conceptos depende poco de los valores de α y β. Sin embargo, los conceptos con mejor calidad se obtienen cuando los valores de α y β son pequeños (cercanos a 0). Esto se debe a que

conforme el valor de α crece, se permite que los predicados reconozcan mayor cantidad de objetos fuera del agrupamiento y por lo tanto habrá más predicados que cumplan la condición α-discriminante. Por otro lado, conforme crece el valor de β, habrá menos predicados que reconozcan al menos β objetos del agrupamiento y por lo tanto también habrá menos predicados que cumplan la condición β-caracterizante. Como consecuencia, los conceptos cubrirán mayor número de objetos fuera del agrupamiento y menor número de objetos dentro del agrupamiento y por lo tanto la calidad de estos conceptos será menor.

En la Tabla 27 se muestran las mejores calidades obtenidas por el algoritmo FCKMSF después de aplicarlo sobre las distintas bases de datos, así como el número de predicados que forman los conceptos obtenidos por el algoritmo FCKMSF.

Algoritmo FCKMSF		
Base de Datos	**No. de predicados**	**Calidad**
Diabetes	230	0.66
Glass	69	0.69
Iris	37	0.78
Wine	99	0.62
Hayes	66	0.71
Lenses	11	0.77
Zoo	41	0.64
Auto-mpg	96	0.60
Bridges	92	0.67
Echocardiogram	84	0.64
Hepatitis	43	0.74
Import85	112	0.60
Tae	96	0.72
Promedio	**83**	**0.68**

Tabla 27. Resultados obtenidos por el algoritmo FCKMSF utilizando el retículo nuevo para las bases de datos cuantitativas y mezcladas que contienen información numérica.

En la Tabla 27 podemos ver que la mejor calidad se obtiene con la base de datos Iris (0.78) y la calidad más baja se obtiene con las bases de datos Auto-mpg e Import85 (0.60). Por otro lado, para la base de datos Diabetes el algoritmo FCKMSF genera conceptos con un gran número de predicados.

En las Figuras 61 y 62 se muestran de manera gráfica los resultados de la Tabla 27. En la Figura 61 se muestra la mejor calidad (cuando se varían α y β) obtenida por el algoritmo k-means conceptual difuso con funciones de similaridad para cada una de las bases de datos. En la Figura 62 se muestra el número de predicados obtenidos por el algoritmo FCKMSF para los conceptos de mejor calidad.

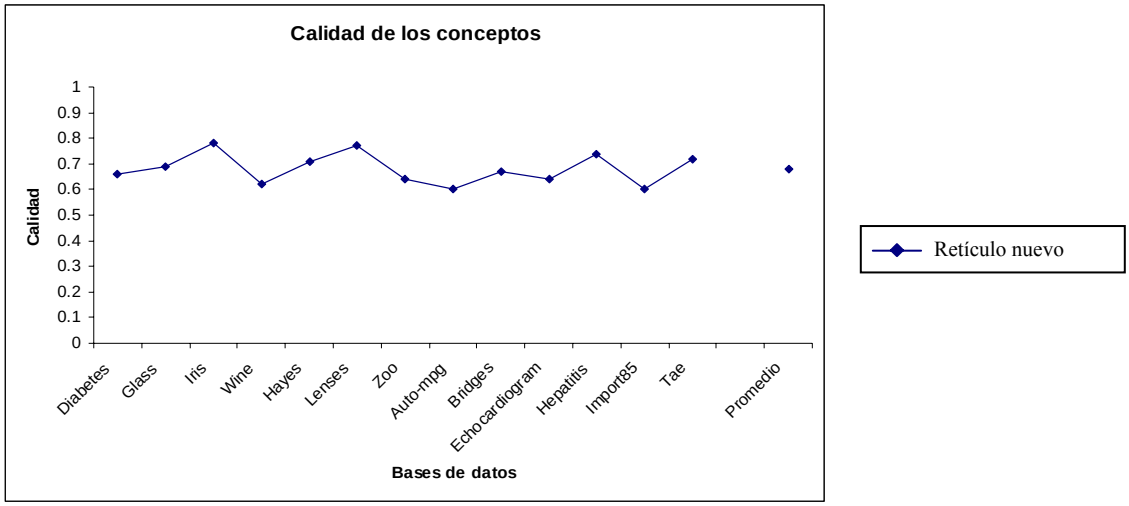


Figura 61. Calidades de los conceptos obtenidos por el algoritmo FCKMSF tomando la mejor calidad obtenida para cada base de datos.

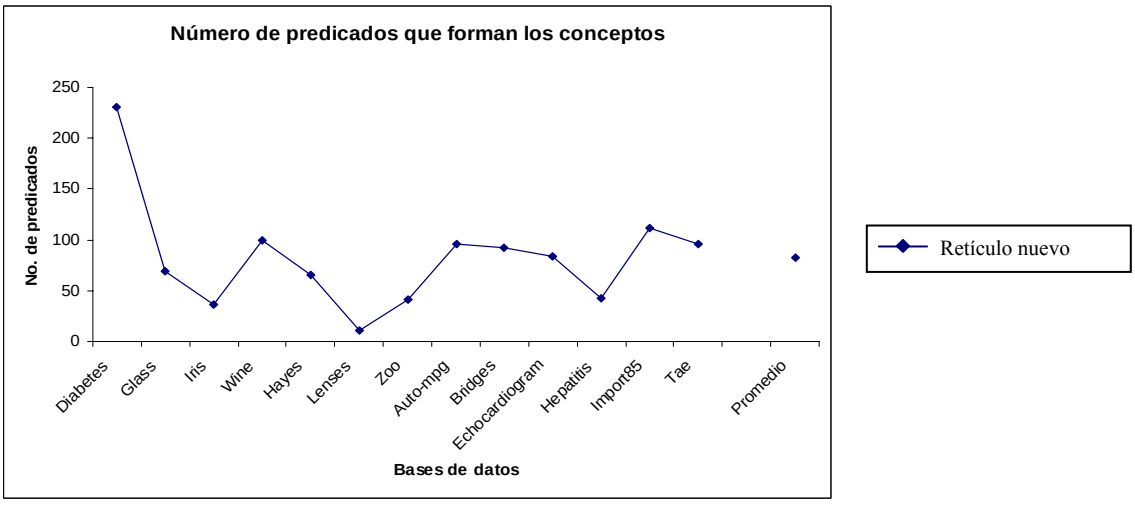


Figura 62. Número de predicados que forman los conceptos obtenidos por el algoritmo FCKMSF tomando la mejor calidad obtenida para cada base de datos.

4.5.5. Discusión

El algoritmo k-means conceptual difuso con funciones de similaridad (FCKMSF) permite resolver problemas de agrupamiento conceptual difuso cuando se conoce *a priori* el número de agrupamientos. Este algoritmo consta de dos fases: una fase de agrupamiento y una fase de caracterización. En la fase de agrupamiento utiliza el algoritmo FKMSF, el cual permite construir agrupamientos difusos con descripciones de objetos dadas en términos de atributos cualitativos y cuantitativos mezclados. La fase de caracterización es una generalización de la fase de caracterización del algoritmo CKMSF propuesto en la Sección 3.3. Con esta modificación se pueden construir conceptos difusos a partir de los agrupamientos difusos obtenidos en la fase de agrupamiento.

Con base en los resultados experimentales observamos que los conceptos obtenidos tienen la característica de que objetos que pertenecen con alto grado hacia un agrupamiento, son cubiertos con alto grado por el concepto; mientras que objetos que pertenecen con bajo grado, son cubiertos con grado bajo por el concepto.

Por otra parte, para las bases de datos en que se observa ausencia de información se obtienen mejores resultados cuando se completa la información antes de agrupar que completando la información después de agrupar.

Al igual que en el caso duro, un inconveniente de este algoritmo es el uso de retículos de generalización. Por esta razón, se propone un nuevo algoritmo conceptual restringido difuso que no depende de retículos de generalización para la generación de los conceptos, el cual se presenta en la siguiente sección.

4.6. Algoritmo K-means Conceptual Difuso con Rasgos Complejos

En esta sección se propone una versión difusa del algoritmo k-means conceptual con rasgos complejos (CKMCF). Este algoritmo, al igual que el algoritmo k-means conceptual con rasgos complejos (CKMCF), consta de una fase de agrupamiento y una fase de caracterización.

En la fase de agrupamiento se utiliza el algoritmo k-means difuso con funciones de similaridad. En la fase de caracterización, se utilizan los rasgos complejos difusos para la construcción de los conceptos que caracterizan a los agrupamientos difusos.

4.6.1. Fase de Agrupamiento

En esta fase, al igual que en el algoritmo FCKMSF, se utiliza el algoritmo k-means difuso con funciones de similaridad para formar los agrupamientos difusos (ver Sección 4.5.1).

4.6.2. Fase de Caracterización

En esta fase, se usan los rasgos complejos difusos para generar los conceptos. Para calcular los rasgos complejos difusos es necesario definir conjuntos de apoyo.

Existe una relación directa entre la semejanza de los grados de pertenencia y la semejanza de los objetos, es decir, intuitivamente si dos objetos tienen grados de pertenencia parecidos, esto debe corresponder al hecho de que ellos fuesen parecidos o viceversa. El objetivo es tener conjuntos de apoyo que conserven o mejoren la relación *semejanza de objetos - semejanza entre grados de pertenencia* que existe en el conjunto *R*. Esta idea la podemos considerar como una generalización del hecho de que la semejanza entre objetos de un agrupamiento debe ser mayor que la semejanza entre objetos de agrupamientos diferentes para el caso duro. Un tipo de conjuntos de apoyo que cuenta con esta característica son los Φ-testores difusos (Alba-Cabrera, 1997).

Por esta razón, en esta tesis se utilizan, además de los conjuntos Γ-diferenciantes, Γ-caracterizantes y Γ-testores, los Φ-testores difusos como conjuntos de apoyo.

Para obtener los conjuntos de apoyo aplicamos un algoritmo genético (Santos-Gordillo, 2003; Santos-Gordillo et al., 2003) donde cada individuo representa un subconjunto de atributos (Ω) formado por *m* genes y cada gen representa un atributo. Un gen vale 1 si el atributo es tomado en cuenta y 0 si no es tomado en cuenta. Este algoritmo utiliza el

operador de cruza en un punto, es decir, se selecciona un punto de cruza y a partir de ahí se intercambia la información de los individuos a cruzar, para aplicar el operador de cruza se seleccionan el individuo más apto y el individuo menos apto; y la mutación uniforme, la cual consiste en seleccionar aleatoriamente un gen de un individuo y cambiar su valor (0 por 1 o 1 por 0). Como función de aptitud, se usa el grado en que un subconjunto Ω satisface la definición de Φ-testor difuso. La población para la siguiente generación se obtiene seleccionando los individuos con mejor aptitud de entre los obtenidos por medio de la cruza y la mutación, unidos con la población original. Los conjuntos de apoyo serán los mejores individuos de la última generación.

Ejemplo 4.5: Supongamos que después de aplicar la fase de agrupamiento para los objetos de la Tabla 28 se obtienen los agrupamientos difusos mostrados en las columnas A_1 y A_2 de la Tabla 28.

Objetos	Atributos				Agrupamientos Difusos	
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)	A_1	A_2
O_1	rojo	chico	20	redondo	1.00	0.00
O_2	rojo	mediano	20	redondo	0.94	0.06
O_3	azul	chico	25	redondo	0.90	0.10
O_4	azul	mediano	25	cuadrado	0.50	0.50
O_5	verde	grande	30	triangular	0.06	0.94
O_6	verde	chico	20	redondo	0.94	0.06
O_7	amarillo	grande	30	triangular	0.00	1.00
O_8	amarillo	mediano	35	triangular	0.06	0.94
O_9	verde	grande	35	redondo	0.31	0.69

Tabla 28. Muestra con 9 objetos descritos por 4 atributos.

Para estos agrupamientos el algoritmo genético obtiene los conjuntos de apoyo mostrados en la Tabla 29.

Φ-testores difusos
{C,T,P,F}
{F}
{P,F}
{T,F}
{P}
{T,P}
{C,P}
{C}
{T}
{T,P,F}
{C,F}
{C,P,F}
{C,T}

Tabla 29. Conjuntos de apoyo obtenidos por el algoritmo genético para la muestra de la Tabla 28.

Una vez que se obtuvieron los conjuntos de apoyo, se calculan los rasgos complejos difusos aplicando la siguiente definición.

Definición 4.1*:** Sea $\Omega = \{x_{s_1},...,x_{s_p}\}$ un conjunto de apoyo y $(a_1,...,a_p)$ valores asociados a los atributos $x_{s_1},...,x_{s_p}$ tomados de un objeto de la muestra, entonces $\{x_{s_1},...,x_{s_p}\}$ y $(a_1,...,a_p)$ forman un ***rasgo complejo difuso del agrupamiento A_i (De-la-Vega-Doria, 1994; De-la-Vega-Doria, 1998), si y sólo si:

1) $\sum_{O_j \in X} [\Gamma(\Omega O_j, (a_1,...,a_p))\mu_i(O_j)] \geq \beta_i$

2) $\sum_{O_j \in X} [\Gamma(\Omega O_j, (a_1,...,a_p))(1-\mu_i(O_j))] < \lambda_i$

donde $\mu_i(O_j)$ es el grado de pertenencia del objeto O_j al agrupamiento A_i ; β_i y λ_i se definen y calculan de la misma manera que en el caso duro (ver Sección 3.4.2 del Capítulo 3).

En esta fase, la función de similaridad que se utiliza para calcular los rasgos complejos es la misma que se utilizó en la fase de agrupamiento. Una ventaja de utilizar la misma

función de similaridad en ambas fases del algoritmo es que, la forma en que se generan los conceptos mantiene una estrecha relación con la construcción de los agrupamientos.

Los rasgos complejos difusos obtenidos con los conjuntos de apoyo de la Tabla 29, para el ejemplo 4.5, se muestran en la Tabla 30.

Agrupamiento	Rasgos Complejos con Φ-testores
1	{F} = (redondo)
	{P,F} = (20,redondo)
	{P,F} = (25,redondo)
	{P} = (20)
	{T,P} = (mediano,25)
	{C,P} = (azul,25)
	{C} = (azul)
2	{P} = (30)
	{P} = (35)
	{F} = (triangular)
	{P,F} = (30,triangular)
	{P,F} = (35,triangular)
	{T,P} = (grande,30)
	{T,P} = (grande,35)
	{T} = (grande)

Tabla 30. Rasgos complejos difusos para el ejemplo.

Una vez que se obtuvieron los rasgos complejos difusos, para obtener los conceptos, a cada rasgo complejo se asocia un predicado difuso (P,μ_P), el cual se construye de la siguiente manera: a cada atributo $x_s \in R$ que aparece en el rasgo complejo se le asigna el valor a_s asociado a ese atributo en el rasgo complejo y para los atributos $x_s \in R$ que no aparecen en el rasgo complejo se le asigna *. El símbolo * significa "cualquier valor es posible". El grado μ_P asociado a este predicado será el promedio de los grados de

pertenencia de los objetos que son cubiertos por dicho predicado, para obtener conceptos que cubran con alto grado (cercano a 1) a los objetos que pertenecen con mayor grado al agrupamiento.

Para el ejemplo 4.5, los predicados formados a partir de los rasgos complejos difusos con Φ-testores son:

Para A_1:

P_1: $((C,*)\wedge(T,*)\wedge(P,*)\wedge(F,redondo),0.82)$

P_2: $((C,*)\wedge(T,*)\wedge(P,20)\wedge(F,redondo),0.95)$

P_3: $((C,*)\wedge(T,*)\wedge(P,25)\wedge(F,redondo),0.95)$

P_4: $((C,*)\wedge(T,*)\wedge(P,20)\wedge(F,*),0.86)$

P_5: $((C,*)\wedge(T,mediano)\wedge(P,25)\wedge(F,*),0.70)$

P_6: $((C,azul)\wedge(T,*)\wedge(P,25)\wedge(F,*),0.70)$

P_7: $((C,azul)\wedge(T,*)\wedge(P,*)\wedge(F,*),0.70)$

Para A_2:

P_1: $((C,*)\wedge(T,*)\wedge(P,30)\wedge(F,*),0.69)$

P_2: $((C,*)\wedge(T,*)\wedge(P,35)\wedge(F,*),0.89)$

P_3: $((C,*)\wedge(T,*)\wedge(P,*)\wedge(F,triangular),0.96)$

P_4: $((C,*)\wedge(T,*)\wedge(P,30)\wedge(F,triangular),0.96)$

P_5: $((C,*)\wedge(T,*)\wedge(P,35)\wedge(F,triangular),0.96)$

P_6: $((C,*)\wedge(T,grande)\wedge(P,30)\wedge(F,*),0.88)$

P_7: $((C,*)\wedge(T,grande)\wedge(P,35)\wedge(F,*),0.88)$

P_8: $((C,*)\wedge(T,grande)\wedge(P,*)\wedge(F,*),0.88)$

El conjunto de predicados obtenido a partir de los rasgos complejos puede contener predicados que no contribuyan a mejorar la calidad de los conceptos. Por lo tanto, este

conjunto de predicados puede reducirse utilizando la misma estrategia utilizada en el algoritmo FCKMSF, descrita en la Sección 4.5.2.

Después de aplicar el proceso de reducción sobre los predicados del ejemplo, los conceptos obtenidos son los siguientes:

Para A_1:

$\boldsymbol{C_1}$: $((Peso = 20) \wedge (Forma, redondo), 0.95)$

$\vee$ $((Peso = 20), 0.86)$

Para A_2:

$\boldsymbol{C_2}$: $((Forma, triangular), 0.96)$

$\vee$ $((Peso = 35), 0.89)$

El grado en que un concepto difuso cubre a un objeto se determina de la misma manera que en el algoritmo FCKMSF (ver Sección 4.5.2).

Para el ejemplo 4.5, los grados en que el concepto difuso cubre a cada uno de los objetos son los que se muestran en las columnas C_1 y C_2 de la Tabla 31.

Objetos	Atributos				Agrupamientos Difusos		Grados en que un concepto cubre a los objetos	
	Color (C)	Tamaño (T)	Peso (P)	Forma (F)	A_1	A_2	C_1	C_2
O_1	rojo	chico	20	redondo	1.00	0.00	0.95	0.00
O_2	rojo	mediano	20	redondo	0.94	0.06	0.95	0.00
O_3	azul	chico	25	redondo	0.90	0.10	0.95	0.00
O_4	azul	mediano	25	cuadrado	0.50	0.50	0.86	0.00
O_5	verde	grande	30	triangular	0.06	0.94	0.00	0.96
O_6	verde	chico	20	redondo	0.94	0.06	0.95	0.00
O_7	amarillo	grande	30	triangular	0.00	1.00	0.00	0.96
O_8	amarillo	mediano	35	triangular	0.06	0.94	0.00	0.96
O_9	verde	grande	35	redondo	0.31	0.69	0.00	0.89

Tabla 31. Grados de pertenencia de los objetos a los agrupamientos y grados en que el concepto cubre a los objetos.

Finalmente, se evalúa la calidad de los conceptos difusos utilizando la expresión (4.1). La calidad obtenida, para el ejemplo es: 0.82, la cual es mejor que la obtenida por el algoritmo FCKMSF (0.75).

El algoritmo FCKMCF es el siguiente:

4.6.3. Algoritmo FCKMCF

Entrada: Un conjunto T de objetos a ser agrupados.
Un número k de agrupamientos deseados.

Salida: Una partición difusa $\{A_1,\ldots,A_k\}$ en k agrupamientos difusos de T y el concepto difuso C_i que caracteriza a cada agrupamiento difuso A_i, $i = 1,\ldots,k$.

Fase de agrupamiento

Paso 1: Aplicar el algoritmo k-means con funciones de similaridad, para generar los agrupamientos A_i, $i = 1,\ldots,k$.

Fase de caracterización

Paso 1: Para cada agrupamiento A_i, $i = 1,\ldots,m$ hacer

Paso 2: Calcular los conjuntos de apoyo para el agrupamiento A_i.

Paso 3: Calcular los rasgos complejos difusos para el agrupamiento A_i.

Paso 4: Asociar a cada rasgo complejo difuso un predicado difuso (P,μ_P).

Paso 5: Reducir el número de predicados utilizando el procedimiento propuesto.

Paso 6: Construir el concepto difuso C_i como la disyunción de los predicados obtenidos en el paso 5.

4.6.4. Resultados Experimentales

En esta sección se muestran los resultados obtenidos con el algoritmo k-means conceptual difuso con rasgos complejos (FCKMCF). Inicialmente se realizaron pruebas usando bases de datos sintéticas que contienen objetos descritos por atributos numéricos, con el objetivo de ilustrar el comportamiento del algoritmo. Posteriormente, para mostrar el desempeño del algoritmo se utilizaron las bases de datos de la Tabla 25.

***Ejemplo 4.6*:** En este ejemplo se muestra el comportamiento del algoritmo cuando los agrupamientos que se desean obtener están bien definidos, aplicado a los datos mostrados en el ejemplo 4.3 (ver Figura 33 a)). En la Figura 33 b) se muestran los agrupamientos obtenidos después de aplicar la fase de agrupamiento del algoritmo FCKMCF.

Sobre los agrupamientos difusos obtenidos en la fase de agrupamiento (ver Figura 33 b)) se aplicó la fase de caracterización y los conceptos obtenidos fueron los siguientes:

Para el agrupamiento A_1:

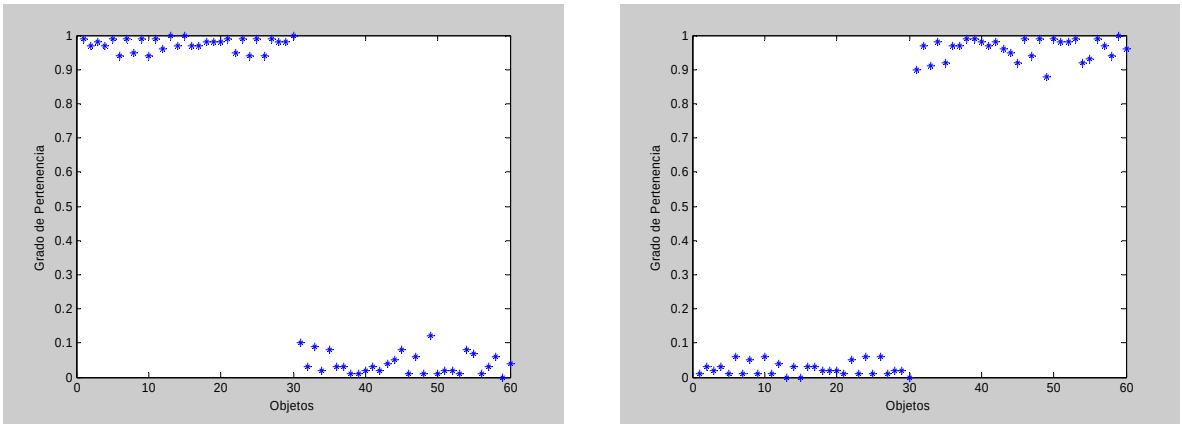
$\boldsymbol{C_1}$**:** $((x = 8.98), 0.97)$
$\vee\ ((x = 7.00), 0.96)$

Para el agrupamiento A_2:

$\boldsymbol{C_2}$**:** $((x = 3.00) \wedge (y = 1.00), 0.98)$
$\vee\ ((x = 2.98) \wedge (y = 1.21), 0.98)$
$\vee\ ((x = 3.67) \wedge (y = 2.74), 0.98)$

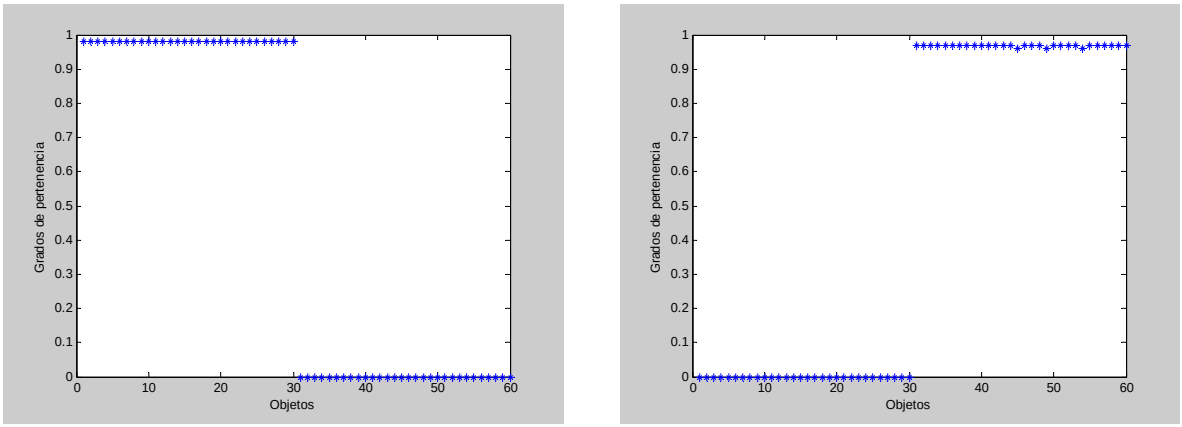
La calidad de estos conceptos es: 0.95.

En la Figura 63 se muestran los grados de pertenencia asignados por el algoritmo FKMSF a cada objeto de la muestra de datos de la Figura 33 a) y en la Figura 64 se muestran los grados en que los objetos de la Figura 33 a) son cubiertos por los conceptos.



a) **b)**

Figura 63. a) Grados de pertenencia de los objetos al agrupamiento A_1, **b)** Grados de pertenencia de los objetos al agrupamiento A_2.



a) **b)**

Figura 64. a) Grados en que los objetos son cubiertos por el concepto C_1, **b)** Grados en que los objetos son cubiertos por el concepto C_2.

En la Figura 64 observamos que los grados en que los objetos son cubiertos por el concepto son similares a los grados de pertenencia de los objetos a los agrupamientos en el sentido de que objetos con pertenencias altas son cubiertos con alto grado por el concepto mientras que objetos con pertenencias bajas son cubiertos con grado bajo por el concepto.

Los grados en que los objetos son cubiertos por los conceptos se muestran en la Figura 65.

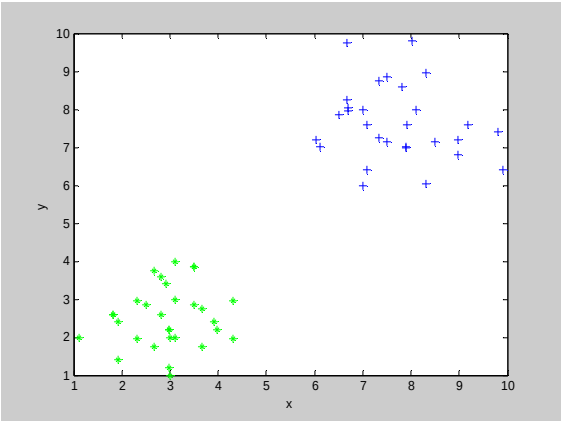


Figura 65. Grados en que los objetos son cubiertos por los conceptos.

En este ejemplo, podemos observar que, al igual que en algoritmo FCKMSF, los grados en que los objetos son cubiertos por los conceptos (ver Figura 65) son similares a los grados de pertenencia de los objetos a los agrupamientos (Figura 33 b)).

***Ejemplo 4.7*:** En este ejemplo se muestra el comportamiento del algoritmo cuando los agrupamientos que se desean obtener se traslapan, aplicado a los datos mostrados en el ejemplo 4.4 (ver Figura 37 a)). En la Figura 37 b) se muestran los agrupamientos obtenidos después de aplicar la fase de agrupamiento del algoritmo FCKMCF.

Sobre los agrupamientos obtenidos en la fase de agrupamiento (ver Figura 37 b)) se aplicó la fase de caracterización y los conceptos obtenidos fueron los siguientes:

Para el agrupamiento A_1:

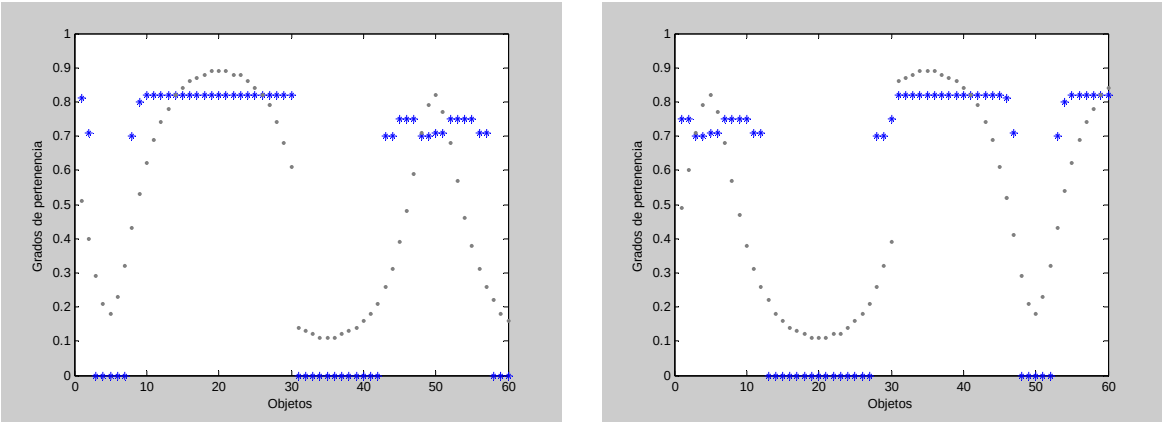
$\boldsymbol{C_1}$**:** $((y = 1.95), 0.82)$
$\vee$ $((x = 1.98), 0.82)$
$\vee$ $((y = 1.87), 0.81)$
$\vee$ $((x = 1.91), 0.80)$
$\vee$ $((x = 1.81), 0.75)$
$\vee$ $((y = 1.74), 0.75)$
$\vee$ $((y = 1.59), 0.71)$
$\vee$ $((x = 1.67), 0.70)$

Para el agrupamiento A_2:

C_2: $((y = -0.95), 0.82)$
$\vee$ $((x = -0.98), 0.82)$
$\vee$ $((y = -0.87), 0.81)$
$\vee$ $((x = -0.91), 0.80)$
$\vee$ $((x = -0.81), 0.75)$
$\vee$ $((y = -0.74), 0.75)$
$\vee$ $((y = -0.59), 0.71)$
$\vee$ $((x = -0.67), 0.70)$

La calidad de estos conceptos es: 0.74.

En la Figura 66 se sobreponen los grados en que los objetos de la Figura 37 a) son cubiertos por los conceptos.



a) **b)**

Figura 66. a) Grados en que los objetos son cubiertos por el concepto C_1, **b)** Grados en que los objetos son cubiertos por el concepto C_2.

Los grados en que los objetos son cubiertos por los conceptos se muestran en la Figura 67.

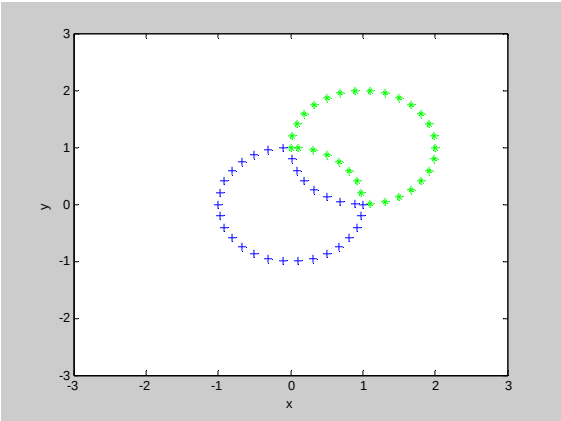


Figura 67. Grados en que los objetos son cubiertos por los conceptos.

En este ejemplo, podemos observar que se obtiene una aproximación de los grados en que los objetos son cubiertos por los conceptos a los grados de pertenencia de los objetos a los agrupamientos.

Los ejemplos anteriores nos muestran el comportamiento del algoritmo FCKMCF cuando los agrupamientos que se desean obtener están bien definidos y cuando se traslapan. En estos ejemplos podemos observar que cuando los agrupamientos están bien definidos los grados de pertenencia asignados por los conceptos son más similares a los grados de pertenencia de los objetos a los agrupamientos, que en el caso en que los agrupamientos se traslapan.

A continuación se muestran los resultados obtenidos por el algoritmo FCKMCF aplicado sobre las bases de datos de la Tabla 25.

Al igual que en el algoritmo FCKMSF, para las bases de datos que contienen ausencia de información se realizaron pruebas completando los datos faltantes. Los resultados obtenidos se muestran en las Tablas 32 y 33.

En la Tabla 32 se muestran las calidades de los conceptos obtenidos por el algoritmo FCKMCF con los diferentes conjuntos de apoyo cuando se completa la información antes de agrupar, después de agrupar y sin completar la información. Mientras que en la Tabla 33

se muestra el número de predicados obtenidos con los diferentes conjuntos de apoyo completando la información antes y después de agrupar y sin completar la información.

Base de datos	Algoritmo FCKMCF											
	Sin completar la información				Completando la información antes de agrupar				Completando la información después de agrupar			
	Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt
Auto-mpg	0.72	0.72	0.72	0.75	0.61	0.61	0.61	0.74	0.46	0.46	0.46	0.73
Bridges	0.50	0.45	0.50	0.61	0.58	0.49	0.58	0.60	0.49	0.43	0.50	0.60
Echocardiogram	0.57	0.57	0.56	0.63	0.57	0.56	0.56	0.60	0.57	0.55	0.56	0.60
Hepatitis	0.45	0.58	0.48	0.62	0.35	0.33	0.34	0.60	0.55	0.58	0.55	0.61
Import85	0.45	0.50	0.47	0.58	0.46	0.51	0.48	0.58	0.46	0.50	0.46	0.58
Promedio	**0.54**	**0.56**	**0.55**	**0.64**	**0.51**	**0.50**	**0.51**	**0.62**	**0.51**	**0.50**	**0.51**	**0.62**

Tabla 32. Calidades de los conceptos obtenidos con el algoritmo CKMCF sin completar la información y completando la información antes y después de agrupar los objetos.

En la Tabla 32 podemos observar que se obtienen conceptos con mejor calidad cuando se seleccionan los Φ-testores como conjuntos de apoyo y no se completa la información. Para los conjuntos Γ-diferenciantes, Γ-caracterizantes y Γ-testores, los mejores resultados se obtienen cuando no se completa la información.

Base de datos	Algoritmo FCKMCF											
	Sin completar la información				Completando la información antes de agrupar				Completando la información después de agrupar			
	Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt
Auto-mpg	22	22	22	60	24	24	24	65	25	25	25	65
Bridges	15	6	18	50	20	10	25	56	18	10	20	54
Echocardiogram	58	58	70	70	60	59	65	75	58	60	65	74
Hepatitis	40	73	49	60	40	70	45	64	40	72	49	63
Import85	18	37	31	90	23	42	40	110	24	42	40	115
Promedio	**31**	**39**	**38**	**66**	**33**	**41**	**40**	**74**	**33**	**42**	**40**	**74**

Tabla 33. Número de predicados obtenidos con el algoritmo CKMCF sin completar la información y completando la información antes y después de agrupar los objetos.

En la Tabla 33 podemos observar que, en promedio, el algoritmo obtiene menor número de predicados cuando se utilizan las bases de datos sin completar la información, para los cuatro tipos de conjuntos de apoyo. Además observamos que, cuando se seleccionan los Φ-testores como conjuntos de apoyo el número de predicados es mayor que cuando se seleccionan los conjuntos Γ-diferenciantes, Γ-caracterizantes y Γ-testores.

En las Figuras 68 y 69 se muestran gráficamente los resultados de las Tablas 32 y 33. En la Figura 68 se muestran las calidades de los conceptos obtenidos por el algoritmo FCKMCF cuando se completa la información antes y después de agrupar y sin completar la información. En la Figura 69 se muestra el número de predicados obtenidos por el algoritmo FCKMCF completando la información antes y después de agrupar y sin completar la información.

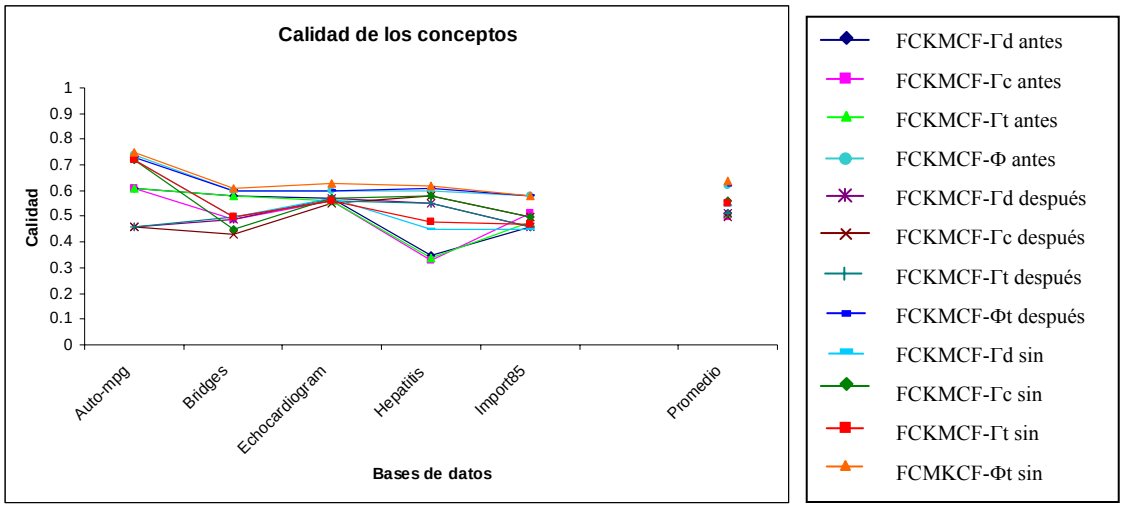


Figura 68. Calidades de los conceptos obtenidos con el algoritmo FCKMCF utilizando diferentes conjuntos de apoyo.

En la gráfica de la Figura 68 podemos observar que, en promedio, se obtienen mejores resultados cuando se aplica el algoritmo FCKMCF utilizando los Φ-testores como conjuntos de apoyo sin completar la información.

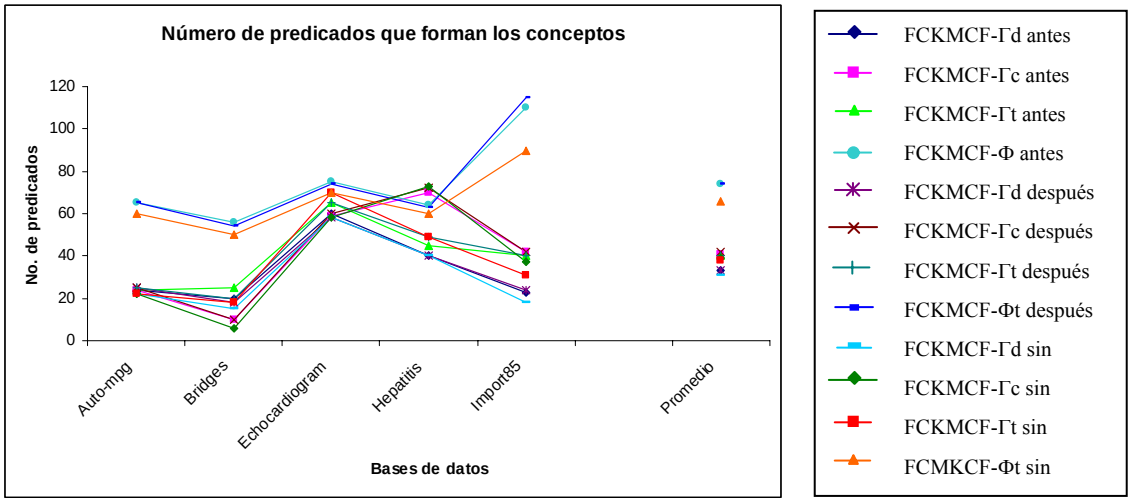


Figura 69. Número de predicados obtenidos con el algoritmo FCKMCF utilizando diferentes conjuntos de apoyo.

En la gráfica de la Figura 69 podemos observar que, en promedio, se obtiene mayor número de predicados cuando se seleccionan los Φ-testores como conjuntos de apoyo completando la información antes de agrupar, después de agrupar y sin completar la información.

Para estas pruebas se aplicó el algoritmo FCKMCF utilizando los conjuntos Γ-diferenciantes (Γd), Γ-caracterizantes (Γc), Γ-testores (Γt) y Φ-testores difusos. Para obtener los conjuntos Γd, Γc y Γt es necesario endurecer los agrupamientos, para lo cual se toman los objetos que pertenecen con mayor grado a cada agrupamiento.

Posteriormente, se realizaron pruebas con todas las bases de datos de la Tabla 25. Para seleccionar los conjuntos de apoyo se realizaron pruebas con 10, 20 y 30 iteraciones y con 10, 20, 30, 40, 50, 100 y 500 individuos. Los mejores resultados se obtuvieron con 20 iteraciones y 500 individuos y se muestran en las Tablas 34y 35

En la Tabla 34 se muestran las calidades obtenidas por el algoritmo FCKMCF con cada uno de los conjuntos de apoyo y en la Tabla 35 se muestra el número de predicados obtenidos para cada conjunto de apoyo.

Bases de Datos	Algoritmo FCKMCF			
	Γd	Γc	Γt	Φ-testores
Diabetes	0.68	0.68	0.68	0.70
Glass	0.65	0.65	0.65	0.68
Iris	0.75	0.75	0.75	0.78
Wine	0.57	0.57	0.57	0.60
Hayes	0.63	0.63	0.63	0.65
Lenses	0.66	0.66	0.66	0.70
Zoo	0.59	0.50	0.58	0.60
Auto-mpg	0.72	0.72	0.72	0.75
Bridges	0.50	0.45	0.50	0.61
Echocardiogram	0.57	0.57	0.56	0.63
Hepatitis	0.45	0.58	0.48	0.62
Import85	0.45	0.50	0.47	0.58
Tae	0.52	0.52	0.52	0.60
Promedio	**0.60**	**0.60**	**0.58**	**0.65**

Tabla 34. Calidades de los conceptos obtenidos con el algoritmo FCKMCF usando diferentes conjuntos de apoyo.

En la Tabla 34 observamos que se obtienen mejores resultados al seleccionar los Φ-testores como conjuntos de apoyo, esto debido a que este tipo de conjuntos de apoyo toman en cuenta que la semejanza entre los grados de pertenencia de los objetos así como la semejanza entre las descripciones de los objetos. Mientras que, los conjuntos Γ-diferenciantes y Γ-caracterizantes toman en cuenta sólo la semejanza entre los objetos.

En la Tabla 35 se puede ver que, en la mayoría de los casos, el número de predicados obtenidos cuando se seleccionan los Φ-testores como conjuntos de apoyo es mayor que cuando se seleccionan los conjuntos Γ-diferenciantes, Γ-caracterizantes o Γ-testores.

Bases de Datos	Algoritmo FCKMCF			
	Γd	Γc	Γt	Φ-testores
Diabetes	72	72	72	105
Glass	21	21	21	40
Iris	3	3	3	10
Wine	99	94	112	99
Hayes	23	23	23	25
Lenses	14	14	14	11
Zoo	17	13	15	22
Auto-mpg	22	22	22	60
Bridges	15	6	18	50
Echocardiogram	58	58	70	70
Hepatitis	40	73	49	60
Import85	18	37	31	90
Tae	21	21	21	50
Promedio	**33**	**35**	**36**	**53**

Tabla 35. Número de predicados obtenidos con el algoritmo FCKMCF usando diferentes conjuntos de apoyo.

En las Figuras 70 y 71 se muestran gráficamente los resultados de las Tablas 34y 35. En la Figura 70 se muestran las calidades de los conceptos obtenidos por el algoritmo FCKMCF y en la Figura 71 se muestra el número de predicados obtenidos por el algoritmo FCKMCF.

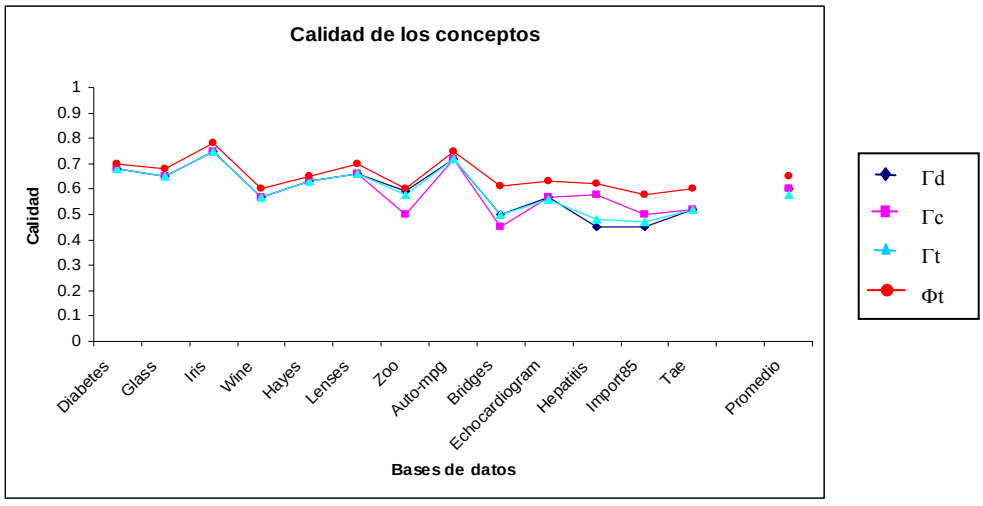


Figura 70. Calidades de los conceptos obtenidos con el algoritmo FCKMCF utilizando diferentes conjuntos de apoyo.

En la gráfica de la Figura 70 podemos observar que se obtienen mejores resultados cuando se aplica el algoritmo FCKMCF utilizando los Φ-testores como conjuntos de apoyo.

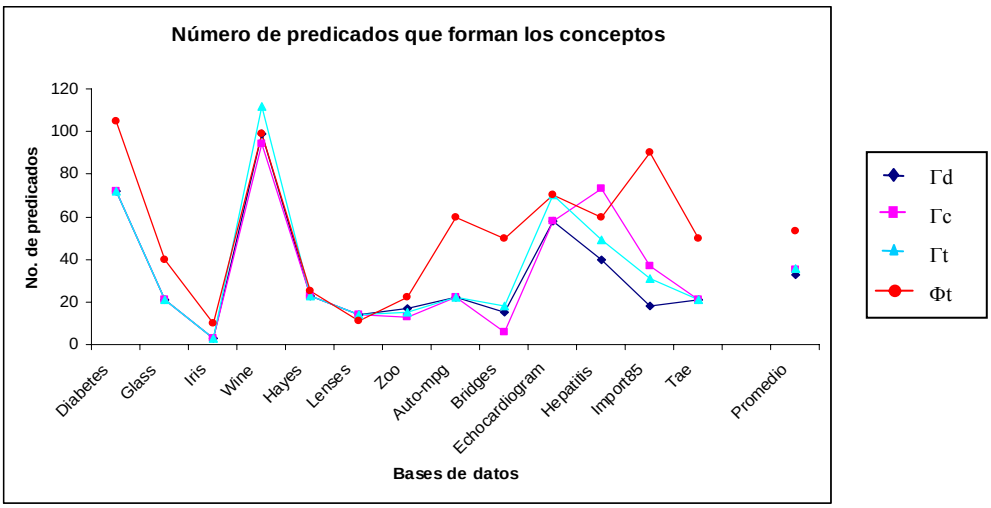


Figura 71. Número de predicados obtenidos con el algoritmo FCKMCF utilizando diferentes conjuntos de apoyo.

En la gráfica de la Figura 71 podemos observar que, en promedio, se obtiene mayor número de predicados cuando se seleccionan los Φ-testores como conjuntos de apoyo.

4.6.5. Discusión

El algoritmo k-means conceptual difuso con rasgos complejos usa, en la fase de agrupamiento, una función de similaridad dada en términos de funciones de comparación, las cuales permiten expresar la forma en que los valores de los atributos son comparados dependiendo del contexto del problema a resolver.

En la fase de caracterización se utilizó el concepto de rasgo complejo difuso para generar los conceptos que caracterizan a los agrupamientos obtenidos en la fase de agrupamiento. Los rasgos complejos difusos son subdescripciones de objetos que permiten diferenciar a objetos con alto grado a distintos agrupamientos y al mismo tiempo permiten caracterizar a objetos con alto grado al mismo agrupamiento.

Con base en los experimentos observamos que el algoritmo FCKMCF obtiene mejores resultados cuando se utilizan los Φ-testores difusos como conjuntos de apoyo, que cuando se utilizan los conjuntos Γ-diferenciantes, Γ-caracterizantes o Γ-testores. Esto se debe a que los Φ-testores difusos toman en cuenta no sólo la semejanza entre los objetos sino además la semejanza entre los grados de pertenencia; estos conjuntos de apoyo guardan una estrecha relación entre la semejanza de los objetos y la semejanza entre los grados de pertenencia.

El algoritmo FCKMCF tiene la ventaja de que puede ser aplicado sobre bases de datos que contengan descripciones de objetos incompletas. Además, no requiere de retículos de generalización los cuales, como se mencionó anteriormente, no siempre están disponibles.

4.7. Comparación entre los Algoritmos Propuestos

En esta sección se presenta una comparación entre los algoritmos k-means conceptual difuso con funciones de similaridad (FCKMSF) y k-means conceptual difuso con rasgos complejos (FCKMCF).

Para las bases de datos en las que se observa ausencia de información se realizó una comparación entre los algoritmos FCKMSF y FCKMCF cuando se completa la información antes de agrupar, completando la información después de agrupar y sin completar la información. Esta comparación se muestra en las Tablas 36-39.

En la Tabla 36 se muestra la comparación entre los algoritmos FCKMSF y FCKMCF cuando se completa la información antes de agrupar y en la Tabla 37 se muestra la comparación entre los algoritmos FCKMSF y FCKMCF completando la información después de agrupar y sin completar la información.

Base de datos	Algoritmo FCKMSF	Algoritmo FCKMCF			
		Γd	Γc	Γt	Φ-testores
	Completando la información antes de agrupar				
Auto-mpg	0.60	0.61	0.61	0.61	0.74
Bridges	0.67	0.58	0.49	0.58	0.60
Echocardiogram	0.64	0.57	0.56	0.56	0.60
Hepatitis	0.74	0.35	0.33	0.34	0.60
Import85	0.60	0.46	0.51	0.48	0.58
Promedio	**0.68**	**0.51**	**0.50**	**0.51**	**0.62**

Tabla 36. Calidades de los conceptos obtenidos por los algoritmos FCKMSF y FCKMCF completando la información antes de agrupar.

Base de datos	Algoritmo FCKMSF	Algoritmo FCKMCF							
		Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt
	Completando la información después de agrupar					Sin completar la información			
Auto-mpg	0.57	0.46	0.46	0.46	0.73	0.72	0.72	0.72	0.75
Bridges	0.59	0.49	0.43	0.50	0.60	0.50	0.45	0.50	0.61
Echocardiogram	0.64	0.57	0.55	0.56	0.60	0.57	0.57	0.56	0.63
Hepatitis	0.73	0.55	0.58	0.55	0.61	0.45	0.58	0.48	0.62
Import85	0.60	0.46	0.50	0.46	0.58	0.45	0.50	0.47	0.58
Promedio	**0.63**	**0.51**	**0.50**	**0.51**	**0.62**	**0.54**	**0.56**	**0.55**	**0.64**

Tabla 37. Calidades de los conceptos obtenidos por los algoritmos FCKMSF y FCKMCF completando la información después de agrupar y sin completar la información.

En las Tablas 36 y 37 se puede ver que los mejores resultados se obtienen con el algoritmo FCKMSF cuando se completa la información antes de agrupar.

En la Figura 72 se muestra de manera gráfica los resultados de las Tablas 36 y 37.

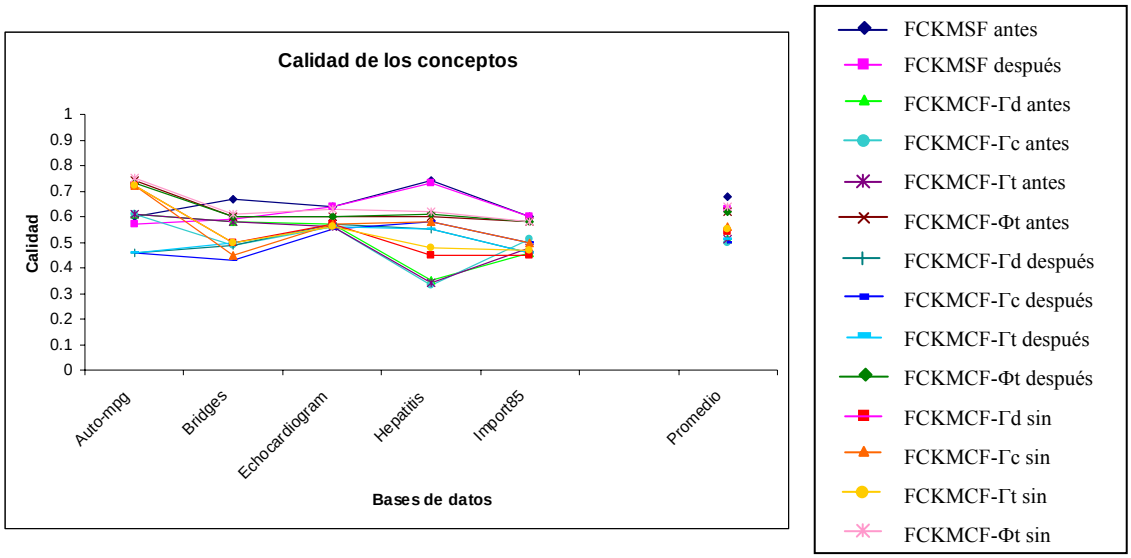


Figura 72. Calidades de los conceptos obtenidos con los algoritmos FCKMSF y FCKMCF completando la información antes y después de agrupar y sin completar la información.

En la gráfica de la Figura 72 observamos que, en promedio, el algoritmo FCKMSF obtiene mejores resultados cuando se completa la información antes de agrupar.

En la Tabla 38 se muestra una comparación entre el número de predicados obtenidos por los algoritmos FCKMSF y FCKMCF cuando se completa la información antes de agrupar y en la Tabla 39 se muestra la comparación entre el número de predicados obtenidos completando la información después de agrupar y sin completar la información.

Base de datos	Algoritmo FCKMSF	Algoritmo FCKMCF			
		Γd	Γc	Γt	Φ-testores
	Completando la información antes de agrupar				
Auto-mpg	96	24	24	24	65
Bridges	92	20	10	25	56
Echocardiogram	84	60	59	65	75
Hepatitis	43	40	70	45	64
Import85	112	23	42	40	110
Promedio	**83**	**33**	**41**	**40**	**74**

Tabla 38. Número de predicados obtenidos por los algoritmos FCKMSF y FCKMCF completando la información antes de agrupar.

Base de datos	Algoritmo FCKMSF	Algoritmo FCKMCF							
		Γd	Γc	Γt	Φt	Γd	Γc	Γt	Φt
	Completando la información después de agrupar					Sin completar la información			
Auto-mpg	190	25	25	25	65	22	22	22	60
Bridges	59	18	10	20	54	15	6	18	50
Echocardiogram	84	58	60	65	74	58	58	70	70
Hepatitis	133	40	72	49	63	40	73	49	60
Import85	112	24	42	40	115	18	37	31	90
Promedio	**116**	**33**	**42**	**40**	**74**	**31**	**39**	**38**	**66**

Tabla 39. Número de predicados obtenidos por los algoritmos FCKMSF y FCKMCF completando la información después de agrupar y sin completar la información.

En las Tablas 38 y 39 se observa que el algoritmo FCKMSF completando la información después de agrupar obtiene el mayor número de predicados mientras que el algoritmo FCKMCF con conjuntos Γ-diferenciantes sin completar la información obtiene el menor número de predicados.

En la Figura 73 se muestra de manera gráfica los resultados de las Tablas 38 y 39.

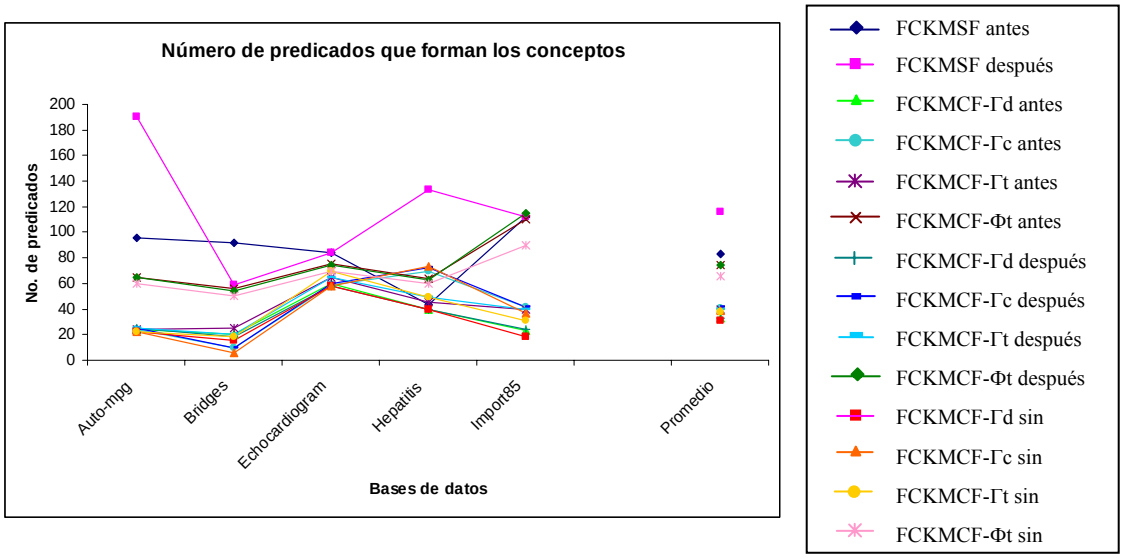


Figura 73. Número de predicados obtenidos con los algoritmos FCKMSF y FCKMCF.

En la gráfica de la Figura 73 podemos ver que, en promedio, el algoritmo FCKMSF obtiene mayor número de predicados cuando se completa la información después de agrupar.

Finalmente, para realizar la comparación entre los algoritmos, las bases de datos Auto-mpg, Bridges, Echocardiogram, Hepatitis e Import85 se toman con las descripciones de los objetos completadas antes de aplicar la fase de agrupamiento. Los resultados obtenidos con los algoritmos FCKMSF y FCKMCF se muestran en las Tablas 40 y 41.

En la Tabla 40 se muestran las calidades de los conceptos obtenidos con los algoritmos FCKMSF y FCKMCF completando la información antes de agrupar. En la Tabla 41 se muestra una comparación del número de predicados que forman los conceptos.

En la Tabla 40 observamos que, en promedio, se obtienen mejores resultados con el algoritmo FCKMSF. Sin embargo, los resultados obtenidos con el algoritmo FCKMCF con Φ-testores son similares a los resultados obtenidos con el algoritmo FCKMSF.

Bases de Datos	Algoritmo FCKMSF	Algoritmo FCKMCF			
		Γd	Γc	Γt	Φ-testores
Diabetes	0.66	0.68	0.68	0.68	0.70
Glass	0.69	0.65	0.65	0.65	0.68
Iris	0.78	0.75	0.75	0.75	0.78
Wine	0.62	0.57	0.57	0.57	0.60
Hayes	0.71	0.63	0.63	0.63	0.65
Lenses	0.77	0.66	0.66	0.66	0.70
Zoo	0.64	0.59	0.50	0.58	0.60
Auto-mpg	0.60	0.72	0.72	0.72	0.75
Bridges	0.67	0.50	0.45	0.50	0.61
Echocardiogram	0.64	0.57	0.57	0.56	0.63
Hepatitis	0.74	0.45	0.58	0.48	0.62
Import85	0.60	0.45	0.50	0.47	0.58
Tae	0.72	0.52	0.52	0.52	0.60
Promedio	**0.68**	**0.60**	**0.60**	**0.58**	**0.65**

Tabla 40. Calidades de los conceptos obtenidos por los algoritmos FCKMSF y FCKMCF.

Bases de Datos	Algoritmo FCKMSF	Algoritmo FCKMCF			
		Γd	Γc	Γt	Φ-testores
Diabetes	230	72	72	72	105
Glass	69	21	21	21	40
Iris	37	3	3	3	10
Wine	99	99	94	112	99
Hayes	66	23	23	23	25
Lenses	11	14	14	14	11
Zoo	41	17	13	15	22
Auto-mpg	96	22	22	22	60
Bridges	92	15	6	18	50
Echocardiogram	84	58	58	70	70
Hepatitis	43	40	73	49	60
Import85	112	18	37	31	90
Tae	96	21	21	21	50
Promedio	**83**	**33**	**35**	**36**	**53**

Tabla 41. Número de predicados obtenidos con los algoritmos FCKMSF y FCKMCF.

En la Tabla 41 podemos observar que el algoritmo FCKMSF obtiene mayor número de predicados que forman los conceptos que los obtenidos por el algoritmo FCKMCF.

En las Figuras 74 y 75 se muestran de manera gráfica los resultados de las Tablas 40 y 41. En la Figura 74 se muestra una comparación entre las calidades obtenidas por los algoritmos FCKMSF y FCKMCF, mientras que en la Figura 75 se muestra una comparación entre el número de predicados que forman los conceptos.

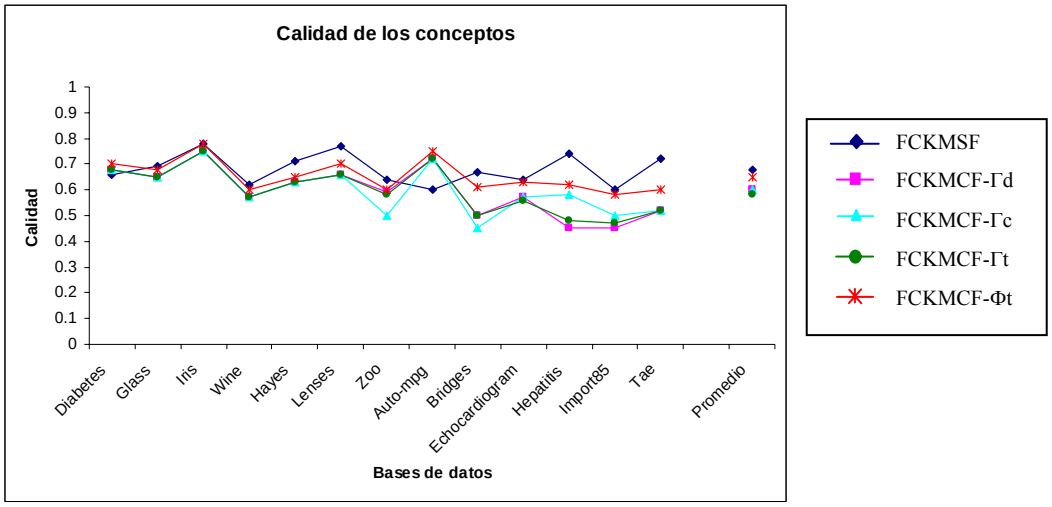


Figura 74. Calidades de los conceptos obtenidos con los algoritmos FCKMSF y FCKMCF.

En la gráfica de la Figura 74 observamos que las calidades obtenidas por el algoritmo FCKMSF son mejores, en algunos casos, que las calidades obtenidas por el algoritmo FCKMSF. Sin embargo, en promedio, los algoritmos FCKMSF y FCKMCF con Φ-testores tienen un desempeño similar, siendo un poco mejor el algoritmo FCKMSF.

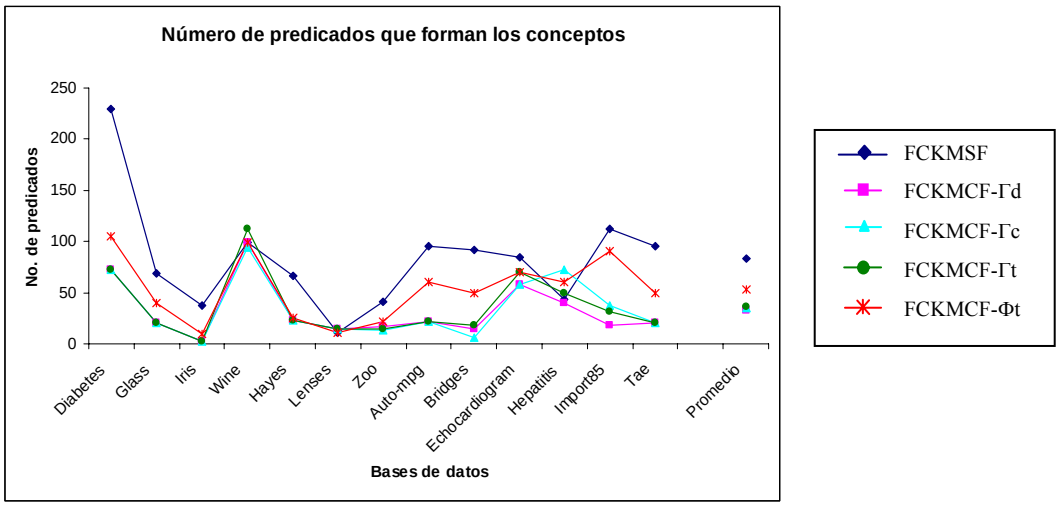


Figura 75. Número de predicados obtenidos con los algoritmos FCKMSF y FCKMCF.

En la gráfica de la Figura 75 observamos que, en la mayoría de los casos, el algoritmo FCKMSF obtiene mayor número de predicados que el algoritmo FCKMCF, mientras que el menor número de predicados se obtiene con el algoritmo FCKMCF cuando se utilizan como conjuntos de apoyo los conjuntos Γ-diferenciantes, Γ-caracterizantes y Γ-testores.

4.8. Análisis de Complejidad de los Algoritmos Conceptuales Difusos

En esta sección se hace un análisis de la complejidad de los algoritmos FCKMSF y FCKMCF.

Sea n el número de objetos, k el número de agrupamientos, n_i el número de objetos en el agrupamiento A_i, m el número de atributos.

Fase de agrupamiento

Los algoritmos FCKMSF y FCKMCF utilizan en su fase de agrupamiento el algoritmo k-means difuso con funciones de similaridad (FKMSF). El algoritmo FKMSF depende del número de iteraciones que el algoritmo requiera para que los centroides se estabilicen. Para cada iteración, n objetos son comparados con k centroides de los agrupamientos, para lo cual se requieren $n \times k$ llamadas a la función de similaridad. Para la ejecución completa del algoritmo KMSF, el tiempo requerido es $O(lnk)$, donde l es el número de iteraciones, en nuestros experimentos fijamos 10 como número máximo de iteraciones, pero en la mayoría de los casos se detenía en 4 o 5 iteraciones. La complejidad en espacio es $O(kn)$, que es el espacio requerido para almacenar los objetos.

Fase de caracterización

Algoritmo FCKMSF

Inicialmente se generan predicados difusos a partir de los objetos. A cada objeto que obtiene su máximo grado en A_i se asigna un predicado duro y esto se hace para cada agrupamiento, esto es $\sum_{i=1}^{k} n_i = n$. Cada predicado está formado por m atributos, además se asigna a cada predicado un grado, lo cual es constante con respecto al número de objetos. Por lo tanto, el tiempo requerido para generar los predicados iniciales es $O(nm)$.

Posteriormente se generan nuevos predicados difusos que deben satisfacer la condición α-discriminante.

Sea q_i el número de valores posibles para x_s, s=1,…,m entonces se tiene que el máximo número posible de predicados difusos es $\prod_{s=1}^{m} q_s$. Si q es el máximo de los q_s, s = 1,…,m, entonces la complejidad en el peor caso es $O(q^m)$. En este paso, se combinan sólo aquellos predicados que tienen grado similar. En el peor de los casos se combinan todos los predicados.

Para cada uno de los q^m predicados difusos generados se verifica si al agregar este nuevo predicado difuso se mejora la calidad; la función de calidad se evalúa sobre los n objetos, en los m atributos y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para generar los predicados α-discriminantes es, en el peor caso, $O(knmq^m)$.

El siguiente paso consiste en verificar la condición β-caracterizante para cada predicado α-discriminante, lo cual requiere comparar contra los n_i objetos que alcanzan su máximo grado en el agrupamiento, en los m atributos, lo cual es $O(nm)$ y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para verificar la condición β-caracterizante es $O(knmq^m)$.

Finalmente se aplica el proceso de reducción de predicados.

Primero se ordenan los predicados difusos de forma descendente con base en su grado, lo cual requiere de un tiempo $O(q^m \ log_2 q^m) = O(mq^m)$. Después se van almacenando aquellos predicados que mejoran la calidad de los conceptos, lo cual implica evaluar la función de calidad sobre los n objetos y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para reducir los predicados es $O(knmq^m)$

Por todo lo anterior el tiempo total requerido por el algoritmo FCKMSF, en el peor caso, es $O(knmq^m)$.

El espacio requerido, en el peor caso, es $O(kmq^m)$, que es el espacio necesario para almacenar los predicados difusos.

Algoritmo FCKMCF

Inicialmente se calculan los conjuntos de apoyo utilizando un algoritmo genético. Para el algoritmo genético se fijaron el número de iteraciones y el tamaño de la población. Sea p el tamaño de la población y r el número de iteraciones. La complejidad de este paso es $O(prm)$. En nuestro caso p y r son constantes; por lo tanto, el tiempo requerido para calcular los conjuntos de apoyo es $O(m)$.

Posteriormente se calculan los rasgos complejos. El tiempo requerido para calcular los rasgos complejos difusos (De-la-Vega-Doria, 1994) es $O(akn^3m)$, donde a es el número de conjuntos de apoyo.

El número máximo de rasgos complejos (De-la-Vega-Doria, 1994), que en nuestro caso es fijo, es $O(kamn)$ y a cada rasgo complejo se asocia un predicado difuso lo cual requiere un tiempo de $O(kam^2n)$.

Finalmente se aplica el proceso de reducción de predicados.

Primero se ordenan los predicados difusos de forma descendente con base en su grado, lo cual requiere de un tiempo $O((kam^2n)\ log_2\ (kam^2n))$. Después se van almacenando aquellos predicados que mejoran la calidad de los conceptos, lo cual implica evaluar la función de calidad sobre los n objetos y esto se hace para los k agrupamientos. Por lo tanto, el tiempo requerido para reducir los predicados es $O(kamn^2)$.

Por todo lo anterior, el tiempo total requerido por el algoritmo FCKMSF, en el peor caso, es $O(akn^3m)$.

El espacio requerido, en el peor caso, es $O(kamn)$, que es el espacio necesario para almacenar los predicados difusos.

En la Tabla 42 se muestra el tiempo y el espacio requeridos por cada uno de los algoritmos.

Complejidad	FCKMSF	FCKMCF
Tiempo	$O(knq^m m)$	$O(akn^3 m)$
Espacio	$O(kmq^m)$	$O(amn)$

Tabla 42. Complejidad en tiempo y espacio de los algoritmos difusos

4.9. Sumario

En este capítulo se introdujo una formalización del problema de agrupamiento conceptual difuso, se definió una función para evaluar la calidad de los conceptos difusos y se introdujeron los algoritmos k-means conceptual difuso con funciones de similaridad (FCKMSF) y k-means conceptual difuso con rasgos complejos (FCKMCF).

La función de calidad propuesta para conceptos difusos toma en cuenta qué tan cercano es el grado en que el concepto cubre a un objeto al grado de pertenencia del objeto al agrupamiento para aquellos objetos que pertenecen con alto grado al agrupamiento y al mismo tiempo qué tan lejano es el grado en que el concepto cubre a un objeto al grado de pertenencia del objeto al agrupamiento para aquellos objetos que pertenecen con grado bajo al agrupamiento. Por otra parte, esta función de calidad es una generalización de la función de calidad propuesta para los algoritmos duros, ya que si los grados de pertenencia fueran 1's y 0's como en el caso duro, entonces lo que se estaría tomando en cuenta sería cuántos objetos del agrupamiento son cubiertos por el concepto, y cuántos objetos fuera del agrupamiento son cubiertos por el concepto.

El algoritmo FCKMSF es una versión difusa del algoritmo CKMSF. En la fase de agrupamiento se utilizó el algoritmo k-means difuso con funciones de disimilaridad (FKMSF) para construir los agrupamientos. El algoritmo FKMSF permite trabajar con datos mezclados sin realizar transformaciones de los atributos. Además, permite utilizar funciones de comparación definidas por el especialista, dependiendo del problema que se esté resolviendo. Por otra parte, en la fase de caracterización se introdujo una manera de trabajar con retículos de generalización para generar conceptos difusos. Un inconveniente al utilizar retículos de generalización es que, para algunas aplicaciones, es difícil determinar cuál es el mejor retículo de generalización. Además, no se tienen métodos automáticos para construir los retículos, por lo que esta tarea se deja al especialista.

Por esta razón, se propuso el algoritmo FCKMCF, el cual no depende de retículos de generalización para la generación de los conceptos de los agrupamientos. En este algoritmo, al igual que en el algoritmo FCKMSF, se utilizó el algoritmo FKMSF para construir los agrupamientos. En la fase de caracterización se utilizaron los rasgos complejos para generar los conceptos de los agrupamientos y los Φ-testores difusos como conjuntos de apoyo.

Posteriormente, se mostraron los resultados obtenidos con los algoritmos FCKMSF y FCKMCF, así como una comparación entre ambos algoritmos. En esta comparación se pudo observar que los algoritmos FCKMSF y FCKMCF con Φ-testores difusos tienen un desempeño similar, siendo el algoritmo FCKMSF el que obtuvo ligeramente mejores resultados. Por otra parte, el algoritmo FCKMSF genera, en la mayoría de los casos, conceptos con mayor número de predicados que el algoritmo FCKMCF.

Los algoritmos FCKMSF y FCKMCF son una primera aproximación para la solución de problemas de agrupamiento conceptual restringido difuso y permiten resolver problemas en los cuales los objetos están descritos por atributos cualitativos y cuantitativos mezclados, posiblemente con ausencia de información en las descripciones de los objetos.

Conclusiones

Sumario

El problema de clasificación no supervisada restringida ha sido enfocado principalmente a obtener los agrupamientos en que se estructuran los objetos de una muestra dada, pero sin dar una descripción de estos agrupamientos. Sin embargo, en algunas disciplinas el especialista puede estar interesado en encontrar no sólo los agrupamientos en que se clasifican los objetos sino además las propiedades que caracterizan a estos agrupamientos.

Una característica importante de los problemas que se abordan en estas disciplinas es que, en muchos casos, los objetos están descritos por atributos cualitativos y cuantitativos mezclados y se puede presentar ausencia de información en las descripciones de los objetos.

Los algoritmos conceptuales restringidos basados en semillas propuestos en la literatura presentan algunas limitantes. Entre las que se encuentran las siguientes:

i) Algunos de estos algoritmos, para trabajar con atributos mezclados, realizan transformaciones de los atributos cualitativos en cuantitativos, o viceversa.

ii) Es necesario que las descripciones de los objetos sean completas, es decir no se permite ausencia de información.

iii) Los objetos pertenecen a los agrupamientos siempre en el mismo grado (agrupamientos duros).

Para superar estas limitantes, en esta tesis se propusieron cuatro extensiones del algoritmo k-means conceptual. Las extensiones propuestas son: k-means conceptual con funciones de similaridad (CKMSF), k-means conceptual con rasgos complejos (CKMCF), k-means conceptual difuso con funciones de similaridad (FCKMSF) y k-means conceptual difuso con rasgos complejos (FCKMCF). Estos algoritmos constan de dos fases: una fase de agrupamiento; y una fase de caracterización, en la que se generan los conceptos que caracterizan a los agrupamientos.

El algoritmo CKMSF es una modificación del algoritmo k-means conceptual. En la fase de agrupamiento se utilizó el algoritmo k-means con funciones de similaridad (KMSF) para construir los agrupamientos. El algoritmo KMSF permite trabajar con datos mezclados sin realizar transformaciones de los atributos. Además, utiliza funciones de comparación que pueden ser definidas por el especialista, dependiendo del problema que se esté resolviendo. Por otra parte, en la fase de caracterización se introdujo un nuevo retículo de generalización para los atributos cuantitativos, el cual permite obtener conceptos de mejor calidad. Un inconveniente al usar retículos de generalización es que, para ciertas aplicaciones, es difícil determinar cuál es el mejor retículo de generalización; además no se tienen métodos automáticos para construirlos, por lo que esta tarea se deja al especialista.

Por esta razón, se propuso el algoritmo CKMCF, el cual no depende de retículos de generalización para la construcción de los conceptos. En este algoritmo, al igual que en el algoritmo CKMSF, se propuso utilizar el algoritmo KMSF para construir los agrupamientos. En la fase de caracterización se propuso utilizar el concepto de rasgo complejo para generar los conceptos de los agrupamientos. Para calcular los rasgos complejos se utilizaron tres tipos de conjuntos de apoyo: conjuntos Γ-diferenciantes, conjuntos Γ-caracterizantes y Γ-testores.

Para evaluar la calidad de los conceptos obtenidos por los algoritmos CKMSF y CKMCF se propuso una función de calidad, la cual toma en cuenta el número de objetos del agrupamiento que son cubiertos por el concepto (ejemplos) así como el número de objetos fuera del agrupamiento que son cubiertos por el concepto (contraejemplos). Un

concepto será de mejor calidad en la medida en que reconozca mayor número de ejemplos y menor número de contraejemplos. El caso ideal es cuando el concepto cubre todos los objetos del agrupamiento y no cubre objetos fuera. Esta función puede utilizarse, además, para problemas en los cuales se conoce de antemano los agrupamientos y se desea obtener las propiedades que los caracterizan. Para este tipo de agrupamientos generalmente no se conoce la función de distancia o similaridad utilizada para construirlos. Por lo tanto, no es posible evaluar la calidad de los agrupamientos, sólo se puede evaluar la calidad de los conceptos generados.

Los algoritmos CKMSF y CKMCF generan conceptos duros a partir de agrupamientos duros. Sin embargo, en algunos problemas prácticos se desea que los objetos puedan pertenecer en distinto grado a los agrupamientos. El problema de agrupamiento conceptual restringido difuso ha sido poco estudiado por lo que, en esta tesis, se introdujo una formalización de este problema y se propusieron dos algoritmos para resolverlo.

El algoritmo FCKMSF es una versión difusa del algoritmo k-means conceptual con funciones de similaridad (CKMSF). En la fase de agrupamiento se utilizó el algoritmo k-means difuso con funciones de similaridad (FKMSF) para construir los agrupamientos. Este algoritmo es una versión difusa del algoritmo KMSF utilizado en la fase de agrupamiento de los algoritmos CKMSF y CKMCF. Por otra parte, en la fase de caracterización se introdujo una manera de trabajar con retículos de generalización para generar conceptos difusos a partir de agrupamientos difusos. Un inconveniente de este algoritmo, al igual que en el caso duro, es el uso de retículos de generalización. Por esta razón, se propuso el algoritmo FCKMCF, el cual no depende de retículos de generalización para la generación de los conceptos difusos. En este algoritmo, al igual que en el algoritmo FCKMSF, se utilizó el algoritmo FKMSF para construir los agrupamientos. Por otra parte, en la fase de caracterización se utilizó el concepto de rasgo complejo difuso para generar los conceptos de los agrupamientos. Para calcular los rasgos complejos difusos los mejores resultados se obtuvieron al utilizar los Φ-testores difusos como conjuntos de apoyo.

Para evaluar la calidad de los conceptos difusos obtenidos por los algoritmos FCKMSF y FCKMCF se propuso una función de calidad. Esta función toma en cuenta qué tan cercano es el grado en que el concepto cubre a un objeto, con respecto al grado de pertenencia del objeto al agrupamiento, para aquellos objetos que pertenecen con alto grado al agrupamiento. Al mismo tiempo se considera qué tan lejano es el grado en que el concepto cubre a un objeto, con respecto al grado de pertenencia del objeto al agrupamiento, para aquellos objetos que pertenecen con grado bajo al agrupamiento.

Conclusiones

A partir del análisis realizado del algoritmo k-means conceptual se concluye que el retículo de generalización propuesto por Ralambondrainy para los atributos cuantitativos no satisface la definición de retículo de generalización. Por lo cual, se uso un nuevo retículo de generalización que satisface la definición y permite obtener conceptos con mejor calidad.

Por otra parte, se realizaron pruebas con el algoritmo CKMSF tomando distintos valores para los parámetros α y β, de lo cual se concluye que las mejores calidades de los conceptos se obtienen para valores de α y β cercanos a 0. Además, al utilizar el retículo nuevo las calidades de los conceptos dependen menos de estos parámetros que al utilizar el retículo original.

En la experimentación realizada con los algoritmos conceptuales duros propuestos, observamos que los algoritmos CKMSF y CKMCF obtienen conceptos de mejor calidad que los conceptos obtenidos por el algoritmo CKM. Por otra parte, el algoritmo CKMCF genera conceptos con menor número de predicados que los algoritmos CKM y CKMSF.

Adicionalmente, para las bases de datos en las que se observa ausencia de información se pudo observar que se obtienen mejores resultados cuando se completan los datos faltantes antes de realizar el agrupamiento que cuando se completan los datos después de la fase de agrupamiento.

De esta parte de la tesis podemos concluir que el algoritmo CKMCF es una buena alternativa para la solución de problemas de agrupamiento conceptual restringido duro basado en semillas cuando los objetos están descritos por atributos cualitativos y cuantitativos mezclados y se presenta ausencia de información. Por otra parte, cuando se tienen problemas para los cuales se conocen los retículos de generalización, el algoritmo CKMSF es una alternativa viable para la solución de este tipo de problemas.

Por otra parte, en la experimentación realizada con los algoritmos conceptuales difusos propuestos, observamos que aún cuando los algoritmos FCKMSF y FCKMCF con Φ-testores difusos tienen un desempeño similar, el algoritmo FCKMSF obtuvo mejores calidades de los conceptos que el algoritmo FCKMCF. Mientras que, el algoritmo FCKMCF genera conceptos con menor número de predicados que el algoritmo FCKMSF.

Sobre los algoritmos conceptuales difusos podemos concluir que los algoritmos FCKMSF y FCKMCF son una primera aproximación para la solución de problemas de agrupamiento conceptual restringido difuso basado en semillas y permiten resolver problemas en los cuales los objetos están descritos por atributos cualitativos y cuantitativos mezclados, posiblemente con ausencia de información en las descripciones de los objetos.

Aportaciones

Las principales aportaciones de este proyecto de investigación doctoral son las siguientes:

i. Dos algoritmos conceptuales restringidos duros basados en semillas.
   a. Algoritmo k-means conceptual con funciones de similaridad (CKMSF).
   b. Algoritmo k-means conceptual con rasgos complejos (CKMCF).

ii. Una función para evaluar la calidad de los conceptos duros.

iii. Una formalización del problema de agrupamiento conceptual restringido difuso.

iv. Dos algoritmos conceptuales restringidos difusos basados en semillas.
   a. Algoritmo k-means conceptual difuso con funciones de similaridad (FCKMSF).
   b. Algoritmo k-means conceptual difuso con rasgos complejos (FCKMCF).

v. Una función para evaluar la calidad de los conceptos difusos.

Publicaciones

De este trabajo de investigación doctoral se derivaron las siguientes publicaciones:

i. Ayaquica-Martínez I. O., Martínez-Trinidad J. F. and Carrasco-Ochoa J. A. "*Conceptual k-means algorithm with similarity functions*". Proceedings of X Iberoamerican Congress on Pattern Recognition, LNCS 3773, Springer-Verlag. Havana, Cuba, pp. 368-376, 2005.

ii. Ayaquica-Martínez I. O., Martínez-Trinidad J. F. and Carrasco-Ochoa J. A. "*Conceptual K-Means Algorithm based on Complex Features*". XI Iberoamerican Congress on Pattern Recognition, LNCS 4225, Springer-Verlag, pp. 491-501. Cancún, Mexico, 2006.

Trabajo Futuro

En la fase de caracterización de los algoritmos CKMSF y FCKMSF, para evaluar si un predicado cubre a un objeto se utiliza como función de comparación la coincidencia total, en lugar de utilizar la función de comparación definida para cada atributo; lo cual hace que los conceptos obtenidos no caractericen adecuadamente a los agrupamientos. Para resolver este inconveniente se propone adaptar estos algoritmos para que, en la fase de caracterización, puedan utilizar las mismas funciones de comparación utilizadas en la fase de agrupamiento.

En la fase de caracterización del algoritmo FCKMSF las condiciones α-discriminante y β-caracterizante se verifican de la misma manera que en el caso duro, es decir, no se toman

en cuenta los grados de pertenencia de los objetos ni el valor de μ_P de los predicados, lo cual origina que los predicados cubran mayor número de objetos con grado de pertenencia cercano a 0, lo que reduce la calidad de los conceptos. Por esta razón, se propone generalizar las condiciones α-discriminante y β-caracterizante para tomar en cuenta los grados de pertenencia de los objetos a los agrupamientos así como las pertenencias asignadas por los predicados.

Con base en los resultados experimentales se observó que los algoritmos propuestos obtienen conceptos con buenas calidades; no obstante, éstas aún pueden mejorarse. Por lo cual, se propone buscar nuevas estrategias, en la fase de caracterización, para generar mejores conceptos.

En problemas prácticos se desea obtener conceptos que cubran el mayor número de objetos del agrupamiento y, al mismo tiempo, el menor número de objetos fuera del agrupamiento. Además, se desea que estos conceptos estén formados por el menor número posible de predicados y atributos. La función propuesta para evaluar la calidad de los conceptos toma en cuenta únicamente el número de objetos que son cubiertos por los conceptos, sin tomar en cuenta el tamaño de dichos conceptos. Como trabajo futuro se propone definir una función de calidad que tome en cuenta ambos aspectos.

Referencias

Alba-Cabrera E. (1997), *Nuevas extensiones del concepto de testor para diferentes tipos de funciones de semejanza*. Tesis para obtener el grado de Doctor en Ciencias Matemáticas, ICIMAF, Cuba.

Aldenderfer M., Blashfield R. (1984), *Cluster Analysis*, Sage Publications, USA.

Ayaquica-Martínez I.O. (2002), *Algoritmo C-means Difuso usando Funciones de Disimilaridad*. Tesis para obtener el grado de Maestro en Ciencias de la Computación, CIC, IPN, México.

Ayaquica-Martínez I.O., Martínez-Trinidad J.F. (2001), *Fuzzy c-means algorithm to analyze mixed data*. VI Iber-american Symposium on Pattern Recognition. Florianopolis, Brazil, pp. 27-33.

Béjar J., Cortés U. (1992), *LINNEO+: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados*. En las memorias del 3er. Congreso Iberoamericano de Inteligencia Artificial. La Habana Cuba, pp. 471-481.

Bellman R. E., Kalaba R., Zadeh L. A. (1966). *Abstraction and Pattern Classification*. J. Math. Anal. Appl. 13, pp. 1-7.

Bezdek J. C. (1973), *Fuzzy Mathematics in Pattern Classification*, PhD thesis, Cornell University, Ithaca, New York.

Bezdek J. C. (1974), *Cluster Validity with fuzzy sets*. J. Cybernetics 3, pp. 58-73.

Bezdek J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

Bezdek J. C. (1992), *Fuzzy Models for Pattern Recognition, Methods that search for structures in data*. IEEE Press.

Bezdek J. C., Pal N. R. (1998), *Some new indexes of cluster validity*. IEEE Trans. Syst., Man. Cybernetics – Part B. Cybernetics 28 (3), pp. 301-315.

Bezdek J. C., Li W. Q., Attikiouzel Y. A., Windham M. P. (1997), *A geometric approach to cluster validity for normal mixtures*. Soft Computing 1, pp. 166-179.

Blake C., Keogh E., Merz C.J. (1998), *UCI Repository of Machine Learning Databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine CA: University of California, Department of Information and Computer Science.

Briscoe G., Caelli T. (1996), *A compendium of Machine Learning, Volume 1: Symbolic Machine Learning*, Ed. Ablex.

Cimiano P., Hotho A., Stumme G., Tane J. (2004), *Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies*, ICFCA 2004, LNAI 2961, pp. 189-207.

Dae-Won K., Kwang H. L., Doheon L. (2003), *Fuzzy cluster validation index based on inter-cluster proximity*. Pattern Recognition Letters 24, pp. 2561-2574.

Dave R. N. (1991), *New measures for evaluating fuzzy partitions induced through c-shells clustering*. Proc. SPIE Conf. Intell. Robot Computer Vision X, vol. 1670, Boston, pp. 406-414.

De-la-Vega-Doria L.A. (1994), *Extensión al caso difuso del algoritmo de clasificación Kora-3*. Tesis para obtener el grado de Maestro en Ciencias en especialidad en Ingeniería Eléctrica, CINVESTAV, México.

De-la-Vega-Doria L.A., Carrasco-Ochoa J.A., Ruiz-Shulcloper J. (1998), *Fuzzy Kora-Ω Algorithm*, 6th European Congress on Intelligent Techniques and Soft Computing EUFIT 98, Aachen Germany, vol. 2 pp. 1190-1194.

Dunn J. (1974), *Well-separated clusters and optimal fuzzy partitions*. J. Cybernetics 4, pp. 95-104.

El-Sonbaty Y., Ismail M.A. (1998), *Fuzzy clustering for symbolic data*, IEEE transactions on fuzzy systems 6, pp. 195-204.

Escudero L. F. (1977), *Reconocimiento de Patrones*, Paraninfo, Madrid.

Feigenbaum E.A. (1963), *The simulation of verbal learning behavior*. In E.A. Feigenbaum and J. Feldman editors. Computers and Thought McGraw Hill, New York.

Fisher D. (1990), *Knowledge acquisition via incremental conceptual clustering*. Shavlik and Dietterich editors. Readings in Machine Learning, pp. 267-283.

Flores-Sintas A., Cadenas J. M., Martin F. (2000), *Partition validity and defuzzification*. Fuzzy sets and systems 112, pp. 433-447.

Fu K. S. (1974), *Syntactic Method in Pattern Recognition*, Academic Press, New York.

Fukunaga K. (1990), *Introduction to Statistical Pattern Recognition*, Academic Press, London.

García-Serrano J.R., Martínez-Trinidad J.F. (1999), *Extension to c-means algorithm for the use of similarity functions*. 3rdEuropean Conference on Principles of Data Mining and Knowledge Discovery Proceedings. Prague, Czech. Republic, pp 354-359.

Gennari J.H., Langley P., Fisher D. (1990), *Model of incremental concept formation*. In J. Cabonell. MIT/Elsevier Machine Learning, paradigms and methods, pp. 11-61.

Gowda K. Ch., Diday E. (1991), *Symbolic clustering using a new dissimilarity measure*, Pattern Recognition 24, pp. 567-578.

Gowda K.Ch., Diday E. (1992), *Symbolic clustering using a new similarity measure*, IEEE Trans. on System Man Cybernetic 22, pp. 368-378.

Grimaldi R. P. (1998), *Matemáticas Discretas y Combinatoria. Una introducción con Aplicaciones*, 3ª. Edición, Prentice Hall.

Guevara-Cruz M. E. (2004), *Genetic Algorithm for feature selection and informational weight computation using the fuzzy FS testor concept*. Tesis para obtener el grado de Maestro en Ciencias de la Computación, Facultad de Computación, BUAP, México.

Hanson S.J. (1990), *Conceptual clustering and categorization: bridging the gap between induction and causal models*. In Y. Kodratoff and R.S. Michalski, editors. Machine Learning: an artificial intelligence approach, vol. 3, Morgan Kaufmann, Los Altos CA, pp. 235-268.

Hathaway R.J., Bezdek J.C. (2003), *Visual cluster validity for prototype generator clustering models*. Pattern Recognition Letters 24, pp. 1563-1569.

Hathaway R.J., Bezdek J.C., Pedrycz W. (1996), *A parametric model for fusing heterogeneous fuzzy data*, IEEE Trans. on Fuzzy Systems 4 (3), pp. 270-281.

Jänichen S., Perner P. (2005), *Acquisition of Concept Descriptions by Conceptual Clustering*. In P. Perner and A. Imiya (Eds.): MLDM 2005, LNAI 3587, pp. 153-162.

Jonyer I., Holder L.B., Cook D.J. (2001), *Graph-based hierarchical conceptual clustering*. International Journal on Artificial Intelligence Tools, 10 (1-2), pp. 107-135.

Kwon S. H. (1998), *Cluster validity index for fuzzy clustering*. Electron. Lett. 34 (22), pp. 2176-2177.

Lazo-Cortés M., Ruiz-Shulcloper J., Alba-Cabrera E. (2001), *An overview of the evolution of the concept of testor*. Pattern Recognition 34, pp. 753-762.

Lebowitz M. (1986), *Concept learning in a rich input domain: Generalization based memory.* In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: an artificial intelligence approach, vol.2, Morgan Kaufmann, Los Altos, CA, pp. 193-214.

Mali K., Mitra S. (2003), *Clustering and its validation in a symbolic framework*. Pattern Recognition Letters 24, pp. 2367-2376.

Martínez-Trinidad J.F. (2000), *Herramientas para la Estructuración Conceptual de Espacios*. Tesis para obtener el grado de Doctor en Ciencias de la Computación, CIC, IPN, México.

Martínez-Trinidad J.F., Guzmán-Arenas A. (2001), *The logical combinatorial approach to pattern recognition an overview through selected works*. Pattern Recognition 34/4, pp. 1-11.

Martínez-Trinidad J.F., Ruiz-Shulcloper J. (1998), *Fuzzy LC conceptual algorithm*. In proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing. Aache, Germany, pp. 20-24.

Martínez-Trinidad J.F., Ruiz-Shulcloper J. (1999), *Algoritmo LC conceptual duro*. IV Iberoamerican Symposium on Pattern Recognition. Havana, Cuba, pp. 195-206.

Martínez-Trinidad J.F., Sánchez-Díaz G. (2001), *LC a conceptual clustering algorithm*. International Workshop on Machine Learning and Data Mining in Pattern Recognition. Leipzig, Germany, pp. 117-127.

Martinez-Trinidad J. F., Sánchez-Díaz G., Rugerio B. (2002), *Genetic algorithm to compute fuzzy FS-Testors*, WSEAS Transactions on Systems 1/1 pp. 267-272.

McKusick K., Thompson K. (1990), *Cobweb/3: A portable implementation*. Technical report FIA-90-6-18-2, NASA Ames Research Center.

Michalski R.S. (1980), *Knowledge adquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts,* (special issue on knowledge acquisition and induction). Policy Analysis and Information Systems 3, pp. 219-244.

Michalski R.S. (1983), *Automated construction of classifications: conceptual clustering versus numerical taxonomy*. IEEE transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5 4.

Michalski R.S. (1986), *A theory and methodology of inductive learning*. In R.S. Michalski, J. G. Carbonell and T. M. Mitchell, editors. Machine Learning: An artificial intelligence approach, volume 2, Morgan Kaufmann, Los Altos, CA, pp. 83-129.

Michalski R.S., Diday E. (1981), *A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts*. Progress in Pattern Recognition L.N. Kanal and A. Rosenfeld. North Holland Publishing Company, pp. 33-56.

Michalski R.S., Stepp R.E. (1983), *Learning from observation: Conceptual clustering*. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: An artificial intelligence approach 1, pp. 331-363.

Miin-Shen Y., Pei-Yuan H., De-Hua Ch. (2004), *Fuzzy clustering algorithms for mixed feature variables*, Fuzzy Sets and Systems 141, pp. 301-317.

Mishra N., Ron D., Swaminathan R. (2004), *A new conceptual clustering framework*. Machine Learning 56, pp. 115-151.

Negoita, C. V. (1973), *On the application of the fuzzy sets separation theorem for automatic classification in information retrieval systems*. Information Science 5, pp. 279-286.

Osinski S., Weiss D. (2004), *Conceptual clustering using lingo algorithm: Evaluation on open directory project data*, Advanced in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference, Zakopane, Poland, pp. 369-378.

Pal N. R., Bezdek J. C. (1995), *On cluster validity for the fuzzy c-means model*. IEEE Trans. Fuzzy Syst. 3 (3), pp. 370-379.

Pal S. K., Mitra S. (1999), *Neuro-fuzzy Pattern Recognition, Methods in Soft Computing*, John Wiley & Sons, Inc. USA.

Pal S. K., Wang P. P. (1996), *Genetic Algorithms for Pattern Recognition*, CRC Press. USA.

Pons-Porrata A. (1999), *RGC: Un nuevo algoritmo de caracterización conceptual*. Tesis para obtener el grado de Maestro en Ciencias de la Computación, Universidad de Oriente, Cuba.

Pons-Porrata A., Ruiz-Shulcloper J., Martínez-Trinidad J.F. (2002), *RGC: a new conceptual clustering algorithm for mixed incomplete data sets*. In Mathematical and Computer Modelling 36, pp. 1375-1385.

Quan T. T., Hiu S. C., Cao T. H. (2004), *A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data*, CLA 2004, pp. 1-12.

Quan T. T., Hui S. C. Cao T. H. (2004). *FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web*. In Proceedings of the Knowledge Discovery and Ontologies Workshop, Pisa, Italy.

Ralambondrainy H. (1995), *A conceptual version of the K-means algorithm*. Pattern Recognition Letters 16, pp. 1147-1157.

Ravi T.V., Gowda K.Ch. (1999), *An ISODATA clustering procedure for symbolic objects using a distributed genetic algorithm*, Pattern Recognition Letters 20, pp. 659-666.

Rezaee M. R., Lelieveldt B. P. F., Reiber J. H. C. (1998), *A new cluster validity index for the fuzzy c-mean*. Pattern Recognition Letters 19, pp. 237-246.

Ruiz-Shulcloper J., Guzmán-Arenas A., Martínez-Trinidad J.F. (1999), *Enfoque Lógico Combinatorio al Reconocimiento de Patrones*. Editorial Politécnica.

Ruspini E. H. (1969), *A new approach to clustering*, Information and Control 15, pp. 22-32.

Ruspini E. H. (1973), *New experimental results in fuzzy clustering*. Information Science 6, pp. 273-284.

Santos-Gordillo J. A. (2003), *Algoritmo genético para el cálculo de Φ-testores difusos*. Tesis para obtener el grado de Maestro en Ciencias en la especialidad de Ciencias Computacionales, INAOE, México.

Santos-Gordillo J. A., Carrasco-Ochoa J. A., Martínez-Trinidad J. F. (2003), Computing Fuzzy Φ-Testors using a genetic algorithm, WSEAS Transactions on Systems 4/2 pp. 1068-1072.

Schalkoff R.J. (1992), *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley & Sons, Inc.

Seeman W. D., Michalski R. S. (2006), *The CLUSTER/3 system for goal-oriented conceptual clustering: method and preliminary results*. Proceedings of The Data Mining and Information Engineering 2006 Conference, Prague, Czech Republic, vol. 37, pp. 81-90.

Stepp R.E., Michalski R.S. (1986), *Conceptual clustering: inventing goal oriented classifications of structured objects*. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: an artificial intelligence approach, vol.2, Morgan Kaufmann, Los Altos, CA, pp. 471-498.

Stumme G. (2002), *Efficient Data Mining based on Formal Concept Analysis*, DEXA 2002, LNCS 2453, pp. 534-546.

Stumme G., Taouil R., Bastide Y., Lakhal L. (2001), *Conceptual Clustering with Iceberg Concept Lattices*, In R. Klinkenberg, S. Ruping, A. Fick, N. Henze, C. Herzog, R. Molitor, O. Schroder, editors. Proc. GI-Fachgruppentre#en Maschinelles Lernen '01,Universitat Dortmund 763.

Valtchev P., Missaoui R., Godin R. (2004), *Formal Concept Analysis for Knowledge Discovery and Data Mining: the new challenges*, ICFCA 2004, LNAI 2961, pp. 352-371.

Xie X. L., Beni G. (1991), *A validity measure for fuzzy clustering*. IEEE Trans. Pattern Anal. Mach. Intelligence 13 (8), pp. 841-847.

Zadeh L. A. (1965), *Fuzzy sets*. In Information and Control 8 (3), pp. 338-353.