Algoritmos de Agrupamiento Global para Datos Mezclados

por

Saúl López Escobar

Tesis sometida como requisito parcial para obtener el grado de **Maestro en Ciencias** en la especialidad de **Ciencias Computacionales** en el Instituto Nacional de Astrofísica, Óptica y Electrónica

Supervisada por:

Dr. Jesús Ariel Carrasco Ochoa
Dr. José Francisco Martínez Trinidad

Sta. Ma. Tonantzintla, Puebla

Febrero 2007

Resumen

El agrupamiento es un problema que se presenta en una gran cantidad de aplicaciones prácticas en varios campos tales como Reconocimiento de Patrones, Aprendizaje Automático, Minería de Datos, Procesamiento Digital de Imágenes, etc. El algoritmo k-Means es uno de los algoritmos más frecuentemente usados para resolver el problema de agrupamiento, debido principalmente a su simplicidad, pero tiene varias desventajas entre las que se tienen: i) sólo permite trabajar con datos exclusivamente numéricos y ii) depende fuertemente de las condiciones iniciales con las que sea ejecutado.

Por otro lado, se tiene que en ciencias denominadas "suaves" (soft sciences) tales como Medicina, Geología, Sociología, Mercadotecnia, etc. es común que los datos se encuentren descritos por medio de atributos numéricos y no numéricos (datos mezclados) simultáneamente.

Dentro de este contexto, en este trabajo se proponen dos algoritmos de agrupamiento restringido basados en el algoritmo k-Means. Ambos algoritmos permiten trabajar con datos mezclados y no dependen de las condiciones iniciales con las que sean ejecutados. Los algoritmos propuestos son evaluados usando conjuntos de datos obtenidos de un repositorio público y son comparados contra otros algoritmos de agrupamiento restringido.

Abstract

Clustering problem arises in many practical applications in several areas such as Pattern Recognition, Machine Learning, Data Mining, Digital Image Processing, etc. The k-means algorithm is one of the most frequently algorithms used to solve the clustering problem, this is due its simplicity but, it has many drawbacks such as: i) it only allows working with numeric data and ii) it heavily depends on the initial conditions.

On the other hand, in soft sciences such as Medicine, Geology, Sociology, Marketing, etc, it is common that objects are described in terms of numeric and no numeric features (mixed data).

In this context, we propose two clustering algorithms based in the k-Means algorithm. Both algorithms allow working with mixed data and they don't depend on the initial conditions. The proposed algorithms are tested with data sets obtained from one public repository and they are compared against other clustering algorithms.

Agradecimientos

Deseo expresar mi más sincero agradecimiento a mis asesores de tesis los Drs. **Jesús Ariel Carraso Ochoa** y **José Francisco Martínez Trinidad** por su apoyo y orientación constante durante mi estancia en el INAOE, pero sobre todo, por su amistad invaluable.

Al Consejo Nacional de Ciencia y Tecnología **CONACYT** por su apoyo financiero mediante la Beca No. 189901.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica **INAOE** por las facilidades prestadas tanto en aspectos de investigación como en administrativos, en especial a la Coordinación de Ciencias Computacionales, cuyos investigadores siempre mostraron su apoyo.

A todos los miembros de **mi familia**, ya que sin su apoyo y cariño no habría sido posible llevar a féliz término la elaboración del presente trabajo.

También agradezco a **mis compañeros de generación** por la amistad y apoyo que me otorgaron durante mi estancia en el Instituto.

Dedicatoria

A mis padres quienes me dieron la vida,
por sus sacrificios, buenas enseñanzas,
orientación y ejemplo que junto con su amor
me han brindado ya ...
toda una vida.

A mis hermanos,
por el apoyo que siempre me han brindado
con su impulso, cariño y alegría.

A todos mis amigos,
en especial a Isabel por su inmenso cariño.

A mis abuelitos,
por sus bendiciones y ser un buen ejemplo para mi.

Índice general

Lista de Figuras

Lista de Tablas

Capítulo 1

Introdución

En este capítulo se presenta la descripción del problema, se muestra el contexto en el que se desarrolla la presente tesis y se dan los objetivos tanto general como específicos. Asimismo, se explica de manera breve el contenido de cada uno de los capítulos que conforman la presente tesis.

1.1. Descripción del Problema

El problema de agrupamiento básicamente consiste en dado un conjunto de datos, dividirlo en dos o más grupos. Este problema se presenta en una gran cantidad de aplicaciones prácticas y ha sido ampliamente estudiado en varios campos tales como Reconocimiento de Patrones, Aprendizaje Automático, Minería de Datos, Procesamiento Digital de Imágenes, etc. Existen dos variantes del agrupamiento: en la primera se conoce el número de agrupamientos a formar y se le llama *agrupamiento restringido* y, en la segunda no se conoce el número de agrupamientos a formar y se le denomina *agrupamiento libre*.

Para resolver el problema de agrupamiento restringido se han desarrollado diferentes algoritmos, siendo el más popular de ellos el algoritmo k-Means [1] del cual existe una gran cantidad de variantes. El algoritmo k-Means usa distancia Euclidiana para comparar objetos, promedios para actualizar los centros de los agrupamientos y su objetivo es minimizar una función objetivo. Básicamente consiste en los siguientes pasos:

1. Seleccionar aleatoriamente los centros iniciales.

2. Cada objeto es asignado al agrupamiento cuya distancia de su centro al objeto es mínima.
3. Recalcular los centros.
4. Repetir los pasos 2 y 3 hasta que no haya cambios en los centros.

Este algoritmo goza de enorme popularidad, lo que se debe principalmente a que es simple, fácil de entender y fácil de programar. Pero cuenta con alguna desventajas entre las que se tienen:

- Depende en gran medida de las condiciones iniciales con las que sea ejecutado, es decir, depende de la selección de los centros iniciales, razón por la cual únicamente puede obtener soluciones locales y
- únicamente puede trabajar en espacios continuos, es decir, en problemas en donde los objetos se encuentren descritos exclusivamente por medio de atributos numéricos. Esto de debe principalmente al uso de promedios en la actualización de los centros.

El problema de la dependencia de las condiciones iniciales ha sido resuelto de dos maneras distintas. La primera consiste en realizar una búsqueda de *mejores* centros iniciales que permitan obtener una *mejor* solución local [2, 3, 4] y la segunda consiste en realizar una búsqueda global [5, 6]. Aunque estas soluciones siguen teniendo la desventaja de que únicamente pueden trabajar con datos numéricos.

Por otro lado, se han propuesto distintos algoritmos basados en el k-Means que permiten trabajar con datos mezclados [7, 8, 9, 10], que de igual manera, heredan la desventaja de la dependencia de las condiciones iniciales.

1.2. Objetivos

En esta sección se presenta el objetivo particular y los objetivos específicos de la presente tesis, los cuales son descritos a continuación.

1.2.1. Objetivo General

Proponer algoritmos de agrupamiento restringido que permitan encontrar una solución global en problemas donde los objetos se encuentren descritos por medio de atributos numéricos y no numéricos simultáneamente (datos mezclados).

1.2.2. Objetivos Específicos

- Proponer un algoritmo de agrupamiento restringido basado en el algoritmo k-Means que no dependa de las condiciones iniciales, obtenga una solución global y al mismo tiempo permita trabajar con datos mezclados.
- Proponer una versión rápida del algoritmo anterior, de modo que se reduzca considerablemente el tiempo de ejecución sin afectar demasiado la calidad de la solución. Lo que permita aplicarlo a conjuntos de datos más grandes.

1.3. Organización de la Tesis

Esta tesis se encuentra organizada de la siguiente manera:

En el capítulo 2 se exponen los conceptos básicos del Reconocimiento de Patrones así como los principales enfoques para atacar y resolver problemas. De igual manera se describen brevemente los cuatro problemas principales que se presentan en el Reconocimiento de Patrones. En el capítulo 3 se presenta una breve revisión sobre los trabajos relacionados que buscan resolver las desventajas del algoritmo k-Means. En el capítulo 4 se introduce un nuevo algoritmo que soluciona la dependencia de las condiciones iniciales del algoritmo k-Means con Funciones de Similaridad. Se presentan algunos resultados al aplicar el algoritmo propuesto en varios conjuntos de datos de datos, mismos que son comparados contra los obtenidos al emplear el algoritmo k-Means con Funciones de Similaridad y k-Prototypes. En el capítulo 5 se presenta un algoritmo de agrupamiento restringido que reduce el costo computacional del algoritmo propuesto en el capítulo 4 sin sacrificar demasiado la calidad de los agrupamientos. Ambos algoritmos son aplicados a varios conjuntos de datos y son comparados entre sí, así como con el algoritmo k-Means con Funciones de Similaridad y el algoritmo k-Prototypes. Por último, en el capítulo 6 se exponen las conclusiones y el trabajo futuro en esta línea de investigación.

Capítulo 2

Conceptos Básicos

En este capítulo se presentan, de manera breve, los conceptos importantes referentes al entorno correspondiente al contenido de este trabajo.
Se da una breve introducción acerca del Reconocimiento de Patrones, así como algunos enfoques para resolver los problemas de los que se encarga. Asimismo, se ubica el presente trabajo dentro de la Teoría de Reconocimiento de Patrones precisando el enfoque y problema específico sobre el cual se trabajará.

2.1. Introducción

Antes de dar una definición del Reconocimiento de Patrones (RP) se proporcionan los conceptos básicos que constituyen la base sobre la cual se realiza el estudio de los problemas de esta área.

Objeto: Concepto utilizado para representar los elementos sujetos a estudio. Los objetos pueden ser:

- *Físicos:* fotografías, pacientes, zonas geológicas, señales acústicas, etc.
- *Abstractos:* n-uplos de un cierto producto cartesiano de conjuntos de cualquier naturaleza: duros, difusos, o de cualquier otra teoría de conjuntos.

Patrón: Sinónimo de objeto, en ocasiones se establece una diferencia

entre un objeto a clasificar y uno ya clasificado, en el presente trabajo a éste último se le denominará patrón.

Clase: Conjunto de objetos.

Atributos: También se les conoce como rasgos y son las propiedades, factores o rasgos de un objeto dado. Los atributos son los medios para trabajar con los objetos y pueden ser clasificados de la siguiente manera:

$$\text{Atributos} = \begin{cases} \text{No numéricos} & \begin{cases} \text{Ordinales} \\ \text{Nominales} \\ \text{Intervalo} \end{cases} \\ \text{Numéricos} & \begin{cases} \text{Discretos} \\ \text{Continuos} \end{cases} \end{cases}$$

- **Atributos no numéricos:** También se denominan *categóricos*. Están limitados a un número finito de posibles valores los cuales indican categorías, etiquetas alfanuméricas o nombres y a su vez se clasifican en:
  - *Nominales:* son atributos categóricos los cuales no tienen algún orden *natural*, no tienen significado numérico y sus posibles valores son excluyentes entre sí. Por ejemplo: estado civil (soltero, casado, divorciado, viudo), etc.
  - *Ordinales:* son similares a los atributos nominales, excepto que éstos si tienen un orden. Por ejemplo: calidad (pobre, media, buena, excelente), una escala (0, 1, 2, 3, 4 y 5), etc.
  - *De intervalo:* surgen de discretizar atributos numéricos, sus posibles valores son etiquetas que representan intervalos de escalas continuas. Por ejemplo: categorías basadas en pesos (0-2 gm, 2-10 gm y $>$ 10 gm), etc.
- **Atributos numéricos:** únicamente toman valores numéricos. Las operaciones aritméticas sólamente pueden ser aplicadas sobre este tipo de atributos que a su vez se clasifican en:
  - *Discretos:* pueden tomar sólo ciertos valores en una es-

cala, por lo regular son números enteros. Por ejemplo: edad, número de habitantes en una casa, etc.

- *Continuos:* pueden tomar un número infinito de valores reales. Por lo regular son obtenidos por medio de mediciones, por lo que están sujetos a la precisión de los instrumentos de medición. Por ejemplo: distancia, temperatura, altura de una persona, etc.

Clasificar: Asignar un objeto a una o varias clases.

Reconocimiento: Proceso por medio del cual se pueden determinar las relaciones de pertenencia entre un elemento cualquiera y un conjunto de clases, o la formación de esos conjuntos a partir de las relaciones entre los objetos y además el detectar los atributos en términos de los cuales se estudiarán los objetos y su importancia en dicho estudio.

Con base en los conceptos anteriores en [11] se define al Reconocimiento de Patrones como:

> *La ciencia de carácter multidisciplinario que se ocupa de los procesos sobre ingeniería, computación y matemáticas, relacionados con objetos físicos o abstractos, con el propósito de extraer - mediante dispositivos computacionales y/o el hombre - la información que le permita establecer propiedades y/o vínculos de o entre conjuntos de dichos objetos.*

En términos generales se puede definir al Reconocimiento de Patrones como la ciencia de carácter multidisciplinario que busca categorizar objetos [12].

Durante años se ha intentado que las computadoras reproduzcan habilidades propias del ser humano tales como: reconocer rostros, conversar, leer documentos, etc. Dichas habilidades involucran procesos complejos que implican Reconocimiento de Patrones. Por lo que es necesario contar con técnicas robustas y eficientes para atacar estos problemas.

El RP puede ser aplicado en un gran número de áreas tales como Medicina, Geología, Sociología, etc. Es difícil encontrar un área, tanto teórica como práctica en donde no sea posible aplicar herramientas de RP.

Por problemas de Reconocimiento de Patrones se entiende todos aquellos relacionados con la clasificación de objetos y fenómenos así como con la determinación de los factores que inciden en los mismos.

2.2. Enfoques del Reconocimiento de Patrones

Dentro del Reconocimiento de Patrones existen diversos enfoques para atacar y resolver problemas, entre ellos se puede mencionar el Estadístico, el Sintáctico Estructural, el Neuronal y el Lógico Combinatorio [11, 12, 13, 14]. Estos enfoques no son necesariamente independientes y comparten características y objetivos comunes.

2.2.1. Enfoque Estadístico

Este enfoque se basa en la teoría de probabilidad y estadística, supone que se tiene un conjunto de medidas numéricas con distribuciones de probabilidad conocidas y a partir de ellas se hace el reconocimiento. En este enfoque, los objetos están descritos en términos de mediciones, es decir, atributos numéricos a los que se les presuponen propiedades tales como las de estar definidas sobre un espacio métrico o normado. Se ha abusado de este enfoque y no siempre las suposiciones concuerdan con la realidad, sino que como resulta conveniente hacerlas desde el punto de vista que simplifican el análisis matemático, se hacen, aunque éstas no sean correctas.

2.2.2. Enfoque Sintáctico Estructural

Desarrollado a partir de la Teoría de los Lenguajes Formales. En este enfoque se supone que los objetos están compuestos de elementos simples que a su vez pueden estar constituidos de elementos más simples y así sucesivamente hasta encontrar elementos primarios atómicos. Por ejemplo, a cada conjunto de objetos se puede asociar una gramática que genera sólo elementos de dicho conjunto y el problema consiste en encontrar cuál de las gramáticas genera la estructura correspondiente al objeto a clasificar. Algunas características de este enfoque son: está basado en las descripciones de los objetos en términos de sus partes constitutivas; se apoya en la Teoría de los Lenguajes Formales, la Teoría de Autómatas, las Funciones Recursivas; Teoría de Grafos y en particular en el estudio relacional entre partes constitutivas

de los objetos; con frecuencia emplea medidas de similaridad estructural y se asume que la estructura de los objetos a reconocer es cuantificable.

2.2.3. Enfoque Neuronal

Las Redes Neuronales Artificiales (RNA) intentan modelar la forma en que los sistemas neuronales biológicos almacenan y procesan la información. Este enfoque supone que tiene una estructura de neuronas interconectadas que operan de forma paralela estimulándose unas a otras. Las RNA pueden ser entrenadas para dar una cierta respuesta cuando se le presentan determinados valores numéricos en sus entradas, de este modo es posible dar una respuesta similar cuando se presente una entrada parecida a las que se usaron para entrenarla.

Algunas de las ventajas de las RNA son su capacidad de aprender y generalizar, su adaptabilidad, no linealidad y tolerancia a fallos.

2.2.4. Enfoque Lógico Combinatorio

Este enfoque se fundamenta en la Lógica Matemática, Teoría de Testores, Teoría Clásica de Conjuntos, Teoría de los Subconjuntos Difusos, Teoría Combinatoria y Matemáticas Discretas. La idea básica de este enfoque consiste en suponer que los objetos están descritos por medio de una combinación de atributos numéricos o no numéricos pudiendo además existir ausencia de información. Este tipo de descripciones de objetos son elementos de un producto cartesiano sin alguna propiedad algebraica, lógica o topológica asumida sobre el espacio de representación, por lo que tiene la peculiaridad de ser heterogéneo. Las características de este enfoque permiten trabajar en problemas donde existen atributos numéricos y no numéricos al mismo tiempo (este tipo de problemas es frecuente en ciencias denominadas *suaves* (Soft Sciences) tales como Medicina, Biología, Ciencias Sociales, etc.).

El enfoque Lógico Combinatorio modela los problemas de RP lo más cercano posible a la realidad sin hacer suposiciones que no estén fundamentadas. Por lo que uno de los aspectos esenciales de este enfoque es que las descripciones de los objetos deben ser tratadas cuidadosamente para no realizar operaciones que resulten antinaturales respecto al problema que estén representando. De ahí que en [11] se

diga que: "*El enfoque Lógico Combinatorio es más que un conjunto de técnicas, es una filosofía, una manera de enfrentar los problemas del Reconocimiento de Patrones a partir de una determinada metodología de la modelación matemática de dichos problemas*".

Cabe resaltar que el marco de trabajo sobre el cual se desarrolla esta tesis es el Reconocimiento Lógico Combinatorio de Patrones (RLCP).

2.3. Problemas del Reconocimiento de Patrones

El Reconocimiento de Patrones se ocupa de cuatro problemas principales: clasificación supervisada (con aprendizaje, también reconocimiento de patrones supervisado), clasificación no supervisada (sin aprendizaje, también reconocimiento de patrones no supervisado), clasificación parcialmente supervisada (con aprendizaje parcial, también reconocimiento de patrones parcialmente supervisado) y selección de variables.

Clasificación Supervisada. En este problema se conoce que un conjunto de objetos se encuentra distribuido en un número dado de clases de las cuales se tiene, de cada una, una muestra de objetos que pertenecen a ella. El problema consiste en, dado un nuevo objeto el cual se desea clasificar, establecer las relaciones de éste con cada una de las clases en las que se encuentra dividido el conjunto de datos.

Clasificación No Supervisada. En este tipo de problemas no se conoce cómo se encuentran clasificados los objetos, por lo tanto el objetivo que se persigue es determinar esta clasificación o agrupamiento. Se pueden presentar dos variantes:

- *Clasificación no supervisada restringida*: en esta variante se conoce el número de clases en las que se quiere dividir el conjunto de datos.
- *Clasificación no supervisada libre*: el número de clases en las que se estructurará la muestra se desconoce y depende exclusivamente de los datos.

Clasificación Parcialmente Supervisada. Es una de las familias de problemas menos estudiadas en Reconocimiento de Patrones [11]. El problema es análogo al de clasificación supervisada excepto que hay al menos una clase de objetos de la que no se tiene una muestra y el problema en general sigue siendo el mismo: dado un nuevo objeto, relacionarlo con los ya clasificados. Es una situación en cierto sentido intermedia entre un problema de clasificación no supervisada y uno de clasificación supervisada, ya que si bien es cierto que no se tiene información de al menos una de las clases también lo es que sí se tiene información de otras y esta información debería ser usada para clasificar.

Selección de Variables. Es uno de los problemas más importantes dentro del Reconocimiento de Patrones, en especial dentro de problemas de clasificación supervisada, puesto que la presencia de información redundante o irrelevante dentro de las descripciones de objetos no sólo aumenta el tiempo requerido por los clasificadores ya sea para entrenamiento o clasificación, sino que puede inducir variaciones en los resultados, afectando de este modo la calidad y confiabilidad de la solución [13]. Existen dos razones principales para hacer selección de variables [12]:

- **Para la clasificación:** La selección de atributos relevantes se hace para mejorar la calidad de clasificación y/o aumentar la velocidad de procesamiento.
- **Para la representación:** Decidir cuales atributos representan mejor a cierto tipo de objetos.

En la figura 2.1 se muestra un esquema de los diferentes enfoques y se sombrea el enfoque y los problemas que interesan en este trabajo.

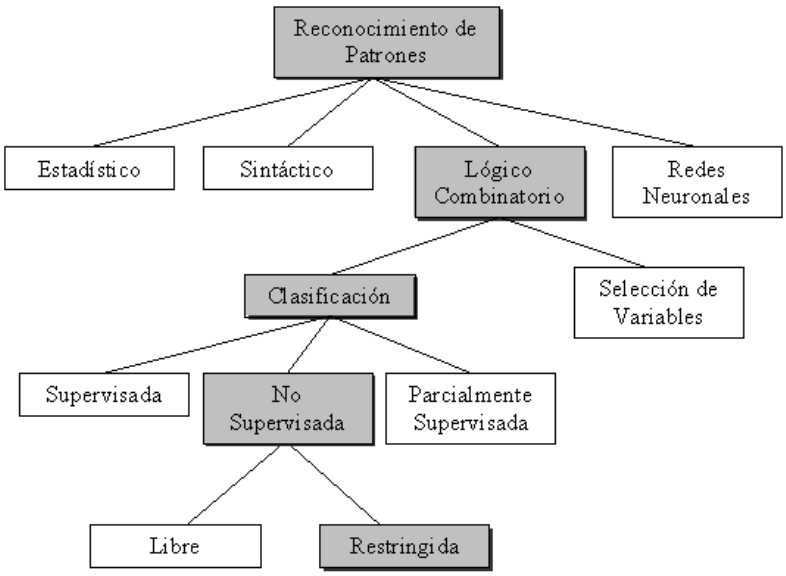


Figura 2.1: Esquema que muestra algunos de los enfoques del Reconocimiento de Patrones y algunos de los problemas abordados por el enfoque Lógico Combinatorio.

2.4. Clasificación No supervisada

2.4.1. Definición Formal de un Problema de Clasificación No supervisada

Sea $X = \{x_1, x_2, \ldots, x_m\}$ un conjunto de datos en donde cada objeto está descrito por un conjunto de atributos $R = \{y_1, \ldots, y_n\}$. Cada atributo toma valores de un conjunto de valores admisibles D_i, $y_i(x_j) \in D_i$, $i = 1, \ldots, n$; estos conjuntos pueden contener subconjuntos de números reales, términos de algún diccionario, proposiciones o predicados de algún lenguaje natural o artificial, etc. Se asume que en cada D_i puede existir un símbolo "?" para denotar ausencia de información en el valor del atributo y_i de un objeto dado. De esta forma, los atributos pueden ser de cualquier naturaleza (no numérico: Booleana, categórica, multivaluada, etc. ó numérico: entero, punto flotante, etc) y se pueden considerar descripciones incompletas.

Se define una función de similaridad $\Gamma : (D_1 \times D_2 \times \ldots \times D_n)^2 \rightarrow L$ la cual permite comparar objetos, donde L es un conjunto totalmente ordenado.

Formalmente, un problema de clasificación no supervisada consiste en, dado un criterio de agrupamiento Π, obtener la estructura interna del conjunto de datos X,

las relaciones entre los objetos y las agrupaciones de los mismos. En este tipo de problemas los tres elementos principales son el espacio de representación inicial (ERI), la función de similaridad Γ y el criterio de agrupamiento Π [14].

2.4.2. Técnicas de Clasificación No Supervisada

Para resolver el problema de clasificación no supervisada se han desarrollado gran cantidad de métodos, los cuales se pueden dividir en los siguientes grandes grupos:

Reagrupamiento

Dado un conjunto de datos de m objetos, un algoritmo de reagrupamiento genera una partición inicial de k agrupamientos y después realiza reorganizaciones sucesivas hasta obtener un agrupamiento que cumpla con un criterio dado [13, 15, 16].

Uno de los principales problemas con estos algoritmos es su alto costo computacional, algunos algoritmos enumeran exhaustivamente todos los posibles agrupamientos para buscar un óptimo global. Incluso para un número pequeño de objetos el número total de posibles particiones es enorme. Ésta es la razón por la que comúnmente se inicia con una partición inicial aleatoria y ésta es refinada posteriormente.

Los algoritmos k-Means [1] y PAM (Partitioning Around Medoids) [17] son ejemplos típicos de este tipo de algoritmos de agrupamiento. Asimismo, el algoritmo k-Means con Funciones de Similaridad [7, 8] es un ejemplo de algoritmos de reagrupamiento que siguen las ideas del Reconocimiento Lógico-Combinatorio de Patrones.

Jerárquicos

Los algoritmos jerárquicos generan una distribución jerárquica de agrupamientos. Su objetivo principal es generar grupos, en donde cada uno de ellos es tratado como si fuése un objeto en un *nuevo* universo y éstos son nuevamente agrupados hasta formar un cierto número de agrupamientos. Pueden ser clasificados en *aglomerativos* o *divisivos* [13, 15, 16].

Aglomerativos. Comienzan considerando cada objeto como un agrupamiento separado, y sucesivamente fusionan agrupamientos simi-

lares hasta llegar al nivel deseado u obtener un sólo agrupamiento con todos los elementos del conjunto de datos.

Divisivos. Siguen una estrategia opuesta a los algoritmos aglomerativos. Comienzan con un sólo agrupamiento el cual contiene a todos los objetos de la muestra y dividen los agrupamientos hasta llegar al nivel deseado o hasta que cada objeto de la muestra sea un agrupamiento separado.

Los algoritmos *simple link* y *complete link* [18] son ejemplos de este tipo de algoritmos de agrupamiento.

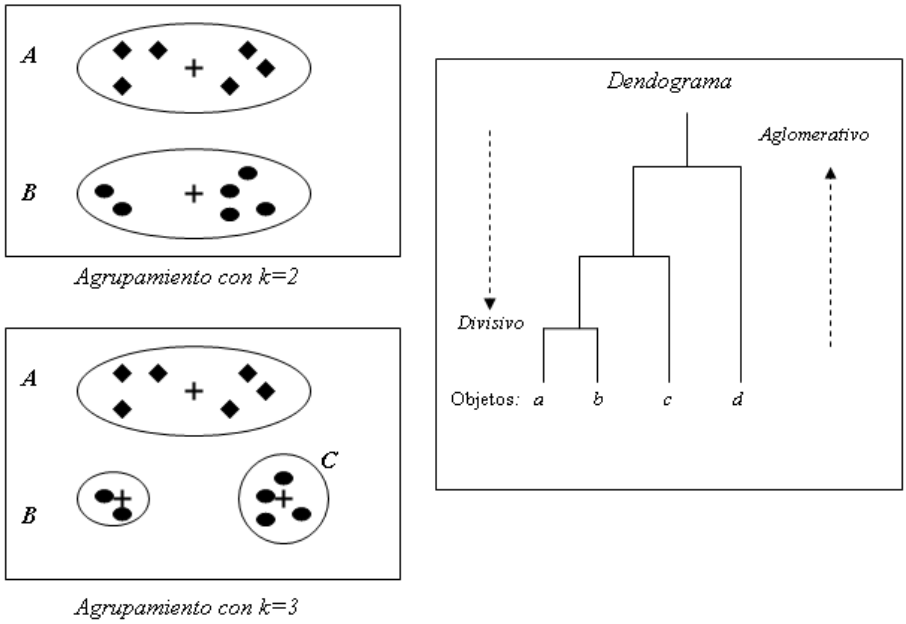
Basados en Teoría de Grafos

Las métodos del Reconocimiento Lógico-Combinatorio de Patrones (RLCP) son alternativas que permiten considerar atributos numéricos y no numéricos simultáneamente, así como usar una función de similaridad, no necesariamente una métrica, en términos de la cual se definen las relaciones entre las diferentes descripciones de los objetos y un cierto criterio de agrupamiento.
Existe un subgrupo importante dentro del RLCP basado en la teoría de grafos, en donde los objetos son tratados como vértices de un grafo y la similaridad entre pares de objetos como aristas. Dichos métodos consideran que se tiene un umbral de semejanza β_0 y que un objeto x_i es β_0-semejante con el objeto x_j si la similaridad entre ambas es mayor o igual a β_0. En este grafo se buscan agrupamientos que cumplan con criterios como conexidad, compacidad, completitud, etc.

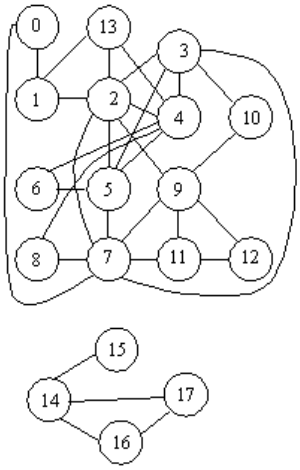
Los algoritmos de Componentes Conexas, Conjuntos Compactos, Conjuntos Fuertemente Compactos [13] son ejemplos de este tipo de algoritmos.

La figura 2.2 muestra ejemplos de las tres principales métodos para resolver el problema de clasificación no supervisada.

La figura 2.3 muestra un esquema que contiene las principales técnicas usadas para resolver el problema de clasificación no supervisada y se sombrea las que interesan en esta tesis.



(a) ***Particional*** *(b)* ***Jerárquico***



(c) ***Basado en Teoría de Grafos. Componentes Conexas***

Figura 2.2: Ejemplos de los 3 principales métodos de clasificación no supervisada, en donde k es el número de agrupamientos

2.5. Funciones de Similaridad

En el enfoque Lógico Combinatorio del Reconocimiento de Patrones es común que los objetos se encuentren descritos en términos de atributos numéricos y no numéricos

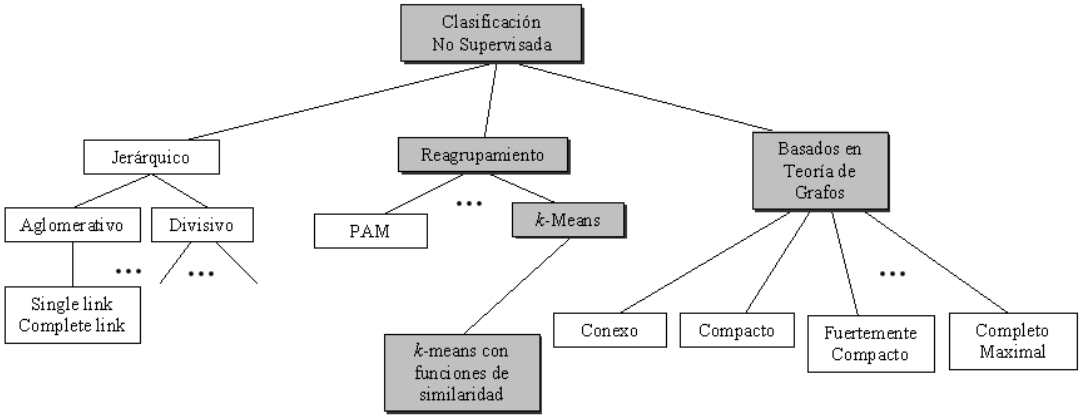


Figura 2.3: Esquema que muestra algunas de las principales estrategias para resolver el problema de clasificación no supervisada

(datos mezclados) pudiendo además existir ausencia de información. Este tipo de descripciones son elementos de un producto cartesiano sin alguna propiedad algebraica, lógica o topológica asumida sobre el espacio de representación, por lo que este espacio tiene la peculiaridad de ser heterogéneo [12]. Además, se usa una *función de similaridad* (o semejanza) la cual permite comparar objetos. En este tipo de problemas resulta de mucha importancia la determinación de dicha función de similaridad.

En [19] se da la siguiente definición de función de similaridad:

> *La función de similaridad es el dispositivo matemático que formaliza un criterio de analogía entre pares de objetos de un universo dado; es la manera en que se expresa, utilizando un lenguaje matemático, el procedimiento que siguen los humanos para comparar un par de objetos de un universo y emitir un criterio acerca de su parecido o similaridad.*

Es claro que este procedimiento es esencial en el proceso de reconocimiento y está siempre colmado de un volumen considerable de subjetivismo [19].

Se denomina *función de similaridad o semejanza* y se denota por Γ a una función que asocia a cada par de descripciones de objetos de un universo dado una magnitud L que evalúa su parecido o semejanza y satisface:

- $\Gamma(x_i, x_j) = \Gamma(x_j, x_i)$
- $\Gamma(x_i, x_i) = \text{máx}(\Gamma(x_i, x_l))$, para $l = 1, \ldots, m$.

2.5.1. Tipos de Funciones de Similaridad

Las funciones de similaridad constituyen abstracciones en el lenguaje matemático de un proceso real que realiza el ser humano al resolver determinado problema relacionado con la obtención de información o conocimiento [19], y pueden ser clasificadas en las siguientes categorias:

Booleanas: poseen un conjunto imagen compuesto por únicamente dos elementos, generalmente 0 y 1, uno de los cuales (por ejemplo el cero) significa que los objetos son diferentes y el otro que son similares.

***k*-valente:** tienen su imagen en el conjunto $E_k = 0, 1, \ldots, k-1$. Los valores 0 y $k-1$ corresponden con los valores de similaridad mínima y máxima respectivamente, el resto de los valores son gradaciones discretas de la similaridad entre un par de objetos.

Reales: La imagen de estas funciones pertenece al conjunto R de los números reales. En ocasiones se realiza una proyección sobre el intervalo [0,1].

Lingüísticas: La imagen de estas funciones constituye un conjunto lingüístico con uno o varios términos, por ejemplo: $\Gamma(x_i, x_j) \in$ {"no semejantes", "poco semejantes", "semejantes", "muy semejantes"}.

2.5.2. Diferencias entre Funciones de Similaridad y Funciones de Distancia

La principal diferencia entre una función de distancia D y una función de similaridad Γ es que D debe satisfacer las siguientes propiedades:

Sea $EspM$ un espacio métrico arbitrario, $x, y, z \in EspM$

1. $D(x, y) \geq 0$, $\forall x, y \in EspM$
2. $D(x, y) = 0$ si y sólo si $x = y$
3. $D(x, y) = D(y, x)$ $\forall x, y \in EspM$ (simetría)
4. $D(x, z) \leq D(x, y) + D(y, z)$ $\forall x, y, z \in EspM$ (desigualdad triangular)

Una función de similaridad puede no cumplir con alguna o algunas de estas propiedades. Por ejemplo, la propiedad de la desigualdad triangular.

Por ejemplo:

Sea $X = \{x_1, x_2, x_3\}$ un conjunto de datos en donde cada objeto está descrito por un conjunto de atributos Booleanos $R = \{y_1, y_2, y_3\}$ denominados $\{sexo, raza, residente - USA\}$. Cada atributo toma valores de un conjunto de valores admisibles D_i, $y_i(x_j) \in D_i$, $i = 1, \ldots, 3$. Los valores admisibles para cada atributo son: $D_1 = \{femenino, masculino\}$, $D_2 = \{blanco, asiático\}$ y $D_3 = \{si, no\}$.

x_1=$\{femenino, blanco, si\}$,
x_2=$\{femenino, asiático, no\}$ y
x_3=$\{masculino, blanco, si\}$.

Si se define la siguiente función de comparación entre atributos:

$$C_p(y_p(x_i), y_p(x_j)) = \begin{cases} 1 & \text{si } y_p(x_i) = y_p(x_j) \\ 0 & \text{en otro caso} \end{cases}$$

y se define la siguiente función de similaridad:

$$\Gamma(x_i, x_j) = C_1(y_1(x_i), y_1(x_j)) + C_2(y_2(x_i), y_2(x_j)) + C_3(y_3(x_i), y_3(x_j))$$

Entonces $\Gamma(x_1, x_2) = 1+0+0 = 1$, $\Gamma(x_1, x_3) = 0+1+1 = 2$ y $\Gamma(x_2, x_3) = 0+0+0 = 0$. Por lo tanto, $\Gamma(x_1, x_3) \nleq \Gamma(x_1, x_2) + \Gamma(x_2, x_3)$, por lo que no se cumple la desigualdad triangular.

Una función de similaridad se ha considerado prácticamente como equivalente a una distancia. Para algunos autores es fácil concebir que si se tiene una función de distancia se puede obtener una función de similaridad por medio del inverso de la distancia, pero existen funciones de similaridad para las cuales no hay funciones de distancia a partir de las cuales puedan ser obtenidas [14].

Capítulo 3

Trabajo Relacionado

En este capítulo se presenta el trabajo relacionado a la línea de investigación seguida en este trabajo.
Se da una breve explicación acerca del algoritmo de agrupamiento restringido k-Means así como de sus principales ventajas y desventajas. Asimismo, se presenta una breve explicación de las extensiones que resuelven las desventajas de dicho algoritmo.

3.1. Introducción

El algoritmo k-Means [1] fué propuesto hace poco más de tres décadas y es uno de los algoritmos de agrupamiento más usados en una amplia variedad de áreas. k-Means es un algoritmo de agrupamiento restringido, por lo que recibe como parámetro el número de agrupamientos a formar y se encuentra definido sobre datos continuos, es decir, únicamente permite trabajar con objetos descritos por medio de un conjunto de atributos numéricos.

Este algoritmo calcula iterativamente los centros de los agrupamientos mientras que al mismo tiempo minimiza una función objetivo. k-Means usa la distancia Euclidiana para comparar objetos y promedios para calcular los centros de los agrupamientos, lo que no le permite trabajar con atributos no numéricos. Frecuentemente es usado para inicializar otros algoritmos de mayor costo computacional [20, 21] (por ejemplo, el algoritmo EM) y básicamente consiste en los siguientes pasos:

1. Seleccionar aleatoriamente los centros iniciales.
2. Asignar cada objeto al agrupamiento cuya distancia con su centro sea mínima.
3. Re-calcular los centros.
4. Repetir los pasos 2 y 3 hasta que no haya cambios en los centros para dos iteraciones consecutivas.

Este algoritmo es simple, fácil de programar, y fácil de entender, pero tiene ciertas desventajas entre las que se encuentran:

- Únicamente puede ser aplicado a conjuntos de datos numéricos.
- No garantiza una solución única, puesto que depende en gran medida de las condiciones iniciales. Por lo que sólamente puede obtener soluciones locales.

En la literatura se han propuesto algoritmos que resuelven las desventajas del k-Means. Algunos de estos algoritmos se enfocan en solucionar la dependencia de las condiciones iniciales, pero no permiten trabajar con datos mezclados; mientras que los restantes resuelven el problema de la aplicabilidad sobre conjuntos de datos mezclados, aunque siguen dependiendo de las condiciones iniciales.

3.2. Algoritmos que Solucionan la Dependencia de las Condiciones Iniciales del *k*-Means

Para resolver la dependencia de la condiciones iniciales del algoritmo k-Means se han propuesto soluciones que siguen distintas estrategias. Algunos algoritmos buscan semillas iniciales que permitan encontrar un mejor mínimo local, mientras que otros realizan una búsqueda global.

A continuación se presentan algunos de los trabajos que se encuentran en la literatura que atacan el problema de la dependencia de las condiciones iniciales del algoritmo k-Means.

3.2.1. Algoritmos de Búsqueda de Semillas Iniciales

En [2] se propone un algoritmo el cual permite refinar los centros iniciales para algoritmos de agrupamiento. Este algoritmo se basa en una técnica de estimación de modas de una distribución y opera sobre un pequeño subconjunto de objetos de la muestra original, por lo que únicamente requiere una pequeña porción del total de la memoria que se necesita para almacenar todo el conjunto de datos, lo que hace a este algoritmo atractivo para conjuntos de datos muy grandes. En [2] sostienen que al inicializar los centros de forma que estos queden cerca de la moda se pueden encontrar agrupamientos correctos más frecuentemente y el algoritmo itera menos veces, por lo que converge más rápido hacia un mejor mínimo local.

El algoritmo **CCIA** [3] (Cluster Center Initialization Algorithm) permite calcular los centros iniciales para el algoritmo k-Means. Este algoritmo está basado en los hechos experimentales de que objetos muy similares forman el núcleo de los agrupamientos, por lo que estos objetos pueden ayudar en la búsqueda de los centros iniciales. El algoritmo genera un número mayor de agrupamientos que el deseado, por lo que agrupamientos similares son unidos usando un método de condensación de datos multiescala. Los centros de estos agrupamientos son usados como los centros iniciales con los que se ejecuta el algoritmo k-Means, lo que permite obtener mejores mínimos locales que los que se pudieran obtener si se ejecuta dicho algoritmo con centros iniciales aleatorios.

En [4] se realiza un estudio comparativo de cuatro métodos de inicialización del algoritmo k-Means. Los métodos de inicialización comparados son: Inicialización Aleatoria, Forgy ([22]), MacQueen [1] y Kaufman [17].

Inicialización Aleatoria. Aleatoriamente divide el conjunto de datos en una partición de k grupos y a partir de ésta genera las semillas con las que se inicializa el algoritmo k-Means.

Forgy. Aleatoriamente elige k objetos del conjunto de datos (semillas) y asigna el resto de los objetos al grupo representado con la semilla más cercana.

MacQueen Elige aleatoriamente k objetos del conjunto de datos. Asigna, siguiendo el orden de los objetos, el resto de los objetos al grupo con el centroide más cercano. Después de cada asignación realiza un recálculo de los centroides.

Kaufman El agrupamiento inicial es obtenido por la selección sucesiva de objetos representativos hasta que k objetos sean seleccionados. El primer objeto representativo es el objeto que se encuentra lo más cerca del centro del conjunto de datos. El resto de los objetos representativos es seleccionado de acuerdo a la regla heurística de elegir las instancias que prometen tener alrededor de ellas el número más grande de objetos.

Existen alguna diferencias interesantes entre los cuatro métodos de inicialización, Kaufman es el único determinístico, el método de Inicialización Aleatoria genera una partición inicial independiente del orden de los objetos y MacQueen genera una partición inicial dependiente del orden de los objetos. Por lo que, existen diferencias obvias entre el costo computacional de los cuatro algoritmos.

En tal estudio comparativo concluyen que Kaufman obtuvo los mejores resultados, ya que hace que el algoritmo k-Means sea más efectivo y un poco más independiente de las condiciones iniciales y del orden de entrada de los objetos. Kaufman induce que el k-Means tenga un mejor comportamiento con respecto a la velocidad de convergencia que el método de inicialización aleatoria. Pero tiene la desventaja de que aún con este método de inicialización el algoritmo k-Means sigue obteniendo una solución local.

3.2.2. Algoritmos de Búsqueda Global

El algoritmo **Global *k*-Means** [5] es un algoritmo de agrupamiento determinístico global el cual minimiza una función objetivo y es completamente independiente de las condiciones iniciales con las que sea ejecutado. Este algoritmo proporciona mejores resultados que el algoritmo k-Means en términos de una función objetivo, lo que se debe principalmente al hecho de que realiza una búsqueda global mientras que el algoritmo k-Means realiza una búsqueda local.

Otra ventaja de este algoritmo es que para resolver el problema de k agrupamientos resuelve todos los problemas intermedios con k' agrupamientos, $k' = 2, \ldots, k-1$, lo que puede ser útil en aplicaciones en donde se busque el número óptimo de grupos a formar para lo cual se prueba con distintos valores de k.

El algoritmo Global k-Means procede de forma incremental y usa al algoritmo k-Means como algoritmo de búsqueda local. Esto es, para resolver el problema con k agrupamientos, resuelve secuencialmente todos los problemas intermedios con

$2, \ldots, k-1$. La idea básica de este algoritmo es que una solución óptima para un problema con k' agrupamientos puede ser obtenida a través de una serie de búsquedas locales (usando el algoritmo k-Means). En cada búsqueda local, los $k'-1$ primeros centros son siempre inicializados en las posiciones óptimas obtenidas al resolver el problema con $k'-1$ agrupamientos. El k'-ésimo centro restante es inicializado en cada objeto de la muestra. Debido a que la posición óptima para $k'=1$ se conoce (el objeto mas cercano al centro del conjunto de datos), el procedimiento descrito puede ser aplicado para resolver todos los problemas de agrupamiento con $k'=2, \ldots, k-1$.

El algoritmo Global k-Means es un algoritmo que resuelve la dependencia de las condiciones iniciales del algoritmo k-Means con la desventaja de que tiene un alto costo computacional, motivo por el cual se han propuesto varias versiones rápidas de este algoritmo [5, 6]. Estas versiones rápidas reducen considerablemente el tiempo de ejecución del algoritmo Global k-Means sin afectar considerablemente la calidad de los resultados. La principal diferencia consiste en que éstos no consideran a todos los objetos del conjunto de datos como posible inserción, para cada nuevo centro solamente usan un pequeño subconjunto de objetos *apropiados* para inicializar el nuevo centro. Dicho subconjunto lo obtienen usando heurísticas tales como particionar el espacio de los datos usando árboles *k-dimensionales* [6], lo que permite reducir el número de objetos que se examinan para un nuevo centro.

Los algoritmos Global k-Means y sus versiones rápidas solucionan la dependencia de las condiciones iniciales del algoritmo k-Means. Asimismo, permiten obtener una solución global pero tienen la desventaja que sólo pueden ser aplicados a conjuntos de datos numéricos, lo que limita su aplicabilidad en problemas con datos mezclados.

3.3. Extensiones del Algoritmo *k*-Means que Permiten Trabajar con Datos Mezclados

El algoritmo k-Means ha mostrado su efectividad en el proceso clasificación no supervisada restringida, bajo la suposición de que los objetos deben estar descritos por atributos que permitan el establecimiento de alguna métrica entre los mismos. Sin embargo, esta suposición no es necesariamente cierta en ciencias denominadas

suaves (Soft Sciences) tales como Medicina, Sociología, Política, etc. En tales ciencias los objetos se encuentran descritos por medio de atributos numéricos, no numéricos y/o ausencia de información, en estos casos no siempre es posible definir una métrica para comparar objetos y sólo el grado de similaridad puede ser determinado.

Dentro de este contexto, se han propuesto los algoritmos k-Means con Funciones de Similaridad [7, 8] y k-Prototypes [9, 10]. Ambos algoritmos realizan agrupamiento restringido y son descritos con mayor detalle a continuación.

3.3.1. Algoritmo k-Means con Funciones de Similaridad

El algoritmo **k-Means con Funciones de Similaridad** [7, 8] es un algoritmo de agrupamiento restringido el cual fué diseñado para ser aplicado en problemas donde no puedan usarse métricas para comparar objetos y sólamente se pueda calcular la similaridad entre objetos. Básicamente constituye una extensión del algoritmo k-Means [1] y sus características lo hacen muy útil en muchos problemas de minería de datos y extracción de conocimiento (Knowledge Discovery).

El algoritmo k-Means con Funciones de Similaridad obtiene agrupamientos con la característica de que los objetos de cada agrupamiento deben ser lo más similares posible entre si, y al mismo tiempo cada agrupamiento debe ser lo menos similar posible de los demás. Esto es, el algoritmo maximiza la similaridad entre objetos que pertenecen a un mismo agrupamiento, y al mismo tiempo minimiza la similaridad entre los diferentes agrupamientos. Este algoritmo usa una función de similaridad en lugar de distancia Euclidiana o cualquier otra métrica definida sobre un espacio continuo, e integra funciones de comparación entre atributos en dicha función de similaridad, lo que lo da una mayor flexibilidad para modelar el problema de forma adecuada.

El algoritmo k-Means con Funciones de Similaridad, a diferencia del algoritmo k-Means, usa objetos del mismo conjunto de datos como *centros* de los agrupamientos y los denomina *objetos representativos*. El objeto representativo de un agrupamiento dado es el objeto que en promedio se parece más a los objetos del mismo agrupamiento y al mismo tiempo, el que menos se parece a los objetos de los demás agrupamientos.

El algoritmo comienza seleccionando aleatoriamente k objetos representativos. Entonces, el resto de los objetos del conjunto de datos son asignados al agrupamiento

cuyo objeto representativo es el más similar al objeto a agrupar. Después de agrupar todos los objetos del conjunto de datos, se actualizan los objetos representativos y el proceso se repite hasta obtener el mismo conjunto de objetos representativos para dos iteraciones consecutivas, o cuando un número máximo de iteraciones sea alcanzado.

Este algoritmo hereda uno de los principales problemas del algoritmo k-Means el cual consiste en una fuerte dependencia de las condiciones iniciales con las que sea ejecutado.

3.3.2. Algoritmo *k*-Prototypes

El algoritmo ***k*-Prototypes** [9, 10] es un algoritmo de agrupamiento restringido que permite agrupar grandes conjuntos de datos mezclados. Este algoritmo básicamente constituye una integración de los algoritmos k-Modes [9] y k-Means [1].

El algoritmo k-Modes fué la primera extensión del algoritmo k-Means orientada al agrupamiento de datos categóricos. Sigue la misma idea que el algoritmo k-Means y la estructura del algoritmo no cambia, siendo la principal diferencia la medida de similitud usada para comparar objetos.

Las principales características de este algoritmo son:

- Usa una medida de disimilaridad para comparar objetos.
- Reemplaza el uso de promedios por el de modas.
- Usa un método basado en frecuencias para actualizar las modas.

El algoritmo k-Modes fué diseñado para agrupar grandes conjuntos de datos categóricos exclusivamente.

El algoritmo k-Prototypes integra al algoritmo k-Means y k-Modes para remover la limitación de poder trabajar únicamente con un sólo tipo de datos. Esto lo hace de la siguiente forma: se asume que s^r es la medida de disimilaridad entre atributos numéricos definida por el cuadrado de la distancia Euclidiana y s^c es la medida de disimilaridad entre atributos categóricos definida por el número de *incoincidencias* (mismatches) de categorías entre objetos. La disimilaridad entre dos objetos se define como:

$$s^r + \gamma s^c$$

en donde γ es un peso usado para equilibrar las dos partes, lo que evita favoritismo entre los dos tipos de atributos. Un pequeño valor de γ indica que el agrupamiento está dominado por los atributos numéricos, mientras que un valor grande implica que los atributos categóricos dominan el agrupamiento.

Los algoritmos k-Prototypes y k-Modes son demasiado *inestables* debido a la no unicidad de las modas, es decir, el resultado depende fuertemente de la selección de las modas durante el proceso de agrupamiento [23]. Por lo que una mala elección de la moda puede llevar a errores en el agrupamiento y considerar todas las modas implica un alto costo computacional. Esto se debe a que un solo valor de un atributo, con la frecuencia más alta, no es suficiente para representar efectivamente la distribución del atributo en el agrupamiento. Asimismo, estos algoritmos heredan uno de los principales problemas del algoritmo k-Means el cual consiste en una fuerte dependencia de las condiciones iniciales.

Capítulo 4

Algoritmo k-Means Global para Datos Mezclados

En este capítulo se presenta un algoritmo de agrupamiento restringido con el que se soluciona la dependencia de las condiciones iniciales del algoritmo k-Means con Funciones de Similaridad. El algoritmo propuesto busca una solución global y no depende de algún otro parámetro externo, dicho algoritmo es probado con conjuntos de datos obtenidos de un repositorio público y se compara contra otros algoritmos de agrupamiento restringido.

4.1. Introducción

El algoritmo k-Means con Funciones de Similaridad [7, 8] es un algoritmo que soluciona una de las principales desventajas del algoritmo k-Means, ya que permite trabajar con datos mezclados (objetos descritos por medio de atributos numéricos y no numéricos) pero, depende en gran medida de las condiciones iniciales con las que sea ejecutado, por lo que el resultado obtenido es una solución local. Por tal motivo en esta sección se propone el algoritmo k-Means Global para Datos Mezclados, el cual también permite trabajar con datos mezclados y/o ausencia de información, y no depende de las condiciones iniciales ni de algún otro parámetro externo. Dicho algoritmo busca una solución global al problema de k agrupamientos a través de una serie de búsquedas locales usando el algoritmo k-Means con funciones de similaridad como procedimiento de búsqueda local. El algoritmo propuesto es descrito a continuación.

4.2. Solución Propuesta

Sea $X = \{x_1, \ldots, x_m\}$ un conjunto de datos en donde cada objeto está descrito por un conjunto de atributos $R = \{y_1, \ldots, y_n\}$. Cada atributo toma valores de un conjunto de valores admisibles D_i, $y_i(x_j) \in D_i$, $i = 1, \ldots, n$; se asume que en D_i existe un símbolo "?" para denotar ausencia de información. De esta forma, los atributos pueden ser de cualquier naturaleza (no numéricos: Booleanos, categóricos, multivaluados, etc. ó numéricos: enteros, punto flotante, etc.) y se pueden considerar descripciones incompletas. Se define una función de similaridad $\Gamma : (D_1 \times D_2 \times \ldots \times D_n)^2 \rightarrow [0, 1]$ la cual permite comparar objetos. En este trabajo la función de similaridad usada fue:

$$\Gamma(x_i, x_j) = \frac{1}{n} \sum_{p=1}^{n} C_p(y_p(x_i), y_p(x_j)) \tag{4.2.1}$$

en donde C_p es una función de comparación entre valores del atributo y_p.

La función de comparación entre atributos no numéricos usada en este trabajo fue:

$$C_p(y_p(x_i), y_p(x_j)) = \begin{cases} 1 & \text{si } y_p(x_i) = y_p(x_j) \\ 0 & \text{en otro caso} \end{cases} \tag{4.2.2}$$

y la función de comparación entre atributos numéricos usada fue:

$$C_p(y_p(x_i), y_p(x_j)) = 1 - \left| \frac{|y_p(x_i)|}{|\operatorname{máx}(y_p)|} - \frac{|y_p(x_j)|}{|\operatorname{máx}(y_p)|} \right| \tag{4.2.3}$$

La ausencia de información puede ser tratada de diferentes maneras, para efectos de esta tesis el tratamiento que se le dio es el siguiente: dados dos objetos x_i, x_j los cuales se desea comparar el atributo p, se tiene que $C_p(y_p(x_i), y_p(x_j)) = 1$ si $y_p(x_i) =$ "?" ó $y_p(x_i) =$ "?".

El problema consiste en particionar el conjunto de datos X en k grupos $M_1, \ldots, M_k$. Para calcular los centros de los grupos, en este tipo de problemas no es posible usar el promedio de los valores de cada atributo de los objetos pertenecientes a un mismo grupo, por lo que objetos del conjunto de datos x_j^r son usados como centros de los agrupamientos M_j, $j = 1, \ldots, k$. Tales objetos son denominados *objetos representativos* y son en promedio los más similares a los objetos del mismo agrupamiento y al mismo tiempo son lo menos similares a los objetos de otros

agrupamientos.

Para determinar los objetos representativos se usan las siguientes expresiones:

$$r_{M_j}(x_i) = \frac{\beta_{M_j}(x_i)}{\alpha_{M_j}(x_i) + (1 - \beta_{M_j}(x_i))} + \eta(x_i) \tag{4.2.4}$$

donde $x_i \in M_j$.

$\beta_{M_j}(x_i)$ evalúa el promedio de similaridad del objeto x_i con el resto de los objetos en el agrupamiento M_j y se calcula de la siguiente manera:

$$\beta_{M_j}(x_i) = \frac{1}{|M_j| - 1} \sum_{\substack{x_i, x_s \in M_j \\ x_i \neq x_s}} \Gamma(x_i, x_s) \tag{4.2.5}$$

Para incrementar el valor informacional de la función (4.2.5) se introduce la expresión $\alpha_{M_j}(x_i)$

$$\alpha_{M_j}(x_i) = \frac{1}{|M_j| - 1} \sum_{\substack{x_i, x_s \in M_j \\ x_i \neq x_s}} \left| \beta_{M_j}(x_i) - \Gamma(x_i, x_s) \right| \tag{4.2.6}$$

la cual evalúa la diferencia entre el promedio (4.2.5) y la similaridad entre el objeto x_i y el resto de los objetos en el agrupamiento M_j, entonces cuando (4.2.6) decrece, el valor de (4.2.4) se incrementa.

La expresión $(1 - \beta_{M_j}(x_i))$ representa el promedio de disimilaridad de x_i con respecto al resto de los objetos en M_j

$$\eta(x_i) = \sum_{\substack{q=1 \\ i \neq q}}^{k} \left(1 - \Gamma(x_q^r, x_i)\right) \tag{4.2.7}$$

Finalmente, la expresión (4.2.7) evalúa la disimilaridad entre el objeto x_i y los otros objetos representativos. Esta función es usada para disminuir los casos donde existen dos objetos con el mismo valor en (4.2.4). Cuando $|M_j| = 1$, entonces el objeto representativo para el agrupamiento M_j es el objeto contenido en él.

De esta manera, el objeto representativo del agrupamiento M_j se define como el objeto x_r el cual alcanza el máximo de $r_{M_j}(x_i)$.

$$r_{M_j}(x_r) = \underset{x_p \in M_j}{\text{máx}} \{r_{M_j}(x_p)\} \tag{4.2.8}$$

El conjunto de datos debe ser clasificado de acuerdo a los objetos representativos de cada agrupamiento, es decir, dado un conjunto de objetos representativos, primero se obtiene una pertenencia $I_j(x_i)$ del objeto x_i al agrupamiento M_j, después de esto, se calcula el objeto representativo para la nueva k-partición. Este procedimiento es repetido hasta que no haya cambios en los objetos representativos.

Así, la función objetivo la cual se busca maximizar es:

$$J(x_1^r, \ldots, x_k^r) = \sum_{j=1}^{k} \sum_{i=1}^{m} I_j(x_i)\Gamma(x_j^r, x_i) \tag{4.2.9}$$

donde

$$I_j(x_i) = \begin{cases} 1 & \text{si } \Gamma(x_j^r, x_i) = \text{máx}\{\Gamma(x_q^r, x_i)\}, 1 \leq q \leq k \\ 0 & \text{en otro caso} \end{cases} \tag{4.2.10}$$

Esto es, un objeto x_i será asignado al agrupamiento el cual su objeto representativo es el más parecido a x_i.

Para resolver un problema con k agrupamientos el algoritmo k-Means Global con Funciones de Similaridad comienza buscando la solución para el problema con un agrupamiento ($k' = 1$), esto se realiza buscando el objeto que tiene en promedio la mayor similaridad a los objetos de conjunto de datos, dicho objeto es denotado por x_1^r. Posteriormente para resolver el problema con 2 agrupamientos ($k' = 1$) se coloca el primer objeto representativo como la solución del problema con un agrupamiento y se ejecuta el algoritmo k-Means con Funciones de Similaridad colocando el segundo objeto representativo como cada objeto x_i del conjunto de datos, $x_i \neq x_1^r$, $i = 1, \ldots, m$. Después de las m-1 ejecuciones del algoritmo k-Means con Funciones de Similaridad se considera como la solución del problema con 2 agrupamientos a la solución que maximiza la función objetivo (4.2.9). En general, sea $\left(x_1^r(k-1), x_2^r(k-1), \ldots, x_{k-1}^r(k-1)\right)$ la solución para el problema con $k-1$ agrupamientos. Una vez que se obtiene la solución para $k-1$ agrupamientos ésta es usada para resolver el problema con k agrupamientos ejecutando $m-(k-1)$ veces el algoritmo k-Means con Funciones de Similaridad, en donde cada ejecución comienza con los objetos representativos ini-

ciales $\left(x_1^r(k-1), x_2^r(k-1), \ldots, x_{k-1}^r(k-1), x_i\right)$, $x_i \neq x_p^r$, $p = 1, \ldots, k-1$, $i = 1, \ldots, m$. La mejor solución después de las $m-(k-1)$ ejecuciones (la cual maximiza la función objetivo (4.2.9)) es considerada como la solución para el problema de k agrupamientos. El pseudo código del algoritmo k-Means Global con Funciones de Similaridad propuesto es mostrado en el algoritmo 4.1.

Algoritmo 4.1: k-Means Global para Datos Mezclados

```
Datos: k = número de agrupamientos a formar, X = conjunto de datos.
Resultado: Valor de la función objetivo, Conjunto de objetos representativos, Partición.
inicio
    m ← número de objetos;
    CrearVector Semillas[0,...,k − 1] /* Crea un vector de longitud k      */
    Semillas[0]← # del objeto más parecido en promedio a todos los objetos del
    conjunto de datos;
    para k' ← 1 a k − 1 hacer
        Aux ← 0;
        para i ← 0 a m − 1 hacer
            si i ≠ Semillas[z], z = 0,...,k' − 1 entonces
                Semillas[k']← i;
                [J, OR, Par]←kMeansFuncionesSimilaridad(Semillas);
                /* J valor de la función objetivo                          */
                /* OR objetos representativos                              */
                /* Par partición                                           */
                si J>Aux entonces
                    Aux ← J;
                    ORAux ← OR;
                    ParAux ← Par;
                fin si
            fin si
        fin para
        Semillas ← ORAux;
    fin para
    FuncObj ← Aux;
    ObjRep ← ORAux;
    Particion ← ParAux;
    retornar FuncObj,ObjRep,Particion;
fin
```

4.3. Resultados Experimentales

4.3.1. Consideraciones Generales

El algoritmo propuesto fue evaluado usando ocho conjuntos de datos, los cuales fueron obtenidos del repositorio de bases de datos de aprendizaje automático de la Universidad de California, Irvine [24]. En todos los conjuntos de datos se ignoraron las etiquetas de los objetos y solamente se uso la información de los atributos. La calidad de los resultados obtenidos fue evaluada en términos de la función objetivo (4.2.9).

Los conjuntos de datos usados fueron Bands, Credit, Ecoli, Flags, Glass, Iris, Machine y Wine. La descripción de tales conjuntos de datos se muestra en la tabla 4.1.

Conjunto de datos	Número de objetos	Atributos no numéricos	Atributos numéricos	Ausencia de información
Bands	512	18	21	si
Credit	690	15	0	si
Ecoli	336	0	7	no
Flags	194	21	8	no
Glass	214	0	9	no
Iris	150	0	4	no
Machine	209	1	7	no
Wine	178	0	13	no

Tabla 4.1: Descripción de los conjuntos de datos usados para evaluar el desempeño del algoritmo k-Means Global para Datos Mezclados y para realizar una comparación con otros algoritmos

Para cada conjunto de datos se realizaron los siguientes experimentos:

- Una ejecución del algoritmo k-Means Global para Datos Mezclados para cada problema con $k = 2, \ldots, 15$ agrupamientos.
- m ejecuciones (donde m es el número de objetos en el conjunto de datos) del algoritmo k-Means con Funciones de Similaridad para cada problema con $k = 2, \ldots, 15$ agrupamientos, en donde cada ejecución inicia con objetos representativos obtenidos aleatoriamente. Para cada problema con i agrupamientos, $i = 2, \ldots, k$, se calculó el máximo de las m ejecuciones.

- m ejecuciones del algoritmo k-Prototypes para cada problema con $k = 2, \ldots, 15$ agrupamientos, en donde cada ejecución comienza con prototipos iniciales obtenidos aleatoriamente del conjunto de datos. De igual manera, únicamente se consideró el máximo de las m ejecuciones.

Para poder comparar los resultados del algoritmo k-Prototypes con los resultados obtenidos de los demás algoritmos a evaluar, se normalizaron los valores de la función de comparación que usa este algoritmo a [0,1] (para mayor detalle ver [9, 10]).En la función objetivo se toma la sumatoria de uno menos el valor normalizado de la función de comparación, por lo que ahora ésta debe ser maximizada. Asimismo, se usó $\gamma = 1$, es decir, no se favoreció a algún tipo de atributo.

4.3.2. Resultados Obtenidos y Comparación con otros Algoritmos

Los resultados experimentales obtenidos al aplicar el algoritmo k-Means Global para Datos Mezclados, las m ejecuciones del algoritmo k-Means con Funciones de Similaridad y las m ejecuciones del algoritmo k-Prototypes a los conjuntos de datos Bands, Credit, Ecoli, Flags, Glass, Iris, Machine y Wine se muestran en las figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8 respectivamente. Asimismo, en las mismas figuras se muestra el tiempo de ejecución de dichos experimentos.

De los resultados obtenidos en los experimentos se puede concluir que el algoritmo k-Means Global para Datos Mezclados obtuvo mejores resultados que los obtenidos por los algoritmos k-Means con Funciones de Similaridad y k-Prototypes.

En las figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8 se puede observar que el algoritmo propuesto obtuvo, en la mayoría de los casos, mejores resultados que el máximo de las m ejecuciones del algoritmo k-Means con Funciones de Similaridad. En algunos casos, el máximo de las m ejecuciones fue el mismo resultado que el obtenido por el algoritmo k-Means Global para Datos Mezclados, lo que ocurre principalmente en problemas con pocos agrupamientos y en ningún caso, las m ejecuciones del algoritmo k-Means con Funciones de Similaridad obtuvieron mejores

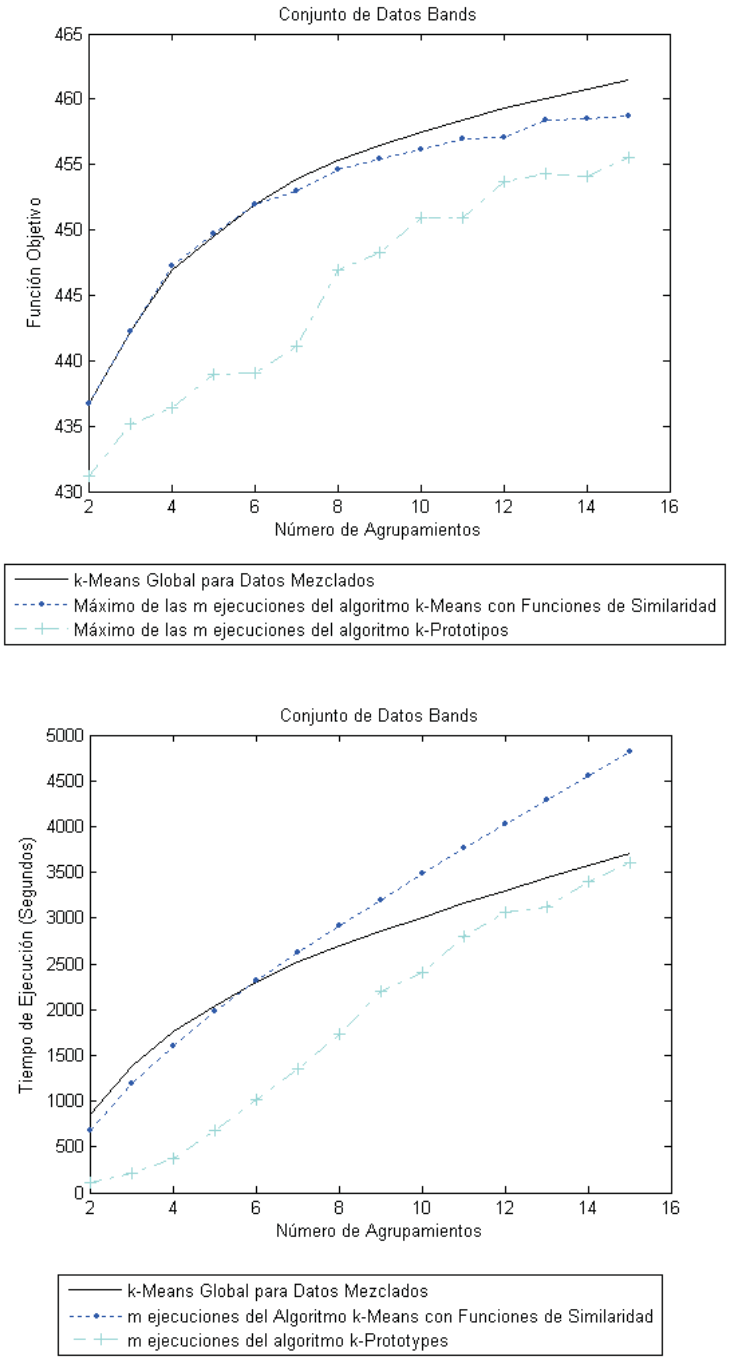


Figura 4.1: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Bands

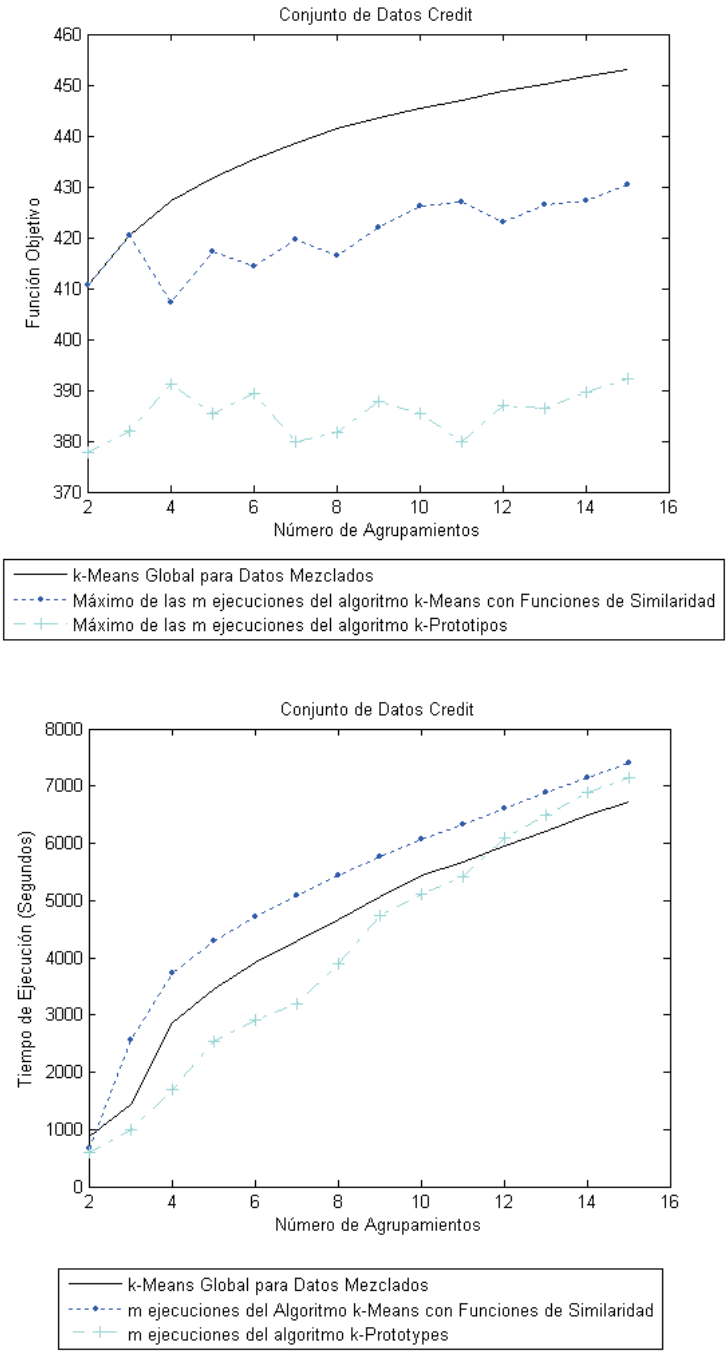
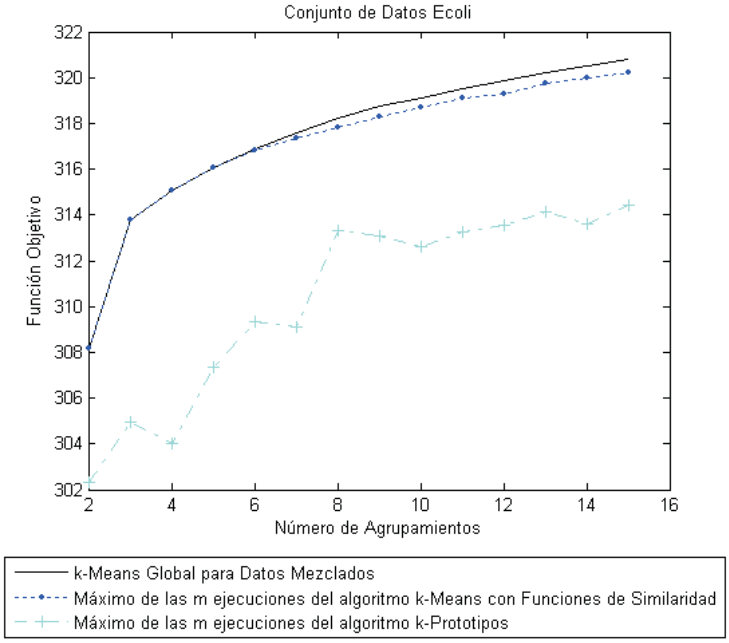


Figura 4.2: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Credit



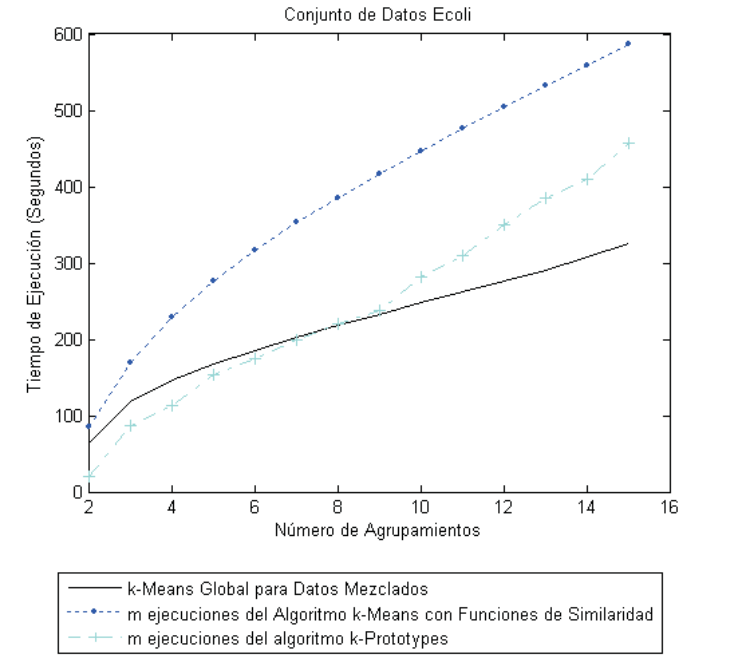
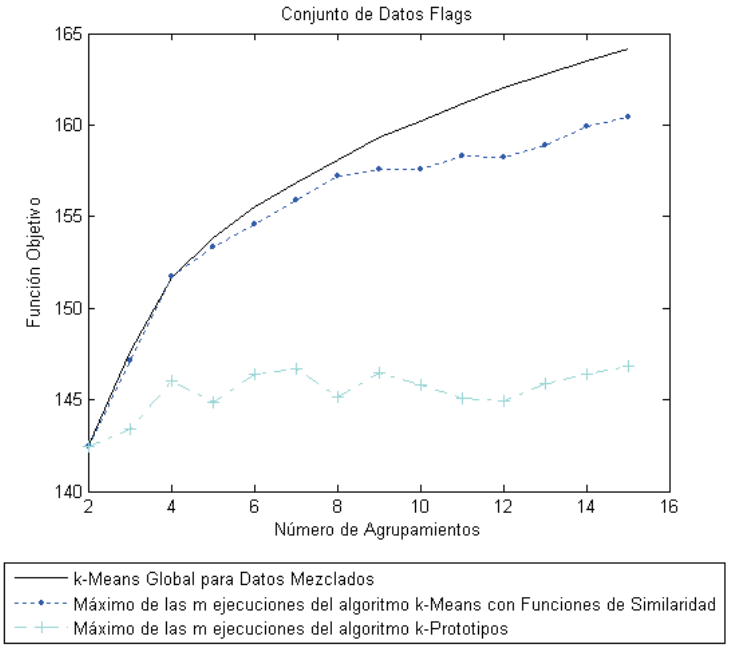


Figura 4.3: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Ecoli



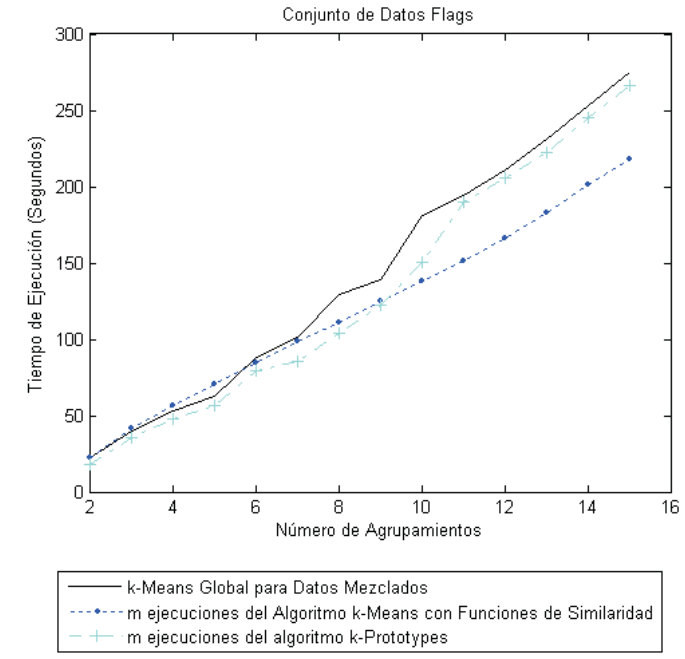
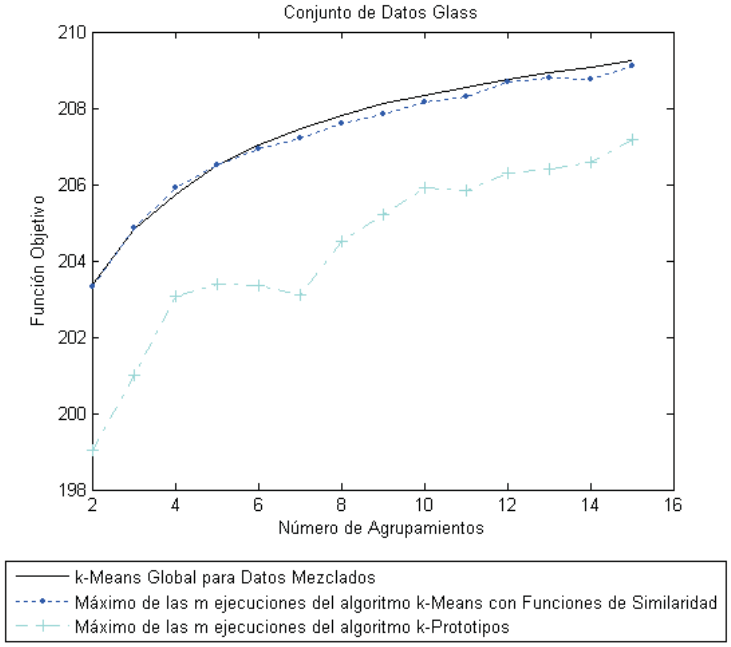


Figura 4.4: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Flags



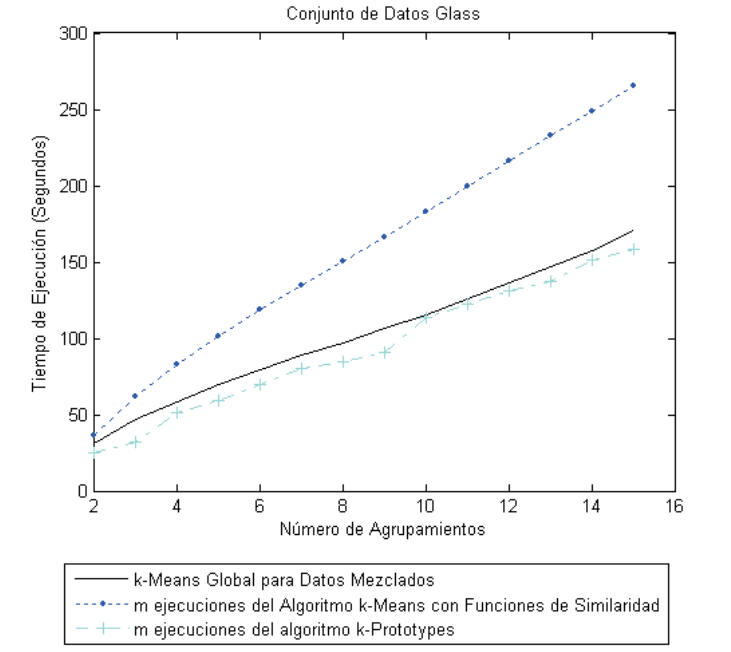


Figura 4.5: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Glass

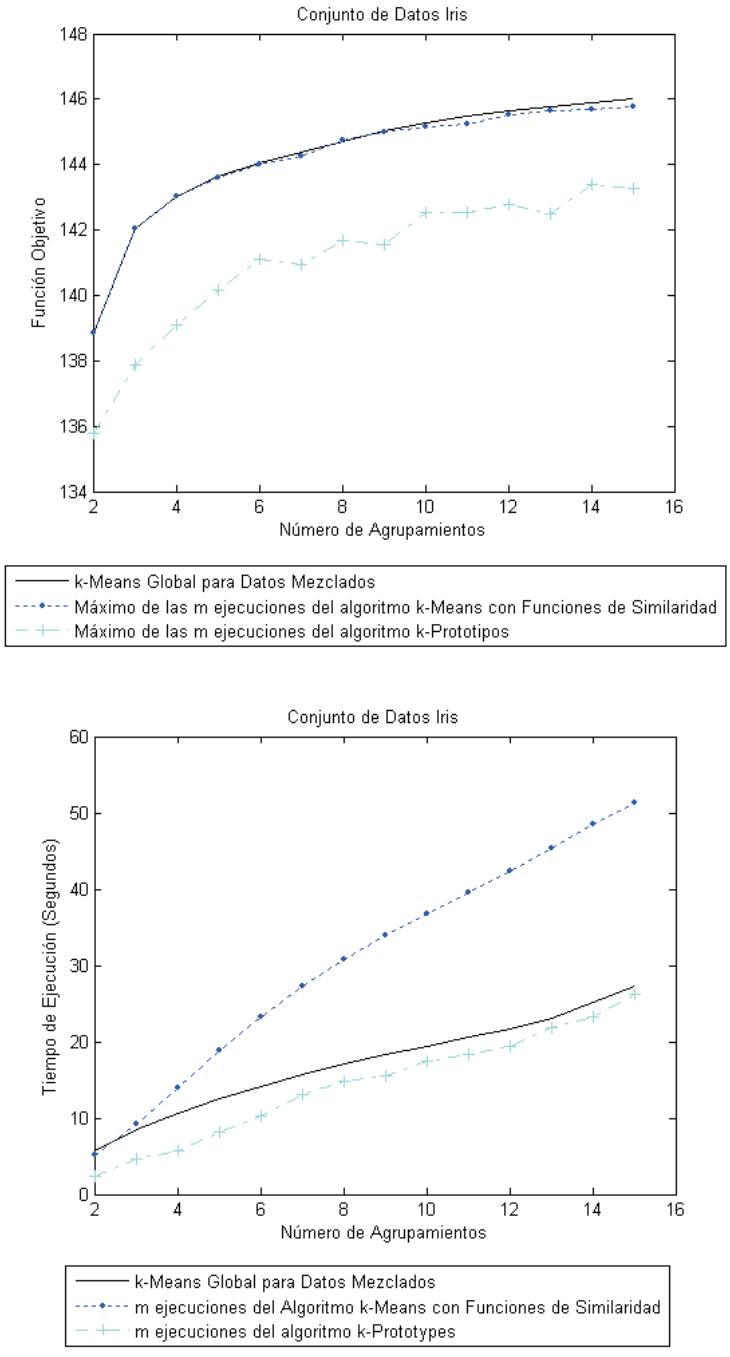


Figura 4.6: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Iris

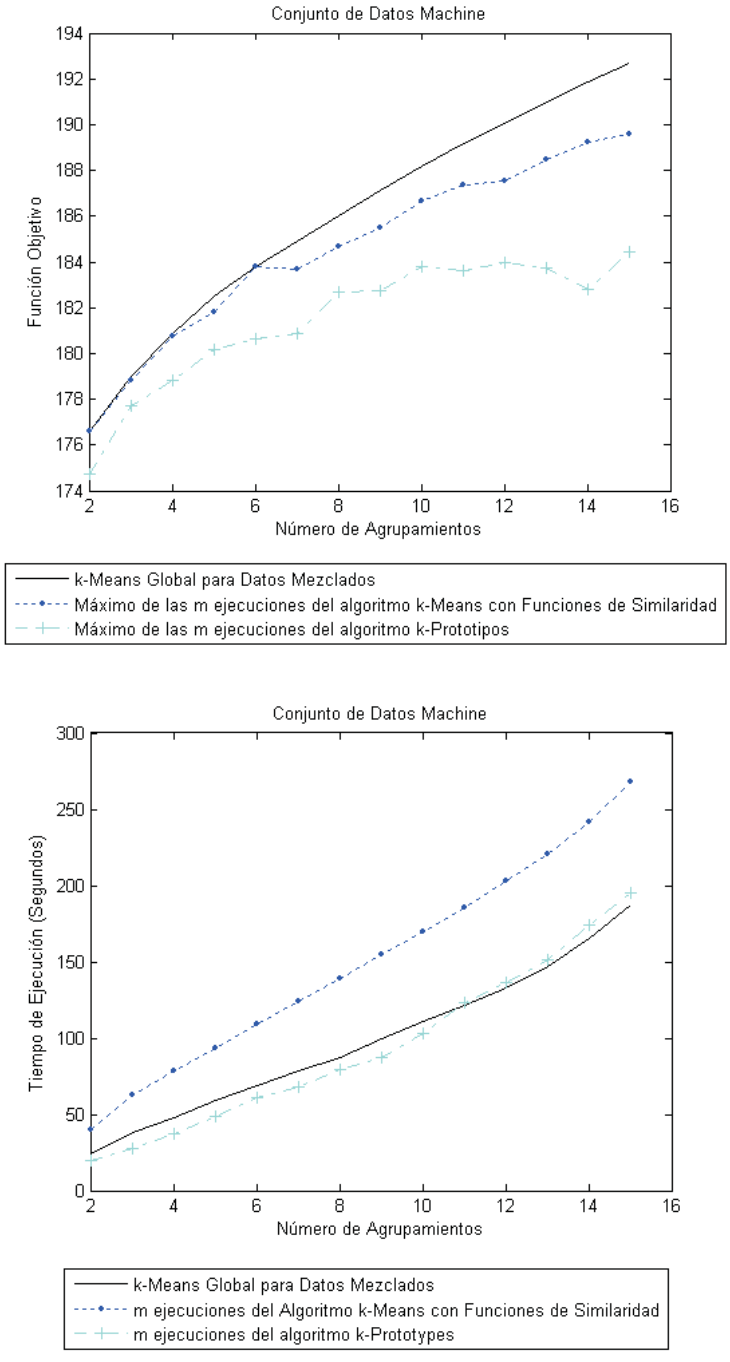


Figura 4.7: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Machine

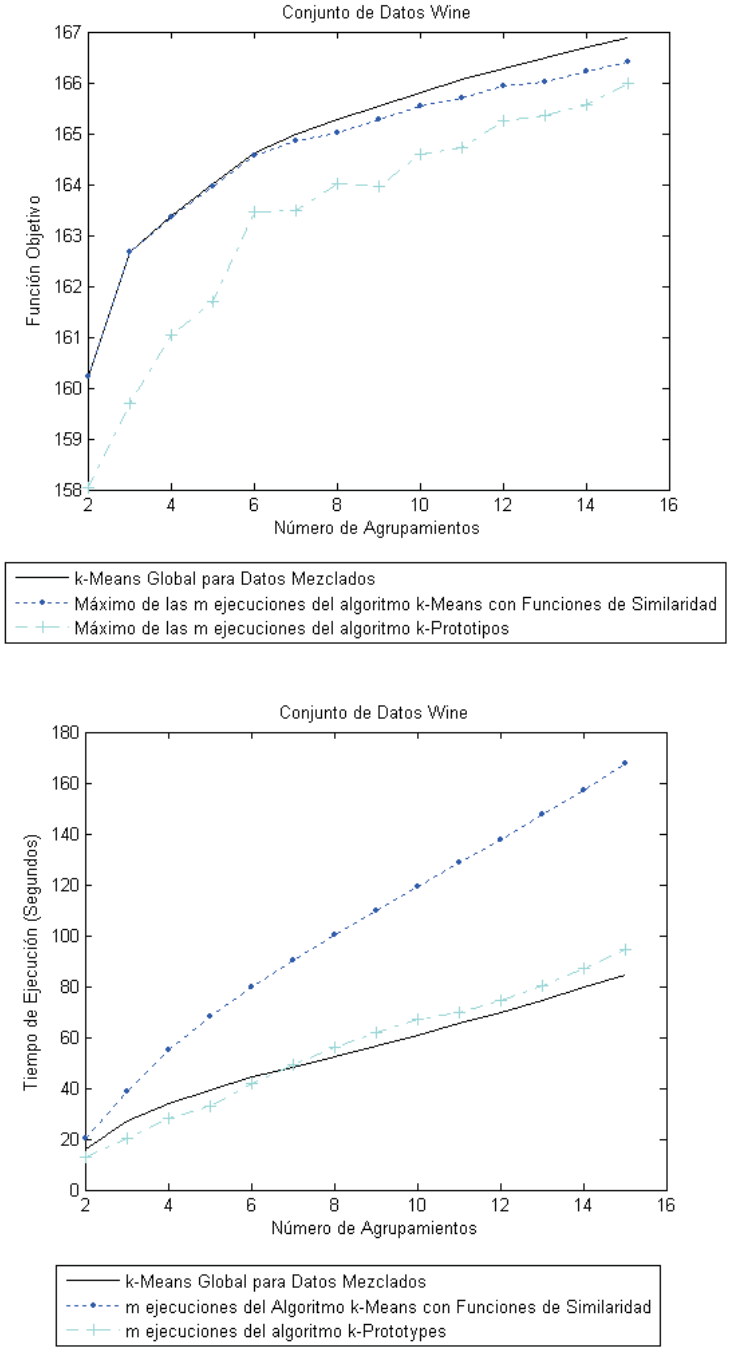


Figura 4.8: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes y k-Means Global para Datos Mezclados al conjunto de datos Wine

resultados que el algoritmo k-Means Global para Datos Mezclados.

En comparación con el algoritmo k-Prototypes, el algoritmo k-Means Global para Datos Mezclados siempre obtuvo mejores resultados. En las figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8 se puede observar que el algoritmo k-Prototypes es el que obtiene los peores resultados, ya que el valor máximo de la función objetivo obtenido a partir de sus m ejecuciones siempre está por debajo del máximo de las m ejecuciones del algoritmo k-Means con Funciones de Similaridad y del resultado obtenido con el algoritmo k-Means Global para Datos Mezclados.

En términos del tiempo de ejecución, en las figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8 se puede observar que el algoritmo k-Means Global para Datos Mezclados tuvo un menor tiempo de ejecución que las m ejecuciones del algoritmo k-Means con Funciones de Similaridad. Esto se debe principalmente a cada vez que el algoritmo k-Means Global para Datos Mezclados hace un llamado al algoritmo k-Means con Funciones de Similaridad como procedimiento de búsqueda local, éste comienza con *mejores* objetos representativos iniciales, a diferencia de cuando se inicia con objetos elegidos aleatoriamente, lo que permite que converja más rápido. Asimismo, en cada iteración del algoritmo k-Means Global para Datos Mezclados no se ejecuta m veces el algoritmo k-Means con Funciones de Similaridad, ya que solamente lo ejecuta $m-(k-1)$ veces. Por lo que en total lo ejecuta $\sum_{i=1}^{k} m-(i-1)$ veces.

En las figuras 4.2, 4.3, 4.5, 4.5, 4.7 y 4.8 se puede observar que el tiempo de ejecución del algoritmo propuesto es mayor que el del algoritmo k-Prototypes en casos con pocos agrupamientos y, en casos con muchos agrupamientos el tiempo de ejecución del algoritmo k-Prototypes es mayor que el del algoritmo propuesto. Asimismo, en las figuras 4.1, 4.4 y 4.6 el tiempo de ejecución del algoritmo k-Means Global para Datos Mezclados es siempre mayor que el del algoritmo k-Prototypes.

4.3.3. Discusión

Con base en los resultados obtenidos en los experimentos realizados, se pueden concluir los siguientes puntos:

- El algoritmo k-Means Global para Datos Mezclados obtuvo mejores resultados que los algoritmos k-Means con Funciones de Similaridad y k-Prototypes. Sólo en algunos casos, principalmente en casos con

pocos grupos a formar, los resultados obtenidos por el máximo de las m ejecuciones del algoritmo k-means con funciones de similaridad fueron iguales a los obtenidos por el algoritmo propuesto.

- El tiempo de ejecución del algoritmo k-Means Global con Funciones de Similaridad fue menor al de las m ejecuciones del algoritmo k-Means con Funciones de Similaridad y en algunos casos menor al de las m ejecuciones del algoritmo k-Prototypes, sin embargo, este último es el que obtuvo los peores resultados.
- Dadas las características del algoritmo propuesto en esta sección, éste no puede ser aplicado a grandes conjuntos de datos mezclados. Lo que se debe principalmente al hecho de que ejecuta demasiadas veces el algoritmo k-Means con Funciones de Similaridad, lo que conlleva un alto costo computacional.

Capítulo 5

Algoritmo *k*-Means Global Rápido para Datos Mezclados

En este capítulo se propone un algoritmo de agrupamiento restringido de conjuntos de datos mezclados el cual reduce considerablemente el costo computacional del algoritmo k-Means Global para Datos Mezclados sin sacrificar demasiado la calidad de los resultados obtenidos. El algoritmo propuesto es probado con diferentes conjuntos de datos mezclados obtenidos de un repositorio público y comparado contra otros algoritmos de agrupamiento restringido.

5.1. Introducción

El algoritmo k-Means Global para Datos Mezclados soluciona la dependencia de las condiciones iniciales del algoritmo k-Means con Funciones de Similaridad, busca una solución global al problema de k agrupamientos y permite trabajar con datos mezclados. Sin embargo, este algoritmo tiene un alto costo computacional, lo que se debe principalmente a que en cada iteración, el algoritmo k-Means Global para Datos Mezclados ejecuta demasiadas veces el algoritmo k-Means con Funciones de Similaridad como procedimiento de búsqueda local, lo cual representa su principal desventaja. Por tal motivo es necesario reducir el número de ejecuciones del algoritmo k-Means con Funciones de Similaridad sin afectar significativamente la calidad de los resultados, por lo que en este capítulo se propone una modificación de dicho algoritmo, la cual

es descrita a continuación.

5.2. Solución Propuesta

Sea $X = \{x_1, \ldots, x_m\}$ un conjunto de datos en donde cada objeto está descrito por un conjunto de atributos $R = \{y_1, \ldots, y_n\}$. Cada atributo toma valores de un conjunto de valores admisibles D_i, $y_i(x_j) \in D_i$, $i = 1, \ldots, n$; se asume que en D_i existe un símbolo "?" para denotar ausencia de información. De esta forma, los atributos pueden ser de cualquier naturaleza (no numéricos: Booleanos, categóricos, multivaluados, etc. ó numéricos: enteros, punto flotante, etc.) y se pueden considerar descripciones incompletas. Se define una función de similaridad $\Gamma : (D_1 \times D_2 \times \ldots \times D_n)^2 \rightarrow [0, 1]$ la cual permite comparar objetos. En este trabajo la función de similaridad usada fue:

$$\Gamma(x_i, x_j) = \frac{1}{n} \sum_{p=1}^{n} C_p(y_p(x_i), y_p(x_j)) \tag{5.2.1}$$

en donde C_p es una función de comparación entre valores del atributo y_p.

La función de comparación entre atributos no numéricos usada en este trabajo fue:

$$C_p(y_p(x_i), y_p(x_j)) = \begin{cases} 1 & \text{si } y_p(x_i) = y_p(x_j) \\ 0 & \text{en otro caso} \end{cases} \tag{5.2.2}$$

y la función de comparación entre atributos numéricos usada fue:

$$C_p(y_p(x_i), y_p(x_j)) = 1 - \left| \frac{|y_p(x_i)|}{|\operatorname{máx}(y_p)|} - \frac{|y_p(x_j)|}{|\operatorname{máx}(y_p)|} \right| \tag{5.2.3}$$

La ausencia de información puede ser tratada de diferentes maneras, para efectos de esta tesis el tratamiento que se le dio es el siguiente: dados dos objetos x_i, x_j los cuales se desea comparar el atributo p, se tiene que $C_p(y_p(x_i), y_p(x_j)) = 1$ si $y_p(x_i) =$ "?" ó $y_p(x_i) =$ "?".

El problema consiste en particionar el conjunto de datos X en k grupos $M_1, \ldots, M_k$. Para calcular los centros de los grupos, en este tipo de problemas no es posible usar el promedio de los valores de cada atributo de los objetos pertenecientes a un mismo grupo, por lo que objetos del conjunto de datos x_j^r son usados como centros

de los agrupamientos M_j, $j = 1, \ldots, k$. Tales objetos son denominados *objetos representativos* y son en promedio los más similares a los objetos del mismo agrupamiento y al mismo tiempo son lo menos similares a los objetos de otros agrupamientos.

En este contexto, el problema de agrupamiento consiste en maximizar la siguiente función objetivo:

$$J(x_1^r, \ldots, x_k^r) = \sum_{j=1}^{k} \sum_{i=1}^{m} I_j(x_i)\Gamma(x_j^r, x_i) \tag{5.2.4}$$

donde

$$I_j(x_i) = \begin{cases} 1 & \text{si } \Gamma(x_j^r, x_i) = \text{máx}\{\Gamma(x_q^r, x_i)\}, 1 \leq q \leq k \\ 0 & \text{en otro caso} \end{cases} \tag{5.2.5}$$

Esto es, un objeto x_i será asignado al agrupamiento el cual su objeto representativo es el más parecido a x_i.

La reducción en el costo computacional del algoritmo k-Means Global para Datos Mezclados puede lograrse calculando una cota $J_i \geq J + b_i^*$ sobre el valor que la función objetivo J_i puede alcanzar para cada posible posición x_i con la cual se busca una solución al problema de k agrupamientos, en donde J representa el valor de la función objetivo (5.2.4) en la solución del problema con $k-1$ agrupamientos y, b_i^* se define cómo:

$$b_i^* = \sum_{j=1}^{m} \text{máx}(\Gamma(x_i, x_j) - \Gamma_{k-1}^j, 0) \tag{5.2.6}$$

donde Γ_{k-1}^j es la similaridad entre el objeto x_j y el objeto representativo del agrupamiento al cual dicho objeto pertenece en la solución del problema con $k-1$ agrupamientos.

Cada vez que un nuevo agrupamiento es agregado, la posición del nuevo objeto representativo es inicializada en los objetos que maximizen J_i o equivalentemente que maximizen b_i^* y posteriormente se ejecuta el algoritmo k-means con funciones de similaridad para obtener la solución al problema de k agrupamientos.

El término b_i^* mide el incremento garantizado en la función objetivo (5.2.4) al insertar un nuevo agrupamiento cuyo objeto representativo es el objeto x_i.

Para resolver el problema con k agrupamientos dada la solución para el problema

con $k-1$ agrupamientos $\left(x_1^r(k-1), x_2^r(k-1), \ldots, x_{k-1}^r(k-1)\right)$, se agrega un nuevo agrupamiento cuyo objeto representativo es inicializado en la posición x_i, de esta forma sólo los objetos x_j cuya similaridad con x_i sea mayor que Γ_{k-1}^j pertenecerán al nuevo agrupamiento. Por ejemplo, si un objeto tiene 0.7 por similaridad con el objeto representativo del agrupamiento al cual pertenece y la similaridad con el objeto x_i es 0.9, entonces este objeto pertenecerá al agrupamiento cuyo objeto representativo es x_i. Así, habrá un incremento de al menos 0.2 en la función objetivo. De esta forma, la sumatoria $\sum_{j=1}^{m} \text{máx}(\Gamma(x_i, x_j) - \Gamma_{k-1}^j, 0)$ acota el incremento mínimo que se obtendrá en la función objetivo al agregar un nuevo agrupamiento en la posición x_i.

En este trabajo se uso una implementación del algoritmo k-means con funciones de similaridad la cual almacena el mejor valor de la función objetivo con sus respectivos objetos representativos en cada iteración, con lo que se garantiza que haya un incremento en la función objetivo de al menos b_i^*. Por lo que los objetos que maximizen la función b_i^* pueden producir un mayor incremento en la función objetivo.

Debido a lo anterior, si se ejecuta el algoritmo k-means con funciones de similaridad con los objetos representativos iniciales $\left(x_1^r(k-1), x_2^r(k-1), \ldots, x_{k-1}^r(k-1), x_i\right)$, en donde $\left(x_1^r(k-1), x_2^r(k-1), \ldots, x_{k-1}^r(k-1)\right)$ es la solución del problema con $k-1$ agrupamientos y x_i son los objetos que maximizan b_i^*, dado que está garantizado que la función objetivo se incrementará en al menos b_i^*, el incremento obtenido en la función objetivo al ejecutar el algoritmo k-means con funciones de similaridad después de agregar un nuevo agrupamiento cuyo objeto representativo es x_i puede ser acotado inferiormente por $J + b_i^*$.

Con esto, al seleccionar los objetos que maximicen b_i^* se estarán seleccionando aquellos objetos con mayor probabilidad de maximizar la función objetivo, ya que al menos se alcanza un incremento igual a b_i^*, sin embargo, no se garantiza obtener el máximo, por lo que el resultado final puede ser ligeramente afectado.

El pseudo código de este algoritmo se muestra en el algoritmo 5.1.

5.3. Resultados Experimentales

5.3.1. Consideraciones Generales

Para evaluar el desempeño del algoritmo propuesto en esta sección se realizaron dos experimentos. En el primero, el algoritmo propuesto se comparó contra los algoritmos

Algoritmo 5.1: k-Means Global Rápido para Datos Mezclados

Datos: k = número de agrupamientos a formar, X = conjunto de datos.
Resultado: Valor de la función objetivo, Conjunto de objetos representativos, Partición.

```
inicio
    m ← número de objetos;
    CrearVector Semillas[0,...,k-1] /* Crea un vector de longitud k      */
    Semillas[0]← # del objeto más parecido en promedio a todos los objetos del
    conjunto de datos;
    para k' ← 1 a k-1 hacer
        Aux ← 0;
        B_i ← 0;
        BAux ← 0;
        para i ← 0 a m-1 hacer
            /* Obtiene B_i usando la ecuación 5.2.6                       */
            Baux ← ObtieneB_i(i);
            si BAux > B_i entonces
                B_i ← BAux
            fin si
        fin para
        /* Obtiene los objetos que maximizan B_i y los almacena en un
           vector                                                         */
        VecAux ← ObtObjMaxB_i();
        w ← |VecAux|;
        para i ← 0 a w-1 hacer
            x_i ← VecAux[i];
            si x_i ≠ Semillas[z], z = 0,...,k'-1 entonces
                Semillas[k']← x_i;
                [J, OR, Par]←kMeansFuncionesSimilaridad(Semillas);
                /* J valor de la función objetivo                         */
                /* OR objetos representativos                             */
                /* Par partición                                          */
                si J>Aux entonces
                    Aux ← J;
                    ORAux ← OR;
                    ParAux ← Par;
                fin si
            fin si
        fin para
        Semillas ← ORAux;
    fin para
    FuncObj ← Aux;
    ObjRep ← ORAux;
    Particion ← ParAux;
    retornar FuncObj,ObjRep,Particion;
fin
```

k-Means Global para Datos Mezclados, k-Means con Funciones de Similaridad y k-Prototypes usando ocho conjuntos de datos. En el segundo, el algoritmo propuesto fue comparado únicamente contra el algoritmo k-Prototypes usando tres conjuntos de datos más grandes en donde el algoritmo k-Means Global para Datos Mezclados no puede ser aplicado debido a su alto costo computacional. Todos los conjuntos de datos fueron obtenidos del repositorio de bases de datos de aprendizaje automático de la Universidad de California, Irvine [24]. Se ignoraron las etiquetas de los objetos y solamente se utilizó la información de los atributos. Asimismo, la calidad de los resultados obtenidos fue evaluada en términos de la función objetivo (4.2.9).

Los conjuntos de datos usados para el primer experimento fueron Bands, Credit, Ecoli, Flags, Glass, Iris, Machine y Wine. La descripción de tales conjuntos de datos se muestra en la tabla 5.1.

Conjunto de datos	**Número de objetos**	**Atributos no numéricos**	**Atributos numéricos**	**Ausencia de información**
Bands	512	18	21	si
Credit	690	15	0	si
Ecoli	336	0	7	no
Flags	194	21	8	no
Glass	214	0	9	no
Iris	150	0	4	no
Machine	209	1	7	no
Wine	178	0	13	no

Tabla 5.1: Descripción de los conjuntos de datos usados para evaluar el desempeño del algoritmo k-Means Rápido Global para Datos Mezclados y para realizar una comparación con otros algoritmos

Para cada uno de estos conjuntos de datos se realizaron los siguientes experimentos:

- Una ejecución del algoritmo k-Means Global Rápido para Datos Mezclados para cada problema con $k = 2, \ldots, 15$ agrupamientos.
- Una ejecución del algoritmo k-Means Global para Datos Mezclados para cada problema con $k = 2, \ldots, 15$ agrupamientos.
- m ejecuciones (donde m es el número de objetos en el conjunto de datos) del algoritmo k-Means con Funciones de Similaridad para cada

problema con $k = 2, \ldots, 15$ agrupamientos, en donde cada ejecución inicia con objetos representativos obtenidos aleatoriamente. Para cada problema con i agrupamientos, $i = 2, \ldots, k$, se calculó el máximo, el mínimo y el promedio de las m ejecuciones.

- m ejecuciones del algoritmo k-Prototypes para cada problema con $k = 2, \ldots, 15$ agrupamientos, en donde cada ejecución comienza con prototipos iniciales obtenidos aleatoriamente del conjunto de datos. De igual manera, únicamente se consideró el máximo de las m ejecuciones.

Los conjuntos de datos para el segundo tipo de experimento fueron CH y HY. Su descripción se muestra en la tabla 5.2.

Conjunto de datos	Número de objetos	Atributos no numéricos	Atributos numéricos	Ausencia de información
CH	3196	36	0	no
HY	3163	18	7	si

Tabla 5.2: Descripción de los conjuntos de datos usados para evaluar el desempeño del algoritmo k-Means Global Rápido para Datos Mezclados y para realizar una comparación con el algoritmo k-Prototypes

Para cada conjunto de datos se realizaron los siguientes experimentos:

- Una ejecución del algoritmo k-Means Global Rápido para Datos Mezclados para cada problema con $k = 2, \ldots, 15$ agrupamientos.
- 100 ejecuciones del algoritmo k-Prototypes para cada problema con $k = 2, \ldots, 15$ agrupamientos, en donde cada ejecución comienza con prototipos iniciales obtenidos aleatoriamente del conjunto de datos. Únicamente se consideró el máximo de las m ejecuciones.

Al igual que en el capítulo anterior, para poder comparar los resultados del algoritmo k-Prototypes con los resultados obtenidos de los demás algoritmos a evaluar, se normalizaron los valores de la función de comparación que usa este algoritmo a [0,1] (para mayor detalle ver [9, 10]).En la función objetivo se toma la sumatoria de uno menos el valor normalizado de la función de comparación, por lo que ahora ésta debe ser maximizada. Asimismo, se usó $\gamma = 1$, es decir, no se favoreció a algún tipo de atributo.

5.3.2. Resultados Obtenidos y Comparación con otros Algoritmos

Los resultados del primer experimento obtenidos al aplicar el algoritmo k-Means Global Rápido para Datos Mezclados, k-Means Global para Datos Mezclados, las m ejecuciones del algoritmo k-Means con Funciones de Similaridad y las m ejecuciones del algoritmo k-Prototypes a los conjuntos de datos Bands, Credit, Ecoli, Flags, Glass, Iris, Machine y Wine se muestran en las figuras 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 y 5.8 respectivamente. Asimismo, en las mismas figuras se muestra el tiempo de ejecución de dichos experimentos. Nótese que el tiempo de ejecución del algoritmo k-Means Global Rápido para Datos Mezclados se encuentra muy cercano al eje x.

Los resultados del segundo experimento obtenidos al aplicar el algoritmo k-Means Global Rápido para Datos Mezclados y 100 ejecuciones del algoritmo k-Prototypes a los conjuntos de datos CH y HY se muestran en las figuras 5.9 y 5.10 respectivamente. De igual manera, en las mismas figuras se muestra el tiempo de ejecución de dichos experimentos.

Con base en los resultados obtenidos en los experimentos se puede concluir que el algoritmo k-Means Global Rápido para Datos Mezclados obtuvo mejores resultados que los obtenidos por los algoritmos k-Means Global para Datos Mezclados, k-Means con Funciones de Similaridad y k-Prototypes.

En las figuras 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 y 5.8 se puede observar que el algoritmo k-Means Global Rápido para Datos Mezclados obtuvo resultados similares a los obtenidos por el algoritmo k-Means Global para Datos Mezclados con un costo computacional mucho menor. Únicamente con los conjuntos de datos Credit y Machine (figuras 5.2 y 5.7) el resultado es significativamente menor, aunque con el conjunto de datos Credit los resultados obtenidos con este algoritmo son mayores que los obtenidos con el máximo de las m ejecuciones del algoritmo k-Means con Funciones de Similaridad.

Asimismo, los resultados obtenidos con el algoritmo k-Means Global Rápido para Datos Mezclados son, en la mayoría de los casos, iguales o mejores que los obtenidos con las m ejecuciones del algoritmo k-Means con Funciones de similaridad y, son siempre mejores que los obtenidos con las m ejecuciones del algoritmo k-Prototypes, siendo este último el que obtuvo los peores resultados.

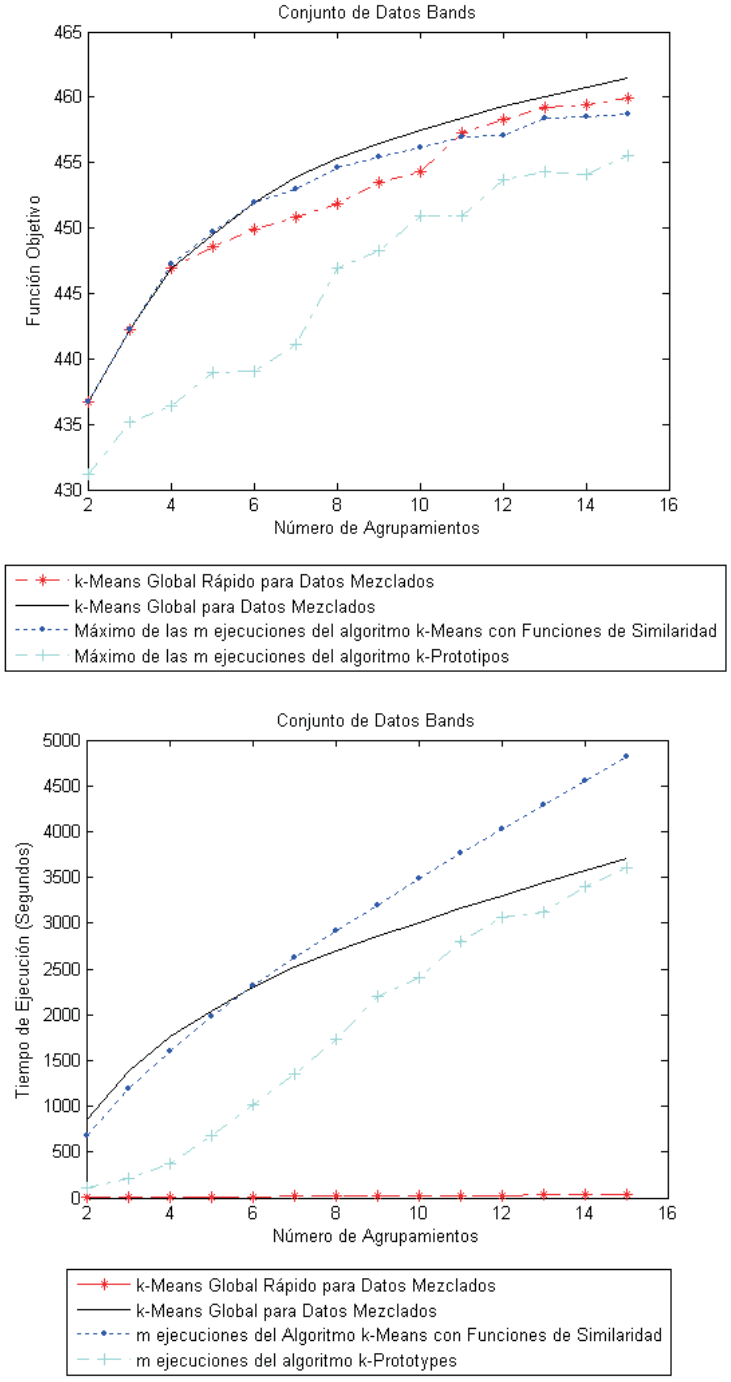


Figura 5.1: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Bands

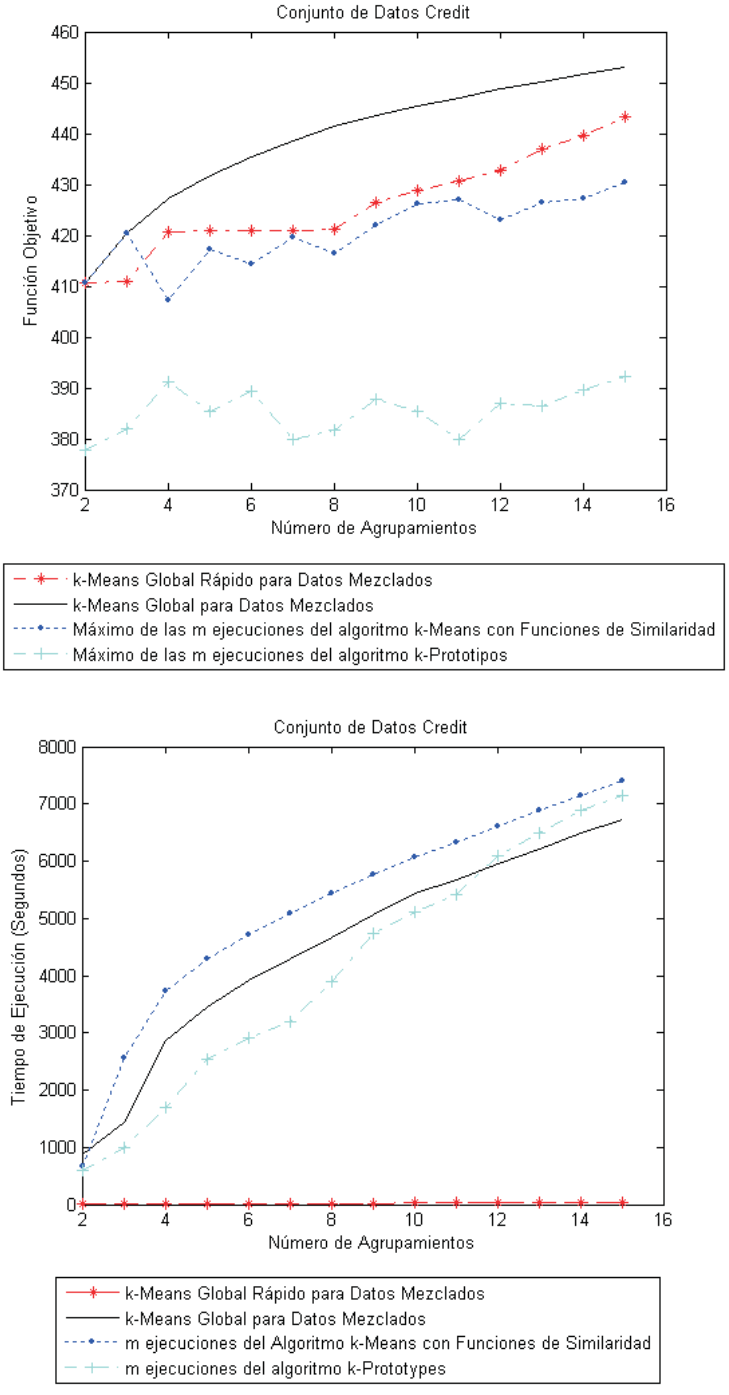


Figura 5.2: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Credit

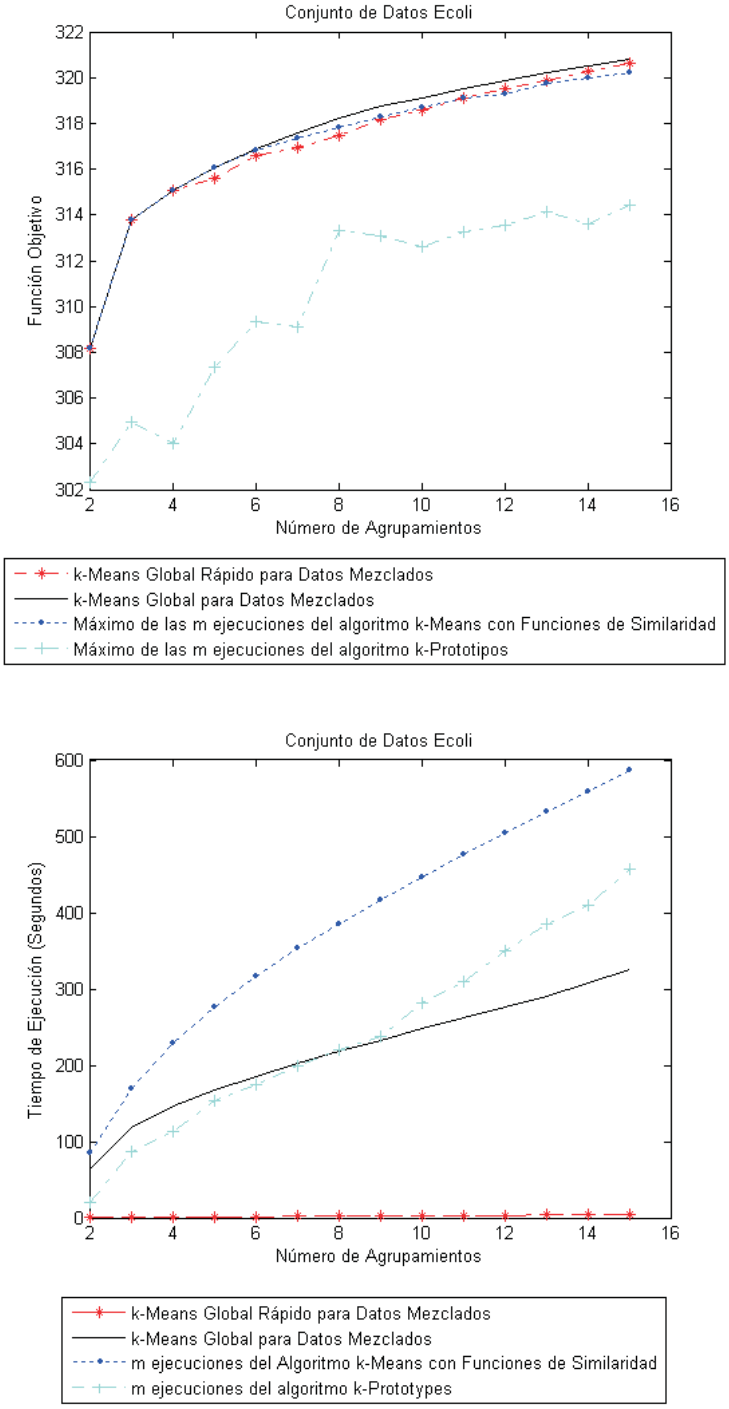
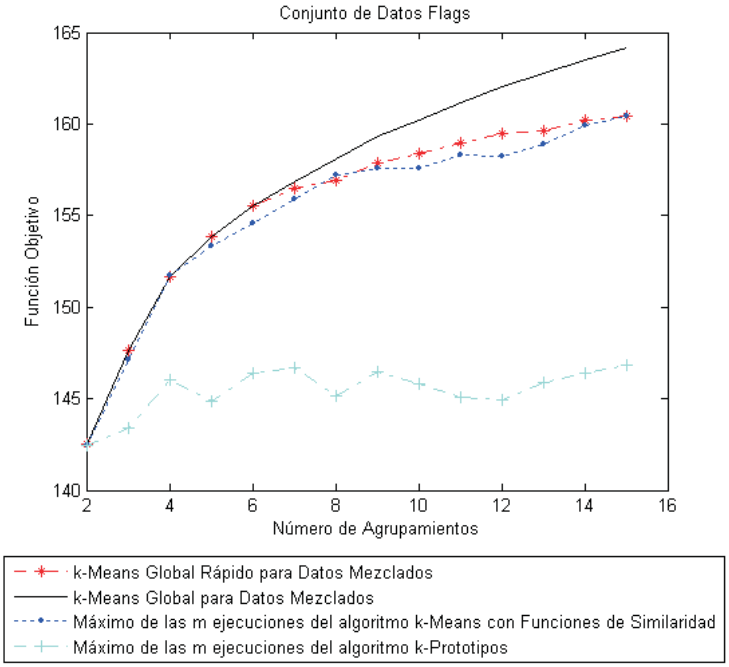


Figura 5.3: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Ecoli



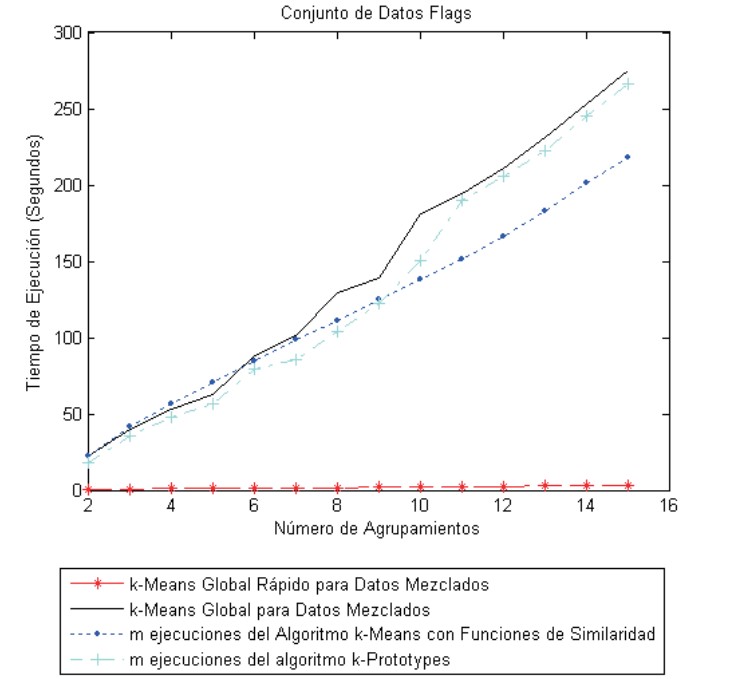
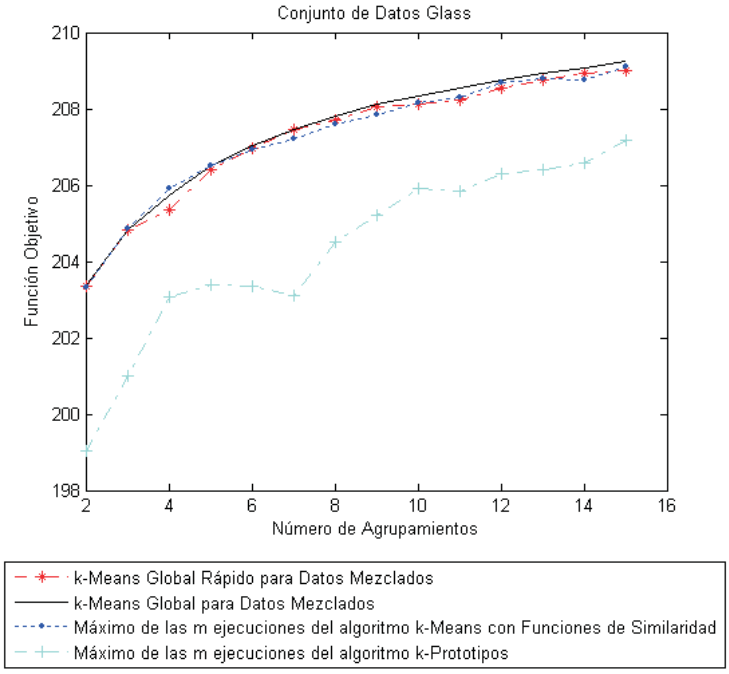


Figura 5.4: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Flags



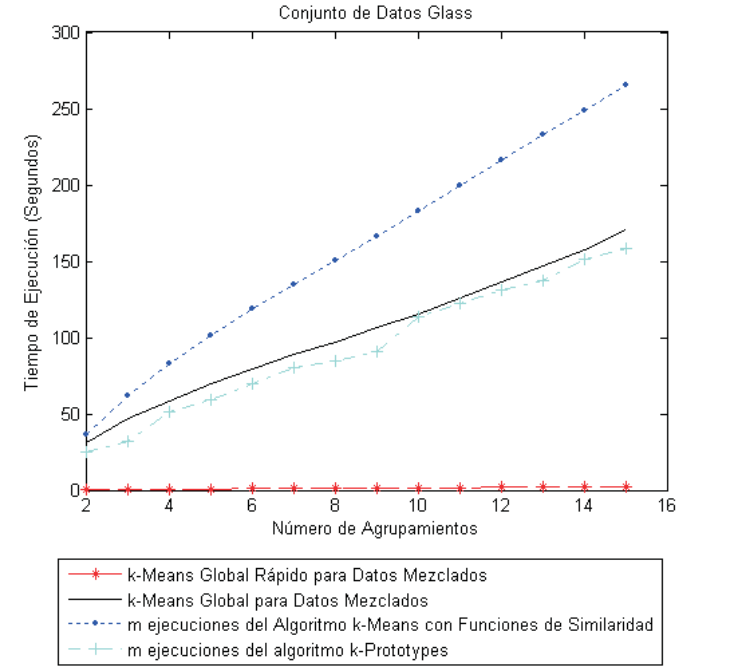


Figura 5.5: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Glass

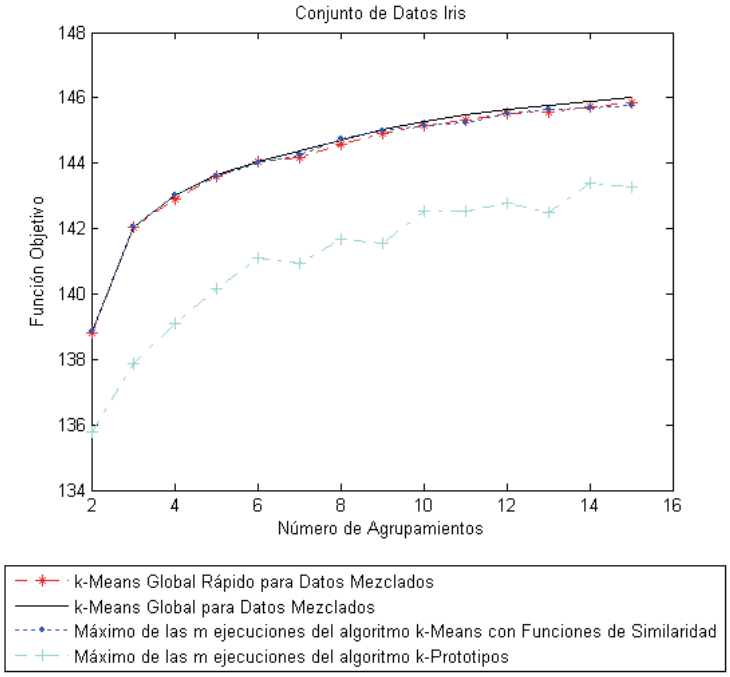


Figura 5.6: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Iris

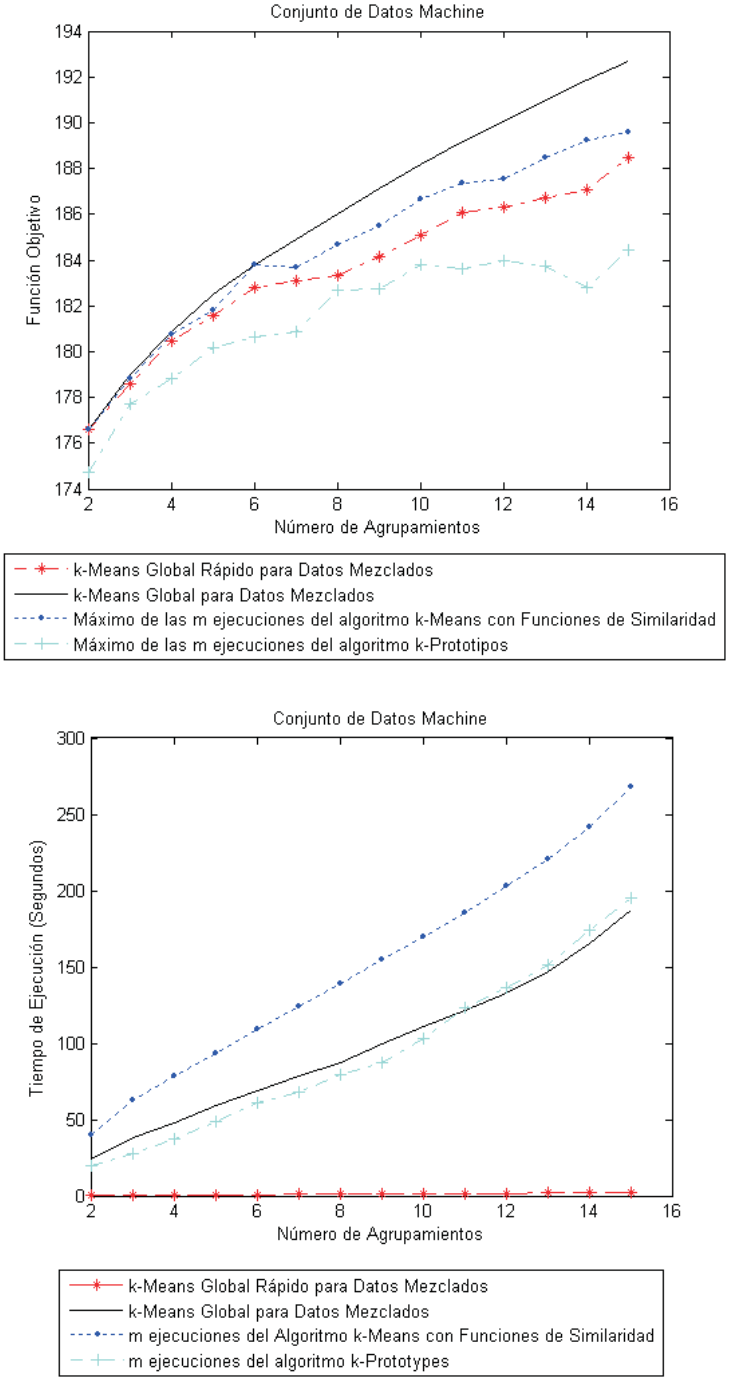
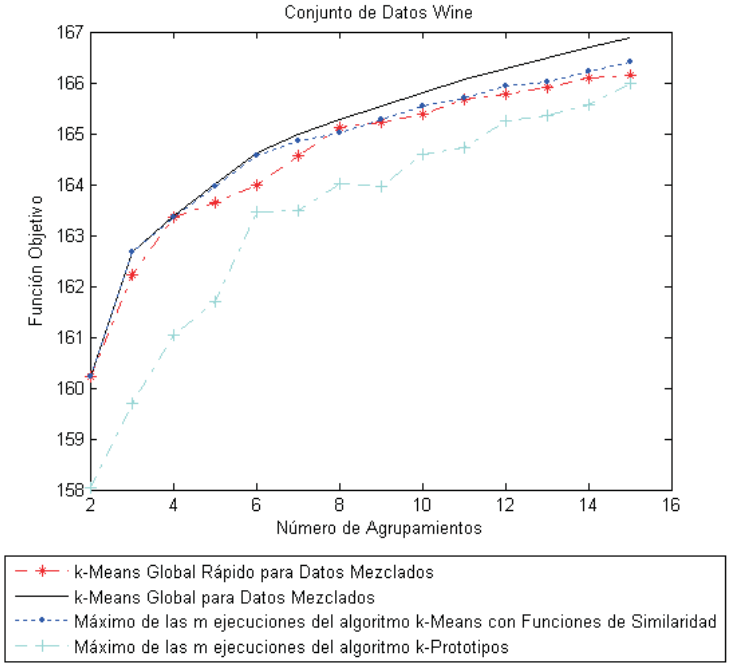


Figura 5.7: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Machine



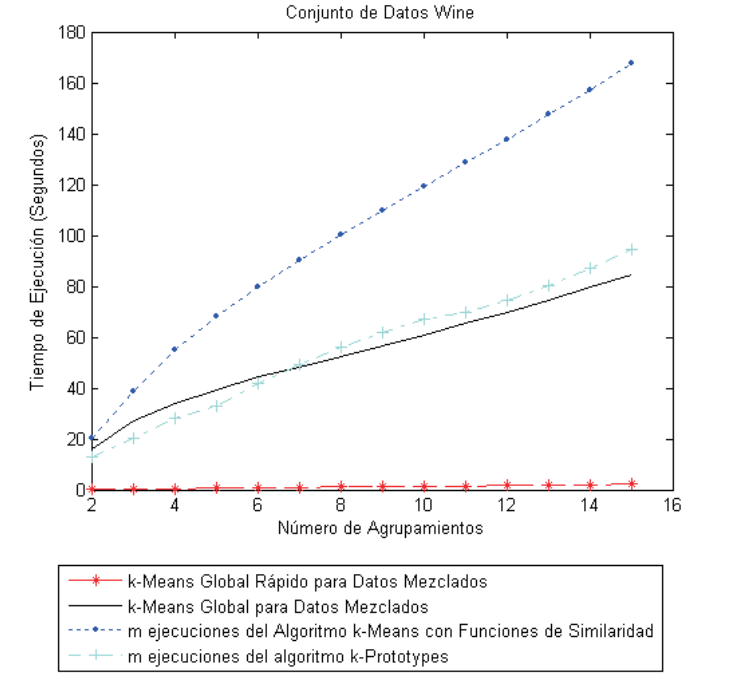


Figura 5.8: Resultados obtenidos al aplicar los algoritmos k-Means con Funciones de Similaridad, k-Prototypes, k-Means Global para Datos Mezclados y k-Means Global Rápido para Datos Mezclados al conjunto de datos Wine

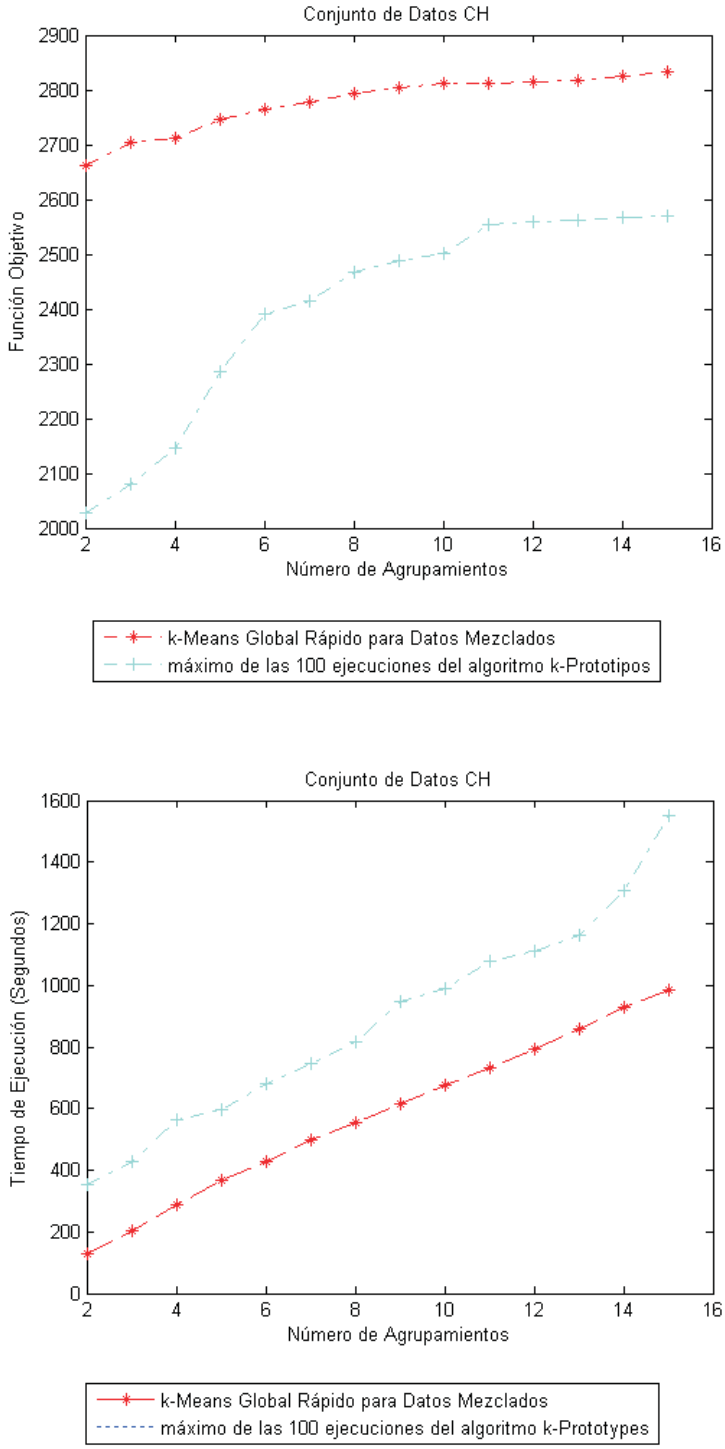


Figura 5.9: Resultados obtenidos al aplicar los algoritmos k-Means Global Rápido para Datos Mezclados y k-Prototypes al conjunto de datos Mushrom

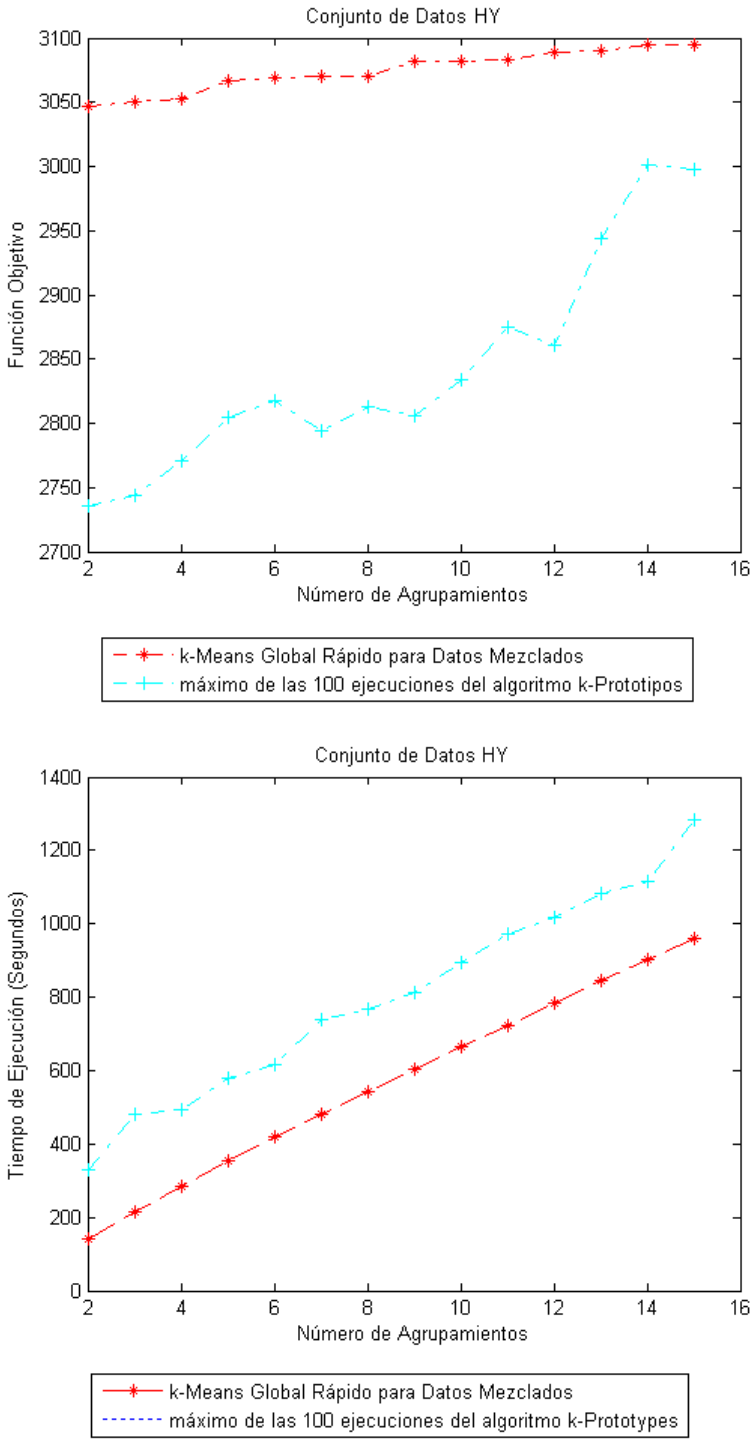


Figura 5.10: Resultados obtenidos al aplicar los algoritmos k-Means Global Rápido para Datos Mezclados y k-Prototypes al conjunto de datos HY

Por otro lado, a partir de las figuras 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 y 5.8 se puede observar que el algoritmo k-Means Global Rápido para Datos Mezclados tuvo un tiempo de ejecución mucho menor que el del algoritmo k-Means Global para Datos Mezclados, lo que se debe principalmente a que en cada iteración no ejecuta $m-(k-1)$ veces el algoritmo k-Means con Funciones de Similaridad.

En las mismas figuras, se puede observar que el tiempo de ejecución del algoritmo propuesto en esta sección es mucho menor que el de las m ejecuciones de los algoritmos k-Means con Funciones de Similaridad y k-Prototypes.

En las figuras 5.9 y 5.10 y se muestran los resultados obtenidos al usar el algoritmo k-Means Global Rápido para Datos Mezclados en conjunto de datos más grandes (CH y HY) en donde el algoritmo k-Means Global para Datos Mezclados no puede ser usado debido a su alto costo computacional. En las mismas figuras también se muestran los resultados obtenidos al ejecutar 100 veces el algoritmo k-Prototypes sobre dichos conjuntos de datos.

A partir de tales resultados se puede observar que el algoritmo k-Means Global Rápido para Datos Mezclados obtiene mejores resultados que las 100 ejecuciones del algoritmo k-Prototypes con un tiempo de ejecución mucho menor.

5.3.3. Discusión

Con base en los resultados obtenidos en los experimentos realizados, se pueden concluir los siguientes puntos:

- El algoritmo k-Means Global Rápido para Datos Mezclados reduce considerablemente el tiempo de ejecución del algoritmo k-Means Global para Datos Mezclados sin afectar considerablemente la calidad de la solución.
- Debido a que el algoritmo k-Means Global Rápido para Datos Mezclados reduce considerablemente el costo computacional del algoritmo k-Means Global para Datos Mezclados, puede ser usado en conjuntos de datos en donde este último no.
- El algoritmo k-Means Globl Rápido obtiene, en la mayoría de los

casos, mejores resultados que los obtenidos con el máximo de las m ejecuciones del algoritmo k-Means con Funciones de Similaridad y, en todos los casos, mejores resultados que los obtenidos con las m ejecuciones del algoritmo k-Prototypes a un costo computacional mucho menor.

Capítulo 6

Conclusiones

El Reconocimiento Lógico Combinatorio de Patrones se basa en un manejo adecuado de la información. Este enfoque, permite plantear modelos y procesos que se apeguen mejor a la realidad de los problemas que se desea resolver, dando a las variables un trato apropiado.

Dentro de este contexto, en esta tesis se propuso el algoritmo ***k*-Means Global para Datos Mezclados**, el cual es un algoritmo de agrupamiento restringido que permite trabajar con conjuntos de datos en donde los objetos se encuentran descritos por medio de atributos numéricos y no numéricos simultáneamente así como con ausencia de información. Este algoritmo busca una solución global y no depende de las condiciones iniciales ni de algún otro parámetro externo. Siendo su principal desventaja su alto costo computacional, por lo que no puede ser aplicado a grandes conjuntos de datos.

Con base en los experimentos realizados en la sección 4.3 (página 32) se puede observar que el algoritmo k-Means Global para Datos Mezclados obtiene mejores resultados que los obtenidos con los algoritmos k-Means con Funciones de Similaridad y k-Prototypes.

Debido al alto costo computacional del algoritmo k-Means Global para Datos Mezclados, también se propuso otro algoritmo de agrupamiento restringido cuyo objetivo es reducir el tiempo de ejecución del algoritmo k-Means Global para Datos Mezclados. En este sentido, el algoritmo propuesto fue el algoritmo ***k*-Means Global Rápido para Datos Mezclados** el cual reduce considerablemente el tiempo de ejecución del algoritmo k-Means Global para Datos Mezclados sin afectar

considerablemente la calidad de los agrupamientos, por lo que puede ser aplicado a conjuntos de datos más grandes.

Con base en los experimentos realizados en la sección 5.3 (página 48) se puede concluir que este algoritmo siempre obtiene resultados similares o ligeramente inferiores a los obtenidos con el algoritmo k-Means Global para Datos Mezclados y en la mayoría de los casos superiores a los mejores resultados obtenidos con m ejecuciones de los algoritmos k-Means con Funciones de Similaridad y k-Prototypes.

Los algoritmos propuestos en esta tesis son una buena opción en problemas en donde los objetos se encuentren descritos por medio de un conjunto de atributos numéricos y no numéricos simultáneamente, este tipo de problemas es frecuente en ciencias poco formalizadas. Siendo especialmente útiles en problemas donde se desee una alta calidad en el resultado.

Ambos algoritmos fueron programados en C++ usando bibliotecas de funciones estándar, por lo que tienen gran portabilidad y pueden ser usados en distintas plataformas, como por ejemplo Linux© y Windows©.

Es importante señalar que los resultados de esta tesis fueron publicados en:

- Saúl López Escobar, Jesús Ariel Carrasco Ochoa and José Francisco Martínez Trinidad. Global k-Means with Similarity Functions. In Alberto Sanfeliu and Manuel Lazo-Cortés, editors, *Proceeding of the 10th Iberoamerican Congress on Pattern Recognition*, volume 3773 of Lecture Notes In Computer Science, pages 392–399, La Havana, Cuba, November 2005. Springer–Verlag.

- Saúl López Escobar, Jesús Ariel Carrasco Ochoa and José Francisco Martínez Trinidad. Fast Global k-Means with Similarity Functions Algorithm. In E. Corchado et al, editors, *Proceedings of the7th Internacional Conference on Intelligent Data Engineering and Automated Learning*, volume 4224 of Lecture Notes In Computer Science, pages 512–521, Burgos, Spain, September 2006. Springer–Verlag.

6.1. Trabajo Futuro

Los algoritmos propuestos no pueden ser aplicados a grandes conjuntos de datos mezclados, por lo que como trabajo futuro se propone buscar una estrategia que permita aplicar el algoritmo k-Means Global Rápido para Datos Mezclados, u otros algoritmos que permitan trabajar con este tipo de datos, a grandes conjuntos de datos.

Otra línea de investigación dentro de este tema son las técnicas de sumarización de conjuntos de datos, esto es, técnicas que permiten definir nuevos objetos de los cuales cada uno representa un grupo de objetos dentro del conjunto de datos, y por consiguiente pueden ser usados como centros en los agrupamientos. Actualmente las técnicas de sumarización de datos sólo permiten trabajar con datos numéricos o categóricos, por lo que también se propone como trabajo futuro extender una técnica de sumarización de datos (por ejemplo, Information Bottleneck Method [25]) para trabajar con datos mezclados. Posteriormente ésta será usada en un algoritmo que lea el conjunto de datos una sola vez y actualice los centros del agrupamiento de forma dinámica mientras los datos son leídos.

Referencias

[1] J. B. MacQueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. Berkeley, University of California Press, 1967.

[2] Paul S. Bradley and Usama M. Fayyad. Refining initial points for K-means clustering. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 91–99, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[3] Shehroz S.Khan and Amir Ahmad. Cluster center initialization algorithm for K-means clustering. *Pattern Recognition Letters*, volume 25(number 11):pages 1293–1302, 2004.

[4] J. M. Pe na, J. A. Lozano, and P. Larra naga. An empirical comparison of four initialization methods for the K-means algorithm. *Pattern Recognition Letters*, volume 20(number 10):pages 1027–1040, 1999.

[5] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, volume 36(number 2):pages 451–461, 2003.

[6] N. Hussein. A fast greedy k-means algorithm. Master's thesis, University of Amsterdam. Faculty of Mathematics, Computer Sciences, Physics and Astronomy, Plantage muidergracht 24, 1018 TV Amsterdam, the Netherlands, November 2002.

[7] Javier Raymundo García Serrano and José Francisco Martínez Trinidad. Extension to C-means algorithm for the use of similarity functions. In *PKDD '99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, pages 354–359, London, UK, 1999. Springer-Verlag.

[8] José Francisco Martínez Trinidad, Javier Raymundo García Serrano, and Irene Olaya Ayaquica. C-means algorithm with similarity functions. *Computación y Sistemas*, volume 5(number 4):pages 241–246, 2002.

[9] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, volume 2(number 3):pages 283–304, 1998.

[10] Zhexue Huang. Clustering large data sets with mixed numeric and categorical values. In *Proceeding of the First Pacific-Asia Conference Knowledge Discovery and Data Mining*, pages 21–34, February 1997.

[11] José Ruiz Shulcloper, Adolfo Guzmán Arenas, and José Francisco Martínez Trinidad. *Enfoque Lógico Combinatorio al Reconocimiento de Patrones I. Selección de Variables y Clasificación Supervisada*. Editorial Politécnica, 1999.

[12] Erika Danaé López Espinoza. Selección de variables y clasificación de imágenes en eigenespacios usando reconocimiento lógico combinatorio de patrones. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2004.

[13] Jesús Ariel Carrasco Ochoa. *Sensibilidad en Reconocimiento Lógico Combinatorio de Patrones*. PhD thesis, Instituto Politécnico Nacional, 2001.

[14] Guillermo Sanchez Díaz. *Desarrollo de Algoritmos para el Agrupamiento de Grandes Volúmenes de Datos Mezclados*. PhD thesis, Instituto Politécnico Nacional, Centro de Investigación en Computación, 2001.

[15] José Ruiz Shulcloper and José Francisco Martínez Trinidad. Clasificación sin aprendizaje y con aprendizaje parcial (enfoque lógico - combinatorio). Centro de Investigaciones y de Estudios Avanzados, Instituto Politécnico Nacional, 1995.

[16] Periklis Andristos. Data clustering techniques. Technical Report CSRG-443, University of Toronto, Department of Computer Science, March 2002.

[17] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[18] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[19] Alba Cabrera E. *Nuevas Extensiones del Concepto de Testor para Diferentes Tipos de Funciones de Semejanza*. PhD thesis, Instituto de Cibernética, Matemática y Física, ICIMAF, 1997.

[20] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, November 1996.

[21] Peter Cheeseman and John Stutz. Bayesian classification (autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, pages 153–180, 1996.

[22] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, NY, 1973.

[23] Ohn Mar San, Van-Nam Huynh, and Yoshiteru Nakamori. An alternative extension of the k-means algorithm for clustering categorical data. *International Journal of Applied Mathematics and Computer Science*, volume 14(number 2):pages 241–247, 2004.

[24] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley and Sons, New York, NY, USA, 1991.