Revista Colombiana de Estadística
Diciembre 2010, volumen 33, no. 2, pp. 321 a 339

Una revisión de los algoritmos de partición más comunes en el análisis de conglomerados: un estudio comparativo

A Review of the Most Common Partition Algorithms in Cluster Analysis: A Comparative Study

Susana A. Leiva-Valdebenito[a], Francisco J. Torres-Avilés[b]

Departamento de Matemática y Ciencia de la Computación, Facultad de Ciencia, Universidad de Santiago de Chile, Santiago, Chile

Resumen

Este estudio está enfocado en comparar diversos métodos de partición del análisis de conglomerados, usualmente conocidos como métodos no jerárquicos. En este trabajo, se realizan estudios de simulación para comparar los resultados obtenidos al implementar los algoritmos k-medias, k-medianas, PAM y Clara cuando los datos son multivariados y de tipo continuo. Adicionalmente, se efectúa un estudio de simulación con el fin de comparar algoritmos de partición para datos cualitativos, confrontando la eficiencia de los algoritmos PAM y k-modas. La eficiencia de los algoritmos se compara usando el índice de Rand ajustado y la tasa de correcta clasificación. Finalmente, se aplican los algoritmos a bases de datos reales, las cuales poseen clases predefinidas.

Palabras clave: algoritmos de conglomerados, medida de similaridad, simulación.

Abstract

This study is oriented to compare several partition methods in the context of cluster analysis, which are also called non hierarchical methods. In this work, a simulation study is performed to compare the results obtained from the implementation of the algorithms k-means, k-medians, PAM and CLARA when continuous multivariate information is available. Additionally, a study of simulation is presented to compare partition algorithms qualitative information, comparing the efficiency of the PAM and k-modes algorithms. The efficiency of the algorithms is compared using the Adjusted Rand Index and the correct classification rate. Finally, the algorithms are applied to real databases with predefined classes.

Key words: Clustering algorithm, Similarity measure, Simulation.

[a]Estudiante de Ingeniería Estadística. E-mail: susanaleivav@gmail.com
[b]Profesor asistente. E-mail: francisco.torres@usach.cl

1. Introducción

El análisis de conglomerados es una técnica de análisis exploratorio, definido dentro de los métodos multivariantes de clasificación, que permite separar en diferentes clases o grupos a un conjunto de objetos o individuos, de modo que todos los que pertenecen a una misma clase son homogéneos entre sí y diferentes de aquellos objetos que pertenecen a una clase distinta. Este método de agrupación es muy utilizado hoy en día en diferentes áreas, tales como, estudios de segmentación de clientes en el área financiera (Abonyi & Feil 2007), biología (Quinn & Keough 2002, Der & Everitt 2006), ecología (McGarigal, Cushman & Stafford 2000), entre otros, puesto que la mayoría de las veces no utiliza ningún supuesto estadístico para llevar a cabo el proceso de agrupación. Los algoritmos de agrupamiento más conocidos son los métodos jerárquicos y los métodos de partición o no jerárquicos, aunque existen otros métodos basados en densidades o modelos probabilísticos (Kamber & Han 2006).

En los métodos de partición, el conjunto de datos es inicialmente distribuido en un número pre-especificado de conglomerados k, el que puede ser aleatorio, y luego iterativamente se asignan las observaciones a los conglomerados hasta que se satisface algún criterio de parada. Entre estos métodos, el más utilizado es el algoritmo k-medias (MacQueen 1967, Anderberg 1973), sin embargo, existen otros algoritmos denominados k-medianas, PAM y Clara (Kaufman & Rousseeuw 1990), los cuales no han sido ampliamente difundidos, razón por la cual no se utilizan con frecuencia.

El presente trabajo tiene como propósito entregar un panorama acerca de los métodos de partición más comunes, abordando su estructura algorítmica e implementación. La contribución está relacionada con comparar la efectividad de los algoritmos para determinar y separar los grupos, usando distintos esquemas de simulación que incluyen datos generados a partir de distribuciones normales y Skew-Normales para el caso cuantitativo y distribuciones multinomiales para el caso cualitativo (ver Leiva 2008).

En una primera etapa, el presente trabajo entrega resultados relacionados con la comparación de los métodos de clasificación k-medias, k-medianas, PAM y Clara cuando se dispone de datos numéricos continuos, en diversos escenarios de agrupación. Luego, se comparan los algoritmos k-modas y PAM cuando se dispone de datos cualitativos. A partir de estudios de simulación se extraen las características y diferencias más relevantes, para posteriormente aplicarlos a bases de datos reales expuestos en la literatura. La implementación y aplicación se realizan usando el software libre R (R Development Core Team 2010).

El trabajo estará organizado de la siguiente manera. En la sección 2 se definirán los algoritmos de partición que intervendrán en el presente estudio. La sección 3 presentará los esquemas de simulación y resultados bajo los cuales se implementaron considerando la presencia de datos continuos y categóricos separadamente. La sección 4 mostrará los resultados de aplicar los algoritmos a bases de datos reales, expuestos en la literatura, para finalmente, en la sección 5, presentar las conclusiones más relevantes.

2. Algoritmos de partición

Se definen como algoritmos que permiten construir k particiones de las observaciones, donde cada partición representará a un conglomerado, segmento o grupo, y son útiles cuando existe ignorancia del investigador frente a la clasificación de observaciones, y éste dispone simultáneamente de un significativo número de variables o características. El algoritmo inicia considerando una división inicial, luego, busca encontrar el mejor agrupamiento reubicando los objetos de un grupo a otro, hasta que se optimice una función objetivo específica, la cual actúa como criterio de parada (Kamber & Han 2006). Intuitivamente, una clasificación adecuada debería considerar que la dispersión dentro de los grupos sea la menor posible.

Estos algoritmos son en general de implementación rápida, pero sufren inconvenientes en la especificación de semillas o particiones iniciales (Hartigan 1975). Este problema puede ser solucionado aplicando algún método jerárquico previo, tal como se hará más adelante. En las subsecciones siguientes se presenta la mecánica y las principales características de los algoritmos involucrados en el presente estudio.

2.1. k-medias

El algoritmo k-medias es uno de los métodos de partición más difundidos y populares. La génesis del algoritmo menciona a MacQueen (1967) como su precursor, a partir del cual se presentan diversas variaciones tales como aquellas propuestas por Anderberg (1973) y Hartigan (1975).

El algoritmo funciona como sigue. Dado un número inicial de conglomerados k, el objetivo del algoritmo es minimizar la distancia euclídea de los elementos dentro de cada conglomerado, respecto a su centro. La similitud de los objetos dentro de cada conglomerado es medida respecto a su vector de promedios, llamado generalmente centroide. El criterio de parada usado es el error cuadrático medio definido por

$$E = \sum_{l=1}^{k} \sum_{\mathbf{X} \in c_l} (\mathbf{X} - M_l)'(\mathbf{X} - M_l) \tag{1}$$

donde $\mathbf{X}$ es el punto en el espacio que representa a los objetos dados y M_l es el vector de promedios del conglomerado c_l, ambos vectores de $\mathbb{R}^p$. Existen varias formas de implementar el algoritmo, pero básicamente sigue los pasos expuestos en el algoritmo 2.1.1.

2.1.1. Algoritmo

1. Seleccionar arbitrariamente los k objetos que serán los centros o centroides iniciales de los conglomerados.

2. Se asigna cada objeto al conglomerado con el centroide más cercano, con base en el valor medio de los objetos en el conglomerado.

3. Se recalculan los centros de los conglomerados es decir se actualiza la media.

4. Se iteran los pasos 2 y 3 hasta que se alcance la convergencia del criterio de parada, o hasta que los centroides se modifiquen levemente.

Características relevantes de este algoritmo es que a menudo termina en un óptimo local, es escalable y eficiente en procesos que involucran grandes conjuntos de datos, y la complejidad computacional del algoritmo es del orden de nkt, donde t es el número de iteraciones y n el número de objetos o individuos (Han, Kamber & Tung 2001). Es de cálculo rápido y trabaja bien con valores faltantes o "missing"; sin embargo, es sensible a valores extremos, ya que distorsiona la media y el criterio de parada.

Entre sus mayores debilidades podemos señalar su sensibilidad a la selección de los centroides iniciales. Más aún, si las elecciones de diferentes centroides producen finales diferentes o la convergencia es muy lenta, quizás no existan agrupaciones naturales en los datos. La mayoría de las variaciones que existen del algoritmo difieren en la elección de los centroides iniciales, no obstante, es usual que esta selección se realice de forma aleatoria dentro del conjunto total de datos. Otra opción está relacionada con que el propio investigador especifique los centroides.

Una alternativa más objetiva para seleccionar el número de centroides iniciales se basa en determinarlos a partir de la aplicación previa de algún método de conglomerados jerárquico aglomerativo (Peña 2002), induciendo al algoritmo 2.1.2.

2.1.2. Algoritmo

1. Aplicar un método de conglomerado jerárquico y guardar los k centros que resulten del análisis.

2. Se asigna cada objeto al conglomerado con el centroide más cercano, con base en el valor medio de los objetos en el conglomerado.

3. Se recalculan los centros de los conglomerados, es decir, se actualiza la media.

4. Se iteran los pasos 2 y 3 hasta que se alcance la convergencia del criterio de parada, o hasta que los centroides se modifiquen levemente.

El algoritmo 2.1.2 será denominado más adelante k-medias jerárquico.

2.2. k-medianas

Descriptivamente, la mediana es una medida más robusta que la media, puesto que no se ve influida por valores extremos; el algoritmo k-medianas funciona de forma similar al algoritmo k-medias, sustituyendo el vector de promedios por el correspondiente vector de medianas como centro del conglomerado. En este caso se utiliza la distancia de Manhattan en vez de la distancia euclídea al cuadrado como medida de disimilitud (Anderson, Gross, Musicant, Ritz, Smith & Steinberg 2006).

Los vectores de medianas de los objetos en cada conglomerado serán denotados por $Me = \{Me_1, \ldots, Me_k\}$.

Usando la medida de distancia de Manhattan, la función por minimizar está dada por la siguiente expresión:

$$P(W, Me) = \sum_{l=1}^{k} \sum_{\mathbf{X} \in c_l} W_l' |\mathbf{X} - Me_l| \tag{2}$$

donde Me_l es el vector de medianas del l-ésimo conglomerado y $W = [W_1, \ldots, W_k]$ es una matriz de pesos con dimensión $n \times k$, cuyos vectores columnas son $W_l = (w_{1l}, \ldots, w_{kl})'$, para $l = 1 \ldots, k$, con $\sum_{l=1}^{k} w_{il} = 1$, donde $w_{il} \in (0, 1)$, para todo $i = 1, \ldots, n$, $j = 1, \ldots, k$.

2.2.1. Algoritmo

1. Seleccionar arbitrariamente los k objetos que serán los centros o centroides iniciales de los conglomerados. El tipo de selección inicial de centroides es análogo a los presentados en el algoritmo k-medias.

2. Asignar cada punto al conglomerado con el centroide más cercano, a través de la distancia de Manhattan.

3. Calcular el nuevo conjunto de centroides de los conglomerados, calculando la mediana de los nuevos grupos formados.

4. Se iteran los pasos 2 y 3 hasta que se minimice la función objetivo (2), o hasta que los centroides apenas se modifiquen.

El algoritmo k-medianas tiene la misma complejidad computacional que el algoritmo k-medias. Al ser bastante similar al algoritmo k-medias, es también sensible a la selección de los centroides iniciales; la ventaja que presenta es que la mediana no está influida por los valores extremos, por lo que se logra un método, en teoría, más robusto. Nótese que este algoritmo no está implementado en los softwares tradicionales.

2.3. k-medoids

Estos métodos fueron introducidos por Kaufman & Rousseeuw (1987). Están basados en el uso de objetos actuales del conjunto de datos para ser los representantes de los conglomerados, denominados medoids. Estos medoids se definen como los puntos localizados lo más al centro posible de cada conglomerado y son representativos de la estructura de los datos. Cada objeto restante es agrupado con el medoid más cercano, e iterativamente estos algoritmos realizan todos los intercambios posibles entre los objetos representativos y los que no lo son, hasta que se minimice una medida de disimilitud entre los k-medoids y los vectores de observaciones que forman los conglomerados (Kamber & Han 2006).

En este grupo encontramos los algoritmos "Partition Around Medoids" (PAM) y "Clustering Large Applications" (Clara). Tanto el algoritmo PAM como el Clara se encuentran implementados en el software R-gui y permiten ingresar una matriz de disimilitudes definida por el analista. Esto presenta una ventaja, pues es posible incorporarlo al proceso de partición cuando se dispone de datos cualitativos.

2.3.1. Algoritmo PAM

El algoritmo "Partitioning Around Medoids" (PAM) es un método tipo k-medoid que intenta determinar k particiones de n objetos determinando los objetos representativos de cada conglomerado (Ng & Han 1994). Para encontrar los k medoids, PAM empieza con una selección arbitraria de k objetos representativos. En cada iteración hace un intercambio entre un objeto seleccionado, O_i, y uno no seleccionado, O_h, si y solo si el intercambio mejora la calidad del agrupamiento.

El efecto de tal intercambio entre O_i y O_h se mide a través de una función de costo, es decir, el algoritmo calcula los costos C_{jih} para todos los objetos no seleccionados O_j. Según el caso en el cual O_j se encuentre, C_{jih} puede ser definido por una de las siguientes expresiones:

Caso 1: Suponga que actualmente O_j pertenece al conglomerado representado por O_i. Además, O_j es más similar a O_{j2} que a O_h, donde O_{j2} es el segundo medoid más similar a O_j. Entonces, si O_i es remplazado por O_h como un medoid, O_j pertenecería al conglomerado representado por O_{j2}. De esta forma el costo del intercambio es dado por:

$$C_{jih} = d(O_j, O_{j2}) - d(O_j, O_i)$$

Esta ecuación siempre da un valor no negativo.

Caso 2: Suponga que O_j pertenece actualmente al conglomerado representado por O_i. En este caso O_j es más similar a O_h que a O_{j2}. Entonces, si O_i es remplazado por O_h, O_j pertenecería al conglomerado representado por O_h. El costo es dado por:

$$C_{jih} = d(O_j, O_h) - d(O_j, O_i)$$

A diferencia del caso anterior, C_{jih} puede ser positivo o negativo.

Caso 3: Suponga que O_j pertenece actualmente al conglomerado representado por O_{j2} y O_j es más similar a O_{j2} que a O_h. Entonces, aun si O_i es remplazado por O_h, O_j permanecería en el conglomerado representado por O_{j2}. De esta manera el costo sería:

$$C_{jih} = 0$$

Caso 4: Suponga que O_j pertenece actualmente al conglomerado representado por O_{j2}, pero O_j es menos similar a O_{j2} que a O_h. Entonces, remplazando O_i por O_h causaría que O_j fuese representado por O_h desde el conglomerado O_{j2}. De esta manera el costo es dado por:

$$C_{jih} = d(O_j, O_h) - d(O_j, O_{j2})$$

Este costo es siempre negativo.

Combinando los cuatro casos, el costo total de remplazar O_i por O_h está dado por $TC_{ih} = \sum_j C_{jih}$. La función $d(\cdot)$ es una la medida de distancia o disimilitud que se utiliza entre los objetos. Por defecto se usa la distancia euclídea. La estructura del algoritmo PAM se puede visualizar en el algoritmo 2.3.2.

2.3.2. Algoritmo

1. Seleccionar arbitrariamente k objetos representativos, los cuales serán los k-medoids iniciales.

2. Calcular TC_{ih} para todos los pares de objetos O_i, O_h donde O_i es actualmente un medoid, y O_h no lo es.

3. Seleccionar el par O_i, O_h el cual corresponda al $\text{mín}_{O_iO_h}\{TC_{ih}\}$. Si el mínimo TC_{ih} es negativo, se intercambia O_i con O_h; regresar al paso 2.

4. Repetir el paso 2 y 3 hasta que no haya cambio.

5. Asignar cada objeto a su medoid más cercano.

La principal ventaja del algoritmo PAM es la robustez del método en presencia de ruido u "outliers", pues el cálculo del medoid está menos influido por ellos u otros valores extremos. PAM comienza a ser muy costoso a medida que el tamaño muestral n y el número de iteraciones k aumentan, siendo una de las principales desventajas de este algoritmo, razón por la cual es eficiente sólo para bases de datos pequeñas (Han et al. 2001).

2.3.3. Algoritmo Clara

El algoritmo Clara, separa múltiples muestras de la base completa y aplica el algoritmo PAM sobre cada una de ellas; luego, encuentra los conjuntos de k-medoids de las muestras. El principal motivo de Kaufman & Rousseeuw (1990) para proponer este algoritmo fue debido a la deficiencia del algoritmo PAM para trabajar con bases de datos con grandes volúmenes de información. Si estas muestras son realmente representativas de toda la base de datos, los medoids de las muestras deberían acercarse a aquellos que se hubiesen escogidos de la base de datos completa. Según estos autores, los resultados experimentales indican que 5 muestras con $(40 + 2k)$ objetos cada una, producen resultados satisfactorios. La calidad del agrupamiento es medida con la disimilitud media de todos los datos, y no sólo aquellos objetos considerados en las muestras (Ng & Han 1994).

2.3.4. Algoritmo

1. Para $i = 1$ a 5 repetir los siguientes pasos.

2. Seleccionar una muestra aleatoria de $s = (40 + 2k)$ objetos de la base completa.

3. Ejecutar el algoritmo PAM sobre la muestra, para encontrar los k medoids de esta muestra.

4. Para cada objeto O_j de la base completa, determinar su medoid más cercano y agruparlos.

5. Calcular la disimilaridad media del agrupamiento obtenido. Si este valor es menor al mínimo actual, usar este valor como el mínimo actual y conservar los k medoids obtenidos en el paso 3 como el mejor conjunto de medoids obtenidos.

6. Retornar al paso 1 y comenzar con la próxima iteración.

La efectividad de Clara depende tanto del tamaño de la muestra como de su calidad. Note que PAM busca los mejores k medoids entre un conjunto total de datos, mientras que Clara busca los mejores k medoids entre las muestras seleccionadas del conjunto total de datos. Clara no podría encontrar la mejor agrupación si los mejores k medoids no son seleccionados dentro de las muestras.

El algoritmo Clara presenta una complejidad mayor en cada iteración, puesto que depende adicionalmente del tamaño de la muestra seleccionada (Han et al. 2001).

2.4. k-modas

La mayoría de las aplicaciones realizadas bajo los algoritmos de partición se ha centrado en datos numéricos, cuyas propiedades pueden ser explotadas para definir naturalmente las funciones de distancia entre los objetos. Sin embargo, las aplicaciones en segmentación contienen conjuntos de datos que incluyen datos categóricos cuyas funciones de distancia o disimilitud no están definidas naturalmente.

Una debilidad importante del algoritmo k-medias es su incapacidad de trabajar con datos que no sean numéricos. Huang (1998) presentó un algoritmo para trabajar con un entorno categórico. La idea de Huang fue extender el algoritmo k-medias al ámbito categórico denominándolo k-modas, teniendo en cuenta algunas modificaciones, tales como: usar una medida de disimilitud de correspondencia simple para datos categóricos, remplazar las medias de los grupos por sus respectivas modas y utilizar un método basado en frecuencias para actualizarlas.

El algoritmo k-modas utiliza el mismo proceso que el k-medias, lo que preserva su eficiencia y es altamente deseable en los análisis de agrupación de datos.

Es así como $X = (X_1, \ldots, X_n)$ representa un conjunto de n objetos descritos por un conjunto de m atributos y frecuencias denotadas por A_j y p_j, respectivamente. Entonces, los objetos X_i son representados por un vector del tipo $[x_{i1}, x_{i2}, \ldots, x_{im}]$. Además $X_i = X_r$, si y solo si $x_{ij} = x_{rj}$ para $1 \leq j \leq m$, que significa que los dos objetos tienen iguales categorías para los distintos atributos.

Como medida de disimilitud entre dos objetos categóricos, se usará el total de discordancias de los correspondientes valores de los atributos entre dos objetos (Kaufman & Rousseeuw 1990), definido por:

$$d(X_i, X_j) = \sum_{k=1}^{m} \delta(x_{ik}, x_{jk}) \tag{3}$$

donde

$$\delta(x_{ik}, x_{jk}) = \begin{cases} 0, & (x_{ik} = x_{jk}) \\ 1, & (x_{ik} \neq x_{jk}) \end{cases}$$

Al usar la medida de disimilitud para objetos categóricos (3), la función objetivo por minimizar es:

$$P(W, Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{il} \delta(x_{ij}, q_{lj}) \tag{4}$$

sujeto a

$$\sum_{l=1}^{k} w_{il} = 1 \text{ y } w_{il} \in (0, 1) \text{ con } 1 \leq i \leq n, 1 \leq l \leq k$$

donde $w_{il} \in W$, W es una matriz de $n \times k$, y los elementos $w_{il} = 1$ indica que el objeto X_i es asignado al clúster C_l. Además, $Q = [Q_1, \ldots, Q_k]$ es un conjunto de vectores de modas representantes de los k grupos, es decir, la matriz Q actúa como centroide, donde $Q_l = (q_{l1}, \ldots, q_{lm})'$, $l = 1, \ldots, k$. De acuerdo con lo propuesto por Andreopoulos, An & Wang (2006), en la práctica es utilizado el algoritmo 2.4.1.

2.4.1. Algoritmo

1. Seleccionar las k modas iniciales para cada clúster.
2. Para cada objeto X
   - Calcular la disimilitud entre el objeto X y las modas de todos los clústers.
   - Insertar el objeto X dentro del clúster cuya moda sea la más similar al objeto X
   - Actualizar las modas de los clústers.
3. Calcular de nuevo la disimilitud de los objetos con las modas actuales. Si un objeto es más similar a la moda de otro clúster que su actual moda, reasignar el objeto a ese clúster y actualizar ambas modas.
4. Repetir 3 hasta que los objetos no hayan cambiado después de probar el ciclo completo del conjunto de datos.

Tal como el algoritmo k-medias, los resultados arrojados por el k-modas produce óptimos locales, éstos dependen de las modas iniciales y del orden de los objetos en el conjunto de datos. Su complejidad computacional es igual a la del algoritmo k-medias.

El algoritmo k-modas original (Huang 1998) presenta ciertas debilidades tanto en la asignación de los objetos como en la selección de los centros de los conglomerados, ya que la medida de disimilitud usada no considera la relación implícita integrada en los valores categóricos y esto produce grupos con una similitud interna más débil (He, Xu & Deng 2007).

Por las razones antes mencionadas se han realizado diversos estudios para mejorar el algoritmo k-modas original, como He, Deng & Xu (2005) que modifica el algoritmo considerando las frecuencias de los valores del atributo en la medida de disimilitud. Ng, Li, Huang & Zengyou (2007) abordan de manera más pertinente el algoritmo presentado en He et al. (2005). Recientemente He et al. (2007) presentaron diversos esquemas para ponderar el valor del atributo en el agrupamiento k-modas.

3. Simulación

3.1. Simulación para datos continuos

Los algoritmos por comparar en esta sección son k-medias, k-medianas, PAM y Clara, y adicionalmente un quinto algoritmo, el cual es una variación del algoritmo k-medias, pero con la elección de los centroides iniciales resultantes de un agrupamiento por método jerárquico (método de Ward), que para efectos de notación será denominando "k-medias jer.". La literatura usual aborda la aplicación de estos métodos considerando sólo datos reales. Recientemente, un estudio desarrollado por Velmurugan & Santhanam (2010), compara la eficiencia de los algoritmos k-medias y k-medoids a través de datos simulados en términos del tiempo, concluyendo que el algoritmo k-medias demora más que k-medoids.

Lo anterior motivó a realizar el estudio usando siete tipos de esquemas de conglomerados y aplicándolos a 5 y 8 grupos, respectivamente, tal como se ilustra en la figura 1, siguiendo aquellos expuestos en algunos textos especializados y las referencias que éstos contienen (SAS Institute Inc. 2008). El trabajo de Velmurugan & Santhanam (2010) no presenta los distintos esquemas analizados en esta investigación y que efectivamente se pueden dar en la práctica. Esto hace la diferencia de otros artículos, donde la aplicación de éstos considera sólo tres conglomerados en datos reales y simulados (Hae & Chi 2009), siendo diez el máximo de grupos analizados bajo sólo dos esquemas de conglomerados, encontrado en Velmurugan & Santhanam (2010). Para esta investigación, la eficiencia de los algoritmos se mide en cada escenario, a través de los índices de Rand ajustado o "ARI" (Hubert & Arabie 1985) y las respectivas tasas de correcta clasificación (TCC) considerando un determinado algoritmo. El total de simulaciones por cada esquema fue de 50, considerando que se obtuvieron los mismos resultados al aplicar una cantidad mayor de éstas.

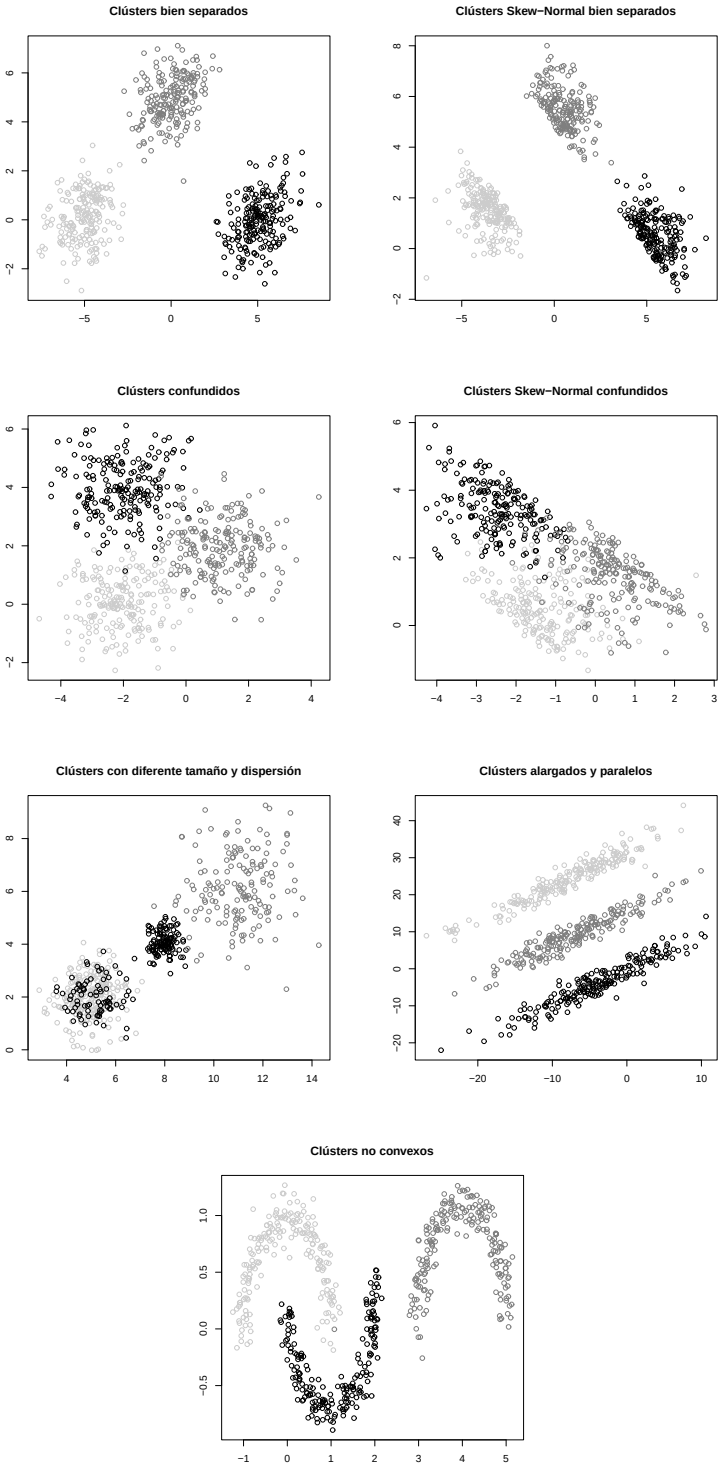

FIGURA 1: Esquemas de simulación para datos continuos.

3.1.1. Resultados de las simulaciones

Analizando las tablas 1 y 2 en forma global, es posible observar que el algoritmo que tiene, en promedio, ambos índices menores tanto para cinco como para ocho grupos, considerando todos los esquemas, es el algoritmo k-medias. A pesar de no existir grandes diferencias entre los otros cuatro algoritmos, el que presenta índices medio mayores es el algoritmo PAM y k-medias jerárquico.

Tabla 1: Medias del índice de Rand ajustado y de la tasa de correcta clasificación en 5 grupos. Datos simulados - caso cuantitativo.

		5 clústers				
Esquema		k-medias	k-medias jer.	k-medianas	PAM	Clara
Bien separados	ARI	0,9141	*0,9991*	*0,9991*	*0,9991*	0,9985
	TCC	0,9279	*0,9996*	*0,9996*	*0,9996*	0,9995
Skew-Normal	ARI	0,8706	0,9995	0,9995	*0,9996*	0,9995
Bien Sep.	TCC	0,8726	*0,9998*	*0,9998*	*0,9998*	0,9998
Confundidos	ARI	0,7843	*0,7884*	0,7820	0,7873	0,7611
	TCC	0,9035	*0,9084*	0,9053	*0,9078*	0,8945
Skew-Normal	ARI	0,8156	*0,8286*	0,8067	0,8281	0,8049
confundidos	TCC	0,9173	*0,9257*	0,9155	0,9219	0,9139
Diferente Tam.	ARI	0,8490	0,9537	0,9529	*0,9547*	0,9504
y dispersión	TCC	0,8604	0,9729	0,9721	*0,9736*	0,9712
Alargados	ARI	0,6181	0,6389	0,6130	*0,7251*	0,7122
	TCC	0,7449	0,7716	0,7568	*0,8762*	0,8673
No convexos	ARI	0,6927	0,7181	0,7535	*0,7553*	0,7356
	TCC	0,8193	0,8641	0,8859	*0,8848*	0,8721
Media	ARI	0,7921	0,8466	0,8438	**0,8642**	0,8517
	TCC	0,8637	0,9203	0,9193	**0,9377**	0,9312
SD	ARI	0,1041	0,1423	0,1454	0,1177	0,1268
	TCC	0,0642	0,0825	0,0851	0,0528	0,0580

Al realizar la comparación entre el ARI y el TCC por cada esquema, se nota que las conclusiones sobre los algoritmos son semejantes, teniendo en cuenta que los valores de las tasas de correcta clasificación son valores más altos que los del índice de Rand ajustado. Considerando lo anterior, a continuación se analizan los resultados de forma independiente para cada esquema sólo por medio del índice de Rand ajustado.

Se analizan los índices medio obtenidos por cada algoritmo en cinco y ocho clústers, resaltando el algoritmo k-medias jerárquico para los cuatro primeros esquemas, seguido muy de cerca por el algoritmo basado en medoids, PAM. Este último presenta los mejores índices para los esquemas de grupos con "Diferente tamaño y dispersión" y "Alargados". Finalmente, el esquema de grupos "No convexo" muestra que el índice asociado al algoritmo k-medianas es el que presenta una mejor agrupación de los datos simulados.

TABLA 2: Medias del índice de Rand ajustado y de la tasa de correcta clasificación en 8 grupos. Datos simulados - caso cuantitativo.

		8 clusters				
Esquema		k-medias	k-medias jer.	k-medianas	PAM	Clara
Bien separados	ARI	0,8740	*0,9997*	*0,9997*	*0,9997*	0,9994
	TCC	0,8972	*0,9999*	*0,9999*	*0,9999*	0,9997
Skew-Normal	ARI	0,8171	*0,9994*	0,9990	*0,9994*	0,9989
Bien Sep.	TCC	0,8148	*0,9998*	0,9996	0,9997	0,9995
confundidos	ARI	0,7272	*0,7396*	0,7264	0,7371	0,6468
	TCC	0,8550	*0,8714*	0,8600	0,8695	0.7877
Skew-Normal	ARI	0,7691	*0,8333*	0,8156	0,8316	0,7899
Confundidos	TCC	0,8548	*0,9213*	0,9127	0,9202	0,8925
diferente Tam.	ARI	0,8901	0,9655	0,9653	*0,9665*	0,9600
y Dispersión	TCC	0,9083	0,9799	0,9826	*0,9831*	0,9797
Alargados	ARI	0,5986	0,5925	0,5901	*0,6878*	0,6743
	TCC	0,6956	0,7521	0,7431	*0,8354*	0,8049
No convexos	ARI	0,7368	0,8364	0,8514	*0,8498*	0,7429
	TCC	0,7659	0,9167	0,9274	*0,9235*	0,8175
Media	ARI	0,7733	0,8523	0,8497	**0,8674**	0,8303
	TCC	0,8274	0,9201	0,9179	**0,9330**	0,8974
SD	ARI	0,0997	0,1511	0,1538	0,1262	0,1534
	TCC	0,0756	0,0884	0,0927	0,0649	0,0955

3.2. Simulación para datos cualitativos

En presencia de datos cualitativos, se propone comparar los algoritmos k-modas y PAM, puesto que este último permite la incorporación de la matriz de distancias o disimilitudes, en lugar de los datos originales. Respecto a la simulación, estas variables se generaron asumiendo distribuciones multinomiales con 2, 3 y 4 categorías. Además, se consideraron probabilidades para las categorías, de tal forma que tres grupos estén bien definidos, variando el número de objetos en cada grupo ($k = 3$).

En la figura 2 se presenta un resumen de los esquemas de simulación utilizados. Para comparar ambos algoritmos se emplea una matriz de disimilitudes construida a través del cálculo del coeficiente general propuesto por Gower (1971), medida adecuada en presencia de este tipo de datos.

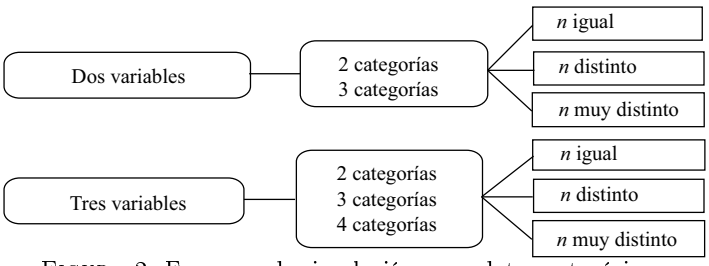

FIGURA 2: Esquema de simulación para datos categóricos.

3.2.1. Resultados de las simulaciones

El índice de Rand ajustado y la tasa de correcta clasificación media obtenida en los diferentes esquemas se presentan en la tabla 3.

Tabla 3: Medias del índice de Rand ajustado y tasa de correcta clasificación. Datos simulados - caso categórico.

		Dos variables		Tres variables	
n		k-moda	PAM	k-moda	PAM
Igual	ARI	0,5234	0,3977	*0,5598*	0,5539
	TCC	*0,8140*	0,7469	*0,8286*	0,8215
Distinto	ARI	0,4486	0,4109	0,4028	*0,4592*
	TCC	*0,7737*	0,7582	0,7613	*0,7811*
Muy distinto	ARI	0,3984	0,3419	0,2970	*0,3006*
	TCC	*0,7547*	0,7332	*0,6718*	0,6685
Media	ARI	**0,4568**	0,3835	0.4199	**0,4379**
	TCC	**0,7808**	0,7461	0,7539	**0,7570**
SD	ARI	0,0629	**0,0366**	0,1322	**0,1280**
	TCC	0,0302	**0,0126**	**0,0787**	0,0793

El valor medio del índice es relativamente bajo en todos los esquemas, aproximadamente menor a 0,56; sin embargo, se puede decir que para el caso de dos variables, el algoritmo que obtuvo mejores resultados fue el k-modas, y al aumentar la cantidad de variables a tres, se presenta una mejora en los índices medios obtenidos por PAM, considerando los esquemas en donde n es "Distinto" o "Muy distinto". Los índices medios más altos fueron obtenidos cuando n era igual para todos los grupos, y apunta a que el algorimto k-modas presenta la mejor agrupación de la información. Esto último sucede de la misma manera con la tasa de correcta clasificación, con la diferencia de que toma valores más altos, alcanzando un 83 % aproximado como máximo, en el caso de tres variables con n igual.

4. Aplicaciones

Las bases de datos reales se obtuvieron del UCI Machine Learning Repository (Asuncion & Newman 2007). Para el caso continuo se utilizan seis conjuntos de datos del UCI Repository, de los cuales todos contienen atributos continuos y atributos de clase. En la implementación categórica se utilizan cinco conjuntos de datos, utilizados por He et al. (2007), de los cuales todos contienen solo atributos categóricos y atributos de clase. Para ambos casos, los resultados son presentados considerando el respectivo índice de Rand ajustado y la tasa de clasificación correcta.

4.1. Resultados obtenidos de bases de datos continuos

La distribución de las frecuencias de las clases se puede observar en la tabla 4. Una vez que se aplican las metodologías, tal como se observa en la tabla 5, los

cinco algoritmos no obtuvieron grandes diferencias en las tres primeras bases de datos; sin embargo, en "Wine" y "Ecoli" PAM presenta una disminución importante en ARI y en su TCC. Estas dos bases de datos tienen las frecuencias con las clases menos homogéneas, lo que puede provocar que PAM no entregue óptimos resultados. En la última base de datos, "Image Segment", PAM presenta mejores resultados en comparación con el resto de los algoritmos.

Tabla 4: Frecuencias de las clases en las bases de datos continuas.

Base de datos	Clases	Atributos	Casos	Distribución de las clases
Iris	3	4	150	50 c/u
Breast Cancer	2	30	569	357/212
Bank	2	6	200	100 c/u
Wine	3	13	178	71/59/48
Ecoli	4	7	336	143/116/52/25
Imag. Segment.	7	19	210	30 c/u

Tabla 5: Índice de Rand ajustado y tasa de correcta clasificación para datos continuos.

Base de datos	k-medias	k-medias jer.	k-medianas	PAM	Clara
Iris	0,893	0,893	0,887	0,893	*0,900*
Breast Cancer	0,854	0,854	0,866	*0,868*	*0,868*
Bank	*1*	*1*	0,990	0,990	0,980
Wine	*0,966*	*0,966*	0,961	0,910	0,938
Ecoli	0,738	0,866	*0,880*	0,676	0,735
Imag. Segment	0,519	0,580	0,595	*0,633*	0,491
Media	0,828	0,860	**0,863**	0,828	0,819

4.2. Resultados obtenidos de bases de datos categóricos

Tal como lo muestra la tabla 6, la distribución de frecuencias de las primeras bases de datos contiene frecuencias homogéneas para cada uno de los atributos. A diferencia de los primeros ejemplos, las dos últimas bases de datos contienen frecuencias desbalanceadas, con valores extremos en algunos casos.

Tabla 6: Frecuencias de las bases de datos categóricas.

Base de datos	Clases	Atributos	Casos	Distribución de las clases
Voting	2	16	435	168/267
Breast Cancer	2	9	699	241/458
Soybean	4	35	47	10/10/10/17
Lymphography	4	18	148	2/4/61/81
Zoo	7	17	101	4/5/8/10/13/20/41

Como se observa en la tabla 7, para "Breast Cancer", "Soybean" y la base de datos de los votos del congreso ("Voting"), el algoritmo PAM muestra mejores resultados que los entregados por el algoritmo k-modas. Sin embargo, en los dos últimos conjuntos de datos, el algoritmo k-modas funciona mejor que el algoritmo de los medoids.

Tabla 7: Tasa de clasificación correcta de datos categóricos.

Base de datos	PAM	k-modas
Voting	*0,864*	0,859
Breast Cancer	*0,937*	0,85
Soybean	*0,936*	0,819
Lymphography	0,412	*0,661*
Zoo	0,743	*0,829*
Media	0,778	**0,804**

Una razón por la cual el uso del algoritmo PAM con datos categóricos no da buenos resultados en los dos últimos conjuntos de datos podría deberse a que las frecuencias de las clases que componen los datos difieren mucho entre sí, es decir, las frecuencias de las clases de "Lymphography" y "Zoo" son menos homogéneas que las del resto.

5. Conclusiones

Dada la importancia que ha tomado el análisis de conglomerados en los diversos estudios de aplicación, se ha querido abordar la presentación, implementación y comparación de los algoritmos de partición de mayor uso en la literatura. En este trabajo se presenta un estudio de simulación y aplicación con el fin de exponer y comparar los algoritmos de partición, según distintos esquemas de agrupación.

En el estudio de simulación realizado, en el caso continuo, los algoritmos que dieron mejores resultados fueron el k-medias jerárquico y PAM, en donde este último sobresale en los esquemas con grupos con "Diferente tamaño y dispersión" y "Alargados". El algoritmo k-medianas da resultados sobresalientes en los esquemas "No convexos", esquema que en general tiene problemas con los valores iniciales y con la convergencia a una única solución. El algoritmo Clara presentó resultados similares a PAM en el esquema de grupos con "Diferentes tamaños y dispersión"; sin embargo, es más inestable ya que trabaja con base a muestras. Para el caso categórico, el algoritmo k-modas obtuvo mejores resultados que PAM; no obstante los resultados de PAM mejoraron al aumentar la cantidad de variables. Cabe acotar que k-modas, a diferencia de PAM, presenta resultados coherentes al considerar esquemas donde las frecuencias de las clases son aproximadamente iguales. Es evidente que si el número de observaciones o el tamaño muestral aumenta, la complejidad de los algoritmos también lo hará, lo que llevará a proponer la aplicación de otros métodos para optimizar el tiempo de ejecución y reducir el costo computacional.

En las aplicaciones realizadas a datos continuos en promedio el algoritmo más débil fue k-medias y en cuatro conjuntos de datos los demás algoritmos dan resultados muy semejantes. Por otro lado, PAM fues el que obtuvo un índice menor que el resto en los datos asociados a "Wine" y "Ecoli". Sin embargo, en "Imag. segment." el algoritmo PAM es el que da una mejor tasa de buena clasificación. Al comparar PAM con k-modas sucede que en las bases de datos en donde la distribución de las frecuencias de las clases son relativamente homogéneas, PAM entrega mejores

resultados que k-modas, pero PAM no es eficiente al tener las frecuencias de las clases muy heterogéneas.

Agradecimientos

Los autores agradecen a los árbitros por tan importantes comentarios que ayudaron a mejorar el contenido de este trabajo.

[Recibido: mayo de 2010 — Aceptado: octubre de 2010]

Referencias

Abonyi, J. & Feil, B. (2007), *Clustering Analysis for Data Mining and System Identification*, Birkhauser Verlag AG, Berlin, Germany.

Anderberg, M. (1973), *Cluster Analysis for Applications*, Academic Press, New York, United States.

Anderson, B., Gross, D., Musicant, D., Ritz, A., Smith, T. & Steinberg, L. (2006), Adapting K-Medians to Generate Normalized Cluster Centers, *in* 'Proceedings of the 2006 SIAM International Conference on Data Mining', Bethesda, pp. 165–175.

Andreopoulos, B., An, A. & Wang, X. (2006), Clustering Mixed Numerical and Low Quality Categorical Data: Significance Metrics on a Yeast Example, *in* 'ACM SIGMOD Workshop on Information Quality in Information Systems, IQIS 2005 Statistics Clustering Session', Baltimore, pp. 87–98.

Asuncion, A. & Newman, D. J. (2007), 'UCI machine learning repository'. *http://www.ics.uci.edu/∼mlearn/MLRepository.html

Der, G. & Everitt, B. S. (2006), *Statistical Analysis of Medical Data using SAS*, CRC Press, Boca Raton, United States.

Gower, J. C. (1971), 'A General Coefficient of Similarity and Some of its Properties', *Biometrics* **27**, 623–637.

Hae, P. & Chi, J. (2009), 'A simple and Fast Algorithm for K-medoids Clustering', *Expert Systems with Applications* **36**, 3336–3341.

Han, J., Kamber, M. & Tung, A. K. H. (2001), Spatial Clustering Methods in Data Mining: A Survey, *in* H. J. Miller & J. Han, eds, 'Geographic Data Mining and Knowledge Discovery', Taylor & Francis.

Hartigan, J. (1975), *Clustering Algorithms*, John Wiley & Sons, Nueva York, United States.

He, Z., Deng, S. & Xu, X. (2005), *Improving K-modes Algorithm Considering Frequencies of Attribute Values in Mode*, Vol. 3801 of *Lecture Notes in Computer Science*, Springer Berlin - Heidelberg, pp. 157–162.

He, Z., Xu, X. & Deng, S. (2007), 'Attribute Value Weighting in K-Modes Clustering', *Computer Science e-Prints* **1**, 1–15.

Huang, Z. (1998), 'Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values', *Data Mining and Knowledge Discovery* **2**, 283–304.

Hubert, L. & Arabie, P. (1985), 'Comparing Partitions', *Journal of Classification* **2**, 193–218.

Kamber, M. & Han, J. (2006), *Data Mining Concepts and Techniques*, Morgan Kaufman Publishers, San Francisco, United States.

Kaufman, L. & Rousseeuw, P. (1987), Clustering by Means of Medoids, *in* D. Y., ed., 'Statistical Data Analysis Based on the L_1 Norm and Related Methods', North-Holland, pp. 405–416.

Kaufman, L. & Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley and Sons, New York, United States.

Leiva, S. (2008), Algoritmos de partición en el análisis de conglomerados: un estudio comparativo, Trabajo de grado, Universidad de Santiago de Chile, Chile.

MacQueen, J. (1967), Some Methods for classification and Analysis of Multivariate Observations, *in* 'Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability', Vol. 1, Symposium on mathematics, pp. 281–297.

McGarigal, K., Cushman, S. & Stafford, S. (2000), *Multivariate Statistics for Wildlife and Ecology Research*, Springer Verlag, New York, United States.

Ng, M. K., Li, M. J., Huang, J. Z. & Zengyou, H. (2007), 'On the Impact of Dissimilarity Measure in k-Modes Clustering Algorithm', *, IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(3), 503–507.

Ng, R. & Han, J. (1994), Efficient and Effective Clustering Methods for Spatial Data Mining, *in* 'Proceeding of the 20^{th} International Conference on Very Large Databases', pp. 144–155.

Peña, D. (2002), *Análisis de datos multivariantes*, McGraw-Hill, Madrid, España.

Quinn, G. & Keough, M. (2002), *Experimental Design and Data Analysis for Biologists*, Cambridge University Press, Cambridge, UK.

R Development Core Team (2010), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
*http://www.R-project.org

SAS Institute Inc. (2008), *SAS/STAT 9.2 User's Guide*, SAS Publishing, Cary, Carolina del Norte.

Velmurugan, T. & Santhanam, T. (2010), 'Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points', *Journal of Computer Science* **6**(3), 363–368.