CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL
DEPARTAMENTO DE COMPUTACIÓN

Diseño e implementación de un algoritmo de aglomeraciones no sensible a valores atípicos

Tesis que presenta

Raymundo Domínguez Colín

Para obtener el grado de

Maestro en Ciencias en Computación

Director de la Tesis: **Dr. Debrup Chakraborty**

México, D. F. Diciembre 2009

A la memoria de
María de la Luz Colín Mendoza
mi madre...
daría
todo lo que soy
por haberte conocido

Agradecimientos

Quiero agradecer sinceramente a muchas personas e instituciones por haber contribuido en mi formación profesional. Primero que nada, comenzaré agradeciendo al Consejo Nacional de Ciencia y tecnología (CONACyT), por el apoyo económico que me proporcionó para la realización de mis estudios de Maestría y mencionar que este trabajo no hubiera sido posible sin su ayuda.

A mi asesor, el **Dr. Debrup Chakraborty**, por ser una de las personas más inteligentes que he conocido y que me dio la oportunidad de trabajar bajo su asesoría en un área muy interesante y desconocida para mí; agradezco todas sus enseñanzas y sobre todo, su gran paciencia.

A los lectores y revisores de esta tesis, Dr. Francisco José Rodríguez Henríquez y Dr. Luis Gerardo de la Fraga, por sus acertados comentarios y correcciones que enriquecieron este trabajo.

Le doy las gracias al Dr. Abdiel Cáceres González por animarme a continuar mis estudios de maestría en esta institución...aunque no me haya advertido de lo duro que sería.

A mi familia, por su apoyo incondicional a toda prueba, a mi padre, por ser mi modelo de sabiduría, tenacidad y esfuerzo, a mis hermanos...ojalá nuestros caminos se reencuentren algún día.

De igual manera, agradezco al COMECyT por la beca terminal que me proporcionó por seis meses, ya que con este apoyo pude concluir mis estudios sin preocupaciones (de dinero claro).

A todos los profesores que nos supieron bajar de la nube de la universidad y mostrarnos que no se deja de aprender jamás. A la Sra. Sofía Reza por su actitud siempre alegre y por ayudarme con los siempre engorrosos trámites administrativos, a Flor que se fue sin su baile y Felipa por sus correos sin red.

Agradezco de manera especial a Lil María Rodríguez por su maravillosa compañía durante estos dos años, por estar conmigo en esas noches de desvelos, enfermedad, malas comidas y pocas horas de sueño.

A mis compañeros Andrés, Beto, Cristian, Julian, Jhonny, Maday, al Paco, Pam, Pau, y a todos los que me faltaron que, por ser demasiados, es imposible mencionarlos por su nombre.

A los siempre incompletos equipos de fútbol y basquet...gracias por participar

Contenido

Índice de Figuras

Índice de Tablas

List of Algorithms

Resumen

El problema de *análisis de aglomeraciones* (cluster analysis) ha estado abierto por décadas. Los algoritmos de aglomeración que se han desarrollado tales como el *k medias, c-medias difuso* entre otros, proponen diversos metodologías para tratar este problema. A la fecha aún no existe una solución óptima que funcione para todos los problemas de aglomeración.

Uno de los factores que más afectan el desempeño de estos algoritmos es la presencia de *valores atípicos* (*outliers*). Los valores atípicos son muestras que numéricamente no pertenecen al conjunto de datos.

Siempre se requiere remover los valores atípicos de un conjunto de datos para que un algoritmo de aglomeración funcione correctamente. En la literatura se ha tratado el problema de los valores atípicos con una metodología llamada *aglomeración de ruido* (*noise clustering*). En ésta se considera una entidad conocida como *aglomerado de ruido*, separada de todos los datos de un conjunto por medio de una *distancia de ruido* (*noise distance*).

Uno de los principales problemas de este método es que considera que el aglomerado de ruido es equidistante a todos los puntos, es decir, que la distancia de ruido que los separa es la misma, pero ésta no es una apreciación realista.

En esta tesis se proponen algunas variantes al método de aglomerado de ruido. Entre estas variantes se maneja una manera distinta de estimar la distancia de ruido y se proponen nuevas ideas para desarrollar este tipo de algoritmos. Nuestro método fue validado con muchas pruebas sobre diferentes conjuntos de datos. Se ha desarrollado una aplicación de este método sobre un problema de la vida real.

Abstract

The problem of cluster analysis has been open for many decades. Clustering algorithms as $k-means, Fuzzyc-means$ among others, propose diverse methodologies to solve this problem. To date there does not exist an optimal solution that works for all kind of clustering problems.

One of the causes that affect the performance of clustering algorithms is the presence of outliers. The outliers are points which are numerically distant from the rest of the data.

In order to make a clustering algorithm works properly, it is necesary to remove the outliers from the data set. In the literature the problem of outliers have been treated with a specific methodology called *noise clustering*. Where it is assume the existance of a separate noise cluster and also there is a *noise distance* from the noise cluster to all data points.

One of the main drawbacks of the previous methods was that they assume that the noise prototype is equidistant from all data points, i.e., the noise distance is the same for all data. This assumption is unrealistic. In this thesis we propose several variants of noise clustering. We propose a new noise distance which does not suffer from this unrealistic assumption.

In this thesis we propose some variants to this noise cluster method. We handle a new noise distance and further propose new ideas to develop this kind of algorithms. We provide extensive simulations on different data sets. Finally we describe an application of the new algorithm in a real life problem.

Capítulo 1

Introducción

Estudio en la duda, acción en la fe

Andrés de la Cruz Michael. UJAT Centro, Villa hermosa Tabasco (1939)

1.1 Análisis de aglomeraciones

Los humanos tenemos la capacidad de reconocer patrones, identificar estructuras y clasificar objetos de acuerdo a un grado de asociación [2]. La práctica de organizar objetos de cualquier naturaleza de acuerdo a su similitud es la base de varias teorías en la ciencia, de hecho el organizar datos en grupos significativos es una manera de entender y adquirir conocimiento.

En el área de la *Inteligencia Artificial* existe la rama del *Aprendizaje Automatizado*, y dentro de esta última existen dos métodos importantes: el aprendizaje supervisado y el no supervisado [3]. La diferencia básica entre uno y otro consiste en la presencia o no, de un valor conocido como *variable dependiente*. Para el primer caso, el identificador del grupo o clase es indispensable, mientras que para el segundo no se requiere conocer la clase a predecir, el propio sistema se organiza para identificar grupos de datos que, bajo características comunes, se agrupen en una misma clase.

Considerado como uno de los problemas más importantes dentro del aprendizaje no supervisado, el *análisis de aglomeraciones* (*cluster analysis*) es el estudio formal de los algoritmos y métodos que sirven para agrupar u organizar conjuntos de datos; ésto es conocido como *análisis exploratorio de datos*, cuyo objetivo principal es descubrir estructuras o ciertos grupos dentro de un conjunto de datos. Este método es aplicado en una gran variedad de disciplinas de ingeniería y científicas tales como biología, psicología, medicina, marketing, reconocimiento de patrones entre muchas otras [4].

Un conjunto de datos u objetos es una colección de valores que muestran las características de los elementos que lo componen; es decir, un objeto es descrito por medidas o por la relación que éste tenga con otros. Si hablamos de conjuntos de objetos que ya han sido clasificados, cada elemento de este conjunto es señalado con una etiqueta, la cuál indica su pertenencia a alguna clase específica (*class label*).

Los algoritmos de *análisis de aglomeraciones* no utilizan dicha etiqueta para analizar los datos. A diferencia de los *problemas de clasificación*, en donde se conoce a cuál clase pertenece cada uno de los elementos, y el objetivo es predecir a cuál grupo pertenece un nuevo punto, el *análisis de aglomeraciones* busca descubrir el número y la composición de los grupos, es decir, una organización válida y conveniente de los datos.

A estas organizaciones se les conoce como *aglomerados o aglomeraciones* (clusters). Un aglomerado consta de un número de objetos *similares* entre sí, que han sido colectados o agrupados juntos. La definición que se utilizó en este trabajo para definir lo que es un aglomerado es la de Everitt [5], que dice lo siguiente:

"*Un aglomerado puede ser descrito como regiones conectadas en un espacio multidimensional, que contienen una* **alta densidad** *de puntos, separados de otros por una región que contiene una relativa* **baja densidad** *de éstos*".

Aunque identificar un aglomerado pueda parecer una tarea sencilla, en realidad no es muy claro la manera en la que se hace. De hecho es difícil llegar a desarrollar un procedimiento general para obtener un aglomerado. Esto es porque los conjuntos de datos revelan diferentes *formas* y *tamaños* de los aglomerados.

Entonces, el problema crucial al identificar aglomerados consiste en especificar qué tan similares son (o serán) y cómo se medirá esta similitud. En la figura 1.1 se aprecian algunos aglomerados en dos dimensiones, y surge la pregunta ¿Cuántos aglomerados hay en la figura?; en un nivel alto de similitud, se observan cuatro aglomerados, pero a un nivel local, digamos, con un bajo umbral de similitud, se perciben nueve aglomerados. ¿Cuál es la respuesta correcta? El mirar la figura tomando en consideración diferentes niveles de similitud, ayuda a analizar su estructura.

Lo que nos conduce a preguntar ¿Cuál es el número de aglomerados que deben buscarse? Esta pregunta no tiene una respuesta definida, ya que en la mayoría de problemas de aglomeraciones, este parámetro es un valor desconocido.

Existen muchas técnicas basadas en *validaciones de aglomerados* (cluster validation) que permiten determinar el número de aglomerados; sin embargo, independientemente de este número el algoritmo es el mismo, sólo el número de aglomerados que se hayan dado como valor de entrada es analizado.

Dado un ejemplo visual como el de la figura 1.1, una persona puede identificar sin problemas los aglomerados que se aprecian. Sin embargo, lo cierto es que distintos individuos apreciarían diferentes aglomerados dentro del mismo conjunto de datos.

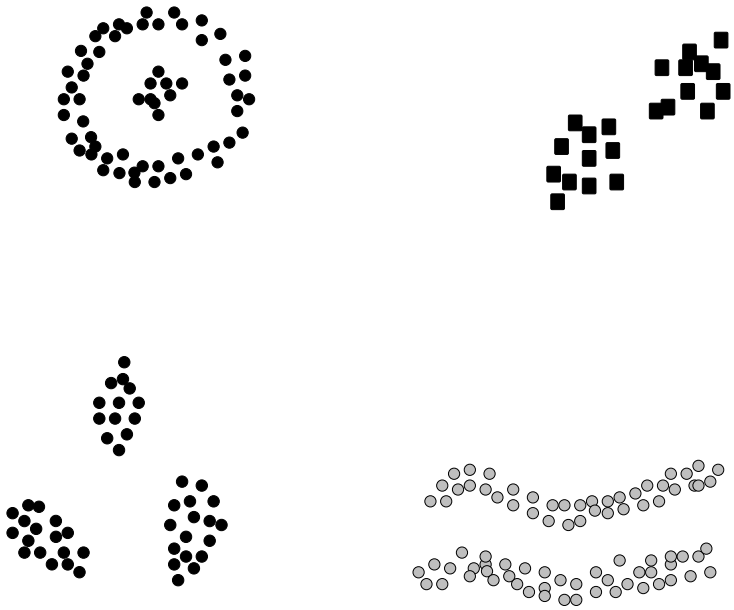


Figura 1.1: Aglomerados de puntos en dos dimensiones

De acuerdo a lo anterior, se establece que el objetivo de los algoritmos de *análisis de aglomeraciones* es el siguiente: Dado un conjunto formado por elementos de cualquier naturaleza (puntos en un plano, documentos, archivos etc), se realice una reagrupación de los mismos en aglomerados distintos, de tal modo que cada uno de éstos contenga aquellos elementos que sean muy semejantes entre sí y a la vez que sean diferentes de los elementos de los otros aglomerados.

Y esto da lugar a otra pregunta ¿qué es lo que constituye un buen aglomerado? la respuesta no es sencilla pues no hay un criterio absoluto que pueda determinar si el resultado final de la aglomeración es el correcto.

En consecuencia, es el usuario quien deberá proveer este criterio, de tal manera que el resultado del algoritmo satisfaga sus necesidades. Y aún cuando esto ocurra, sucede a menudo que un algoritmo de *análisis de aglomeraciones* que funciona bien con un conjunto de datos, puede no funcionar de manera adecuada con otro.

Esta es una de las razones por las que el campo de estudio de este problema es amplio y ofrece problemas interesantes a resolver. Debido a que los elementos que intervienen en el *análisis de aglomeraciones* pueden ser de cualquier naturaleza, tales como conjuntos de puntos en un plano, archivos, *píxeles* de una imagen entre otros; en este trabajo de investigación se referirá a éstos como datos, puntos, objetos o elementos de manera indistinta.

1.2 Algoritmos de aglomeración

Existen diversos algoritmos de *análisis de aglomeraciones* que proporcionan técnicas para lograr una correcta identificación de aglomerados dentro de un conjunto de datos. Éstos pueden ser clasificados como:

- Aglomeración exclusiva
- Aglomeración de traslape
- Aglomeración basada en jerarquías
- Aglomeración probabilística

Dentro de este trabajo de investigación, nos enfocaremos a los primeros dos. El primero corresponde al algoritmo de k-medias (*k-means*) y el segundo al de c-medias difuso (*fuzzy c-means*). Estos algoritmos se explicarán con más detalle en secciones posteriores.

Normalmente, en un conjunto de datos, cada uno de los objetos está compuesto por un cierto número de características (*features*) y una etiqueta (*class label*) que lo identifica como miembro de una clase en particular. El algoritmo que sea utilizado debe lograr identificar aquellos elementos que pertenecen a cierta clase sin conocer previamente la etiqueta de cada elemento.

A manera de ejemplo observe la figura 1.2 en la que se observan dos conjuntos de puntos en dos dimensiones; a primera vista parecen ser dos aglomerados bien definidos (imagen de la izquierda). Como este ejemplo se encuentra en un espacio en dos dimensiones, cada unos de los puntos consta de dos características (las coordenadas x y y). La imagen de la derecha muestra el resultado de aplicar uno de los algoritmos de aglomeración básicos, el de k-medias.

El resultado del algoritmo son dos nuevos puntos llamados centroides o *prototipos*. Un prototipo es un punto cuyas características (en este caso las coordenadas) son iguales al promedio de los valores de todos los elementos que pertenecen a éste (la cruz en color negro que se observa al centro de los aglomerados en la imagen de la derecha).

Aunque este ejemplo básico da una idea del funcionamiento de los algoritmos, es evidente que se han escogido dos aglomerados debido a que se había realizado una inspección visual de las imágenes. La pregunta es ¿Puede desarrollarse un procedimiento general que realice una agrupación similar de los elementos del conjunto cuando existan más de dos variables?

Decidir el número de aglomerados a descubrir no es algo sencillo; una manera simple de hacerlo sería encontrar todos los aglomerados posibles del conjunto de datos e ir

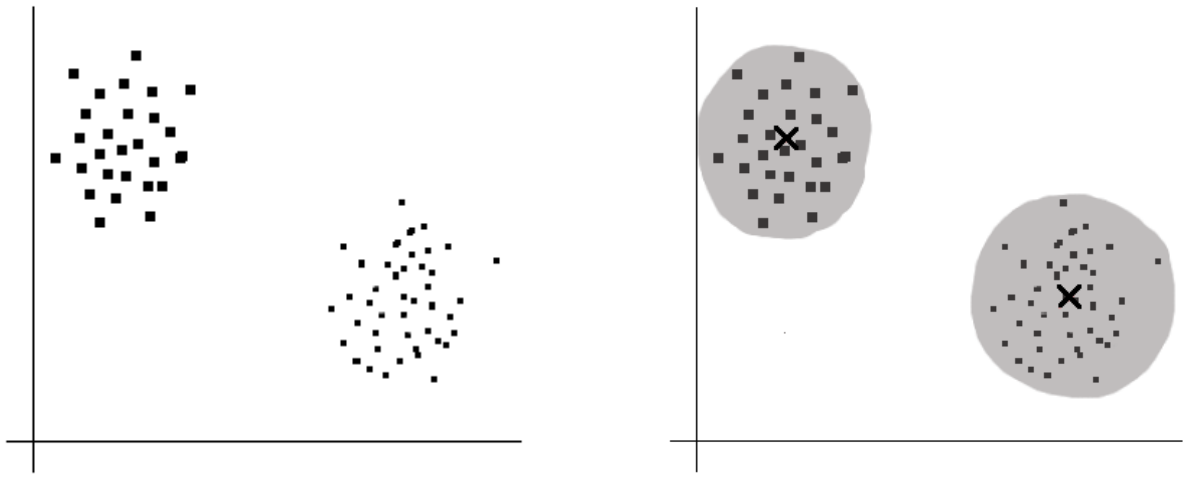


Figura 1.2: Ejemplos de aglomerados. Izq. Conjunto de puntos con características similares. Der. El resultado (esperado) del algoritmo

resumiéndolos en aglomerados más grandes de acuerdo a la distancia que exista entre ellos. Sin embargo esto no es viable porque en la práctica el número de los posibles aglomerados puede ser muy grande y realizar esto puede estar fuera del alcance aún de los equipos de cómputo actuales. Por lo que sigue siendo un problema abierto.

1.3 El problema de los valores atípicos

Dentro del análisis de aglomerados, existen diversos factores que afectan el desempeño de los algoritmos. Uno de estos, y quizás el más importante, es la presencia de los *valores atípicos* (*outilers*). En estadística, un valor atípico es una muestra de los datos que se desvía de las demás muestras de tal modo que parecen haber sido generadas por un mecanismo diferente que la mayor parte de los datos [6].

En algunos casos, el que una muestra se encuentre muy lejos del promedio de las demás se considera razonable, pues pueden ocurrir errores sistemáticos o fallas en la manera de generarlas. Los valor atípicos indican fallas en los datos, procedimientos erróneos o áreas en las que ciertas formas de generación de muestras podrían no ser válidas.

Dentro del análisis de datos, una solución sería que todos los valores atípicos fueran correctamente identificados y removidos del conjunto de datos. Sin embargo, los algoritmos comunes de aglomeración (como los ya mencionados) no consiguen identifi-

carlos correctamente; éstos son considerados como parte de los puntos válidos y como consecuencia se tiene una aglomeración deficiente de los datos.

Considérese la figura 1.3. Se observan dos aglomerados bien definidos y además tres valores alejados de los demás (valores atípicos) exactamente a la mitad entre los dos aglomerados. Un algoritmo limitado como el k-medias asignará los tres puntos centrales en alguno de los dos aglomerados como se ve en la imagen de la derecha.

Al igual que en el ejemplo anterior, la salida del algoritmo son dos centroides, cuyas coordenadas son los cruces de las líneas en la imagen de la derecha. Es evidente que el centroide del aglomerado formado por los símbolos '$\star$', se encuentra muy alejado del aglomerado que se aprecia visualmente. Y esto se debe a que la presencia de esos tres valores atípicos afectan el desempeño del algoritmo.

Este es una de las principales motivaciones para este trabajo de investigación. Se pretende realizar una aportación a los algoritmos existentes, de tal manera que se logre reducir en una medida significativa el efecto de los valores atípicos en el desempeño de los algoritmos.

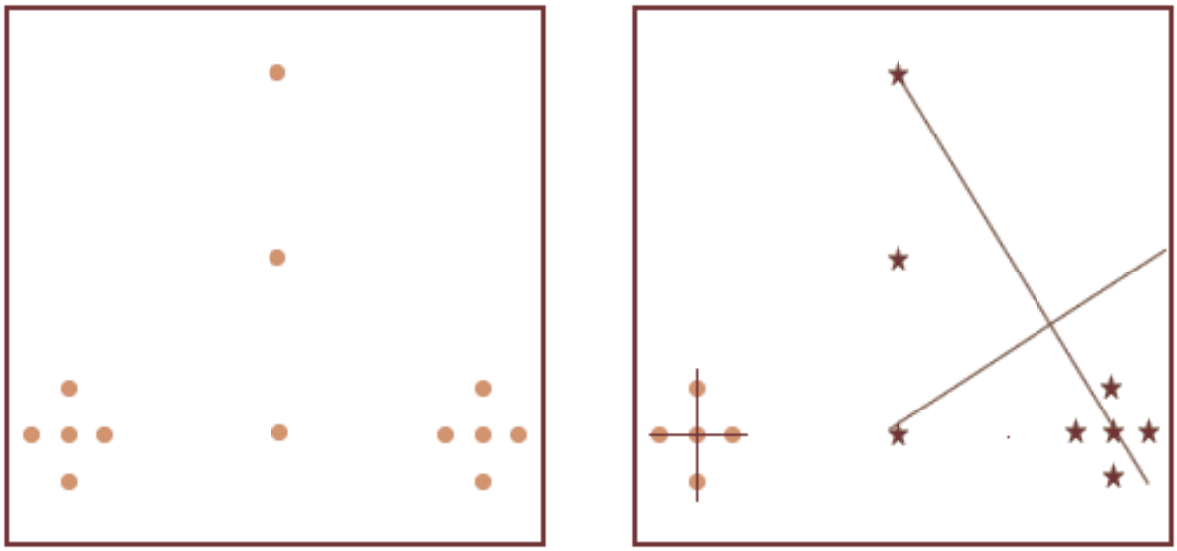


Figura 1.3: Ejemplo básico de aglomeración. Izq. dos aglomerados y tres valores atípicos entre ellos. Der. el Resultado del algoritmo k-medias muestra los tres puntos clasificados en uno de los aglomerados.

1.3.1 Objetivo principal de la tesis

El objetivo general de este trabajo de tesis puede dividirse en dos partes. La primera consiste en diseñar un algoritmo de aglomerado eficiente, que sea menos sensible a los efectos provocados por los valores atípicos existentes y que sea una contribución al

marco teórico en el estudio y desarrollo de los algoritmos de aglomeración. Para ello se ha propuesto una nueva metodología que toma como punto de partida los algoritmos de aglomeración existentes.

Bajo esta propuesta, se han modificado los algoritmos existentes para mejorar la identificación de valores atípicos y lograr un buen proceso de aglomeración a partir de un conjunto de datos dado superando los algoritmos existentes.

La segunda parte consiste en implementar dicho algoritmo en el desarrollo de una aplicación que resuelva un problema de la vida real y cuyo desempeño sea satisfactorio.

1.4 Organización de la tesis

Esta tesis está organizada en siete capítulos. Éste, es introductorio y da un panorama general de lo que se propone realizar. Se presentan conceptos básicos tales como aglomerado y aglomeración; además se explica cuál es el efecto que produce la presencia de valores atípicos en los análisis de aglomeraciones.

El segundo capítulo es el marco teórico de la tesis. Se describe de una manera más formal el objetivo del análisis de aglomeraciones. También se da una breve descripción de los algoritmos de aglomeración que existen, así como de los trabajos que se han desarrollado que atacan el problema de los valores atípicos y las soluciones que han propuesto. Se presenta un artículo que describe una de las aplicaciones más recientes del algoritmo c-medias difuso en una aplicación médica.

En el tercer capítulo se presentan las métricas utilizadas para medir el desempeño del algoritmo desarrollado. Éstas métricas servirán para realizar comparaciones entre éste y los algoritmos que se han desarrollado en la literatura respecto al problema principal. Para realizar estas comparaciones, se implementaron dos algoritmos que atacan el problema de los valores atípicos, además del desarrollado en este trabajo.

El cuarto es el capítulo medular de esta tesis. Se presenta formalmente la metodología utilizada, además se explica cuál es la aportación más importante de este trabajo, el algoritmo desarrollado y los resultados de las pruebas realizadas. Éstas últimas se presentaron en tablas comparativas con los resultados de los algoritmos puestos a prueba.

En el quinto capítulo se describen dos variantes que surgieron durante el desarrollo del algoritmo. En éstas se propone un enfoque ligeramente distinto pero que mantiene la idea original de la propuesta planteada. En la primera se basa en el cálculo de un valor al que se le ha llamado *hiper radio* y la segunda está basada en el *árbol generador mínimo* (*Minimal Spanning Tree*). Se muestran los resultados de la comparación de estas dos variantes.

El sexto capítulo es en donde se presenta el enfoque práctico del algoritmo desarrollado. Se trata del análisis de *imágenes de resonancia magnética*, se extrae información de las imágenes que serán analizadas para formar un conjunto de datos que sirven como entrada para el algoritmo.

En el séptimo y último capítulo se describen las conclusiones del trabajo desarrollado y el trabajo futuro en los que se pretenda aplicar el algoritmo.

Finalmente se incluye un apéndice en el que se describen principalmente detalles de implementación y un manual de usuario en donde se dan las instrucciones para la ejecución del programa desarrollado.

Capítulo 2

Algoritmos de análisis de aglomeración existentes

> Me llevó quince años descubrir que no tengo talento para escribir. Pero no pude dejar de hacerlo, pues para ese entonces ya era demasiado famoso.
>
> *Robert Benchley, narrador*

En este capítulo se muestra el estado del arte de los algoritmos de *análisis de aglomeraciones*. Se describen algunos de estos algoritmos, haciendo énfasis en aquellos que tratan el problema del efecto producido por los valores atípicos. Se introduce el concepto de *aglomerado de ruido* y cuál es su papel dentro del análisis de aglomeraciones.

2.1 Localización particiones dentro de un conjunto de datos

En secciones anteriores se mencionó que los algoritmos de análisis de aglomeraciones requieren de una medida de *similaridad* (*cercanía* o *proximidad*) para la observación de los conjuntos de datos. Es común utilizar el concepto de *distancia*. Regularmente se utiliza la distancia euclidiana (aunque se pueden utilizar otros criterios) cuando se habla de distancias entre puntos, ya que ésta maneja un mejor significado físico (ver apéndice A.1 Medidas de distancia, en la página 71).

La base de un algoritmo de aglomeraciones es encontrar *particiones* dentro de un conjunto de datos, de tal manera que puntos que sean semejantes pertenezcan a la misma partición y al mismo tiempo sean diferentes de los puntos de las otras particiones.

Formalmente, dado un conjunto $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, una partición c en X puede ser convenientemente representada con una matriz U de tamaño $c \times n$ llamada matriz de

particiones o bien *matriz de pertenencia*. Hay varios tipos de particiones definidas en un conjunto. Los dos tipos que son de interés para este trabajo son las particiones *rígidas* y las particiones *difusas*.

Una *partición rígida* (*hard partition*) está representada por una matriz $U = [u_{ij}]$ donde $u_{ij} \in \{0, 1\}$ y $\sum_{i=1}^{c} u_{ij} = 1 \ \forall \ j$. Por otro lado, en una *partición difusa* la matriz U se define en el intervalo $0 \leq u_{ij} \leq 1$ y además $\sum_{i=1}^{c} u_{ij} = 1$.

2.2 Algoritmos de aglomeración

El análisis de aglomeraciones es una herramienta importante en muchas disciplinas científicas. Aunque existen muchos métodos de este tipo, ninguno es capaz de resolver todos los problemas de aglomeraciones que existen. Además, estos métodos se ven afectados por la presencia de los valores atípicos, que como ya se mostró en el capítulo anterior, da lugar a resultados deficientes.

2.2.1 K-medias

En el *análisis de aglomeraciones*, el algoritmo básico es el de $k-$medias. Este algoritmo, aún con sus limitantes, no deja de ser eficiente para algunos casos y debe conocerse dentro del estudio de los algoritmos de aglomeraciones.

El algoritmo de k-medias fue desarrollado en 1967 por J.B. MacQueen [7]. Siendo un algoritmo voráz y también uno de los más simples, básicamente asigna cada objeto del conjunto de datos al aglomerado más cercano de acuerdo a sus características. El objetivo de este algoritmo es minimizar la siguiente función objetivo:

$$V = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in X} \| \boldsymbol{x}_j - \boldsymbol{v}_i \|^2 \tag{2.1}$$

donde $X = \{\boldsymbol{x}_i : \boldsymbol{x}_i \in \mathbb{R}^p, \ i = 1, 2, \ldots, n\}$ es el conjunto de datos. $\|\boldsymbol{x}_j - \boldsymbol{v}_i\|^2$ es una medida de distancia (regularmente es la distancia euclidiana) entre el j-ésimo objeto $\boldsymbol{x}_j$ y el i-ésimo aglomerado $\boldsymbol{v}_i$, e indica la distancia entre cada $\boldsymbol{x}_i$ de X y los k aglomerados, donde k es el número de aglomerados, $n = |X|$ y $k \leq n$.

Entrada: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, k, ε, *band*=true
Salida : Centroides $\boldsymbol{v}_i$, con i=1, $\ldots$, k

Seleccionar k elementos de X y llamarlos $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$;
while *band* **do**
  for $i = 1$ **to** k **do**
    $S_i = 0$;
  end
  for $j = 1$ **to** n **do**
    $d_l = \left\|\boldsymbol{x}_j - \boldsymbol{m}_l\right\|$ para toda $l = 1, 2, \ldots, k$;
    $d_p = min\{d_1, \ldots, d_k\}$;
    $S_p = S_p \cup \{\boldsymbol{x}_j\}$
  end
  for $i = 1$ **to** k **do**
    $\boldsymbol{m}_i^{act} = \frac{1}{|S_i|} \sum_{\boldsymbol{x} \in S_i} \boldsymbol{x}$
  end
  if $\frac{1}{k} \sum_{i=1}^{k} \left\|\boldsymbol{m}_i^{ant} - \boldsymbol{m}_i^{act}\right\| < \varepsilon$ **then** *band*=false;
  else $\boldsymbol{m}_i^{ant} = \boldsymbol{m}_i^{act}$ $\forall$ i;
end
return $\boldsymbol{m}^{act}$;

Algorithm 1: Algoritmo de aglomeración $k-$medias

El algoritmo 1 muestra el algoritmo de $k-$medias. Se tiene el conjunto de datos X, el número k de aglomerados y ε es la condición de paro (normalmente un valor muy pequeño).

La manera de inicializar los prototipos iniciales $\boldsymbol{m}_j$ con $j = 1, 2, \ldots, k$, es seleccionarlos del conjunto de datos de manera aleatoria; sin embargo el resultado depende principalmente de los valores iniciales de los prototipos, y con frecuencia se obtiene más de un buen resultado; por esta razón no es fácil saber cuál es el óptimo global.

Este algoritmo establece partición rígida de los datos, así que todos los puntos y aún los valores atípicos, son asignados a alguno de los aglomerados y esto afecta el desempeño del algoritmo.

El algoritmo $k-$medias es un método simple que puede adaptarse para resolver diferentes tipos de problemas. Su extensión es el algoritmo de c-medias difuso (*Fuzzy c-means*) que se explica a continuación.

2.2.2 c-medias difuso

El algoritmo c-medias difuso (CMD) es una versión general del $k-$medias. Desarrollado en 1973 por Dunn [8] y modificado e implementado por Bezdek [9], este algoritmo permite a los objetos pertenecer a todos los aglomerados con un grado de pertenencia, lo que indica una fuerte asociación entre éstos y los aglomerados.

La idea básica es similar a la del $k-$medias. Se presupone que el número k de aglomerados es conocido e intenta minimizar la función objetivo. Esta función permite una mejor formulación para el criterio de la aglomeración. La minimización funcional es:

$$\boldsymbol{J}(U, v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \left\| \boldsymbol{v}_i - \boldsymbol{x}_k \right\| \tag{2.2}$$

donde $\boldsymbol{x}_k$ es el vector de características del punto k, con $k = 1, 2, \ldots, n$ número de datos, y $\boldsymbol{v}_i$ es el prototipo del i-ésimo aglomerado con $i = 1, 2, \ldots, c$; $\left\| \boldsymbol{v}_i - \boldsymbol{x}_k \right\|$ es la distancia euclidiana; m es un parámetro conocido como *difuminador* y $1<m<\infty$; normalmente se escoge $m = 2$. u_{ik} es el *grado de pertenencia* (*membership*) del punto k al aglomerado i. Este *grado de pertenencia* debe cumplir estrictamente con la siguiente restricción:

$$\sum_{i=1}^{c} (u_{ik}) = 1 \ \ \forall \, k \tag{2.3}$$

El difuminador m no tiene mayor efecto cuando se tiene una partición *rígida*, i.e. cuando sólo se permiten los valores $\{0, 1\}$ para indicar la pertenencia de un punto a un aglomerado como en el caso del $k-$medias. A diferencia de éste, el *grado de pertenencia* es *difuso*, i.e. puede tomar valores dentro del intervalo [0, 1]. La ecuación (2.2) representa un problema de *optimización no lineal* que es resuelto normalmente por los multiplicadores de Lagrange. Sea entonces:

$$J_L = J(\boldsymbol{u}, \boldsymbol{v}) + \sum_{k=1}^{n} \lambda_k \left(1 - \sum_{i=1}^{c} u_{ik} \right) \tag{2.4}$$

donde λ_k son los multiplicadores de Lagrange. Ahora de las condiciones necesarias

$$\frac{\partial}{\partial \boldsymbol{u}_{ik}} J_L = 0, \tag{2.5}$$

y

$$\frac{\partial}{\partial \boldsymbol{v}_i} J_L = 0, \tag{2.6}$$

donde $i = 1, \ldots, c$ y $k = 1, \ldots, n$, se realizan las derivadas para obtener:

$$u_{ik} = \frac{1}{\sum_{j=1}^{n} [(d_{ik})^2/(d_{jk})^2]^{1/(m-1)}} \tag{2.7}$$

y

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{n} (\boldsymbol{u}_{ik})^m \boldsymbol{x}_i}{\sum_{k=1}^{n} (\boldsymbol{u}_{ik})^m} \tag{2.8}$$

Con las ecuaciones (2.7) y (2.8) se van actualizando la matriz u y el vector $\boldsymbol{v}$ durante las iteraciones del algoritmo hasta que éste converge. El algoritmo 2 muestra los pasos del CMD; éste construye una *matriz de pertenencia* $U_{c\times n}$ donde cada una de sus columnas representa el grado de pertenencia de entre los puntos y los aglomerados.

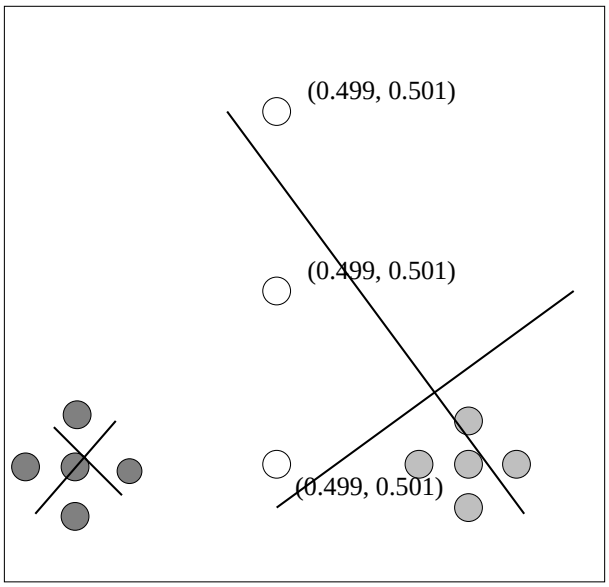


Figura 2.1: Se muestra que los tres valores atípicos se alojan en alguno de los aglomerados

Ahora bien, la restricción impuesta por la ecuación (2.3) implica que incluso los valores atípicos que logren ser identificados se asignarán a alguna de los aglomerados. Así que aún en *particiones difusas*, los valores atípicos pertenecen a todos los aglomerados.

Obsérvese la figura 2.1. Normalmente la clasificación difusa tiene ventajas en situaciones como ésta en donde un punto en el medio tiene un valor igual de pertenencia a todos los aglomerados. Debido a que algunos puntos están exactamente a la mitad de los aglomerados, el grado de pertenencia será el mismo sin ninguna discriminación aún cuando los puntos estén a diferentes distancias. Los valores en la imagen fueron obtenidos con el CMD.

Debido a la restricción de la ecuación (2.3), aún los tres puntos centrales pertenecen a los aglomerados *buenos*. Como se dijo anteriormente, se espera tener un método que consiga identificar y descartar los tres puntos centrales de la imagen.

Entrada: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, k, ε, *band*=true
Salida : Centroides $\boldsymbol{v}_i$, con i=1, ..., c

1 Inicializar U tal que satisfaga la restricción (2.3);
2 **while** band **do**
3 **for** $i = 1$ **to** c **do**
4 $$\boldsymbol{v}_i^{act} = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \quad \forall\ j=1, \ldots, n$$
5 **end**
6 **for** $i = 1$ **to** c **do**
7 $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| \boldsymbol{v}_i^{ant} - \boldsymbol{x}_j \right\|}{\left\| \boldsymbol{v}_k^{ant} - \boldsymbol{x}_j \right\|} \right)^{\frac{1}{m-1}}} \quad \forall\ j=1, \ldots, n$$
8 **end**
9 **if** $\frac{1}{c} \sum_{i=1}^{c} \left\| \boldsymbol{v}_i^{ant} - \boldsymbol{v}_i^{act} \right\| \leq \varepsilon$ **then**
10 *band*=false;
11 **else**
12 $\boldsymbol{v}_i^{ant} = \boldsymbol{v}_i^{act}\ \forall\ i$
13 **end**
14 **end**
15 **return** $\boldsymbol{v}^{act}$

Algorithm 2: Algoritmo de c-medias difuso

2.3 Algoritmos de aglomeración de ruido

En las siguientes secciones se describen algunos trabajos que atacan el problema de los valores atípicos. También se muestra uno de los trabajos más recientes en donde se aplica el algoritmo CMD.

2.3.1 El método de Dave [1]

A partir de la aparición del CMD en 1973, se desarrollaron diversas aplicaciones basadas en este algoritmo. Sin embargo, éstas continuaron siendo afectadas por la presencia de los valores atípicos. No fue sino hasta 1991 que Rajesh N. Dave desarrolló un método basado en el CMD que lograba reducir el efecto de los valores atípicos.

Este método pretende *relajar* la restricción impuesta por la ecuación (2.3), que obliga a que todos los puntos sean asignados a los aglomerados, sean eśtos valores atípicos o no. Dave propone que exista un aglomerado especial llamado *aglomerado de ruido* (*noise cluster*) en donde se espera que todos los valores atípicos sean descargados en dicho aglomerado.

De acuerdo con Dave, el aglomerado de ruido es una entidad universal, sin dimensiones ni forma definida y que se encuentra siempre a la misma distancia de cada elemento del conjunto de datos; a esa distancia se le conoce como *distancia de ruido* (*noise distance*). Básicamente, todo objeto que no pueda ser asignado a ninguno de los aglomerados *buenos* tendrá que caer irremediablemente en el aglomerado de ruido.

Sea $\boldsymbol{v}_c$ el aglomerado de ruido, $\boldsymbol{x}_k$ un punto del espacio de características y $\boldsymbol{v}_c, \boldsymbol{x}_k \in \mathbb{R}^p$; la distancia de ruido d_{ck} del punto $\boldsymbol{x}_k$ a $\boldsymbol{v}_c$ es

$$d_{ck} = \delta, \quad \forall\, k \tag{2.9}$$

La distancia δ hacia $\boldsymbol{v}_c$ es la misma para todos los puntos; lo que significa físicamente que todos tienen la misma probabilidad de pertenecer al aglomerado de ruido. Esto al menos como condición inicial, pues se espera que a medida que progrese el algoritmo, los puntos *buenos* sean clasificados en aglomerados *buenos*. Entonces hay $c-1$ aglomerados *buenos* y el c-ésimo aglomerado es el aglomerado de ruido. La función objetivo J_N incluyendo el aglomerado de ruido es la misma que la ecuación (2.2)

$$\boldsymbol{J}(U, v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m (d_{ik})^2$$

se toma el mismo criterio para medir la similitud entre los puntos y además

$$(d_{ik})^2 = \delta^2, \text{ cuando } k = c \tag{2.10}$$

Si para un punto $\boldsymbol{x}_i$ se tiene un valor de δ pequeño, significa físicamente que éste se encuentran muy cerca del aglomerado de ruido, por lo que este punto es un potencial valor atípico; si por el contrario, δ es grande, significa que se encuentra muy lejos, por lo que debe de estar muy cerca de alguno de los aglomerados *buenos*.

Entrada: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, k, ε, *band*=true
Salida : Centroides $\boldsymbol{v}_i$, con i=1, $\ldots$, c

Seleccione aleatoriamente $\boldsymbol{v}_1^{ant}, \ldots, \boldsymbol{v}_c^{ant}$ elementos de X;
Calcular δ con Ec. (2.11);
while band **do**
  for $i = 1$ **to** c **do**
    $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| \boldsymbol{v}_i^{ant} - \boldsymbol{x}_j \right\|}{\left\| \boldsymbol{v}_k^{ant} - \boldsymbol{x}_j \right\|} \right)^{\frac{1}{m-1}}} \quad \forall\ j=1, \ldots, n$$
  end
  for $i = 1$ **to** c **do**
    $$\boldsymbol{v}_i^{act} = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \quad \forall\ j=1, \ldots, n$$
  end
  if $\frac{1}{c} \sum_{i=1}^{c} \left\| \boldsymbol{v}_i^{ant} - \boldsymbol{v}_i^{act} \right\| \leq \varepsilon$ **then**
    band=false;
  else
    $\boldsymbol{v}_i^{ant} = \boldsymbol{v}_i^{act}\ \forall\ i$;
    Calcular δ con Ec. (2.11)
  end
end
return $\boldsymbol{v}^{act}$

Algorithm 3: Método de Dave de aglomeración de ruido

El algoritmo requiere la especificación del valor de δ ya que éste es diferente para cada problema. El autor propone calcularlo basándose en el promedio de las distancias de los puntos a los aglomerados *buenos*, tomando el hecho de que esto refleja una relación de la estructura del espacio de puntos. Bajo este argumento, δ se calcula con

$$\delta^2 = \lambda \left[\frac{\sum_{i=1}^{c-1} \sum_{k=1}^{n} (d_{ik})^2}{n(c-1)} \right] \tag{2.11}$$

donde λ es una constante definida por el usuario. De acuerdo a este artículo, el valor que maneja el autor es de $\lambda = 0.1$; sin embargo ese valor no funciona para todos los problemas y debe de calibrarse continuamente.

El algoritmo 3 presenta el *algoritmo de aglomeración de ruido*. En el paso 4 se genera una nueva partición con los prototipos $\boldsymbol{v}_j$. Para cada u_{ij} se calcula la distancia de cada punto $\boldsymbol{x}_j$ hacia los c aglomerados desde k=1, . . . , c; cuando toca el turno del aglomerado de ruido, i.e., cuando k=c, ya no se calcula la distancia sino que se asigna directamente el valor de δ^2 ya que ésta es la misma para todos los puntos. Así que este cálculo se realiza sólo en los siguientes casos:

$$d_{ij} = \begin{cases} \delta^2 & \text{si } k{=}c \\ \left\| \boldsymbol{v}_i^{ant} - \boldsymbol{x}_j \right\| & \text{de otro modo} \end{cases} \tag{2.12}$$

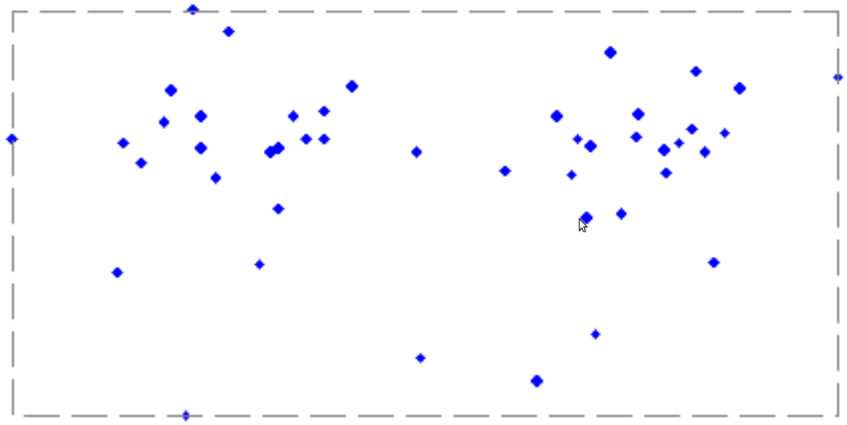
Observe que en cada iteración se realiza el cálculo de δ (paso 14 del algoritmo) utilizando la ecuación (2.11). Este paso se realiza porque los prototipos van cambiando durante todo el algoritmo y por consecuencia, también cambia el valor de δ.

2.3.2 Aglomeración basada en *hiper* volúmenes

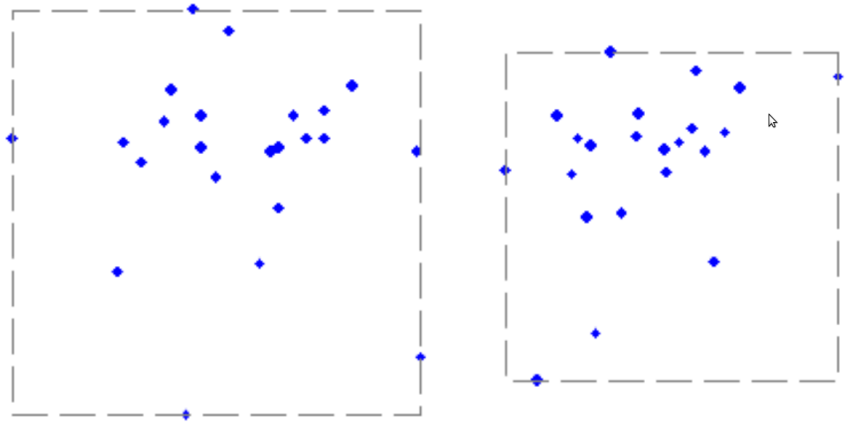
El método de Rehm, explicado en esta sección se basa en el cálculo del *hiper* volumen [10]. Este método está basado en el de Dave; define un *prototipo* y una distancia de ruido. Los autores presentan un método para especificar esta distancia basándose en la preservación del *hiper volumen* del espacio de características. Aquellos puntos cuya distancia de ruido a todos los aglomerados exceda un cierto *umbral*, serán considerados como valores atípicos.

La distancia de ruido depende principalmente del número c de prototipos usados para la aglomeración y también de la expansión del espacio. La idea general es dividir el espacio global en $c-1$ *hiper volúmenes* del mismo tamaño aproximadamente y especificar la distancia δ con un valor que corresponda al *hiper radio* de estos hiper volúmenes.

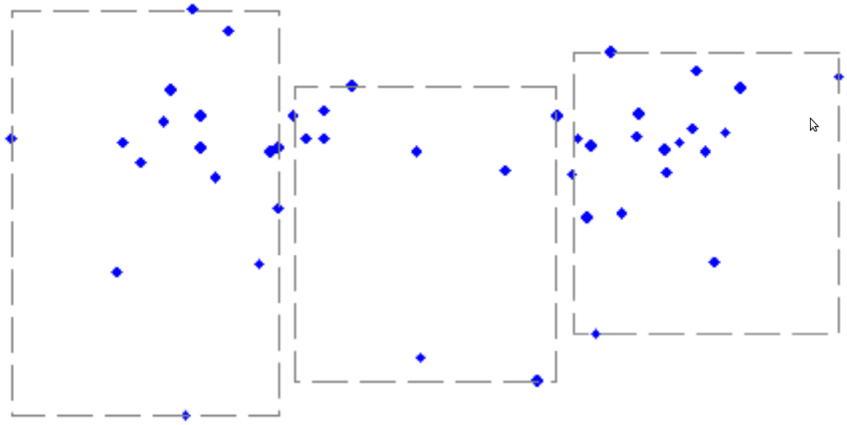
Para obtener el *hipervolumen* considérese el espacio de la imagen (a) de la figura 2.2. El hipervolumen de este espacio se obtiene dividiéndolo en espacios rectangulares más pequeños (estos subespacios deben de tener aproximadamente el mismo tamaño.), ya sea en dos (imagen (b)) o en tres (imagen (c)); esto es dependiendo del número de $c-1$ aglomerados *buenos* (el c-ésimo aglomerado es el de ruido).



(a)



(b)



(c)

Figura 2.2: (a) Conjunto de datos. (b) y (c) Obteniendo dos y tres subespacios respectivamente. Éstos deben de tener aproximadamente el mismo tamaño.

Suponiendo que los subespacios o aglomerados tienen el mismo tamaño, el radio y la distancia δ son aproximadamente el radio r de la hiperesfera con un hipervolumen de $V/(c-1)$, siendo V el volumen global del hiperespacio. Entonces el radio r se calcula con

$$r = \sqrt[n]{\frac{V \cdot \Gamma(\frac{n}{2}+1)}{\pi^{\frac{n}{2}}}} \tag{2.13}$$

Así, la distancia δ, que puede ser calibrada con un parámetro α, se obtiene con

$$\delta = \alpha r \tag{2.14}$$

de acuerdo a los autores, el mejor valor para α es 1.5. Con valores más pequeños de δ, se producen aglomerados más compactos y un número mayor de valores atípicos. Con $\alpha \to \infty$, este método se comporta como el CMD.

Una vez que se ha definido el valor de δ y del radio r, se necesita especificar un método para definir el mínimo grado de pertenencia para declarar a un punto como valores atípicos. Se obtiene el promedio μ (*mean*) de los grados de pertenencia al aglomerado de ruido y su desviación estandar σ; evaluando cada punto con la función $is\_outlier(\boldsymbol{x}_j)$, se considera como un valor atípico si

$$is\_outlier(\boldsymbol{x}_j) = \begin{cases} 1 & \text{if } u_{cj} - \beta\sigma > \mu \\ 0 & \text{de otro modo} \end{cases} \tag{2.15}$$

el valor de μ se calcula con

$$\mu = \frac{1}{n}\sum_{j=1}^{n} u_{cj} \tag{2.16}$$

y la desviación estándar

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(u_{cj} - \mu)^2} \tag{2.17}$$

ajustando el valor de β puede influenciar en la detección de valores atípicos. El mejor valor para β que consideraron los autores es de 1.4.

2.3.3 Aplicación del CMD sobre análisis de imágenes

En esta sección se describe el trabajo de Z. Hou [11], en el que se presenta un método basado en el algoritmo CMD para analizar *imágenes de resonancia magnética* (RM) de tejidos cerebrales. De manera general, el autor propone un parámetro *difuminador* con el que se espera poder reducir el efecto de los valores atípicos en el análisis de las imágenes.

Dentro del *diagnóstico clínico* y en la planeación de *cirugías cerebrales*, contar con un sistema automatizado de procesamiento de imágenes de RM es de gran ayuda. El propósito de la *segmentación de imágenes* es dividir una imagen en sus *partes constituyentes* (*regiones*).

Un algoritmo eficiente debe considerar no sólo la intensidad (el valor) de los datos, sino también el contexto de la imagen. Algoritmos como el CMD consideran los datos como puntos separados sin ninguna conexión entre ellos, y debido a esto, no es posible formar un *mapa de segmentación* adecuado[1].

Además, el CMD es vulnerable ante la presencia de valores atípicos. Para reducir este efecto, algunos métodos han introducido la *teoría de regularización*, tales como la *Regularización por máxima entropía*, la *Regularización por término cuadrático* y el *Método de Pahm* (RCMD).

Esta regularización básicamente introduce un *término de castigo* (penalty term) de tal modo que el grado de pertenencia de un *píxel/voxel*[2] para una clase en particular, esté inversamente relacionado al grado de pertenencia de su espacio vecinal con las demás clases (regiones).

De acuerdo al autor, el problema de estos métodos radica en que siguen basándose sólo en la intensidad de la información, por lo que siguen siendo afectados por los valores atípicos. Para diseñar un método más robusto, se debe incorporar el *efecto espacial* (el contexto) de la imagen al momento de realizar el análisis de las mismas.

En su trabajo, el autor introduce un *filtro* que actúa como un difuminador. Con este valor, el grado de pertenencia de cada punto puede ser determinado por su vecindario y por su intensidad, con lo que se logra un mejor desempeño en imágenes que contengan valores atípicos.

El método desarrollado por los autores fue llamado *regCMD*. Éstos realizaron varias pruebas en imágenes de segmentaciones de tejidos cerebrales en dos (2-D) y tres dimensiones (3-D). Las pruebas fueron hechas aplicando el algoritmo normal CMD, el RCMD y el regCMD.

[1] Los *mapas de segmentación* se usan para definir las descripciones (*labels*) de las regiones de una imagen

[2] Un *voxel* es la unidad básica de una *reconstrucción por tomografía computarizada*.

Las imágenes en 2-D fueron editadas para remover el cráneo y aumentar el contraste. Las imágenes en 3-D eran representaciones tridimensionales de materia gris. En éstas se logró reducir el efecto de los valores atípicos, logrando obtener imágenes más *suaves* y con delineaciones más finas.

Las pruebas que realizaron los autores mostraron que el regCMD es eficiente en el análisis de imágenes de RM. Al comparar su desempeño con los otros métodos, observaron que realiza una segmentación más precisa y con un 30% menos de costo computacional.

2.4 Selección *rígida* sobre una partición difusa

Los algoritmos de aglomeración de ruido están basados en el CMD. En el siguiente capítulo se definen unas métricas con las que se pretende evaluar su desempeño. En la definición de estas métricas debe considerarse que aún cuando el CMD utiliza una técnica difusa, la manera de evaluar las métricas es a través de una *selección rígida* de los grados de pertenencia de los puntos.

Una selección rígida es como la realizada por el $k-$medias. En este algoritmo un dato sólo puede pertenecer a un solo aglomerado. A diferencia de éste, el CMD produce una *matriz de pertenencia difusa*, en la cual se muestra el grado de pertenencia de cada punto a todos los aglomerados. En esta matriz, cada columna corresponde a uno de los puntos del conjunto, mientras que las filas representan a los aglomerados.

En la definición de las métricas, se dice que se utiliza una selección rígida debido a que se toma el valor máximo de cada una de las columnas de dicha matriz; ya que se considera que este valor indica a qué aglomerado pertenece cada punto. Con esta selección se deja a un lado el hecho de tener un grado de pertenencia difuso para cada punto.

Una manera de explicar ésto es la siguiente. Dado un $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, el CMD produce una matriz de pertenencia U, en la que se encuentran definidos los grados de pertenencia de cada $\boldsymbol{x} \in \mathrm{X}$; esta matriz cumple la restricción estipulada por la ecuación (2.3), es decir

$$\sum_{i=1}^{c} u_{ij} = 1, \;\; j = 1, 2, \ldots, n$$

donde c es el número de aglomerados y n el número de datos. A partir de esta matriz U, se puede *separar* al conjunto X en c subconjuntos $\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_c$, de tal manera que cada Y_i contiene puntos en el aglomerado i, es decir, cada punto i tal que $u_{ik} = 1$ para toda k.

Aglomerados	x_1	x_2	x_3	x_4	x_5	x_6	x_7
c_1	0.192698	0.253690	0.421085	0.244917	0.430521	0.551148	...
c_2	0.353232	0.259987	0.118666	0.220843	0.303055	0.150919	...
c_3	0.265020	0.197388	0.060404	0.224773	0.211859	0.155334	...
c_4	0.189049	0.288934	0.399846	0.309467	0.054564	0.142599	...

Tabla 2.1: Tabla de pertenencia U en donde se ven los valores máximos de cada columna.

En la tabla 2.1 se muestra una partición difusa de un conjunto de puntos hacia seis aglomerados. Las celdas sombreadas indican el valor máximo de cada columna. Así que por ejemplo, el máximo de la columna 1 es el índice 2 (c_2), que indica que el punto $\boldsymbol{x}_1$ pertenece al segundo aglomerado, el punto $\boldsymbol{x}_2$ pertenece al cuarto y así para todos los puntos.

De esta manera se realiza una selección rígida de los datos, i.e., no se toman en cuenta los valores difusos de pertenencia de cada uno. Definir ésto es útil para entender cómo se definieron las métricas explicadas en el siguiente capítulo.

2.5 Resumen

En este capítulo se presentó el marco teórico de los algoritmos existentes de *análisis de aglomeraciones*. Se mostraron los algoritmos básicos: $k-$medias y c-medias difuso y se explicó por qué fallan ante la presencia de valores atípicos. Debido al efecto que producen este tipo de valores, surge la necesidad de un método que logre identificarlos de una manera eficiente.

También se explicaron formalmente los *algoritmos de aglomeraciones de ruido* que se han desarrollado hasta el momento. Estos son el método de Dave [1] y el método de Rehm [10]. El primero de éstos introduce el concepto de un aglomerado especial llamado *aglomerado de ruido* en el cual se espera que sean alojados los valores atípicos. El segundo a su vez, está basado en el trabajo de Dave y propone un método de aglomeración de ruido basado en el cálculo del hipervolumen del espacio de puntos.

Además se presentó brevemente el artículo de Z. Hou [11], uno de los trabajos más recientes en los que se aplica el algoritmo CMD. En este artículo se desarrolló el *regFCM*, una modificación al CMD para reducir el efecto de los valores atípicos dentro del análisis de imágenes de resonancia magnética (RM).

Finalmente se describe la manera en la que se hace una selección *rígida* a partir de la partición difusa generada por el CMD. Con esta selección se obtiene a cual aglomerado pertenece cada punto. Con esto se define la manera con la que trabajan las métricas que se definen en el siguiente capítulo.

Capítulo 3

Descripción de las métricas de desempeño

> Todos creen que tener talento es cuestión de suerte, pero nadie piensa que la suerte puede ser cuestión de talento.
>
> Jacinto Benavente

En este capítulo se describen algunas métricas que pueden ser usadas para evaluar la *calidad* de los aglomerados obtenidos por un algoritmo de aglomeraciones de ruido. La definición de valor atípico puede variar de acuerdo a la aplicación, de tal manera que estas métricas no pueden verse como de propósito general.

Las métricas que se han definido cubren de manera intuitiva los criterios necesarios para decidir si un aglomerado es bueno o malo. Éstas fueron llamadas *promedio de distancia de ruido*, *micro precisión* y *diferencia de prototipos*.

3.1 Conjuntos de datos

Para poner a prueba las métricas, fueron utilizados diversos conjuntos de datos. Algunos fueron tomados del *UCI Machine Repository* [12] o UCI[1]. Para una mayor referencia de estos conjuntos, véase el apéndice B.2 Conjuntos de datos en la página 73.

El problema de los conjuntos del UCI es que no es posible conocer su estructura, es decir, la forma o el tamaño que tienen. Al ser datos tomados de muestras reales, es difícil encontrar una forma que pueda ser identificada por algún algoritmo. Debido a

[1]El repositorio UCI es una colección de bases de datos, teórias de dominio y generadores de datos usados por los estudiosos del *Aprendizaje de Máquina* para el *análisis empírico* de sus algoritmos

esto se crearon algunos conjuntos que facilitaran la apreciación del desempeño de los aglomerados de ruido. Estos conjuntos son:

- *normal*. Los puntos forman dos aglomerados bien definidos generados con una distribución normal. Vea la figura 3.1a
- *normal-ruido*. Al conjunto anterior le fue agregado algunos atípicos. Corresponde a la figura 3.1b
- *básico*. Es el ejemplo básico de los algoritmos de análisis de aglomeraciones. Figura 3.1c
- *srand*. Conjuntos con densidades diferentes. Un aglomerado tiene una densidad de puntos mayor que la del otro. Corresponde a la figura 3.1d

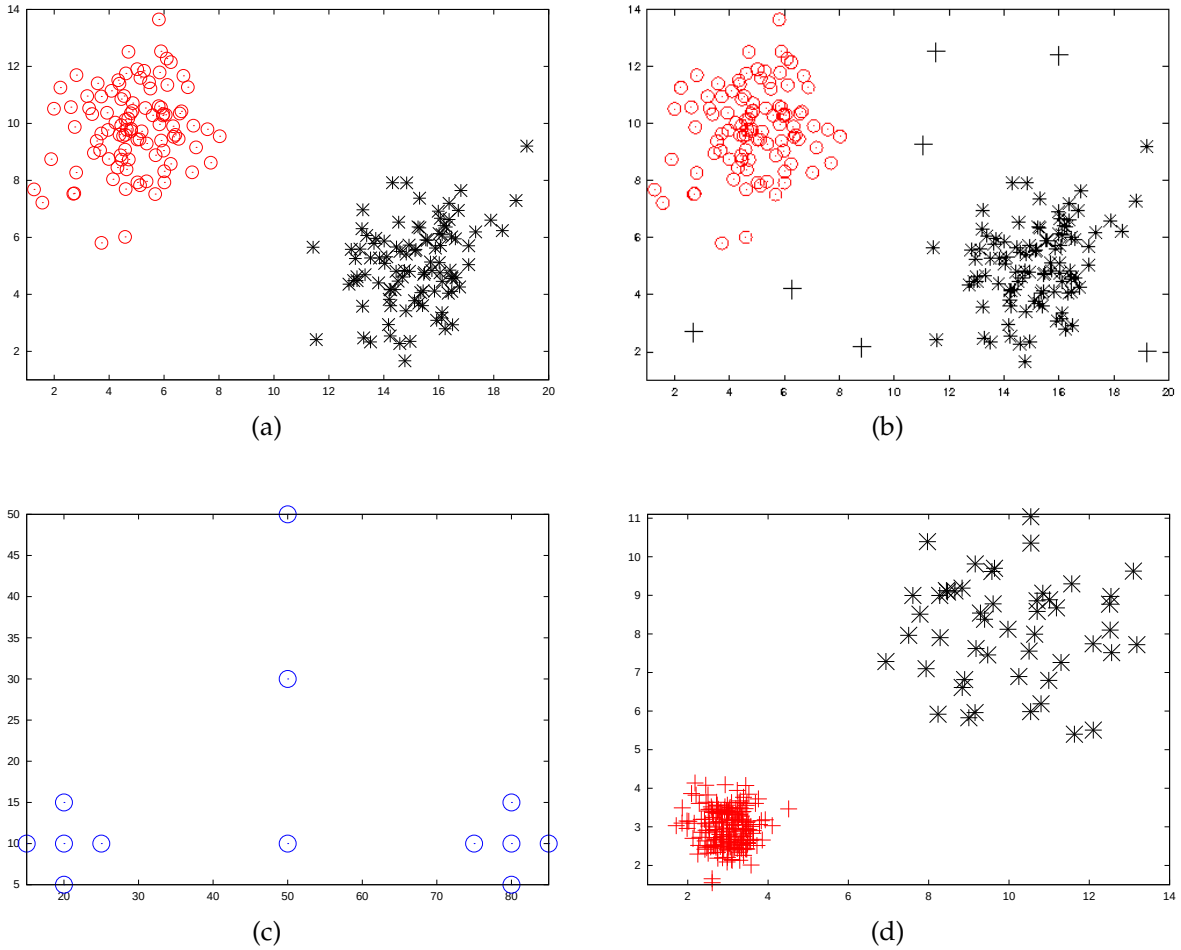


Figura 3.1: Muestran gráficamente los diferentes conjuntos que fueron creados para medir el desempeño del algoritmo

En la figura 3.1, sólo dos de éstas tienen valores atípicos. La figura 3.1c presenta tres valores atípicos y como se ha mencionado, es el ejemplo básico para explicar los algoritmos de aglomeración de ruido. Los aglomerados de la figura 3.1b se generaron con una distribución normal y se le agregaron algunos valores atípicos, aproximadamente el 15% del número de datos.

Todos los conjuntos de datos que se utilizaron para realizar las pruebas están *etiquetados*, i.e., se conoce la clase correspondiente de cada elemento. Las métricas que se definen a continuación están relacionadas con la estructura de clases especificada por el conjunto de datos.

3.2 Métricas de desempeño

En las siguientes secciones se definen formalmente las métricas que miden el desempeño de un algoritmo de aglomeración de ruido. Cada una de éstas fue diseñada para evaluar la calidad de los centroides obtenidos.

3.2.1 Promedio de distancias de ruido

Ya se ha mencionado que las características de los valores atípicos es que son muestras que parecen no pertenecer al resto de los datos del conjunto. Por lo que se considera que éstos deben encontrarse *alejados* de los centroides de los aglomerados *buenos*, i.e., la distancia de los valores atípicos hacia los prototipos es mucho mayor que la distancia de los puntos *buenos*.

Basada en esta suposición, esta métrica A llamada *el promedio de distancias de ruido*, calcula el promedio de distancia de los valores atípicos hacia los prototipos y se define de la siguiente manera.

Dado un conjunto $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, un algoritmo de aglomeración de ruido detectará algunos puntos en X como puntos *buenos* y otros como puntos *malos* o valores atípicos. Sea $N \subseteq X$ el conjunto de los estos últimos y N estrictamente no vacío.

Supóngase que un algoritmo de aglomeración de ruido produce c aglomerados buenos en X con prototipos $\mu_1, \mu_2, \ldots, \mu_c$, y además obtiene el conjunto no vacío $N = \{\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_p\}$, que es el conjunto que contiene a los valores atípicos y $p = |N|$. Entonces el promedio de distancias A, de cada elemento de N hacia los aglomerados buenos se define como sigue:

$$A \leftarrow \frac{1}{|N|} \sum_{\bar{\boldsymbol{x}} \in N} \min_{j=1,\ldots,c} \left(\left\| \bar{\boldsymbol{x}} - \mu_j \right\| \right) \tag{3.1}$$

donde N es el conjunto de valores atípicos encontrados. $\min_{j=1,\ldots,c} \left(\left\|\bar{\boldsymbol{x}} - \mu_j\right\|\right)$ es la distancia más corta de cada $\bar{\boldsymbol{x}} \in N$ hacia los c aglomerados. La ecuación (3.1) sugiere que si todos los elementos de N se encuentran muy alejados de los aglomerados buenos, entonces la distancia de cada uno es grande por lo que el valor de A también lo será; lo que se traduce en una *buena* aglomeración por parte del algoritmo.

Sin embargo se debe hacer la siguiente aclaración. Podría ocurrir que los datos no presenten valores atípicos, como lo sería el conjunto de la figura 3.1a. En este caso el valor de A es indefinido ya que $|N| = 0$. Es por eso que esta métrica sólo tiene sentido cuando $N \neq \emptyset$, de otro modo no se utiliza. Ésto se debe a que dado un valor bajo de A, no significa necesariamente que el algoritmo haya hecho una buena aglomeración.

3.2.2 Micro precisión

En esta sección se define la *micro precisión* Mp, que es una métrica con la que se pretende medir la *calidad* de la aglomeración. De manera general, dada la salida de un algoritmo de ruido, los prototipos $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_c$, se trata de calcular el número de puntos del aglomerado $\boldsymbol{v}_i$ que hayan sido asignados correctamente con su clase *real*.

Se había mencionado que los datos usados en este trabajo están etiquetados, i.e., que se conoce la etiqueta real $l(\boldsymbol{x})$ correspondiente a cada dato $\boldsymbol{x}$. Por lo que la estructura de clases del conjunto de datos debe coincidir con las etiquetas que le asigna a cada dato el algoritmo de aglomeración de ruido.

Debido a que se manejan conjuntos de datos etiquetados, normalmente el número de clases del conjunto es el mismo que el número de aglomerados. La Mp se basa en el *número de errores de clasificación*. Se considera un error de clasificación cuando la etiqueta que le asigna el algoritmo al punto $\boldsymbol{x}_i$ con i=$1, \ldots, n$, no coincide con su respectiva clase $l(\boldsymbol{x}_i)$.

Entonces sea $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ el conjunto de datos y sea $l(\boldsymbol{x}_i)$ la etiqueta *real* de clase de $\boldsymbol{x}_i$y c el número de clases presentes en X. Se presupone que un algoritmo de aglomeración de ruido detecta c aglomerados *buenos* en X.

Lo que se quiere mostrar es la forma en la que coinciden los puntos etiquetados por el algoritmo con sus etiquetas reales. De acuerdo con los resultados de la aglomeración, la *micro precisión* Mp se calcula con

$$Mp = \frac{1}{n} \sum_{h=1}^{c} a_h \tag{3.2}$$

donde c es el número de aglomerados y n el número de datos. a_h indica el número de elementos en el aglomerado h que han sido asignados adecuadamente a su clase

correspondiente. La clase a la que pertenece cada punto se obtiene a partir de la matriz de pertenencia U, tomando el valor máximo de cada columna, tal y como se explicó en la sección 2.4 (página 2.4).

Cada a_h contiene a los datos cuya etiqueta asignada por el algoritmo coincide con la clase real de cada uno; así que de acuerdo a la ecuación (2.4), la suma de todos los puntos en cada a_h es n; por lo que la Mp es un valor dentro del $0 \leq Mp \leq 1$.

Aunque para un algoritmo de aglomeración de ruido, normalmente Mp es siempre menor a 1 pues no se consideran los valores atípicos, i.e., que estos últimos no forman parte de ninguna de las a_h particiones formadas por puntos buenos. Por lo que la unión de todas las a_h no siempre es igual a X.

Entonces la Mp considera que, con valores cercanos a 1, el algoritmo ha realizado una buena aglomeración; en caso contrario, un valor cercano a cero significa que hay un gran número de errores de clasificación en la aglomeración, y por lo tanto, el valor de Mp disminuye, lo que se considera un mal desempeño del algoritmo.

3.2.3 Diferencia de prototipos

Conocer la distribución con la que han sido generados los datos de un conjunto es una tarea muy difícil o incluso imposible. Sin embargo es posible realizar un cálculo que permita obtener el *centroide real* de cada aglomerado. En este caso se trabaja con conjuntos de datos etiquetados, de modo que conoce la clase a la que pertenece cada elemento del conjunto.

Se tiene un conjunto de datos $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, y un algoritmo de aglomeración de ruido que obtiene $c-1$ aglomerados buenos. Los puntos en X se separan de acuerdo a sus clases, i.e,

$$\mathrm{C}_i = \{\boldsymbol{x} \in X : l(\boldsymbol{x}) = i\}$$

donde Y_i contiene los puntos correspondientes a la i-ésima clase. Se define entonces μ_i como el promedio de cada una de las clases, i.e.,

$$\mu_i = \frac{1}{|\mathrm{C}_i|} \sum_{\boldsymbol{x} \in Y_i} \boldsymbol{x}, \qquad \text{para } i{=}1 \text{ a } c{-}1$$

Ahora el cálculo de la diferencia de prototipos consiste en saber qué tan cerca se encuentran los aglomerados buenos (los obtenidos por el algoritmo) de los centroides reales, i.e., los μ_i. Sea, $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ los prototipos de los aglomerados buenos dados por el algoritmo de aglomeración de ruido. Entonces se define la diferencia de prototipos DP como

$$DP = \frac{1}{c-1} \sum_{i} \min_{j} \left\| \boldsymbol{v}_i - \mu_j \right\| \tag{3.3}$$

Esta métrica supone que si el valor de DP es grande, significa entonces que los prototipos de los aglomerados se encuentran alejados de los promedios μ_i reales, lo que sugiere una mala aglomeración. Así que si el algoritmo realiza una buena aglomeración, se espera que el valor de DP sea pequeño.

3.3 Resumen

En este capítulo se describieron las métricas que se diseñaron para probar el desempeño de un algoritmo de aglomeración de ruido. Todas estas métricas fueron definidas para tomar las salidas del algoritmo y evaluar diferentes parámetros. Además presuponen que se está trabajando con conjuntos de datos etiquetados, es decir, que se conoce la clase a la que pertenece cada uno de sus elementos.

La primera de éstas es la del *promedio de distancias de ruido* A; consiste en tomar el conjunto de los valores atípicos identificados por el algoritmo y calcular el promedio de sus distancias hacia los prototipos de los aglomerados buenos. Se espera que este valor sea grande si se ha desarrollado una buena aglomeración.

La segunda métrica es la del calcúlo de la *micro precisión* Mp. En ésta se comparan las etiquetas de clase reales de los datos, contra las que el algoritmo asigna a cada uno. Si se realiza una buena aglomeración, el valor de Mp debe ser cercano a 1.

La tercera fue definida como *diferencia de prototipos* DP. En ésta se forman aglomerados con los elementos de la misma clase y se calcula el centroide real de cada una. Estos prototipos son comparados con los que obtiene el algoritmo. Se espera que esta diferencia sea pequeña si el desempeño del algoritmo ha sido bueno.

Capítulo 4

Un nuevo método de aglomeración para detectar valores atípicos

> Si estás sentado junto a una linda chica, una hora te parecerá un minuto. Si te sientas en una estufa caliente, un minuto te parecerá una hora. Eso es la relatividad.
>
> Albert Einstein

En este capítulo se aborda de una forma detallada el método de aglomeración desarrollado en este trabajo de investigación.

En los capítulos anteriores se explicó en qué consiste el problema de los valores atípicos dentro del análisis de aglomeraciones y los trabajos que se han publicado para resolverlos. Principalmente este nuevo método se basa en el trabajo desarrollado por Dave [1], el cual como se mencionó anteriormente, introduce el concepto *aglomerado* y *distancia de ruido*.

Estos conceptos ayudaron a mejorar el desempeño de los algoritmos ante la presencia de valores atípicos. Aunque el hecho de considerar que la distancia de ruido es la misma para todos los puntos no es una buena apreciación.

El método presentado en este trabajo propone una suposición más realista (físicamente) del problema y es la aportación más importante de la tesis. Básicamente se trata, a diferencia de los métodos mencionados, de que la *distancia de ruido* no sea la misma para cada dato, sino de que tenga un valor particular para cada uno de acuerdo a su posición en el espacio.

4.1 Descripción del nuevo algoritmo de aglomeración

En esta sección se describe con detalle la metodología utilizada para el desarrollo de este algoritmo de aglomeración de ruido. Se describe la metodología utilizada, la nueva propuesta de la distancia de ruido en base a una función principal y el algoritmo propuesto.

4.1.1 Metodología

En el método propuesto en este trabajo se considera también la existencia de un *aglomerado* y una *distancia de ruido* . La aportación más importante es que se ha definido una forma de *medir* la distancia hacia el aglomerado de ruido.

El aglomerado de ruido es una entidad que no existe físicamente, i.e., no hay parámetros que definan su posición o su tamaño; así que es difícil definir una manera general con la que medir la distancia de los puntos al prototipo de ruido.

En los trabajos de Dave [1] y Rehm [10], los autores proponen el hecho de que todos los puntos son equidistantes hacia el prototipo de ruido y esta suposición no es realista físicamente.

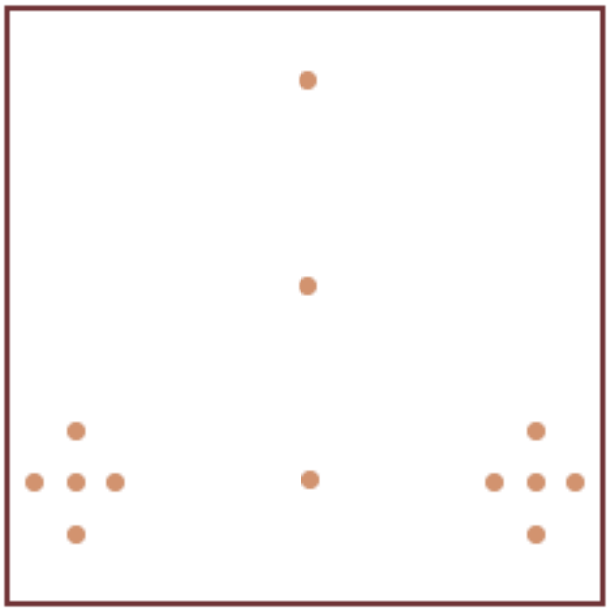


Figura 4.1: Forma básica de un problema de aglomeraciones.

En la figura 4.1 se aprecian dos aglomerados bien definidos y algunos puntos exactamente a la mitad entre éstos. Es evidente que los tres puntos centrales son *más* valores

atípicos que los otros. Si se considera que existe una distancia de ruido de todos los puntos hacia una entidad sin forma ni dimensión, ésta tiene que ser distinta para todos, i.e., distinta para los puntos centrales, como para los que forman los dos aglomerados.

En la propuesta de este trabajo, se considera una manera nueva para identificar y caracterizar los valores atípicos. La idea básica detrás de esta propuesta es considerar que un valor atípico se encuentra en áreas de poca densidad de puntos, i.e., un valor atípico debe estar rodeado de pocos, o más bien, de ningún vecino.

Entonces puede decirse que el *estado* de un valor atípico, es siempre el de estar *aislado*; ésto tiene sentido pues por definición, los valores atípicos son muestras que muy distantes (numéricamente) al resto de los datos. Por otro lado, se considera como puntos buenos a aquellos que se encuentren rodeados de un gran número de puntos, i.e., su estado es *no aislado*.

Definición de la función principal

De acuerdo a lo anterior, se trata de conocer el *estado* de cada punto. Entonces se necesita una manera de *penalizar* drásticamente a cada punto de acuerdo a su posición en el espacio, es decir, obtener su *densidad local*. Por lo que, conociendo este valor de densidad es posible determinar su posición respecto al *aglomerado de ruido*.

Una manera de obtener la densidad local es utilizando la *función de montaña*. Esta función de densidad fue aplicada por primera vez en los problemas de *análisis de aglomeraciones* por Yager y Filev [13] y sirve para principalmente para estimar la aproximación de los centroides basado en el concepto de la densidad de los puntos.

Dado un conjunto $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, se define una función $M(\boldsymbol{x})$ para todo $\boldsymbol{x} \in X$ como

$$M(\boldsymbol{x}_i) = \sum_{j=1}^{n} e^{-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|} \quad \forall\ \boldsymbol{x}_i \in X \tag{4.1}$$

donde $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|$ es la distancia euclidiana. La definición de esta función muestra que si un punto $\boldsymbol{x}_i$ tiene muchos puntos vecinos, entonces $M(\boldsymbol{x}_i)$ será un valor alto, mientras que si el punto tiene pocos o ningún vecino, entonces el valor de la función será bajo. De esta manera, la función M ayuda a caracterizar la distancia de ruido.

Aquí se utiliza una variante de M para definir la distancia de $\boldsymbol{x}$ hacia el aglomerado de ruido. Es mejor normalizar este factor, de otra manera cuando el número de puntos es grande, el valor de la función M crece de manera considerable. Por lo que se propone la siguiente manera para normalizar M.

Sea X el conjunto de datos, se calcula

$$\begin{aligned} Z &= \sum_{i=1}^{n} \mathrm{M}(\boldsymbol{x}_i), \text{ y se define} \\ \tilde{M}(\boldsymbol{x}_i) &= \frac{1}{Z}\mathrm{M}(\boldsymbol{x}_i) \quad \forall \ \boldsymbol{x}_i \end{aligned} \tag{4.2}$$

donde se tiene que $0 \leq \tilde{M}(\boldsymbol{x}_i) \leq 1$ para todo $\boldsymbol{x} \in \mathrm{X}$ y también $\sum_{i=1}^{n} \tilde{M}(\boldsymbol{x}_i) = 1$.

Entonces $\mathrm{M}(\boldsymbol{x}_i)$ actúa como un *factor de densidad* para la distancia de ruido. Se propone que la distancia δ_i del punto $\boldsymbol{x}_i$ como

$$\delta_i = \tilde{M}(\boldsymbol{x}_i) \times \delta$$

donde

$$\delta = \lambda \left[\frac{\sum_{j=1}^{c-1} \sum_{k=1}^{n} d_{jk}}{n(c-1)} \right] \tag{4.3}$$

donde λ es una constante; $\frac{\sum_{j=1}^{c-1} \sum_{k=1}^{n} d_{jk}}{n(c-1)}$ es el promedio de la suma de las distancias de todos los puntos a los centroides. Utilizando esta distancia de ruido para cada punto, ahora se puede modificar el algoritmo CMD para la correcta identificación de cada punto.

En algoritmo 4 muestra los pasos del método desarrollado en este trabajo. La inicialización de la matriz U se realiza de forma aleatoria considerando la restricción (2.3). El valor de δ se calcula con la ecuación (4.3) y debe obtenerse de nuevo en cada iteración; esto es por que los prototipos se actualizan y cambian de posición, por lo tanto la distancia δ es diferente para cada uno en cada una de las iteraciones.

Cuando se genera la matriz U (paso 8), se calcula el grado de pertenencia de cada punto a los $c-1$ aglomerados buenos; al llegar el turno del aglomerado de ruido, ya no se realiza el cálculo de este valor, sino que debe tomarse en cuenta el *factor de densidad* calculado por la función M para definir el valor de u_{ij}.

La función M modifica el valor de δ cuando se le aplica el factor de densidad $\mathrm{M}(\boldsymbol{x}_i)$ del i-ésimo punto. Por lo que la distancia de cada punto $\boldsymbol{x}_i$ hacia el j-ésimo aglomerado se define de la siguiente manera

Entrada: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, k, ε, *band*=true
Salida : Centroides $\boldsymbol{v}_i$, con i=$1, \ldots, c$
Inicializar matriz U;
Calcular $\tilde{M}(\boldsymbol{x}_i)\ \forall\ \boldsymbol{x}_i$ con Ec. 4.1 y 4.2
Calcular δ con Ec. (4.3);
while band **do**
  for $i = 1$ **to** c **do**
    $$\boldsymbol{v}_i^{act} = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \quad \forall\ j=1, \ldots, n$$
  end
  for $i = 1$ **to** c **do**
    $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| \boldsymbol{v}_i^{ant} - \boldsymbol{x}_j \right\|}{\left\| \boldsymbol{v}_k^{ant} - \boldsymbol{x}_j \right\|} \right)^{\frac{1}{m-1}}} \quad \forall\ j=1, \ldots, n$$
  end
  if $\frac{1}{c} \sum_{i=1}^{c} \left\| \boldsymbol{v}_i^{ant} - \boldsymbol{v}_i^{act} \right\| \leq \varepsilon$ **then**
    band=false;
  else
    $\boldsymbol{v}_i^{ant} = \boldsymbol{v}_i^{act}\ \forall\ i$;
    Calcular δ con Ec. (4.3)
  end
end
return $\boldsymbol{v}^{act}$

Algorithm 4: Nuevo método de aglomeración de ruido basado en la función de montaña

$$d_{ij} = \begin{cases} \delta \times \tilde{M}(\boldsymbol{x}_j) & \text{si } k{=}c \\ \left\| \boldsymbol{v}_k^{ant} - \boldsymbol{x}_j \right\| & \text{de otro modo} \end{cases} \tag{4.4}$$

La ecuación (4.4) aplica el factor de densidad a la distancia de ruido δ de cada punto hacia el aglomerado de ruido. Por lo que el valor de todo $\boldsymbol{x} \in X$ es distinta de acuerdo a su posición en el espacio.

4.2 Resultados

Para poder evaluar el desempeño del método descrito en este capítulo, se realizaron muchas pruebas con los conjuntos de datos descritos en el capítulo anterior. Se descartó el CMD para ser analizado por ser un algoritmo sensible a los valores atípicos. Se realizaron las mismas pruebas en el algoritmo de Dave y Rehm (ambos descritos en el capítulo 2, página 19). Se consideraron las métricas que evalúan el desempeño del algoritmo descritas en el capítulo anterior y son las siguientes:

- **A**. Corresponde al promedio de distancias de ruido (sección 3.2.1 página 29)
- Mp. Se refiere al número de puntos que han sido correctamente asignados (página 30)
- DP. Diferencia de prototipos. Referente a la diferencia entre los prototipos *reales* y los centroides devueltos por el algoritmo (página 31)

Como se mencionó en el capítulo en el que se describen las métricas, se observan los valores que devuelven y se considera cada uno de manera distinta respecto a lo que se esté evaluando, i.e., el promedio de las distancias debe ser un valor alto, ya que indica una buena aglomeración, mientras que se espera que el valor de la *micro precisión* sea cercano a 1.

4.2.1 Comparativas entre los algoritmos

Para evaluar el desempeño del algoritmo desarrollado, además de éste se implementaron los algoritmos de Dave y el de Rehm; los tres algoritmos analizaron los mismos conjuntos de datos y se realizaron el mismo número de ejecuciones cada uno.

En esta sección se presenta una tabla comparativa con los resultados de los diferentes algoritmo evaluados. Se realizaron treinta ejecuciones de los algoritmos con cada conjuntos de datos. En la tabla se muestra el promedio y la *desviación estándar* de los resultados devueltos por las métricas. Para los algoritmos implementados se utilizaron los siguientes parámetros

- Dave. El valor de la constante λ fue de 0.6 (definido por el autor)
- Rehm. $\delta = r \times \alpha$. El valor de α manejado por los autores es de 1.5. El radio fue calculado con la ecuación (2.13), página 23
- Nuevo método. Se utilizó un valor para la constante λ de 1.3

En la tabla 4.1 se muestran los resultados finales de los algoritmos. Los valores que se consideran los mejores en cada ejemplo se han marcado en *negritas*. La columna 1 muestra el nombre de los datos. Pt son los puntos *buenos* y np los valores atípicos.

Datos	**Dave**	**Rehm**	**Nuevo Método**	Métricas
Iris (150 pt)	1.473333 ± 0.000196	1.602096 ± 0.000348	$\mathbf{1.800824 \pm 0.000086}$	A
	0.804776 ± 0.067894	0.869060 ± 0.099042	$\mathbf{0.881182 \pm 0.074565}$	Mp
	0.089587 ± 0.000149	$\mathbf{0.077153 \pm 0.000141}$	0.077801 ± 0.000014	DP
Breast (683 pt)	9.905846 ± 0.000041	8.749159 ± 0.000096	$\mathbf{14.147696 \pm 0.000182}$	A
	0.781918 ± 0.027219	0.747584 ± 0.067894	$\mathbf{0.836076 \pm 0.045364}$	Mp
	$\mathbf{0.501659 \pm 0.000010}$	1.375708 ± 0.000100	1.8163720 ± 0.000100	DP
Wine (178 pt)	0.799562 ± 0.000010	0.797771 ± 0.000056	$\mathbf{0.934328 \pm 0.000002}$	A
	0.857416 ± 0.000011	0.808989 ± 0.000845	$\mathbf{0.875843 \pm 0.001777}$	Mp
	0.080802 ± 0.000101	$\mathbf{0.080676 \pm 0.000099}$	0.081828 ± 0.000180	DP
Bupa (345 pt)	0.461695 ± 0.001420	0.447692 ± 0.000240	$\mathbf{0.987348 \pm 0.000090}$	A
	0.332464 ± 0.020904	0.316232 ± 0.000185	$\mathbf{0.559565 \pm 0.000031}$	Mp
	0.099283 ± 0.000445	0.109145 ± 0.000199	$\mathbf{0.092815 \pm 0.000082}$	DP
Pima (768 pt)	0.537984 ± 0.006646	0.574808 ± 0.016023	$\mathbf{0.647784 \pm 0.001162}$	A
	0.328125 ± 0.005719	0.397135 ± 0.000422	$\mathbf{0.527604 \pm 0.007412}$	Mp
	1.093270 ± 0.025105	1.096671 ± 0.008758	$\mathbf{0.107508 \pm 0.000461}$	DP
Ionos (351 pt)	$\mathbf{3.664874 \pm 0.000019}$	3.250681 ± 0.000138	1.120544 ± 0.000010	A
	$\mathbf{0.789762 \pm 0.000406}$	0.711766 ± 0.000100	0.691026 ± 0.002410	Mp
	$\mathbf{0.997412 \pm 0.000036}$	1.046735 ± 0.000038	1.160947 ± 0.000003	DP
Segment (2310 pt)	1.183889 ± 0.000441	$\mathbf{1.311357 \pm 0.009185}$	0.914168 ± 0.024418	A
	$\mathbf{0.695844 \pm 0.123718}$	0.662208 ± 0.009570	0.656388 ± 0.020329	Mp
	$\mathbf{0.945982 \pm 0.087912}$	0.953152 ± 0.012196	1.157824 ± 0.018034	DP
Normal (200 pt)	3.035853 ± 1.387966	0.000000 ± 0.000000	$\mathbf{3.413348 \pm 0.000014}$	A
	0.725001 ± 0.083854	$\mathbf{0.995000 \pm 0.005321}$	0.933843 ± 0.031480	Mp
	0.271657 ± 0.283242	$\mathbf{0.068944 \pm 0.000118}$	0.085331 ± 0.000136	DP
Normal-ruido	5.503414 ± 0.000536	9.90742 ± 0.000000	$\mathbf{11.405176 \pm 0.000122}$	A
	0.644635 ± 0.001919	0.804077 ± 0.000084	$\mathbf{0.836910 \pm 0.000860}$	Mp
	0.140427 ± 0.000173	0.059511 ± 0.000968	$\mathbf{0.041073 \pm 0.000036}$	DP
Basico (10pt + 3np)	8.861819 ± 0.368780	7.884823 ± 0.618940	$\mathbf{9.323460 \pm 0.000000}$	A
	0.330769 ± 0.210663	0.176923 ± 0.172005	$\mathbf{0.869231 \pm 0.397201}$	Mp
	6.320933 ± 2.767726	6.870089 ± 2.500510	$\mathbf{0.001134 \pm 0.000000}$	DP
Srand (250pt)	8.705178 ± 0.231697	2.292340 ± 0.000149	$\mathbf{2.590066 \pm 0.000162}$	A
	0.646003 ± 0.386012	0.819240 ± 0.032660	$\mathbf{0.832151 \pm 0.333919}$	Mp
	6.708216 ± 0.000000	0.634150 ± 0.000157	$\mathbf{0.254820 \pm 0.000254}$	DP

Tabla 4.1: Se muestra los resultados de los tres algoritmo: Dave, Rehm y el nuevo método.

Observe que en casi todos los ejemplos se mantiene la tendencia de que el valor de A es el mayor para el nuevo método, así como también el los valores de Mp cercanos a 1 ocurren en éste; también la diferencia de prototipos DP es muy pequeña aún cuando ésta se inclina para alguno de los otros dos algoritmos.

Considere por ejemplo el caso del primer ejemplo: *iris*. De acuerdo a la definición de la primera métrica (la del promedio de distancias de ruido), el valor de A debe ser grande, ya que se está evaluando el promedio de las distancia de los elementos en N hacia los aglomerados buenos. El mejor resultado corresponde al nuevo método. Observe también que el valor de Mp mas alto (cercano a 1) lo tiene el mismo algoritmo. En el caso de DP, el algoritmo Hiper tiene el mejor resultado, aunque la diferencia respecto al nuevo método es muy pequeña.

El ejemplo *normal* (figura 4.2) es un conjunto de datos del que se sabe de antemano que no tiene valores atípicos. La métrica A se descarta para hiper, pues no se puede asegurar una buena aglomeración tal y como se explicó en la sección 3.2.1 página 29.

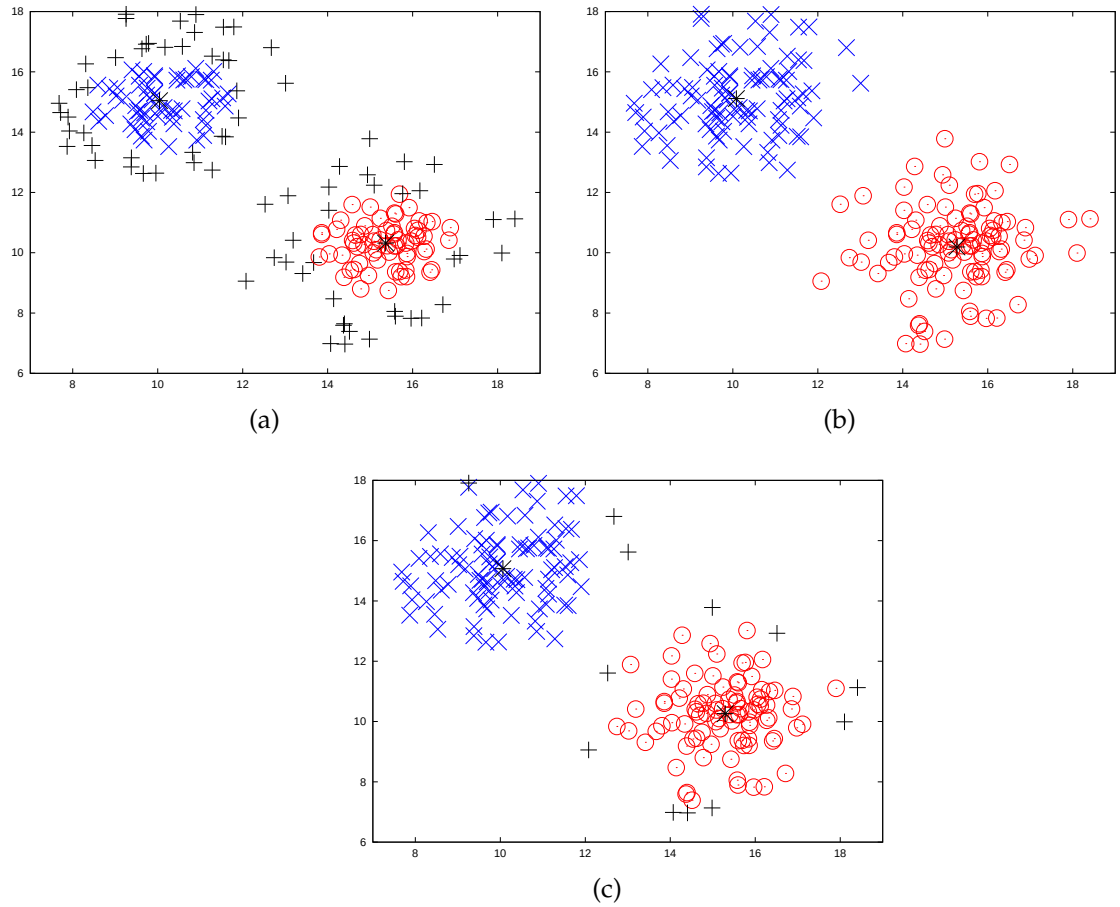


Figura 4.2: Resultados del conjunto *normal*. (a) Método de Dave. (b) Método de Rehm. (c) Nuevo método.

Obsérvese que tanto Dave como el nuevo método muestran valores altos de A, lo que sugiere un buen desempeño. El mejor valor de A es el del nuevo método. Ya que los

puntos que ha detectado se encuentran más alejados de los aglomerados buenos, por lo que el promedio de su distancia es mayor.

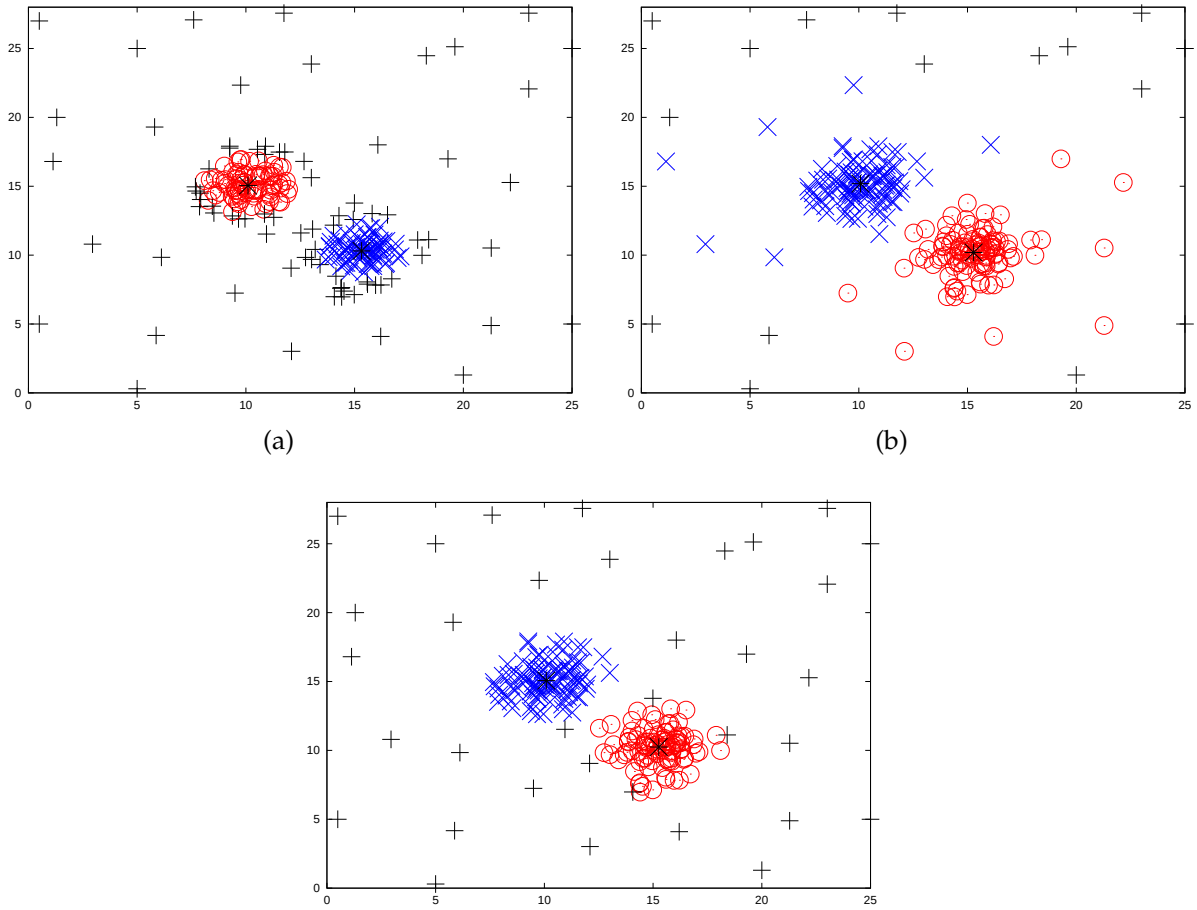


Figura 4.3: Resultados del conjunto *normal-ruido*. (a) Método de Dave. (b) Método de Rehm. (c) Nuevo método.

Para observar cuál algoritmo se desempeñó mejor en el manejo de los valores atípicos, obsérvese la figura 4.3. Allí se muestran los resultados gráficos de los algoritmos para el ejemplo *normal-ruido*. Los puntos buenos se simbolizan con '○' y '×', y los valores atípicos con '+'. La imagen 4.3c, que es la salida del nuevo método, muestra que todos los valores atípicos fueron detectados. La imagen 4.3b, que corresponde al método de Rehm, muestra algunos puntos malos que fueron identificados por error como puntos buenos.

El método de Dave (figura 4.3a) no lo hace tan mal. Lo que sucedió es que la amplitud del *alcance* del aglomerado es más reducido que el de los otros algoritmos. En este caso se requiere *calibrar* algunos parámetros para ampliar un poco esta amplitud.

Ahora observe el caso del ejemplo *srand* (figura 4.4), los mejores resultados son para el nuevo método. Dave (figura 4.4a) no realiza una buena aglomeración, ya que los prototipos son atraídos por el aglomerado de mayor densidad; lo que provoca un valor bajo en la micro precisión. Rehm no lo hace tan mal; consigue identificar los dos aglomerados aunque la amplitud de uno de éstos no es muy grande. El nuevo método (figura 4.4c) tambiénén consigue identificar a los dos aglomerados. Tuvo un valor bueno de Mp y logró detectar 34 de 50 puntos buenos.

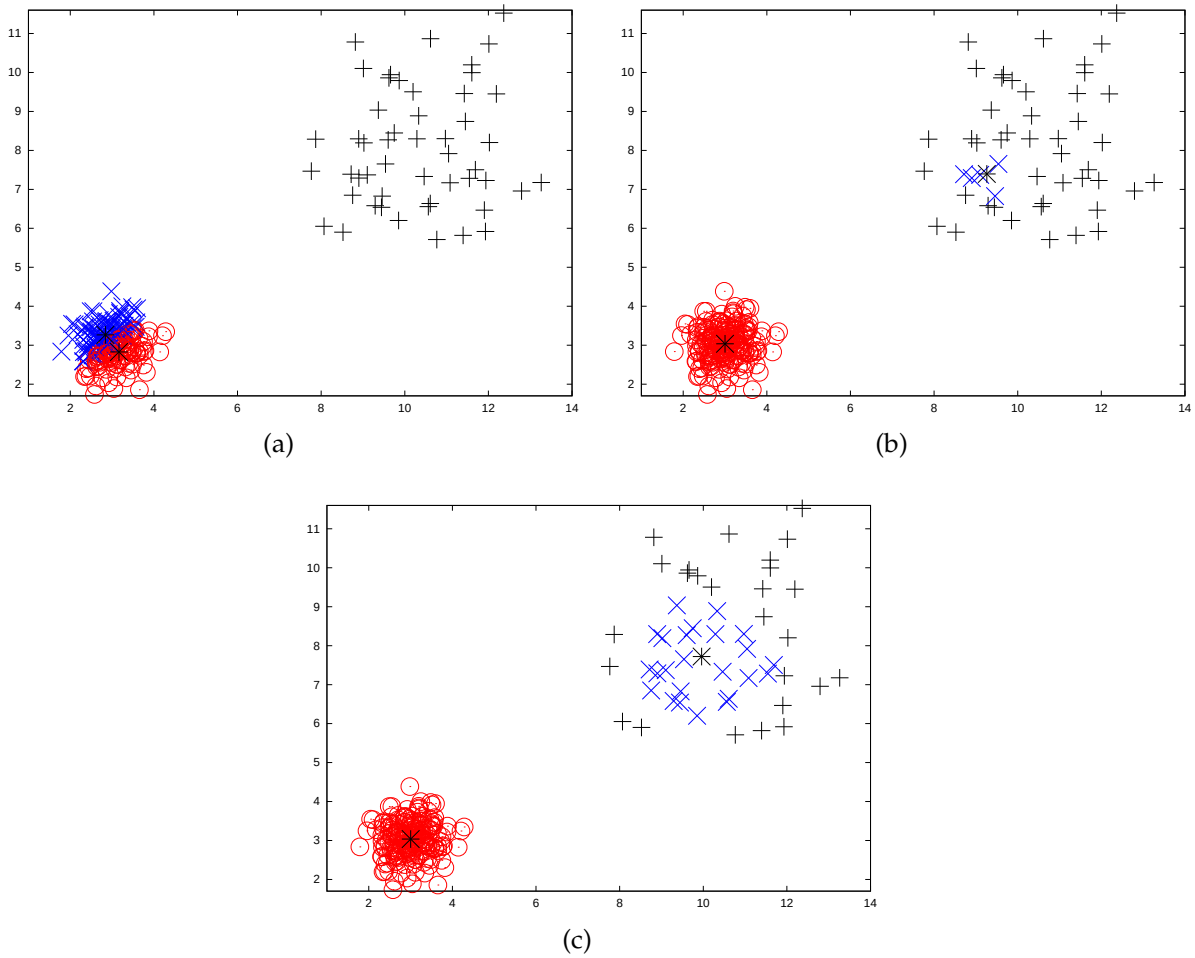


Figura 4.4: Resultados del conjunto *srand*. Dave realizan una mala aglomeración. Con Rehm (b) y el nuevo método (c) se observan la buena ubicación de los dos prototipos.

4.3 Resumen

En este capítulo se presentó de manera formal el algoritmo diseñado en este trabajo de tesis. Se explicó la metodología utilizada y cuál es la idea principal propuesta para el cálculo de la distancia de ruido. A diferencia de los otros trabajos, en éste se ha propone una apreciación distinta para este valor. Ésta es diferente para cada punto dependiendo del *estado* en el que se encuentre dentro del espacio de puntos. Estos dos estados se definen como aislado y no aislado.

El estado de cada punto se calcula utilizando la función de la montaña, la cual es una función que penaliza drásticamente a cualquier punto que se encuentre alejado de los demás; por lo que sirve para determinar un factor de densidad de cada punto. Con este factor de densidad, se estima la distancia cada punto hacia el aglomerado de ruido.

También se realizaron pruebas en los algoritmo de Dave, el del Hiper volumen y el nuevo método. Estas pruebas fueron hechas con los mismos conjuntos de datos y los resultados presentados en una tabla comparativa, en la que se muestra el desempeño de los tres algoritmos que fueron evaluados. Puede observarse que los mejores resultados son devueltos por el algoritmo desarrollado en este trabajo de investigación.

Capítulo 5

Nuevas variantes del algoritmo

> Para todo problema hay una solución fácil, que suele ser ingeniosa, plausible...y equivocada.
>
> H.L. Mencken

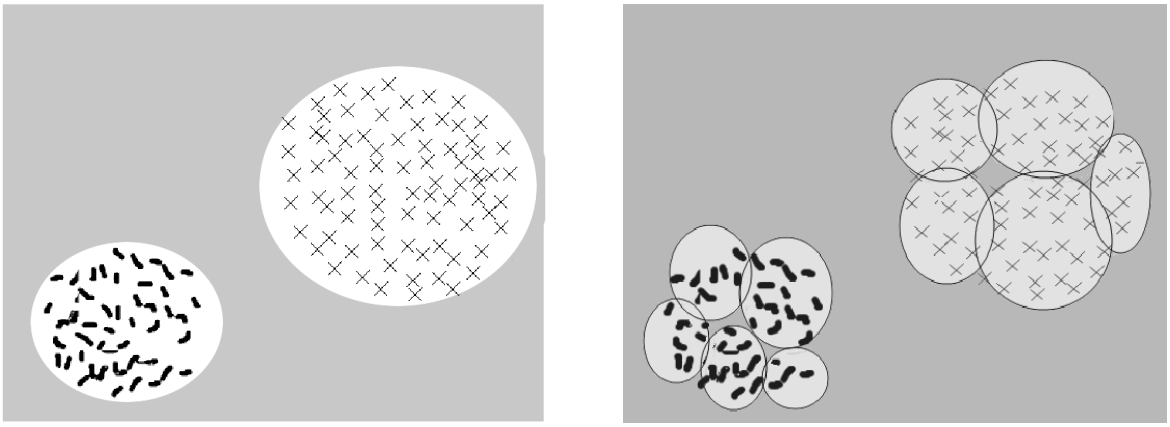
Durante el desarrollo del algoritmo se desarrollaron distintas variantes para resolver el problema de aglomeración. Éstas surgieron principalmente por la necesidad de resolver el problema de la *diferencia de densidad* que había en algunos de los conjuntos de datos. Este problema ocurre normalmente en aquellos conjuntos de datos en los que la densidad de los aglomerados que los forman es diferente.

5.1 Cálculo del híper radio

En la mayoría de los conjuntos de aglomeración, se puede apreciar que mientras hay zonas del espacio ocupadas por datos, también existen zonas *vacías* o con poca o nula presencia de éstos. Se ha considerado que si se logra eliminar estos espacios vacíos y enfocar el desempeño del algoritmo sólo a las zonas ocupadas por puntos, se puede realizar una mejor aglomeración de los datos.

Entonces la idea principal de esta variante consiste en calcular un *volumen global real* del espacio de puntos y descartar los espacios vacíos. A manera de ejemplo, observe la imagen a) de la figura 5.1a. El espacio en color gris no contiene elementos y se considera *vacío*; sólo los espacios en blanco (en donde se encuentran los puntos) son considerados durante el cálculo del *volumen global*.

De una manera general, se hace lo siguiente. Sea $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, se calculan c=10 aglomerados en X usando CMD. El valor de c es un valor que puede ajustarse de acuerdo al problema, sin embargo, dio buenos resultados para los conjuntos que fueron evaluados.



(a) (b)

Figura 5.1: (a) El espacio en gris se considera *vacío*, sin puntos. b) El CMD calcula c aglomerados con los cuales se calcula el *volumen real* del espacio.

Se tienen entonces diez prototipos μ_i, cada uno un determinado número de puntos. Ahora se requiere el radio i-ésimo radio de cada prototipo. Éstos se obtienen con

$$r_i = \max_{\boldsymbol{x} \in \mu_i} (\boldsymbol{x}_j - \mu_i) \;\; \forall \;\; \boldsymbol{x} \in \mu_i$$

donde r_i es la distancia del punto $\boldsymbol{x} \in \mu_i$ más alejado del centroide de μ_i. Entonces el i-ésimo volumen se calcula con

$$v_i = \frac{\pi^{\frac{n}{2}} r_i^n}{\Gamma(\frac{n}{2}+1)}$$

que es la formula para calcular el volumen de una *híper esfera* de dimensión n y de radio r. Ahora, considerando todos los volúmenes v_i, se calcula el *volumen global* V con

$$V = \frac{1}{gc} \sum_{i=1}^{c} v_i$$

donde gc son los aglomerados *buenos*; en el ejemplo de la figura 5.1b, $gc = 2$.

Finalmente, se obtiene el valor del *híper radio* h_r con

$$h_r = \sqrt{\frac{V \cdot \Gamma(\frac{n}{2}+1)}{\pi^{(\frac{n}{2})}}}$$

Este valor proporciona una idea de la estructura de los datos. Con éste se realiza una modificación al cálculo de la función M, la cual se define con

$$M(\boldsymbol{x}_i) = \begin{cases} M(\boldsymbol{x}_i) + 1 & \text{si } \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\| \leq h_r \times n \\ 0 & \text{de otro modo} \end{cases} \tag{5.1}$$

y se normaliza con

$$Z = \sum_{i=1}^{n} \mathrm{M}(\boldsymbol{x}_i), \text{ y}$$
$$\tilde{M}(\boldsymbol{x}_i) = \frac{1}{Z}\mathrm{M}(\boldsymbol{x}_i) \ \ \forall \ \boldsymbol{x}_i$$

que es la misma ecuación que normaliza la función del algoritmo 4 (página 37). Lo que se propone con esta modificación es obtener un factor de densidad que esté en función del número de elementos que se localicen dentro del área híper esférica de radio h_r que rodea cada $\boldsymbol{x}_i$.

El algoritmo 5 muestra los pasos del método explicado en esta versión. En el paso 4, δ es calculada con $\delta = h_r \times n$. Sin embargo, en el desarrollo del algoritmo, δ se calcula con la ecuación (4.3) que aquí se muestra de nuevo

$$\delta = \lambda \left[\frac{\sum_{j=1}^{c-1} \sum_{k=1}^{n} d_{jk}}{n(c-1)} \right]$$

En el paso 9, se utiliza el mismo procedimiento definido en la ecuación (2.12) para calcular la actualización de la matriz u_{ij}. Es decir

$$d_{ij} = \begin{cases} \delta \times \tilde{M}(\boldsymbol{x}_i) & \text{si } k{=}c \\ \left\| \boldsymbol{v}_k - \boldsymbol{x}_j \right\| & \text{de otro modo} \end{cases}$$

donde sólo se realiza este cálculo cuando se trata de los aglomerados buenos; llegado el turno al aglomerado de ruido, se aplica el factor de densidad al punto $\boldsymbol{x}$.

Para las pruebas, se utilizó un valor de $\lambda = 1$. No se consideró el conjunto *básico* porque tiene pocos puntos y el método explicado en esta sección no tendría ningún sentido para este ejemplo. Los resultados las pruebas de este algoritmo se presentan en la tabla 5.1 de la página 51.

Entrada: $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, k, ε, *band*=true
Salida : Centroides $\boldsymbol{v}_i$, con i=$1, \ldots, c$

Inicializar matriz U;
Calcular h_r con el método de la sección 5.1;
Obtener $\tilde{M}(\boldsymbol{x}_i)\ \forall\ \boldsymbol{x}_i$ con Ec. (5.1);
Calcular $\delta \times h_r$;
while band **do**
  for $i = 1$ **to** c **do**

$$\boldsymbol{v}_i^{act} = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m} \quad \forall\ j=1, \ldots, n$$

  end
  for $i = 1$ **to** c **do**

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| \boldsymbol{v}_i^{ant} - \boldsymbol{x}_j \right\|}{\left\| \boldsymbol{v}_k^{ant} - \boldsymbol{x}_j \right\|} \right)^{\frac{1}{m-1}}} \quad \forall\ j=1, \ldots, n$$

  end
  if $\frac{1}{c} \sum_{i=1}^{c} \left\| \boldsymbol{v}_i^{ant} - \boldsymbol{v}_i^{act} \right\| \leq \varepsilon$ **then**
    band=false;
  else
    $\boldsymbol{v}_i^{ant} = \boldsymbol{v}_i^{act}\ \forall\ i$;
    Calcular δ con Ec. (4.3)
  end
end
return $\boldsymbol{v}^{act}$

Algorithm 5: Variante del algoritmo que calcula un híper radio para obtener el factor de densidad de cada punto.

5.2 Usando el árbol de conexión mínima

Aunque el uso del *árbol de conexión mínima* (ACM) como una técnica de aglomeración ha tenido buena aceptación, también es cierto que su uso se recomienda para aquellos ejemplos en donde los aglomerados esán bien definidos. El cálculo del ACM requiere de una gran capacidad de cómputo cuando se trata de conjuntos de datos grandes, por ejemplo, el tamaño de una imagen.

De manera intuitiva, el uso de ACM como un método de aglomeración se debe principalmente a que es fácil de implementar y su desempeño ha sido aceptable en una variedad de distribuciones [14].

De manera general, un conjunto de datos $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, puede verse como un grafo $G = (V, E)$, en donde V lo forman los elementos de X, y E son las aristas asociadas con un *peso* w_i para toda $\boldsymbol{x}_i$ (véase el apéndice A.2 para una mejor referencia). En el caso de los problemas de aglomeración, el peso de una arista es la distancia de un punto contra todos los demás. Para una referencia más formal de *grafo*, véase el apéndice A.2

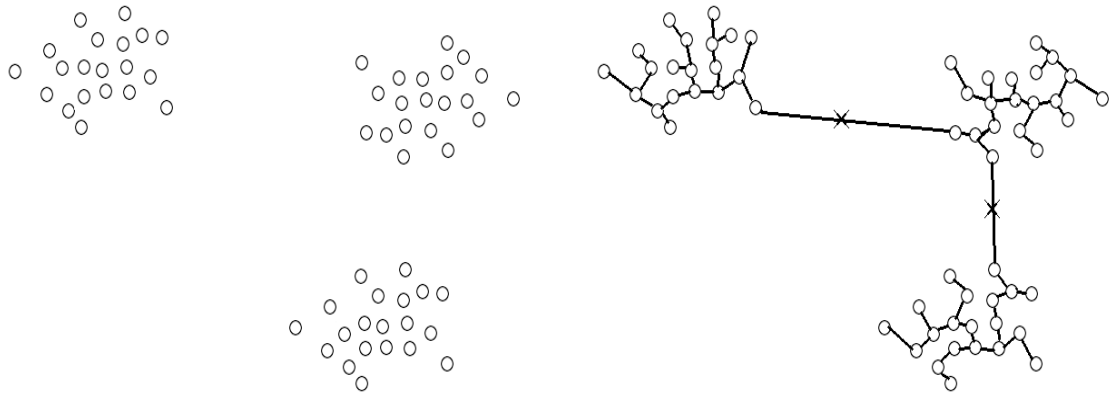


Figura 5.2: (a) Tres aglomerados bien definidos b) ACM con las aristas más largas marcadas.

Zahn [15] demostró cómo puede usarse el ACM para detectar aglomerados. Por definición, una arista del ACM es aquella con el valor más bajo (su peso) que conecta a un par de nodos. Debido a que el peso de las aristas es la distancia entre un par de puntos, la posición que ocupen éstos es importante. Se considera que una arista que una dos elementos del mismo aglomerado debe ser pequeña, mientras que será muy grande para un par que se encuentren en diferentes aglomerados.

La idea básica puede verse en la figura 5.2. La imagen de la izquierda muestra tres aglomerados bien definidos. Al obtener el ACM, se tiene un camino (*path*) que recorre todos los nodos. Las aristas marcadas con una $\times$, que son las más largas, se eliminan y los componentes conectados restantes serán los aglomerados buscados.

El problema, sin embargo, es que el eliminar las aristas más largas puede no ser suficiente para obtener los aglomerados correctos; en ocasiones éstos no tienen un tamaño razonable para considerarse aglomerados o bien, su nivel de densidad es bajo. Para obtener aglomerados *buenos* se necesita agregar ciertas restricciones a las aristas que serán eliminadas.

Para aplicar el ACM se realiza lo siguiente. Sea $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ el conjunto de puntos; se forma el grafo G utilizando X, donde el conjunto de vértices es el mismo X y el conjunto de aristas es $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ para toda $i \neq j$. El peso de cada arista $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ es $\|(\boldsymbol{x}_i - \boldsymbol{x}_j)\|$

Se calcula entonces el ACM en G, que queda denotado por $\bar{G} = (V, E^*)$. El promedio p de todas las aristas $e \in E^*$ con peso mínimo se calcula con

$$p = \frac{1}{|E^*|} \sum_{e^* \in E^*} e^*$$

El valor de p da una idea de la estructura de los datos. El uso que se le da es similar al del híper radio explicado en la sección anterior. La función M se calcula de la misma manera con la ecuación (5.1), sólo que se substituye el valor de h_r por el de p. Los pasos restantes del algoritmo son los mismos. El valor de δ se calcula en el paso 16 con $\delta = p \times n$ y durante el desarrollo del algoritmo, en el paso 11, se utiliza la ecuación (4.3). En general se sigue el mismo proceso que en el algoritmo explicado en la sección anterior.

En esta versión se realizaron pruebas calibrando el valor de λ y se consideró finalmente $\lambda = 1$. Los resultados, al igual que los de la variante anterior, se muestran en la tabla 5.1.

5.3 Comparación final

En esta sección se presenta una tabla comparativa mostrando los resultados obtenidos con los nuevos métodos de aglomeración de ruido: el método basado en la función de montaña y las dos variantes, la que utiliza el híper radio y la que calcula el ACM. En la tabla 5.1 se presentan los mejores resultados de cada uno de éstos. En el caso de los algoritmos del híper radio y el ACM, no se consideró el ejemplo *básico* por ser un conjunto con muy pocos elementos.

Obsérvese que las versiones explicadas en este capítulo muestran mejoras en algunos ejemplos, superando los resultados del nuevo método. Tal es el caso del conjunto *pima* e *ionos*; en el primero, Dave obtiene un mejor valor de A, aunque su Mp no es muy bueno. Para el caso de *Ionos* los mejores resultados los obtiene la variante del ACM sobre los otros dos algoritmos.

Datos	**Hiper radio**	ACM	**Nuevo Método**	Métricas
Iris (150 pt)	1.672933 ± 0.000000	1.348077 ± 0.000001	$\mathbf{1.800824 \pm 0.000086}$	A
	0.866667 ± 0.115931	0.806667 ± 0.075895	$\mathbf{0.881182 \pm 0.074565}$	Mp
	0.086736 ± 0.000000	0.105079 ± 0.000013	$\mathbf{0.077801 \pm 0.000014}$	DP
Breast (683 pt)	10.729219 ± 0.000000	13.099127 ± 0.001014	$\mathbf{14.147696 \pm 0.000182}$	A
	0.778917 ± 0.034219	0.8175840 ± 0.463571	$\mathbf{0.836076 \pm 0.045364}$	Mp
	$\mathbf{0.680110 \pm 0.000245}$	1.444180 ± 0.005910	1.8163720 ± 0.000100	DP
Wine (178 pt)	0.887232 ± 0.000311	0.797771 ± 0.048541	$\mathbf{0.954328 \pm 0.000002}$	A
	0.845640 ± 0.000103	0.808989 ± 0.002911	$\mathbf{0.875843 \pm 0.001777}$	Mp
	0.095203 ± 0.000001	$\mathbf{0.071022 \pm 0.000002}$	0.081828 ± 0.000180	DP
Bupa (345 pt)	0.709592 ± 0.011390	0.447692 ± 0.000001	$\mathbf{0.987348 \pm 0.000090}$	A
	0.521739 ± 0.001021	0.336232 ± 0.018835	$\mathbf{0.559565 \pm 0.000031}$	Mp
	0.073176 ± 0.007703	0.122914 ± 0.000332	$\mathbf{0.092815 \pm 0.000082}$	DP
Pima (768 pt)	$\mathbf{0.807937 \pm 0.002941}$	0.587949 ± 0.007781	0.647784 ± 0.001162	A
	$\mathbf{0.636719 \pm 0.000024}$	0.397183 ± 0.901620	0.527604 ± 0.007412	Mp
	0.133568 ± 0.000049	$\mathbf{0.101672 \pm 0.241781}$	0.107508 ± 0.000461	DP
Ionos (351 pt)	3.250667 ± 0.000117	$\mathbf{3.942022 \pm 0.000000}$	1.120544 ± 0.000010	A
	0.766781 ± 0.094322	$\mathbf{0.814387 \pm 0.144177}$	0.691026 ± 0.002410	Mp
	1.046739 ± 0.000032	$\mathbf{1.010014 \pm 0.000000}$	1.160947 ± 0.000003	DP
Segment (2310 pt)	0.755103 ± 0.000084	0.713005 ± 0.002746	$\mathbf{0.914168 \pm 0.024418}$	A
	0.621860 ± 0.077445	0.598268 ± 0.003162	$\mathbf{0.656388 \pm 0.020329}$	Mp
	$\mathbf{0.146755 \pm 0.001175}$	0.147104 ± 0.001526	1.157824 ± 0.018034	DP
Normal (200 pt)	0.000000 ± 0.000000	3.227759 ± 0.000631	$\mathbf{3.413348 \pm 0.000014}$	A
	$\mathbf{0.944831 \pm 0.000724}$	0.905001 ± 0.000122	0.933843 ± 0.031480	Mp
	0.862522 ± 0.000013	0.130936 ± 0.000000	$\mathbf{0.085331 \pm 0.000136}$	DP
normal-ruido (200 + 30 np)	10.421244 ± 0.000016	9.036639 ± 0.003259	$\mathbf{11.405176 \pm 0.000122}$	A
	0.814077 ± 0.000010	0.828326 ± 0.000011	$\mathbf{0.836910 \pm 0.000860}$	Mp
	0.074178 ± 0.000030	0.059149 ± 0.002854	$\mathbf{0.041073 \pm 0.000036}$	DP
basico (10pt + 3np)	–	-	$\mathbf{9.323460 \pm 0.000000}$	A
	–	-	$\mathbf{0.869231 \pm 0.397201}$	Mp
	–	-	$\mathbf{0.001134 \pm 0.000000}$	DP
srand (250pt)	2.177735 ± 0.000000	2.121864 ± 0.000000	$\mathbf{2.590066 \pm 0.000162}$	A
	0.797203 ± 0.171362	0.769231 ± 0.174520	$\mathbf{0.832151 \pm 0.333919}$	Mp
	$\mathbf{0.191416 \pm 0.000000}$	0.473698 ± 0.003462	0.254820 ± 0.000254	DP

Tabla 5.1: Se muestran los resultados de los tres algoritmo desarrollados.

En general puede observarse que los algoritmo de este capítulo ofrecen un mayor equilibrio en cuanto a buenos o malos resultados. Hay algunos ejemplos en los que sus resultados son mejores que los del nuevo método y otros en los que predomina el buen desempeño de este último. De hecho, en los conjuntos básicos *normal, normal-ruido, básico* y *srand*, que son los que pueden comprobarse visualmente, los mejores resultados corresponden al algoritmo basado en la función de montaña.

En el caso del ejemplos *srand* (que corresponde a la diferencia de densidad de puntos), las dos variantes no lo hacen tan mal, ya que consiguen identificar a los dos aglomerados, aunque con un valor bajo de Mp en comparación con el nuevo método.

En los resultados mostrados en la tabla 5.1 se observa que las variantes explicadas en este capítulo tienen un desempeño aceptable. Sus mejores resultados fueron similares y por esta razón es difícil tomar una desición respecto a cúal es el mejor. En los conjuntos *normal, normal-ruido, básico* y *srand*; ambos no tienen un mal desempeño, si bien no superan los resultados del nuevo método, éstos pueden considerarse como buenos.

5.4 Resumen

En este trabajo se presentaron las dos versiones que surgieron durante el desarrollo del algoritmo. En una de éstas se calcula el volumen global sin tomar en cuenta los espacios vacíos de la estructura de los datos. Con este volumen se obtiene un valor llamado *híper radio* con el que se calcula la región de vecindad de cada punto para conocer su estado.

La otra utiliza el *árbol de conexión mínima* para tener una idea de la estructura de los elementos del conjunto. A partir del ACM, se calcula el promedio p de las aristas con peso mínimo; este valor se utiliza del mismo modo que el híper radio y se hace uso del mismo procedimiento para el cálculo de la función M.

A diferencia del nuevo método de aglomeración explicado en el capítulo 4, puede concluirse entonces que en estas dos versiones lo único que cambia es la manera de considerar el espacio de puntos, es decir, el método que usan para calcular el valor que proporciona una idea general de la estructura del conjunto de datos.

Respecto al ACM, una de sus principales desventajas es sin duda el proceso de cómputo que requiere cuando se trabaja con conjuntos de datos grandes. Además de ser costosa, la manera de realizar el análisis de aglomeraciones utilizando el ACM pudiera no ser eficiente en ejemplos en los que los aglomerados no estén bien definidos, no tengan la densidad requerida o bien no sean de un tamaño razonable.

Capítulo 6

Aplicación del algoritmo sobre un problema real

Nada sale tal y como está previsto.

Ley de Murphi #35 acerca de la máxima fatalidad con el mínimo esfuerzo

Los algoritmos de aglomeración han sido altamente utilizados para el análisis de imágenes [11]. Se ha implementado el CMD para el análisis de imágenes dentro de las las áreas biológicas y médicas. En este capítulo se muestra una propuesta de aplicación del algoritmo de aglomeración de ruido desarrollado en este trabajo de investigación, con lo que cubre el segundo objetivo principal de esta tesis.

En esta aplicación se analizan imágenes de tejidos cerebrales. Normalmente, la mayoría de las imágenes de este tipo se presentan en *cortes axiales* en los que pueden apreciarse las distintas partes del cerebro. Para este trabajo, se utilizaron imágenes de cerebros humanos que presentan lesiones visibles y evidentes (tumores).

El objetivo principal es identificar correctamente las distintas lesiones cerebrales que aparecen en una serie de imágenes y observar cómo se comporta el algoritmo ante la presencia de valores atípicos en el análisis de las mismas.

Sin embargo, debe mencionarse que esta es una aplicación limitada, ya que se trabajó con ciertas limitantes. Por ejemplo, no se tuvo a la mano suficiente material, en este caso las imágenes, para realizar pruebas; además la calidad de éstas era baja, ya que se obtuvieron de la red; y quizá la más importante es que no se contó con la opinión de un experto en el tema. Pese a esto, los resultados que se presentan al final de este capítulo se consideran buenos y dan lugar a trabajos futuros siguiendo esta misma línea de investigacíon.

En algunos trabajos que aplican CMD para el análisis de imágenes cerebrales, como el desarrollado por Z. Hou [11], los resultados obtenidos se vieron afectados por la

presencia de valores atípicos. Éstos provocan que se pierda *nitidez* en las imágenes analizadas por el CMD, por lo que la *distorción* en las mismas provoca que se realicen malas lecturas y diagnósticos equivocados.

Esta aplicación no plantea una solución al problema de estos autores, sino más bien pretende probar el desempeño del algoritmo desarrollado en este trabajo en otro tipo particular de datos.

6.1 Acerca de la implementación

La implementación del algoritmo diseñado en este trabajo se realizó principalmente en Octave versión 3.0. El trabajo se desarrolló sobre el sistema operativo Linux, en la versión 9.04 de Ubuntu. La computadora utilizada fue una *Lap top* marca Acer con un procesador *AMD Turion* de 2.2 GHz y una memoria *RAM* de 1.5 G.

El lenguaje utilizado fue Octave, que es la versión libre de Matlab. Se decidió utilizar este *software* debido a que facilita la manipulación de arreglos, matrices y las operaciones que pueden hacerse sobre éstos. Además es muy *amigable* para el caso de efectos visuales, tales como mostrar en pantalla las imágenes y los resultados de los algoritmos en gráficas.

6.2 Síndromes Neoplásicos

Los *síndromes neoplásicos* (*Neoplastic Disease*) son uno de los grupos de lesiones cerebrales que deben (en la mayoría de los casos) ser tratados quirúrgicamente. A estas lesiones se les conoce como *tumores*[16]. Un tumor es un crecimiento de tejido causado por la división anormal y no controlada de *células*; éstos pueden diseminarse a otras partes del cuerpo a través del *sistema linfático* o bien, por el *torrente sanguíneo*.

Los tumores pueden ser *benignos* o *malignos* (cáncer) [17]. Los primeros crecen lentamente y pueden permanecer del mismo tamaño por mucho años. Los tumores malignos, por otro lado, van creciendo implacablemente, minando e invadiendo el tejido a su alrededor, formando *úlceras, abscesos, fracturas* (en huesos) entre otros.

Los tumores malignos que se desarrollan en el cerebro son denominados de acuerdo al tipo de célula que los origina o al tejido donde se desarrollan; los más comunes son *sarcoma, meningioma y ependimomas* [18]. Las imágenes de este tipo de lesiones fueron las que se analizaron en este trabajo.

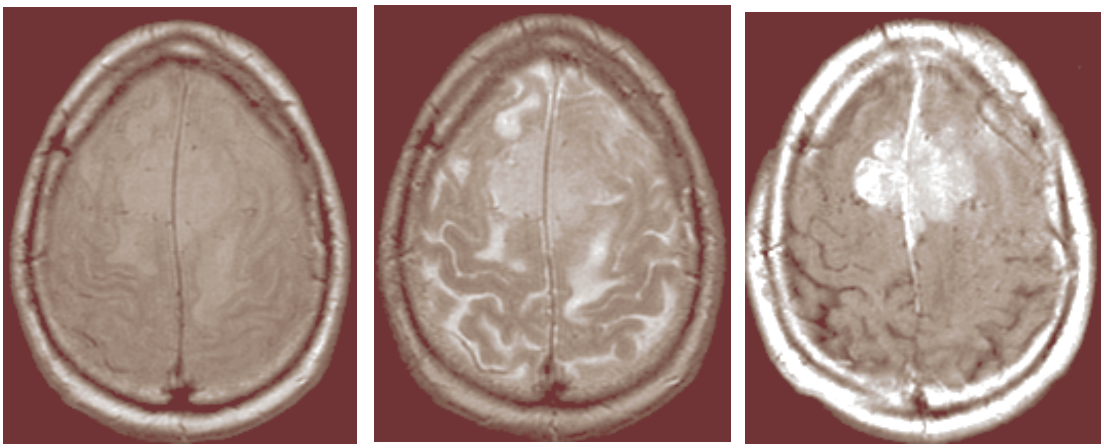


Figura 6.1: Imágenes que presentan el mismo daño cerebral pero tomadas con diferentes tecnologías

6.2.1 Resonancia magnética

La *resonancia magnética* es una técnica de diagnóstico en la cual se introduce a un paciente dentro de un *campo magnético* y mediante la aplicación de determinados estímulos se consigue una *resonancia* de los núcleos de sus *átomos*, recuperando la energía liberada en forma de señales que, tratada adecuadamente se transforma en una imágen *tomográfica*.

Un volumen de tejido del organismo tiene una densidad específica en núcleos de H^+. Así, el agua tendrá una densidad diferente en la sangre, huesos y en cada músculo o víscera. La calidad de las señales de resonancia emitidos por la materia (los tejidos cerebrales) dependen de tres parámetros independientes T_1, T_2 y la *densidad protónica* o DP.

En la figura 6.1 se presenta tres *capas* (*slides*) de un cerebro humano. Las tres capas presentan la misma lesión cerebral pero fueron tomadas con distintas técnicas (T_1, T_2 y DP respectivamente).

6.2.2 Partes principales en tejidos cerebrales

En los análisis de imágenes cerebrales, siempre surgen las siguientes preguntas, ¿Qué es lo que se está buscando? y ¿Cúal es el número c de aglomerados adecuado que debe darse como entrada?. Las tres principales partes del cerebro son el *líquido cefaloraquideo* (LCR), la *sustancia gris* (SG) y la *sustancia blanca* (SB); y normalmente son el objeto de análisis en este tipo de implementaciones.

Las imágenes utilizadas fueron tomadas del sitio web *The whole brain Atlas* [19], el cual es un recurso en línea de imágenes del *sistema nervioso central*. En este sitio se integra información clínica y un gran repositorio de imágenes. Finalmente estas imágenes son archivos, y por lo tanto están formadas por un conjunto de *píxeles*[1]; éstos tienen un valor en *escala de grises* de 0 (negro) hasta 255 (blanco). Cada píxel representa información de cada punto de la imagen.

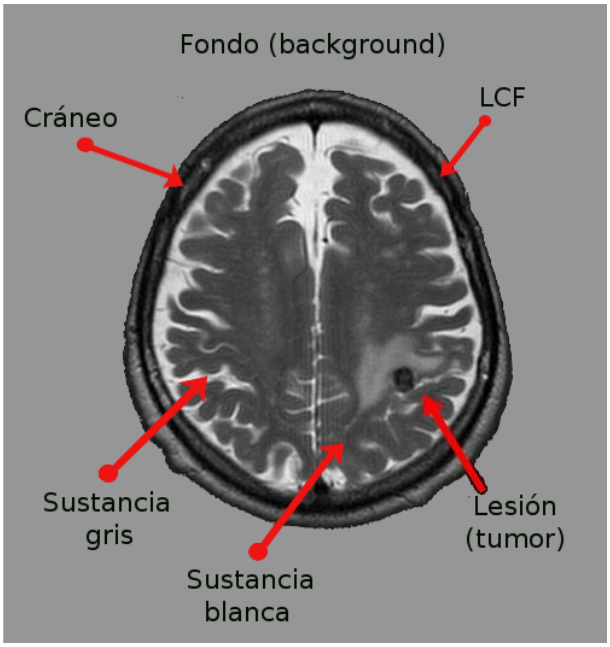


Figura 6.2: Se muestran las seis partes principales de la imagen que serán analizadas.

Para este trabajo, las imágenes que se analizan presentan lesiones cerebrales evidentes y visibles (tumores). Para los estudiosos del cerebro, el cráneo no es de interés y es removido de la imagen; sin embargo, por tratarse de un ejercicio de aglomeraciones, en este trabajo sí es considerado tal y como lo presentan las imágenes originales.

En la imagen 6.2 se aprecia un corte longitudinal de un cerebro humano. Se señalan las partes de éste que serán analizadas por el algoritmo. Así que el *cráneo, el tumor* y *el fondo la imagen*, junto con el *líquido cefaloraquideo*, la *sustancia gris*, la *sustancia blanca* y el *aglomerado de ruido*, son siete los aglomerados que tiene que identificar el algoritmo.

Lo anterior se debe a que tanto el tumor como el fondo de la imagen (*background*), presentan características diferentes al resto de los datos, por lo que el algoritmo los considera como un aglomerado más a cada uno.

[1]Un *píxel* es la menor unidad homogénea en color que forma parte de una imagen digital

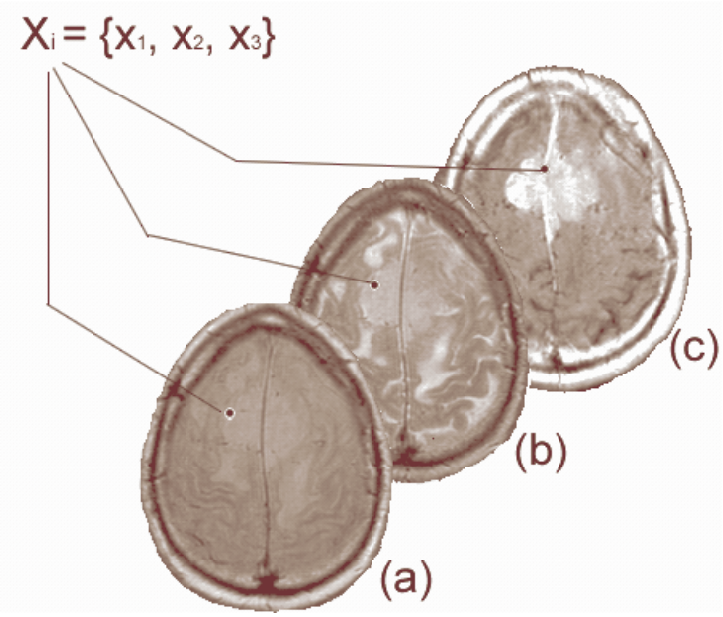


Figura 6.3: Los píxeles de cada imagen forman un vector que es una de las característica del conjunto

Extracción de puntos

El siguiente paso consiste en la extracción de puntos para construir un conjunto de datos D utilizando los valores de los *píxeles* de las imágenes. La manera de hacer ésto es la siguiente. Si se tienen, por ejemplo, tres imágenes y cada imagen contribuye con un píxel, entonces cada elemento de D será un vector de tres características, el valor de los tres píxeles. En la figura 6.3 se muestra cómo se forma un punto de D.

Como las imágenes son del mismo tamaño, se tiene que el píxel marcado con un punto de la figura (a), corresponde al mismo píxel de las figuras (b) y (c). Así que los tres píxeles juntos forman el i-ésimo punto del conjunto D, con $i = 1, 2, \ldots, n$.

Siendo p y q el tamaño de las imágenes y que corresponden al largo (*weight*) y ancho (*height*) respectivamente, el tamaño de n será de $p \times q$ elementos. En los problemas de aglomeración, éste ya es un número considerable de puntos y tiene un costo computacionalmente alto para cualquier programa.

Ahora, véase la ecuación (4.1), que es la función de la montaña

$$M(\boldsymbol{x}_i) = \sum_{j=1}^{n} e^{-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|}$$

Esta ecuación calcula la distancia de un elemento fijo contra todos los demás. Para una imagen de tamaño 512×512 (n = 262,144) resulta un número considerable de operaciones que se tienen por hacer.

Esta es una de las principales desventajas de este método, afortunadamente esta operación sólo se realiza una vez; ya que con ésta se obtiene el vector $\tilde{M}$, el cual se almacena en un archivo que es cargado después por el programa.

6.3 Resultados visuales

Como se mencionó, se tomaron como ejemplos imágenes de tejidos cerebrales en tres diferentes canales $\mathrm{T}_1, \mathrm{T}_2$ y DP. Todas estas imágenes presentan una lesión visible (tumor). El algoritmo utilizado fue el del nuevo método. Esta desición se basó en los resultados que se obtuvieron en el capítulo anterior; ya que aunque el desempeño de las dos variantes (hiper y ACM) fue bueno, se decidió utilizar el nuevo método por ser el que mejor resultados dió en los ejemplos *normal, normal-ruido, básico* y *srand*.

Se aplicó el algoritmo de aglomeración con 6 aglomerados buenos más el aglomerado de ruido, con la suposición de identificar las seis diferentes partes del cerebro y considerarlas como aglomerados buenos: la SG, SB, LCR, el cráneo, el tumor y fondo de la imagen.

Debido a que se está aplicando un algoritmo de aglomeración de ruido, específicamente el descrito en el capítulo cuatro, habrá un séptimo aglomerado, en el que se espera que se alojen los valores atípicos identificados por el algoritmo.

Del sitio web *The whole brain Atlas* fueron seleccionadas imágenes que corresponden a tres tipos de tumor y una metástasis. Cada uno de estos se presentan a continuación con sus respectivos resultados.

6.3.1 Sarcoma

La imagen 6.4 corresponde a un *sarcoma cerebral*. Un sarcoma es una *neoplasia maligna* (tumor) que se origina en un tejido conjuntivo, como pueden ser hueso, cartílago, grasa, músculo, vasos sanguíneos entre otros.

Observe la figura 6.5. Las tres imágenes superiores presentan la misma lesión cerebral en tres canales distintos. Ésta se encuentra en los dos hemisferios tanto en la parte baja como en la parte alta. En el recuadro central se encuentra el proceso principal de la aplicación; en éste se lleva a cabo la extracción de los datos, el proceso que realiza el nuevo método y el tratamiento de la imagen para presentarla visualmente.

La figura central es la salida final de la aplicación. Se aprecian seis aglomerados bien definidos e identificados con un color cada uno. Se puede observar que las distintas partes del cerebro han sido identificadas adecuadamente. A cada una de éstas les fue

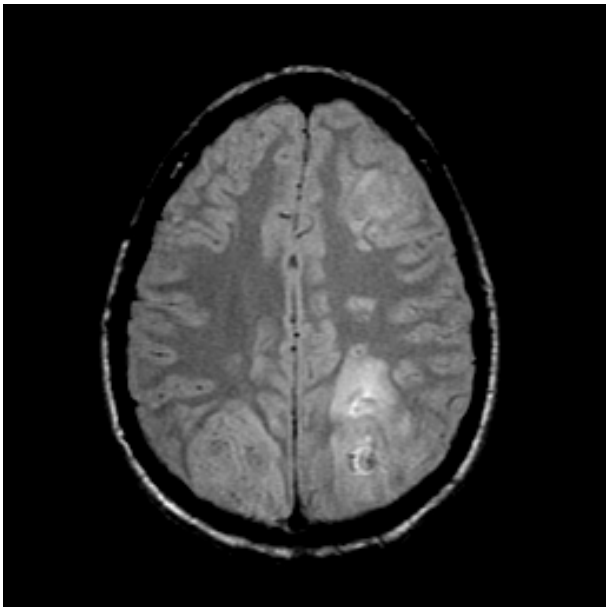


Figura 6.4: Imagen de un sarcoma cerebral

asignado un color distinto de acuerdo al aglomerado al que pertenecen. La SG y la SB presentan los colores amarillo, rojo y verde.

Al cráneo y el LCR también se identifican de una manera clara. Aunque a este último le fue asignado el mismo color del fondo de la imagen (el azul). Esto se debió quizá a la baja calidad de la imagen, ya que originalmente presentan el mismo tono obscuro y los tonos de éste es muy similar.

El tumor se muestra en color blanco y se diferencia plenamente de las otras partes. El algoritmo logra realizar una correcta aglomeración de las partes dominantes. Este ejemplo en particular no presenta muchos valores atípicos. Si acaso se alcanzan a apreciar algunos en la parte inferior izquierda del cráneo y otros esparcidos en toda la imagen.

Esto último es un resultado importante. El algoritmo asignó el color gris a los pocos valores atípicos encontrados. A lo largo de la tesis, se ha mencionado que el algoritmo forma una matriz de pertenencia U; en la cual se define la pertenencia de cada punto con los aglomerados.

La última fila corresponde al aglomerado de ruido. Se espera que los puntos que pertenezcan a este aglomerado (que se encuentran en la última fila de la matriz de pertenencia) se les represente con el color gris.

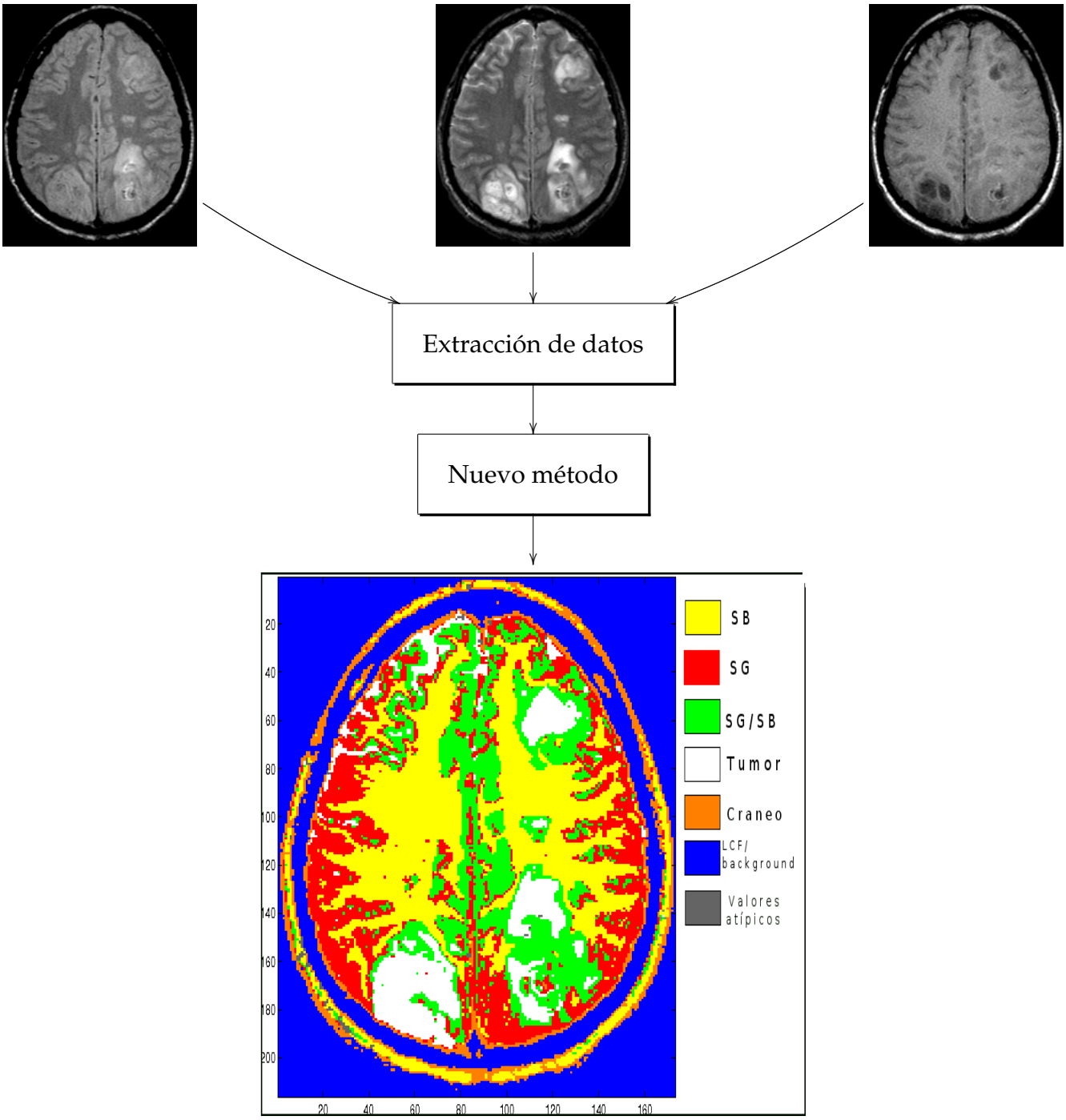


Figura 6.5: Sarcoma cerebral. Se buscaron seis aglomerados buenos y el de ruido. No hay muchos de estos últimos en la imagen.

6.3.2 Metastatic bronchogenic carcinoma

El siguiente resultado corresponde a una *metástasis* provocada por un *carcinoma bronquial* (Metastatic bronchogenic carcinoma). Una metástasis es la propagación de un foco canceroso a un órgano distinto de aquel en el que se inició el cáncer.

Los resultados se aprecian en la figura 6.7. Al igual que en el resultado anterior, se aprecian seis aglomerados distintos representados con un color. Observe que de nuevo el azul domina como color de fondo. Al igual que en el ejemplo anterior, el algoritmo también ha asignado el color azul al LCR y al *background*. Las demás partes del cerebro han sido identificados de manera adecuada.

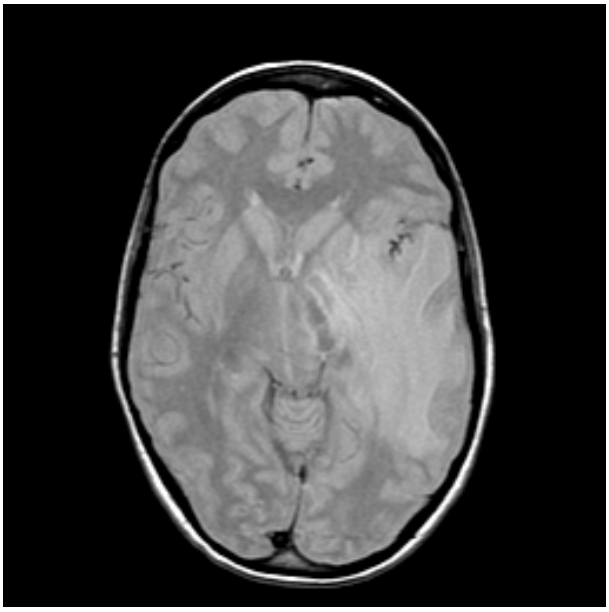


Figura 6.6: Imagen de una metástasis cerebral provocada por un tumor primario ubicado en los bronquios.

La SG y SB se han identificado con los colores amarillo, naranja y verde. Un dato curioso es que el algoritmo asigna de nuevo el color blanco para el tumor, que se aloja del lado derecho de la imagen. Quizá signifique que los valores de los píxeles del tumor de cada imagen presentan ciertas características que lo hacen diferente a los demás tejidos.

Similar al anterior, en este ejemplo no hay muchos valores atípicos. Los pocos que logran apreciarse se ven en color gris, esparcidos por toda la imagen. Para facilitar la apreciación de los mismos, se modificóí su color por un tono más obscuro.

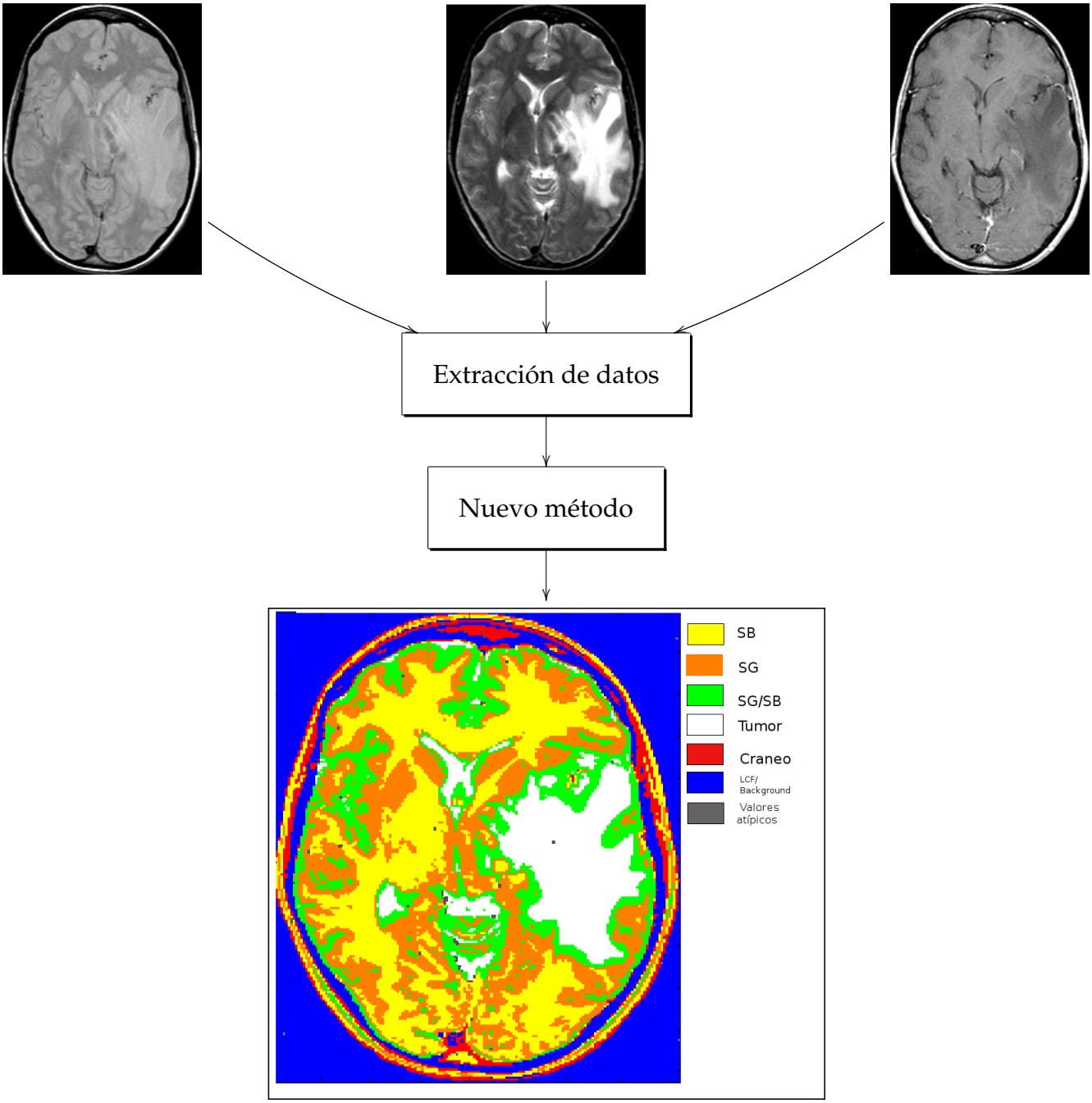


Figura 6.7: Metástasis cerebral. Se aprecian seis aglomerados buenos y algunos valores atípicos esparcidos por toda la imagen.

6.3.3 Meningioma

La figura 6.8 muestra otro tipo de tumor que afecta el cerebro: un *meningioma*. Se presenta en el tejido de las *meninges*, que son unas membranas de *tejido conectivo* que cubren todo el sistema nervioso central. Es el tumor del tipo primario más común que se puede diagnosticar en el cerebro.

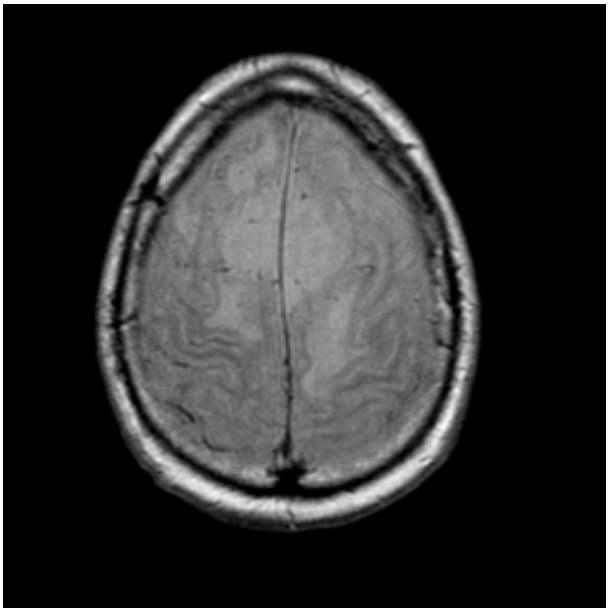


Figura 6.8: Se muestra un meningioma cerebral.

Los resultados de esta aglomeración son mostrados en la figura 6.9. La estructura de la figura completa es la misma que en las anteriores. Las tres imágenes superiores presentan el meningioma ubicado en el cerebro del paciente.

Ahora al fondo de la imagen es de color verde. El algoritmo ha realizado nuevamente una buena aglomeración sobre las diferentes zonas del cerebro. Aunque éste es un caso particular en el que una de las imágenes originales es muy diferente a las demás.

La imagen superior derecha no parece tratarse de un corte longitudinal del cerebro, sino más bien una proyección tridimensional de la misma capa cerebral. Ésta tiene una mayor iluminación sobre todo en la parte del cráneo y del LCR; en este último, el contraste del tono obscuro de las imágenes izquierdas con el color blanco de la tercera, provoca que se pierda la zona del LCR, provocando que no se defina correctamente en la gráfica.

En la imagen central se observan mucho puntos en color gris esparcidos por toda la imagen. Éstos son valores atípicos que han sido identificados adecuadamente, ya que no forman un aglomerado bien definido como los demás. En la parte del cráneo y del

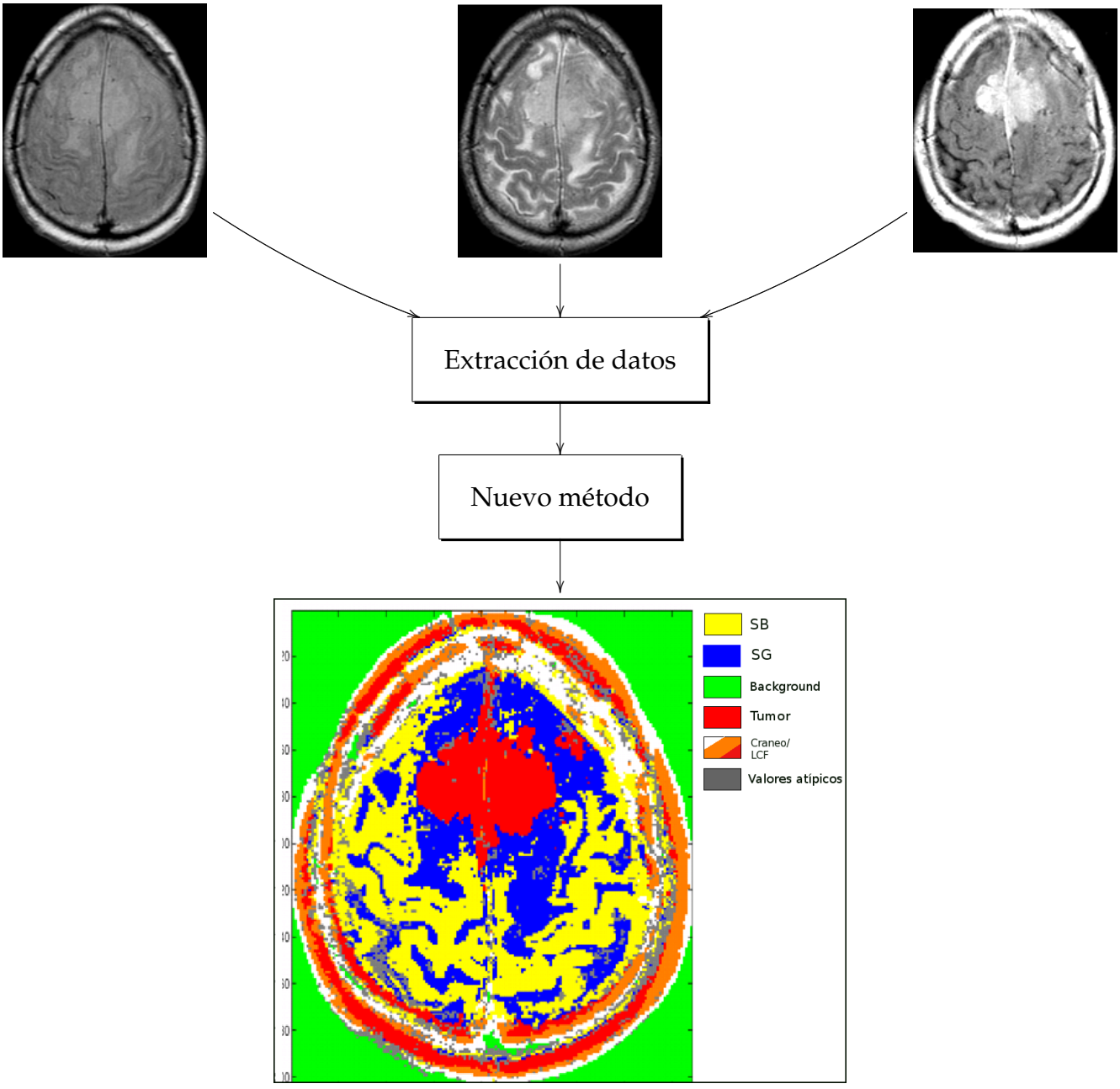


Figura 6.9: Salida de un meningioma cerebral. Uno de los tumores más comunes.

LCF, puede resultar confuso lograr identificarlos aún para el ojo humano. La imagen superior derecha es la que provoca el traslape de estos dos. Es por eso que el algoritmo no realiza un buen desempeño en estas zonas.

Sin embargo en la parte central, que es en donde las tres imágenes parecen ser del mismo segmento, la aglomeración es diferente. El tumor fue identificado con el color rojo y se observa claramente la SB y la SG con colores amarillo y azul respectivamente. Esta parte se haya libre de valores atípicos, lo que indica que el aglomerado lo ha hecho bien, ya que la mayoría de éstos aparecen en las zonas de traslape arriba mencionada.

6.4 Resumen

En este capítulo se presentó una aplicación a un problema real del algoritmo desarrollado en este trabajo de investigación. Se mostraron los detalles técnicos de la implementación. Después se da una breve introducción del problema de los *síndromes neoplásicos*, que son lesiones que aparecen en el cuerpo humano y que comúnmente se les conoce como *tumores*.

En la aplicación del algoritmo se manejan imágenes cerebrales de *resonancia magnética* tomadas de un sitio web. Se seleccionaron aquellas que presentan una lesión visible de un tumor canceroso. A estas imágenes se les extraen los valores de sus píxeles para formar vectores; estos vectores son los datos de entrada del algoritmo.

De los algoritmos desarrollados en este trabajo, se seleccionó el que mejor resultados presentó en las pruebas realizadas y mostradas en el capítulo anterior, es decir el nuevo método. Aunque hubo ejemplos en los que las variantes, hiper y ACM lo hicieron mejor, en general el desempeño del nuevo método fue el más adecuado para todos los conjuntos de prueba.

Para los resultados de la aplicación no se contó con la opinión de un experto en el tema. Así que los resultados se basan en la interpretación del resultado del algoritmo. La salida de la aplicación es visual. Se explican cada uno de los ejemplos de tumores analizados; éstos son *sarcoma, meningioma* y una *metástasis* provocada por un tumor alojado en los bronquios. Los resultados visuales muestran un buen desempeño del nuevo método.

Los resultados mostraron que el algoritmo es capaz de analizar imágenes construyendo conjuntos de datos a partir de éstas. Desafortunadamente no se contó con el suficiente material para realizar más pruebas. Las imágenes con las que se trabajó son de baja calidad, sin embargo fueron suficientes para probar el buen desempeño del nuevo método.

Capítulo 7

Conclusiones

> Ningún alumno debería salir de nuestras universidades sin saber lo poco que sabe.
>
> J. Robert Oppenheimer

En este trabajo de investigación se trabajó en el desarrollo de un algoritmo de aglomeración no sensible a los valores atípicos. Algunos trabajos como los de Dave [1] y Rehm [10], tratan de resolver este problema, sin embargo su propuesta de solución no es muy realista. En estos trabajos, los autores manejan algunos parámetros que deben ser calibrados para cada problema y esto los hace imprácticos.

El contenido de esta tesis es una aportación teórica a los problemas de análisis de aglomeraciones. La aportación más importante es proponer un planteamiento diferente y lógico del problema. Se basa principalmente en calcular un factor de densidad para cada punto, basado en la función de la montaña; esto da una idea individual de su estado.

Con este planteamiento se obtuvieron buenos resultados; se implementaron los algoritmos de los trabajos mencionados y se hicieron las mismas pruebas para los tres. Las tablas comparativas muestran un buen desempeño del algoritmo desarrollado.

Como es complicado conocer la estructura de los datos de un conjunto multidimensional, se crearon algunos conjuntos en dos dimensiones que mostraban diversos problemas; tales como tener dos o más aglomerados bien definidos, la presencia de valores atípicos, la diferencia en la densidad de los aglomerados entre otros.

Las pruebas se realizaron con estos y otros conjuntos tomados del repositorio en línea del *UCI Machine Learning Repository* [12]. Los resultados muestran en todos los casos un mejor desempeño para este algoritmo que para los otros trabajos, sobre todo en los ejemplos en los que existe la diferencia de densidad en las estructura del conjunto de datos.

Puede concluirse que el algoritmo desarrollado mejora el desempeño de los algoritmos que han intentado resolver el problema de los valores atípicos dentro del análisis de aglomeraciones. Los resultados del nuevo método fueron los más regulares en la mayoría de los ejemplos evaluados. Estos por encima del método de Dave, el de Rehm y el de las dos variantes diseñadas.

El enfoque práctico que se le dió al algoritmo fue el de analizar imágenes cerebrales de resonancia magnética. Estas imágenes presentan lesiones visibles de diferentes tipos de tumores que afectan los tejidos cerebrales. El algoritmo las analiza y presenta un resultado visual. En esta imagen final se puede apreciar que aquellos puntos que son atípicos a todos los demás, son los que se asignan al aglomerado de ruido.

De acuerdo a la teoría conocida, el planteamiento propuesto en este trabajo muestra un buen desempeño del algoritmo respecto a los conjuntos de prueba. Como se ha mencionado, consigue lograr una buena aglomeración en todos los conjuntos aún en aquellos en los que la densidad de los aglomerados es diferente. Con esto cumple uno de los objetivos principales de esta tesis.

La implementación mostró de igual manera buenos resultados. El análisis de las imágenes de resonancia magnética muestra que el algoritmo realiza una buena aglomeración de las diferentes partes del cerebro. Los puntos de las imágenes que no se asocian a ninguno de los colores pueden tomarse como valores atípicos. Así que estos puntos no forman parte de ninguno de los aglomerados *buenos* y se encuentras esparcidos por toda la imagen sin definir un nuevo aglomerado.

De los principales inconvenientes que tiene el proceso de análisis de imágenes, es desde luego el elevado tiempo de cómputo que se requiere. Para tener un análisis más preciso, se requiere que las imágenes sean de alta calidad, lo que aumenta el tamaño de los archivos a ser analizados. Se requiere utilizar técnicas más eficientes de extracción de datos. Sin embargo, al tratarse de tomografías cerebrales, no es posible omitir detalles de la imagen que puedan ocasionar un mal diagnóstico por parte del médico.

Las imágenes que se utilizaron en este trabajo son de baja calidad, pues fueron tomadas de la red. La resolución que tienen no es buena y el proceso de análisis se ve afectado por esta razón.

7.1 Trabajo futuro

La correcta identificación de un tumor maligno es una muestra del buen desempeño del algoritmo desarrollado. Sin embargo no es posible determinar con exactitud cuál será el aglomerado destinado a alojar aquellos puntos de la imagen que sean identificados como un tumor. Esto es debido a la naturaleza del algoritmo base, el CMD; la

aleatoriedad de los aglomerados iniciales es un factor importante en el resultado final de la aglomeración.

Un trabajo futuro sería diseñar un clasificador supervisado en el que se tenga información más técnica acerca de las partes que conforman una imagen cerebral. Esto es que se tenga un repositorio completo de imágenes con lesiones cerebrales que ya hayan sido diagnosticadas, de tal manera que se realice un proceso de *aprendizaje* por parte del clasificador. Éste deberá identificar la presencia de un tumor en una imagen nueva.

Un segundo trabajo sería realizar un *clasificador difuso* utilizando técnicas de aglomeración. Un clasificador es un algoritmo que asigna una etiqueta a un objeto, basándose en su descripción. Este clasificador estará basado en reglas del tipo

$$\text{Si} \rightarrow \text{Antecedente}, \text{entonces} \rightarrow \text{Consecuente}$$

Donde el antecedente es una regla que utiliza *valores lingüisticos* y el consecuente puede ser la asignación de la etiqueta del objeto.

Debido a la aglomeración difusa que realiza el algoritmo, en el que cada elemento tiene un grado de pertenencia hacia todos los aglomerados, es posible definir reglas del tipo Antecedente $\rightarrow$ Consecuente. Como el grado de pertenencia define a cuál aglomerado pertenece cada punto, la definición de las reglas será cuestión de tomar los valores máximos de pertenencia de cada elemento.

Apéndice A

Conceptos

En este apéndice se incluyen algunos conceptos básicos así como de los detalles de implementación y de usuario requeridos para el funcionamiento del programa.

A.1 Medidas de distancia

Se habla de distancia entre objetos como una propiedad relacionada con la cercanía o proximidad entre los mismos.

Así pues, se dice que una casa está a una distancia de dos kilómetros del parque o que la mesa está a tres metros de la puerta. Se pueden comparar las distancias entre varios objetos y un punto de referencia fijo. Algunos estarán más cerca que otros y siempre es posible medir la distancia entre ellos, pues esto es algo que se puede cuantificar por medio de una medida o métrica. Si se consideran puntos en un plano euclidiana, la distancia entre éstos se define como el segmento de recta más corto que los une.

Formalmente, la distancia entre puntos se define de la siguiente manera. Sea D un conjunto de puntos; una métrica de distancia M es una función d definida como

$$M : D \times D \rightarrow \mathbb{R} \tag{A.1}$$

que asocia a cada par de elementos $(a, b) \in D$ un número real $\in \mathbb{R}$ que se define como *la distancia entre a y b*. En la figura 1.3, se han graficado los puntos a, b, c, cuyas coordenadas son (x_1, y_1), (x_2, y_2) y (x_3, y_3) respectivamente.

La *distancia euclidiana* entre los puntos a y b, definida como $d(a, b)$ se calcula con:

$$d(a, b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{A.2}$$

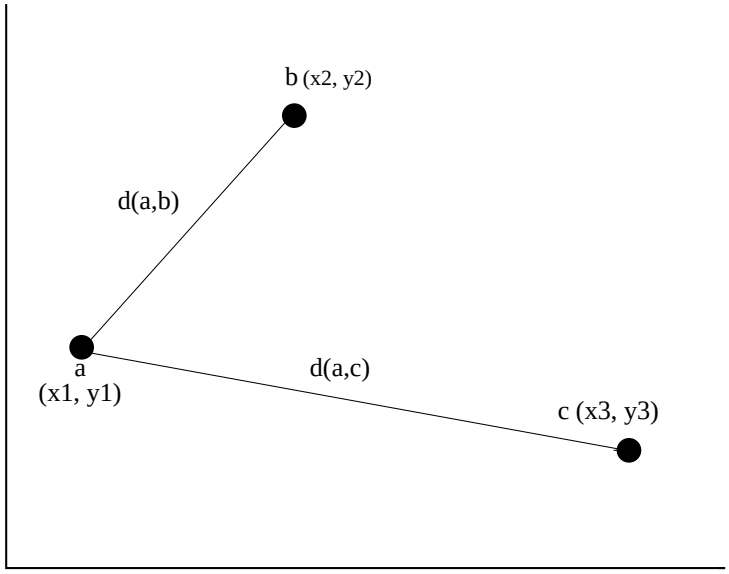


Figura A.1: Se ilustra la medida de las distancias. Claramente se observa que $d(a, b) < d(a, c)$

De la figura puede observarse claramente que la distancia de a a b es menor que la de a a c; por lo que el punto a está más cerca al punto b que al punto c, o en términos de semejanza, a es más "*similar*" a b que a c, lo que puede expresarse como $a \sim b : c$; entonces $a \sim b : c$ si $d(a, b) < d(a, c)$

A.2 Grafos

Formalmente, un grafo se define como $G = (V, E)$, donde V es el conjunto de vértices y E el conjunto de aristas. Se puede considerar a E como una *relación* en V, i.e., $E \subset V \times V$; si la relación es *simétrica*, es decir si $(a, b) \in E$, entonces $(b, a) \in E$. Cada arista tiene dos puntos (*endpoints*) en V y se dice que *conecta* o *une* dos vértices. El grado de un grafo es el número de vértices. Un *ciclo* es una arista cuyos puntos finales conectan al mismo vértice.

Un grafo que no contiene *ciclos* y que conecta a todos los nodos se le conoce como *árbol*. Un *subgrafo* de un grafo *no dirigido* en forma de árbol recibe el nombre de *árbol de cobertura* (*spanning tree*) o ST. Un *grafo con peso* tiene asociada una etiqueta (el peso) a cada arista en el grafo. Normalmente los pesos son números reales. El ST de un grafo G con peso es llamado el *árbol de cobertura mínima* (*Minimum Spanning Tree*) si minimiza una cierta *función de pesos* de las aristas de G [20].

Apéndice B

Manuales

B.1 Instalación

La versión 3.0 de Octave es libre y se encuentra disponible en la red. El usuario puede consultar el repositorio de aplicaciones que corresponda a la versión del sistema operativo que esté manejando. Octave se ejecuta como un intérprete de comandos en el terminal del sistema, pudiendo acceder desde ahí a las diferentes funciones y procedimientos predefinidos o a los módulos que pueden importarse para ampliar el lenguaje.

- El usuario tiene que estar en su terminal como *root* y ejecutar el siguiente comando:
  root@∼$ apt-get install octave3.0

- Para ejecutar el programa, deberá ir a `Applications->Education->GNU Octave`; o bien ejecutar en consola lo siguiente
  root@∼$ octave

B.2 Conjuntos de datos

Los conjuntos de datos que son las entradas del algoritmo son archivos de texto con extensión `%.dat`. La estructura de cada archivo es basada en columnas; donde cada columna define una de las características de los datos. La fila completa se lee como el vector de características que define un elemento del conjunto. El usuario cuenta con los siguientes conjuntos de datos.

Datos	# Características	# Clases	# Instancias
iris	4	3	150
breast-cancer	9	2	683
wine	9	13	178
bupa	6	2	345
pima	8	2	768
ionosphere	34	2	351
image-segmentation	19	7	2310
basic	2	2	13
normal	2	2	200
noisenormal	2	2	230
srand	2	2	250

Tabla B.1: Descripción de los conjuntos de datos utilizados en las pruebas.

B.3 Manual de usuario

Para la ejecución del programa se hace lo siguiente. El usuario debe escribir en un archivo las siguientes líneas:

#!/bin/bash
/usr/bin/octave -q - -eval genfcm.m

donde `/usr/bin/octave` es la ruta en donde se encuentra (normalmente) octave; *genfcm.m* es el programa principal. El usuario debe guardar el archivo con cualquier nombre de su agrado con la extensión `sh`. Este archivo debe guardarse en el mismo *directorio* en donde se encuentra el programa principal. Si desea ejecutarlo desde otro directorio, deberá incluir la ruta del programa *genfcm.m* en el archivo `*.sh`

Ahora en una terminal se escribe lo siguiente:

root@∼$ ./filename.sh

donde `filename.sh` es el nombre con el que se guardo el archivo. Si no se ejecutase el programa, tal vez sea necesario extender los permisos de ejecución al archivo `sh` creado. Esto se logra escribiendo en la terminal lo siguiente:

root@∼$ chmod 755 filename.sh

Una vez hecho esto, el programa hace lo siguiente:

- Muestra al usuario tres opciones: `Dave, Hyper, Mountain`. El programa da la opción de escoger cuál algoritmo ejecutar

- Una vez seleccionado el algoritmo, se le pide al usuario que introduzca el nombre del dato de entrada (los conjuntos mostrados en la tabla B.1)
- Ahora el programa pide el número c de aglomerados deseados. Normalmente este valor es el número de clases. Sin embargo, por la naturaleza de estos algoritmos, se debe dar el valor de c incrementado en 1 para el aglomerado de ruido. Por ejemplo, si el conjunto de prueba es el conjunto *iris*, que está compuesto de tres clases, el valor de c=4
- El usuario debe esperar la salida del programa. El tiempo que tome depende del conjunto que se está analizando

Referencias

[1] Rajesh N. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12(11):657–664, 1991.

[2] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall PTR, Upper Saddle River, New Jersey 07458, 1995.

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

[4] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data (Prentice Hall Advanced Reference Series : Computer Science)*. Prentice Hall College Div, March 1988.

[5] B. S. Everitt. *Cluster Analysis*. Edward Arnold and Halsted Press, 1993.

[6] Graham Williams, Rohan Baxter, Hongxing He, Simon Hawkins, and Gu & Lifang. A comparative study of rnn for outlier detection in data mining. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, page 709, Washington, DC, USA, 2002. IEEE Computer Society.

[7] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[8] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics and Systems*, 3(3):32–57, 1973.

[9] James. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algoritms*. Plenum Press, 1981.

[10] Frank Rehm, Frank Klawonn, and Rudolf Kruse. A novel approach to noise clustering for outlier detection. *Soft Comput.*, 11(5):489–494, 2007.

[11] Z. Hou, W. Qian, S. Huang, Q. Hu, and W. L. Nowinski. Regularized fuzzy c-means method for brain tissue clustering. *Pattern Recogn. Lett.*, 28(13):1788–1794, 2007.

[12] A. Asuncion and D.J. Newman. Uci. machine learning repository. `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`, 2007. [Online; accesed on 14-January-2009].

[13] R. R. Yager and D. P. Filev. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1279–1284, August 1994.

[14] R. L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *IEEE Ann. Hist. Comput.*, 7(1):43–57, 1985.

[15] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86, 1971.

[16] Anne-Sophie Capelle, Olivier Alata, C. Fernandez, Sébastien Lefèvre, and J. C. Ferrie. Unsupervised segmentation for automatic detection of brain tumors in mri. In *ICIP*, 2000.

[17] Bela Bodey Stuart E. Siegel & Hans E. Kaiser. *Molecular Markers of Brain Tumor Cells, Implications for Diagnosis, Prognosis and Anti-Neoplastic Biological Therapy*, chapter I Brain Tumors, pages 3–12. Springer Netherlands, 2005.

[18] Francis Ali-Osman. *Brain Tumors*. Humana Press, Totowa, New Yersey, 2005.

[19] Keith A. Hohnson & J. Alex Becker. The whole brain atlas. `http://www.med.harvard.edu/AANLIB/home.html`, 2009. [Online; accessed 01-August-2009].

[20] Victor N. Kasyanov and Vladimir A. Evstigneev. *Graph theory for programmers: algorithms for processing trees*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.