UNIVERSIDAD NACIONAL AGRARIA LA MOLINA

FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

Departamento Académico de Estadística e Informática

TRABAJO MONOGRÁFICO

"DESCRIPCIÓN DEL PROCEDIMIENTO METODOLÓGICO DEL ANÁLISIS CLUSTER NO JERÁRQUICO CON EL ALGORITMO CLARANS"

Presentado para optar el título de Ingeniero

ESTADÍSTICO INFORMÁTICO

CARLOS AGUSTÍN SALVADOR ALFARO

Modalidad Examen Profesional

Lima – Perú
2017

Dedicatoria

Agradezco a mi padre Henry
por confiar en que puedo lograrlo
y a mi madre Ilda por ser
guía desde el cielo

INDICE

ÍNDICE DE FIGURAS

RESUMEN

El algoritmo CLARANS, perteneciente a los métodos clúster no jerárquico. Lo que se pretende describir en este trabajo es explicar el procedimiento del algoritmo CLARANS.

El proceso que realiza este algoritmo es encontrar una muestra con una cierta aleatoriedad en cada paso de la búsqueda. El agrupamiento obtenido después de sustituirlo a un solo medoide se denomina el vecino del agrupamiento actual. Si en el camino el objeto (individuo) encuentra un mejor vecino, CLARANS lo mueve al nodo del vecino y el proceso comienza de nuevo; si ya no lo encuentra entonces el agrupamiento actual para y se produce un óptimo local (Cluster).

Se presenta un ejemplo que ilustra la metodología y se explica el paso a paso del algoritmo CLARANS.

CAPITULO I: INTRODUCCIÓN

Cada año las empresas están más interesadas en conocer a su público objetivo ya sea para realizar campañas de marketing dirigidas, segmentar a sus clientes o posicionar su marca de manera estratégica. Para ello existen diferentes técnicas estadísticas diseñadas para el agrupamiento de individuos según características similares, llamadas a estas técnicas de agrupamiento o clúster . Las cuales tienen como objetivo la clasificación de individuos en grupos homogéneos internamente y heterogéneos entre sí.

En la construcción de clúster existen métodos de agrupamiento jerárquico y no jerárquico dentro de los cuales existe una amplia bibliografía, sin embargo en el algoritmo CLARANS, perteneciente a los métodos clúster no jerárquico, existen pocos estudios que aborden este tema a profundidad o artículos en castellano referidos a este brinden una solución eficiente y eficaz sobre el agrupamiento de individuos.

Los objetivos que se pretenden describir en este trabajo es explicar el procedimiento del algoritmo CLARANS y señalar su importancia frente al resto de algoritmos similares mediante un ejemplo basado en un artículo científico.

El presente trabajo se divide en cuatro capítulos, en el primero se abordan las definiciones y conceptos principales que dan validez teórica a la investigación. Se empezará explicando el análisis clúster, el cual es un procedimiento estadístico de clasificación que pretende identificar grupos relativamente homogéneos de casos (o de variables) basándose en las características seleccionadas. Asimismo se mencionan los procedimientos jerárquicos y los no jerárquicos, en donde el primero precisa que el investigador fije de antemano el número de clúster en que desea agrupar con los datos, y los métodos no jerárquicos, o también llamados como métodos partitivos o de optimización, los cuales tienen como objetivo realizar una sola partición de los individuos en k grupo, lo que conlleva que el investigador

deberá especificar a priori los grupos que deben ser formados. Dentro del análisis clúster no jerárquico, existen familias de algoritmos clustering que optimizan la partición de los grupos.

El segundo capítulo se dará a conocer de forma clara y esquemática cómo funciona el algoritmo CLARANS y luego mediante un ejemplo se pondrá a prueba su alcance y eficiencia en resultados sobre otros algoritmos.

Mediante un ejemplo en el tercer capítulo se explican el paso a paso del algoritmo así como el uso del paquete R en las diferentes librerías.

Las conclusiones que se desglosan del trabajo afirman la importancia de este algoritmo en el procedimiento no jerárquico en la optimización y eficacia en el agrupamiento de individuos frente a otros algoritmos.

Es importante la monografía realizada ya que constituye un estudio realizado sobre la base de la revisión de diversas fuentes bibliográficas y esperamos se convierta en referente de consulta para estudiantes de la misma carrera profesional. Se desea exponer los beneficios de utilizar está técnica estadística en la minería de datos debido a que no existen muchas investigaciones previas (y en castellano) acerca del uso del algoritmo CLARANS, daremos un aporte exponiendo los beneficios de usar este algoritmo que servirá de base para futuras investigaciones acerca del agrupamiento no jerárquico de individuos mediante un algoritmo eficiente y eficaz.

La bibliografía utilizada se basó en artículos, ponencias y libros de minería de datos.
De igual modo dará a conocer a estudiantes nuevas metodologías en agrupamiento para minería de datos en temas comerciales, marketing, finanzas y otros afines.

CAPITULO II: ANÁLISIS CLUSTER

2.1. Definición

También conocido como Análisis de conglomerados, es una técnica estadística multivariante que busca agrupar elementos (o variables) tratando de lograr la máxima homogeneidad en cada grupo y la mayor diferencia entre los grupos. Es un método estadístico multivariante de clasificación automática de datos. A partir de una tabla de casos-variables, trata de situar los casos (individuos) en grupos homogéneos, conglomerados o clusters, no conocidos de antemano pero sugeridos por la propia esencia de los datos, de manera que individuos que puedan ser considerados similares sean asignados a un mismo cluster, mientras que individuos diferentes (disimilares) se localicen en clusters distintos.

Es un método estándar del análisis multivariado que puede reducir una compleja cantidad de información en pequeños grupos o clústers, donde los miembros de cada uno de ellos comparten características similares (Lin & Chen, 2006).

El Análisis clúster se considera una técnica eminentemente exploratoria que no utiliza ningún tipo de modelo estadístico para llevar a cabo el proceso de clasificación (Hair *et al*., 1999; Peterson, 2002) y, por ello, se le podría calificar como una técnica de aprendizaje no supervisado, es decir, una técnica muy adecuada para extraer información de un conjunto de datos sin imponer restricciones previas en forma de modelos estadísticos (Barrios & Carvajal, 2006).

El análisis clúster tiene por objeto formar grupos o clústers homogéneos en función de las similitudes o similaridades entre ellos (Peña, 2002). Los grupos se forman de tal manera que cada objeto es parecido a los que hay dentro del clúster con respecto a algún criterio de

selección predeterminado (Rao & Srinivas, 2006; Hair *et al*., 1999). Las técnicas de agrupamiento en el análisis clúster se pueden clasificar en dos categorías: el clúster jerárquico y el no jerárquico.

La creación de grupos basados en *similaridad* de casos exige una definición de este concepto, o de su complementario *distancia* entre individuos. La variedad de formas de medir diferencias multivariables o *distancias* entre casos proporciona diversas posibilidades de análisis. El empleo de ellas, y el de las que continuamente siguen apareciendo, así como de los algoritmos de clasificación, o diferentes reglas matemáticas para asignar los individuos a distintos grupos, depende del fenómeno estudiado y del conocimiento previo de posible agrupamiento que de él se tenga.

Puesto que la utilización del análisis cluster ya implica un desconocimiento o conocimiento incompleto de la clasificación de los datos, el investigador ha de ser consciente de la necesidad de emplear varios métodos, ninguno de ellos incuestionable, con el fin de contrastar los resultados.

Los procedimientos jerárquicos consisten en la construcción de una estructura en forma de árbol. Existen dos tipos de procedimientos de obtención de clústers jerárquicos: los de aglomeración y los divisivos.

Dentro de los métodos jerárquicos aglomerativos se tienen:
(i) método de encadenamiento simple,
(ii) métodos de encadenamiento completo,
(iii) método de encadenamiento medio,
(iv) método de Ward, y
(v) método del centroide (Hair *et al*., 1999).

Estos procedimientos difieren en la forma como se calcula la distancia entre los conglomerados, entre los que se encuentran la DEC, Manhattan, coeficiente de correlación de Pearson, Chevichev y Cosine.

Sin embargo el clúster encontrados mediante técnicas no jerárquicas no requiere de procesos de construcción de árboles; en su lugar, asignan los objetos a clústers una vez que el número de grupos a formar esté especificado. Los procedimientos de aglomeración no jerárquicos se denominan frecuentemente agrupaciones de k – medias, k –medianas y k – modas.

Una desventaja con respecto a la técnica jerárquica consiste en que debe conocerse a priori el número de clústers a obtener, lo que implica un grado de subjetividad en el proceso (Peterson, 2002). A pesar de lo anterior, se considera un método dinámico en el sentido en que los objetos dentro de los clústers se pueden mover de un clúster a otro, minimizando la distancia entre objetos dentro de un mismo clúster (Rao & Srinivas, 2006).

Una vez finalizado un análisis de clusters, el investigador dispondrá de una colección de casos agrupada en subconjuntos jerárquicos o no jerárquicos. Podrá aplicar técnicas estadísticas comparativas convencionales siempre que lo permita la relevancia práctica de los grupos creados; así como otras pruebas multivariantes, para las que ya contará con una variable dependiente *grupo*, aunque haya sido creada artificialmente.

2.1.1. Etapas de un Cluster

1. Selección de la muestra de datos
2. Selección y transformación de variables a utilizar
3. Selección de concepto de distancia o similitud y medición de las mismas
4. Selección y aplicación del criterio de agrupación
5. Determinación de la estructura correcta (elección del número de grupos)

2.1.2. Técnicas gráficas para determinar la agrupación óptima

La decisión sobre el número óptimo de clusters es subjetiva, especialmente cuando se incrementa el número de objetos pues si se seleccionan pocos, los clusters resultantes son heterogéneos y artificiales, mientras que si se seleccionan demasiados, la interpretación de los mismos suele resultar complicada.

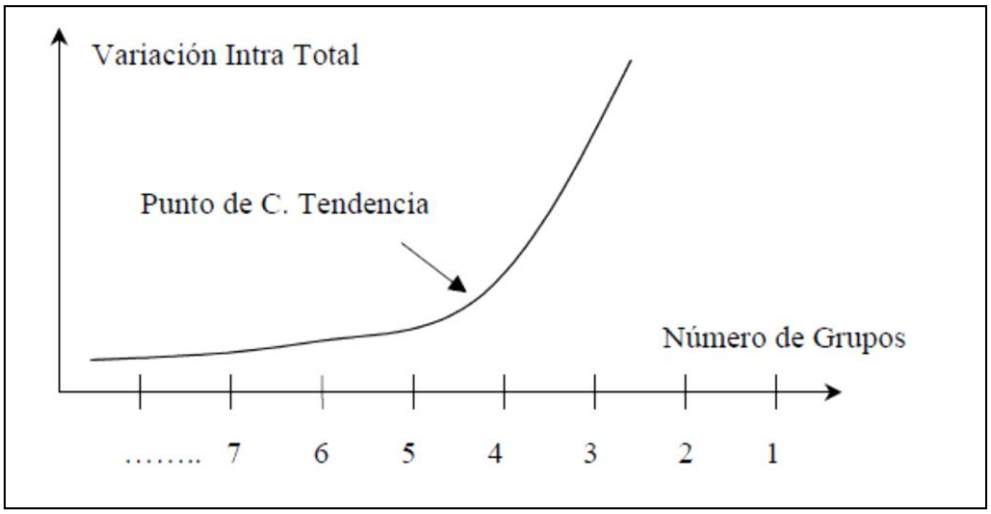

Figura 1: Observación de la variación intragrupal

FUENTE: De la Fuente Crespo (2000)

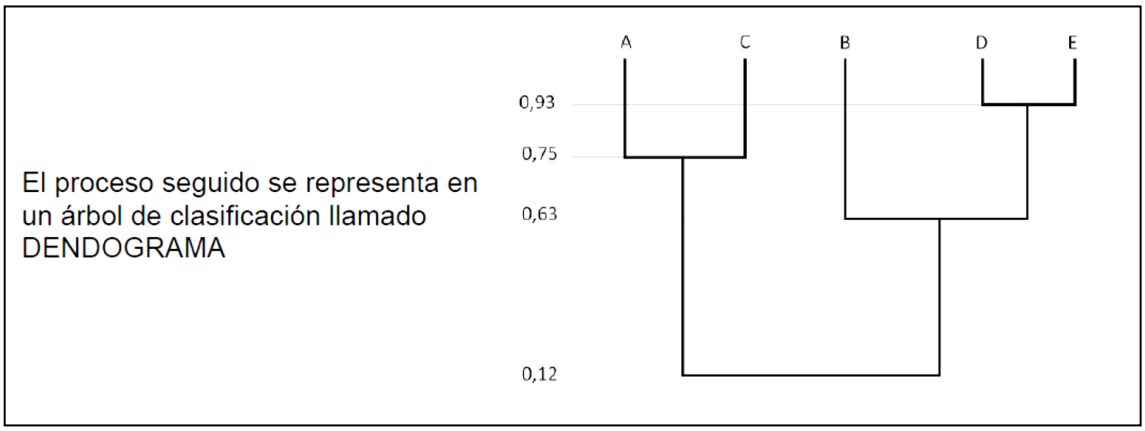

Figura 2: Dendograma: Representación gráfica de una clasificación jerárquica

FUENTE: De la Fuente Crespo (2000)

Un dendograma es una representación gráfica en forma de árbol que resume el proceso de agrupación en un análisis de clusters. Los objetos similares se conectan mediante enlaces cuya posición en el diagrama está determinada por el nivel de similitud/disimilitud entre los objetos.

2.1.3. Número óptimo de grupos

No existen criterios objetivos y ampliamente válidos, sin embargo a medida que se van formando grupos estos son menos homogéneos (las distancias para las que se forman los grupos iniciales son menores que las de los grupos finales) pero la estructura es más clara.

Se puede fijar un objetivo, el cual es identificar el punto de equilibrio entre la estructura incompleta y la estructura mezclada o confusa.

Es difícil definir conceptualmente y más aún estadísticamente la situación de estructura correcta, no confusa, o la contraria de falta de estructura. (Estructura por asociación o diferenciación)

En la observación, tanto de las variables iniciales, como de la definición inicial de los sujetos y el significado de cada una de las etapas del proceso de agrupación.

De igual manera se puede utilizar alguna herramienta técnica discriminante, caída mediante un punto de inflexión en la similitud o en la homogeneidad, dendograma.
Entre los métodos de agrupamiento o clúster se tienen métodos jerárquicos y no jerarquicos:

2.2. Métodos jerárquicos

La clasificación de todos los casos de una tabla de datos en grupos separados configura el propio análisis de clusters no jerárquicos. Esta denominación alude a la no presencia de una estructura vertical de dependencia entre los grupos formados y, por tanto, éstos no se presentan en distintos niveles de jerarquía.

El análisis precisa que el investigador fije de antemano el número de clusters en que desea agrupar los datos. Como puede no existir un número definido de grupos o, si existe, generalmente no se conoce, la prueba debe ser repetida con diferente número de clusters con la finalidad de tantear la clasificación que mejor se ajuste al objetivo del problema, o a la más clara interpretación.

Los métodos no jerárquicos, también se conocen como *métodos partitivos* o de optimización, considerando que tienen por objetivo realizar una sola partición de los individuos en k grupos. Esto conlleva que el investigador debe especificar a priori los grupos que deben ser

formados. Ésta es, probablemente, la principal diferencia respecto de los métodos jerárquicos. La asignación de individuos (casos) a los grupos se realiza mediante algún proceso que optimice el criterio de selección.

Otra diferencia de los métodos no jerárquicos es que trabajan con la matriz de datos originales y no requieren su conversión en una matriz de proximidades.

Resulta muy intuitivo suponer que una clasificación correcta debe ser aquélla en que la dispersión dentro de cada grupo formado sea la menor posible. Esta condición se denomina *criterio de varianza*, y lleva a seleccionar una configuración cuando la suma de las varianzas dentro de cada grupo (varianza residual) sea mínima.

El objetivo de estos métodos es la de agrupar los cluster para formar uno nuevo o separar alguno ya existente para dar origen a otros dos de forma que se maximice una medida de similaridad o se minimice alguna distancia. Los métodos jerárquicos permiten construir un diagrama de árbol de clasificación o dendograma

Estos se pueden clasifican en Asociativos y Disociativos.

2.2.1. Asociativos o Aglomerativos

Se parte de tantos grupos como individuos hay en el estudio y se van agrupando hasta llegar a tener todos los casos en un mismo grupo.

2.2.2. Disociativos

Se parte de un solo grupo que contiene todos los casos y a través de sucesivas divisiones se forman grupos cada vez más pequeños.

2.2.3. Algoritmos de agrupación jerárquicos

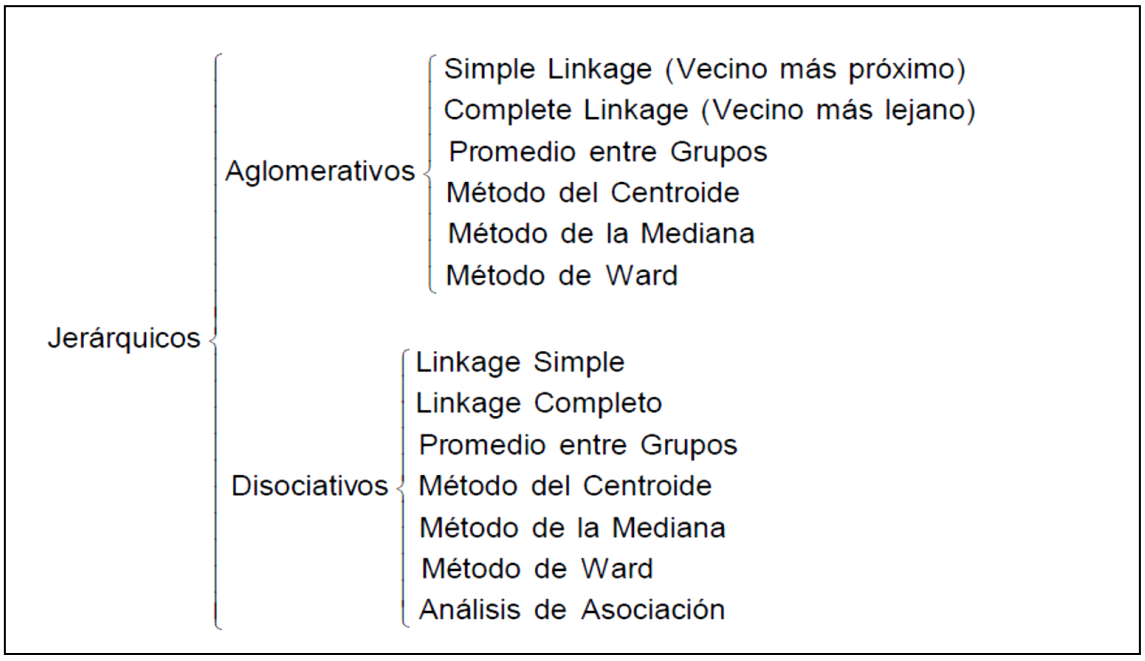

Figura 3: Esquema de Algoritmos de agrupación Jerárquicos

FUENTE: De la Fuente Crespo (2000)

2.2.4. Distancias entre conglomerados

Las distancias entre los conglomerados son funciones de las distancias entre observaciones, hay varias formas de definirlas:

Sean A y B dos conglomerados:

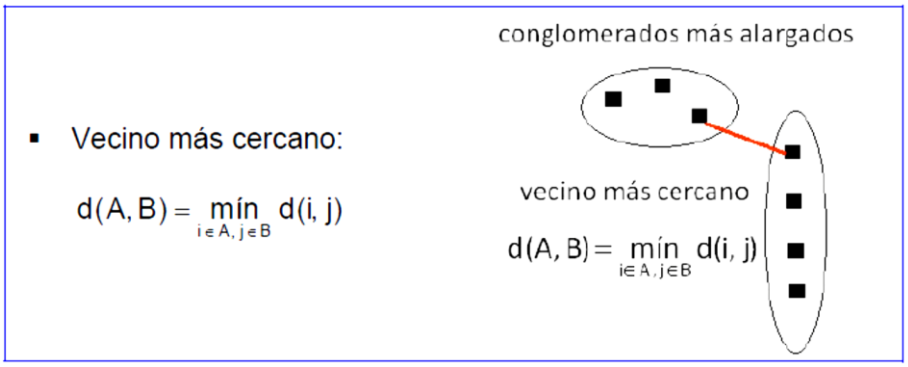

Figura 4: Esquema de Vecino cercano

FUENTE: De la Fuente Crespo (2000)

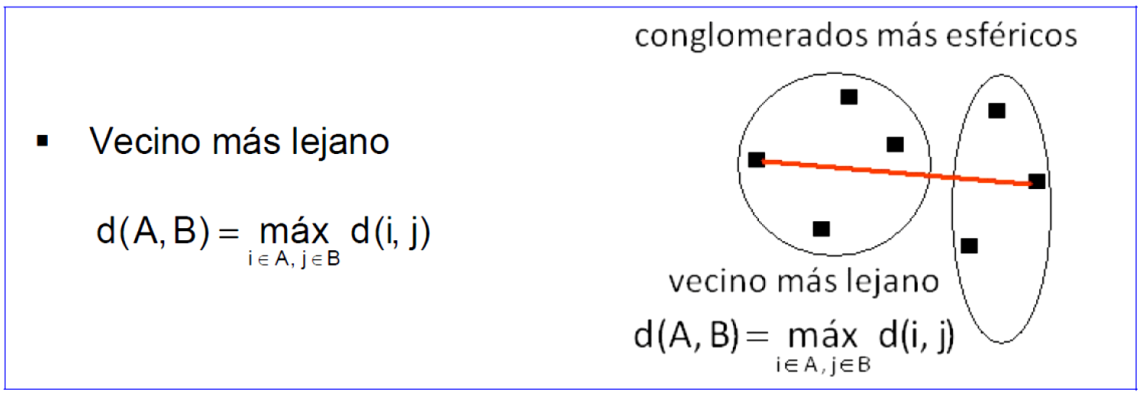

Figura 5: Esquema de Vecino más lejano

FUENTE: De la Fuente Crespo (2000)

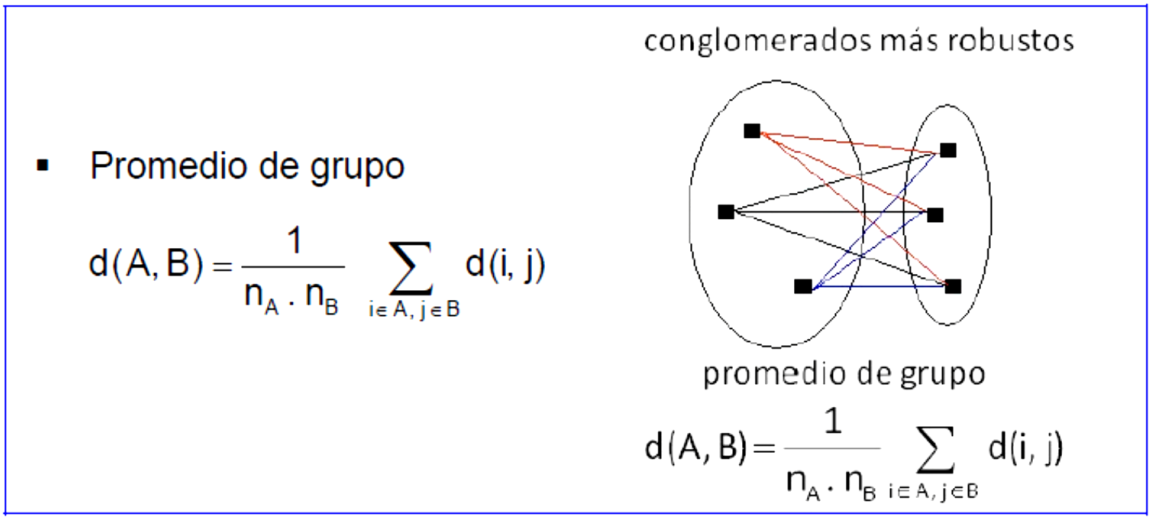

Figura 6: Esquema del Promedio de grupo

FUENTE: De la Fuente Crespo (2000)

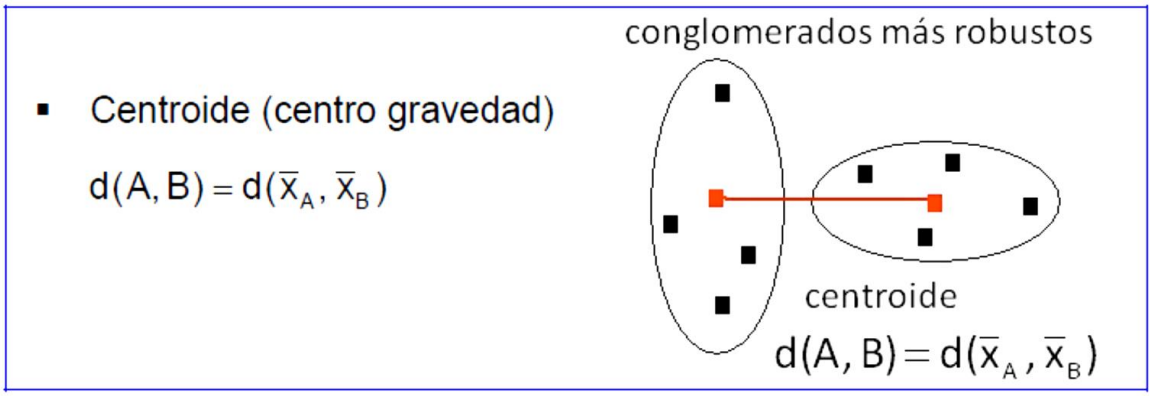

Figura 7: Esquema de Centroide

FUENTE: De la Fuente Crespo (2000)

2.2.5. Método linkage simple aglomerativo (*vecino más cercano*)

Una vez que se conocen las distancias existentes entre cada dos individuos se observa cuáles son los individuos más próximos en cuanto a esta distancia o similaridad (qué dos individuos tienen menor distancia o mayor similaridad). Estos dos individuos forman un grupo que no vuelve a separarse durante el proceso.

Se repite el proceso, volviendo a medir la distancia o similaridad entre todos los individuos de nuevo (tomando el grupo ya formado como sí de un solo individuo se tratara) de la siguiente forma:

- Cuando se mide la distancia entre el grupo formado y un individuo, se toma la *distancia mínima* de los individuos del grupo al nuevo individuo.
- Cuando se mide la *similitud* o *similaridad* entre el grupo formado y un individuo, se toma la *máxima* de los individuos del grupo al nuevo individuo.

2.2.6. Método linkage completo aglomerativo (*Vecino más lejano*)

Conocidas las distancias o similaridades existentes entre cada dos individuos se observa cuáles son los individuos más próximos en cuanto a esta distancia o similaridad (qué dos individuos tienen menor distancia o mayor similaridad). Estos dos individuos formarán un grupo que no vuelve a separarse durante el proceso.

Posteriormente, se repite el proceso, volviendo a medir la distancia o similaridad entre todos los individuos de la siguiente forma:

- Cuando se mide la *distancia* entre el grupo formado y un individuo, se toma la *distancia máxima* de los individuos del grupo al nuevo individuo.
- Cuando se mide la *similitud* o *similaridad* entre el grupo formado y un individuo, se toma la *mínima* de los individuos del grupo al nuevo individuo

2.3. Métodos no jerárquicos

Estos métodos están diseñados para la clasificación de individuos (no de variables) en K grupos. El procedimiento es elegir una partición de los individuos en K grupos e intercambiar los miembros de los clusters para tener una partición mejor

Figura 8: Esquema de Algoritmos de agrupación No Jerárquicos

FUENTE: De la Fuente Crespo (2000)

En los cluster no jerárquicos los datos se dividen en k particiones o grupos donde cada partición representa un cluster. Opuestamente a los métodos jerárquicos el número de cluster debe conocerse a priori.

Básicamente siguen los siguientes pasos:

1. Seleccionar K medias iniciales, siendo K el número de clusters deseados.
2. Asignar cada observación al cluster que le sea más cercano.
3. Reasignar o relocalizar cada observación a uno de los K cluster de acuerdo con alguna regla de parada.

4. Parar si no hay reasignación de los puntos o si la reasignación satisface la regla de parada. En otro caso se vuelve al paso dos.

La mayoría de los algoritmos no jerárquicos difieren con respecto a:

- El procedimiento para obtener los medias iniciales.
- La regla que se usa para reasignar las observaciones

La clasificación de todos los casos de una tabla de datos en grupos separados configura el propio análisis de clusters no jerárquicos. Esta denominación alude a la no presencia de una estructura vertical de dependencia entre los grupos formados y, por tanto, éstos no se presentan en distintos niveles de jerarquía. El análisis precisa que el investigador fije de antemano el número de clusters en que desea agrupar los datos.

Como puede no existir un número definido de grupos o, si existe, generalmente no se conoce, la prueba debe ser repetida con diferente número de clusters con la finalidad de tantear la clasificación que mejor se ajuste al objetivo del problema, o a la más clara interpretación.

Los métodos no jerárquicos, también se conocen como *métodos partitivos* o de optimización, considerando que tienen por objetivo realizar una sola partición de los individuos en k grupos. Esto conlleva que el investigador debe especificar a priori los grupos que deben ser formados. Ésta es, probablemente, la principal diferencia respecto de los métodos jerárquicos. La asignación de individuos (casos) a los grupos se realiza mediante algún proceso que optimice el criterio de selección.

Otra diferencia de los métodos no jerárquicos es que trabajan con la matriz de datos originales y no requieren su conversión en una matriz de proximidades.

Resulta muy intuitivo suponer que una clasificación correcta debe ser aquélla en que la dispersión dentro de cada grupo formado sea la menor posible. Esta condición se denomina *criterio de varianza*, y lleva a seleccionar una configuración cuando la suma de las varianzas dentro de cada grupo (varianza residual) sea mínima.

2.3.1. Algoritmo de las k-medias

Parte de unas medias arbitrarias y, mediante pruebas sucesivas, contrasta el efecto que sobre la varianza residual tiene la asignación de cada uno de los casos a cada uno de los grupos. El valor mínimo de varianza determina una configuración de nuevos grupos con sus respectivas medias. Se asignan otra vez todos los casos a estos nuevos centroides en un proceso que se repite hasta que ninguna transferencia puede ya disminuir la varianza residual; o bien se alcance otro criterio de parada: un número limitado de pasos de iteración prefijado o, simplemente, que la diferencia obtenida entre los centroides de dos pasos consecutivos sea menor que un valor prefijado.

El procedimiento configura los grupos maximizando la distancia entre sus centros de gravedad. Como la varianza total es fija, minimizar la residual hace máxima la factorial o inter-grupos. Y puesto que minimizar la factorial es equivalente a conseguir que sea mínima la suma de distancias al cuadrado desde los casos a la media del cluster al que van a ser asignados, es esta distancia euclídea al cuadrado la utilizada por el método.

Se comprueban los casos secuencialmente para ver su influencia individual, el cálculo puede verse afectado por el orden de los mismos en la tabla. No obstante, es el algoritmo que mejores resultados produce. Otras variantes propuestas a este método llevan a clasificaciones muy similares.

Como cualquier otro método de clasificación no jerárquica, proporciona una solución final única para el número de clusters elegido, a la que llegará con menor número de iteraciones cuanto más cerca estén las *medias* de arranque de las que van a ser finalmente obtenidas. Los programas estadísticos seleccionan generalmente estos primeros valores, tantos como grupos se pretenda formar, entre los puntos más separados de la nube.

Los clusters no jerárquicos están indicados para grandes tablas de datos, y son también útiles para la detección de casos atípicos: Si se elige previamente un número elevado de grupos, superior al deseado, aquéllos que contengan muy escaso número de individuos servirán para detectar casos extremos que podrían distorsionar la configuración. Es aconsejable realizar el análisis definitivo sin ellos, ya que con el número deseado de grupos para después,

opcionalmente, asignar los atípicos al cluster adecuado que habrá sido formado sin su influencia distorsionante.

Cabe mencionar que para clasificar los datos en grupos, es importante la elección de un número adecuado de clúster. Siempre será conveniente efectuar varios tanteos, la selección del más apropiado al fenómeno que se analiza se basa en criterios tanto matemáticos como de interpretación.

2.3.2. Elección de puntos semilla

Supuesto que el número de clúster a formar es *k*, un conjunto de *k* puntos semilla no es más que un conjunto de puntos que puede emplearse como núcleo de los clusters sobre los cuales el conjunto de individuos puede agruparse. Los procedimientos, todos subjetivos, que pueden emplearse para tal hecho son:

1. Elegir los primeros *k* individuos del conjunto de datos, como propone McQueen (1967). Este método es el más simple, siempre y cuando la secuenciación en la que los datos han sido introducidos no influencie el resultado final.
2. Etiquetar los casos de 1 a *m* y elegir aquellos etiquetados como

$$\left[\frac{m}{k}\right], \left[\frac{2m}{k}\right], \ldots, \left[\frac{(k-1)m}{k}\right] \text{ y } m$$

   donde [*x*] representa la parte entera de *x*. Con este sistema se pretende compensar la tendencia natural de ordenar los casos en el orden de introducción o alguna otra secuencia no aleatoria.
3. Etiquetar los casos de 1 a *m* y elegir los casos correspondientes a *k* números aleatorios diferentes, (McRae, 1971).
4. Tomar una partición de casos en k grupos mutuamente excluyentes y usar sus centroides como semillas, (Forgy, 1965).
5. Emplear el algoritmo de Astrahan (1970) según el cual se eligirían las semillas de tal forma que abarcaran todo el conjunto de datos, o sea, los datos estarán relativamente próximos a un punto semilla, pero las semillas estarán bien separadas unas de otras.

Astrahan propuso el siguiente algoritmo para ello: Para cada individuo se calcula la densidad, entendiendo por tal el número de casos que distan de una cierta distancia, digamos d1.

- Ordenar los casos por densidades y elegir aquel que tenga la mayor densidad como primer punto semilla. Elegir de forma sucesiva los puntos semilla en orden de densidad decreciente sujeto a que cada nueva semilla tenga al menos una distancia mínima, *d*2, con los otros puntos elegidos anteriormente.
- Continuar eligiendo semillas hasta que todos los casos que faltan tengan densidad cero, o
- sea, hay al menos una distancia *d*1 de cada punto a otro.
- En el caso de que, por este procedimiento, se produjera un exceso de puntos generados, se agruparán de forma jerárquica hasta que haya exactamente *K*. Por ejemplo, el método del centroide puede ser elegido para tal cuestión.

6. Ball y Hall (1967) proponen tomar el vector de medias de los datos como el primer punto semilla; posteriormente se seleccionan los puntos semilla examinando los individuos sucesivamente, aceptando uno de ellos como siguiente punto semilla siempre y cuando esté, por lo menos, a alguna distancia, *d*, de todos los puntos elegidos anteriormente Se continúa de esta forma hasta completar los *k* puntos deseados o el conjunto de datos se agota.

   Notemos que este método es tan simple que permite probar con diversos valores de la distancia *d* si los anteriormente empleados proporcionaran pocas semillas o examinaran una parte pequeña del conjunto de datos.

2.3.3. Elección de particiones iniciales.

En algunos métodos cluster, el énfasis del método recae en generar una partición inicial de los individuos en *K* clusters exclusivos más que en encontrar un conjunto de puntos semilla. Algunos procedimientos para generar tales particiones son:

1. Para un conjunto de puntos semilla dado, se asigna cada caso al cluster construido sobre el punto semilla más próximo, (Forgy, 1965), permaneciendo los puntos semilla estacionarios durante la asignación. Con ello el conjunto de clusters

resultante es independiente de la secuencia en la cual los individuos han sido introducidos.

2. Dado un conjunto de puntos semilla, sea cada uno de ellos, inicialmente, un cluster unitario. A continuación se asigna cada individuo al cluster con el centroide más próximo. Tras asignarlo, se actualiza el centroide del cluster. Este método tiene una gran semejanza con el método descrito en el tema de métodos jerárquicos. Al igual que en el método del centroide, los clusters pueden irse moviendo, por lo que la distancia entre un individuo y un centroide puede ir variando durante el proceso. Además, el conjunto de clusters resultante es independiente del orden en el que los individuos fueron asignados.
3. Emplear un método jerárquico para producir una partición inicial idónea. Wolfe (1970) emplea el método de Ward para proporcionar un conjunto inicial de clusters para su algoritmo.

2.4. Métodos que fijan el número de clusters.

A continuación veremos varios de estos métodos siguiendo el problema básico de ordenar los individuos en un número fijo de clusters de tal forma que cada individuo pertenezca a un solo cluster. Asimismo plantearemos algunas variantes de estos procedimientos.

2.4.1. Método de Forgy y variante de Jancey.

Forgy (1965), sugiere un algoritmo simple consistente en la siguiente secuencia de pasos:

1. Comenzar con una configuración inicial. Ir al paso segundo si se comienza con un conjunto de puntos semilla. Ir al paso tercero si se comienza con una partición de los casos.
2. Colocar cada individuo en el cluster con la semilla más próxima. Las semillas permanecen fijas para cada ciclo completo que recorra el conjunto de datos.
3. Calcular los nuevos puntos semilla como los centroides de los clusters.
4. Alternar los pasos segundo y tercero hasta que el proceso converja, o sea, continuar hasta que ningún individuo cambie de cluster en el paso segundo.

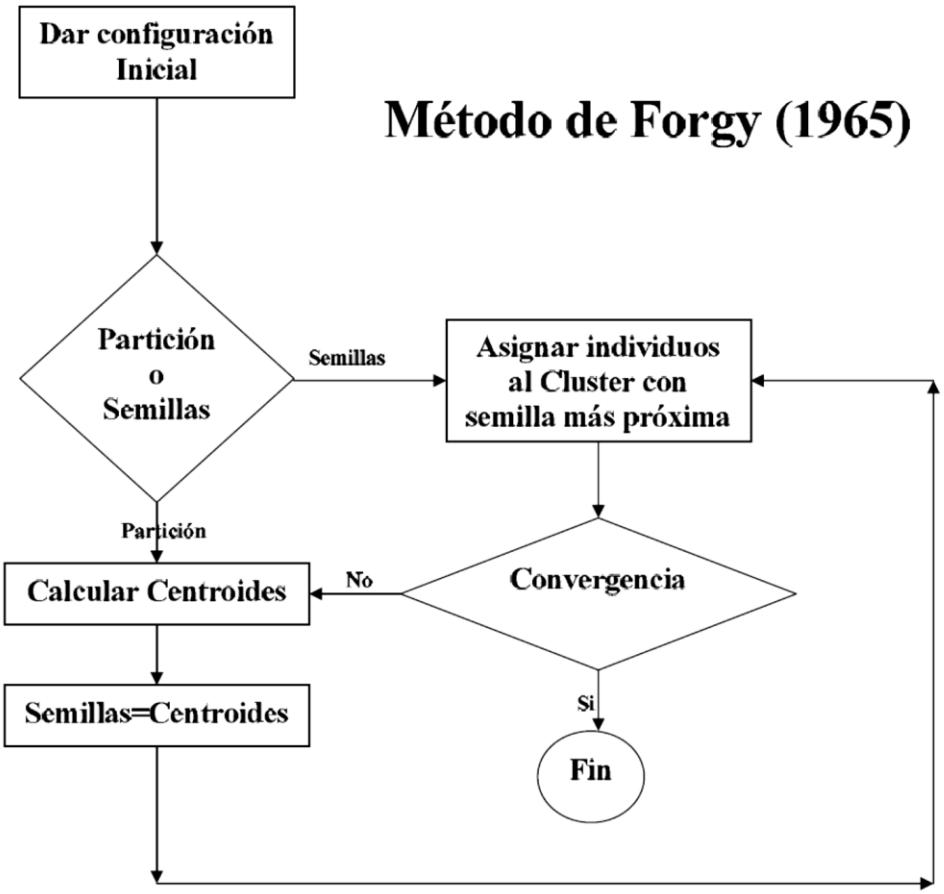

Figura 9: Diagrama de fujo - Método de Forgy

FUENTE: Elaboración propia

2.5. Diferencias entre Cluster Jerárquico y Cluster No Jerárquico

2.5.1. Clúster jerárquicos

- No requieren un conocimiento a priori del número de clúster o de la partición de partida.
- Los jerárquicos se usan a menudo con fines exploratorios y la solución resultante se utiliza en los no jerárquicos para afinar la solución.
- Ambas técnicas podrían verse como métodos complementarios y no como competitivos.

2.5.2. Clúster no jerárquicos

- Necesitan conocimiento previo del número de cluster.
- Hemos de identificar los centros de los cluster antes de que la técnica pueda proceder con las observaciones.
- Los algoritmos son muy sensibles a las particiones iniciales.

CAPITULO III: CLUSTERS BASADOS EN PARTICIONES

3.1. Métodos de Particionamiento

El objetivo del cluster particional es la de obtener una partición de los objetos en grupos o clusters de tal forma que todos los objetos pertenezcan a alguno de los k clusters posibles y que por otra parte los clusters sean disjuntos.

Si denotamos por $O = \{O_1,....,O_N\}$ al conjunto de N objetos, se trata de dividir O en k grupos o clusters, *Cl1*,...., Cl_k de tal forma que:

$$\bigcup_{j=1}^{k} Cl_j = \mathcal{O}$$

$$Cl_j \cap Cl_i = \emptyset \text{ para } i \neq j$$

Entre los algoritmos de partición se encuentran:

- K-medoides
- PAM
- CLARA
- CLARANS

A continuación se mostrará la definición de cada algoritmo dando un mayor énfasis en CLARANS

3.1.1. K-MEDOIDES (Clúster basado en particiones)

- Este algoritmo es efectivo debido a que es invariable frente a los valores atípicos.
- A su vez no depende del orden en que se examinan los puntos de datos.
- El centro del cluster es parte del conjunto de datos, a diferencia de k-means donde el centro del cluster es basado en el centro de gravedad. Los experimentos muestran que los grandes conjuntos de datos se manejan de manera eficiente.

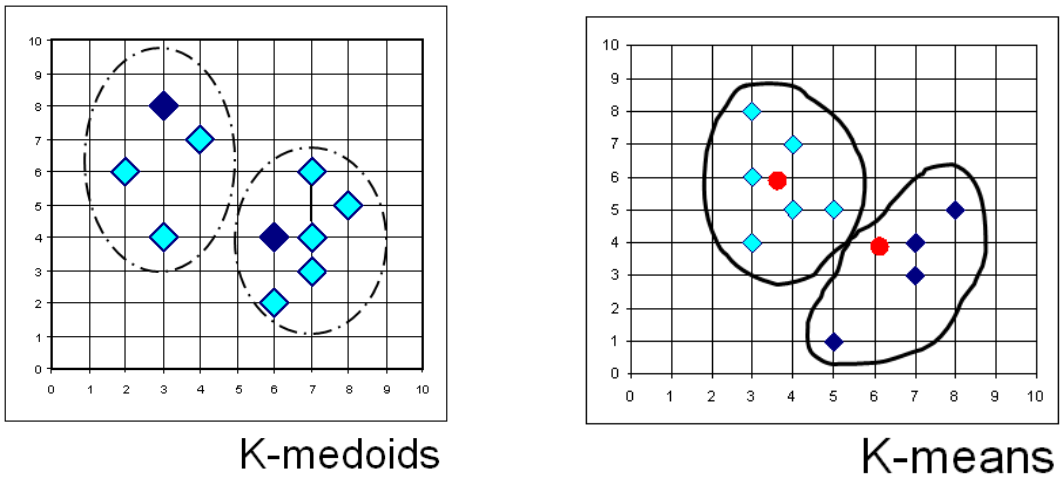

Figura 10: Esquema de agrupamiento no jerárquico

FUENTE: Raymond T. Ng, Jiawei Han (2015)

3.1.2. PAM (Partitioning Around Medoids)

PAM (Partitioning Around Medoids o en español, Partición por Medoides) fue desarrollado por Kaufman y Rousseeuw.

Procedimiento:

Para hallar *k* conglomerados, en enfoque de PAM se basa en determinar un objeto representativo para cada conglomerado. Este objeto representativo, llamado *medoide*, es reconocido como el objeto mejor ubicado cerca de la parte central del conglomerado.

Una vez que los medoides han sido seleccionados, cada objeto no seleccionado es agrupado con el medoid con el que guarda más similitudes.

De manera más precisa, si O_j es un objeto no seleccionado y O_m es un medoide (seleccionado), decimos que O_j pertenecerá al conglomerado representado por O_m.

Si $d(O_j; O_m) = \min_{Oe} d(O_j; O_e)$, donde la notación min denota el mínimo entre todos los medoide O_e y la notación $d(O_1; O_2)$ denota la disparidad o distancia entre los objetos O_1 y O_2.

Todos los valores de disparidad están dados como entradas a PAM.
Finalmente, la calidad de las conglomeraciones, es medida por un promedio de disparidad entre el objeto y el medoide de su conglomerado.

Para hallar los k-medoides, PAM empieza con una selección arbitraria de *k* objetos. Luego, en cada paso, se realiza un intercambio entre el objeto seleccionado O_m y el no seleccionado O_p, siempre y cuando ese intercambio resulte en una mejora de la calidad del conglomerado. Antes que nos embarquemos en un análisis formal, se debe considerar un ejemplo simple.

Ejemplo:
Suponga que hay dos medoides: *A* y B.Y nosotros consideramos reemplazar *A* con un nuevo medoide *M*. Entonces, para poder hacer el reemplazo, se debe encontrar al medoide más cercano para todos los objetos *Y* que han estado originalmente en el conglomerado representado por *A*.

Existen dos casos. En el primer caso, *Y* se mueve al conglomerado representado por *B*, pero no al nuevo representado por *M*.
En el segundo caso, *Y* se mueve al nuevo conglomerado representado por *M*, y el conglomerado representado por *B* no es afectado.
Aparte de reconsiderar todos los objetos *Y* que están originalmente en el conglomerado *A*, también se debe considerar todos los objetos *Z* que originalmente están en el conglomerado *B*. Para poder hacer el reemplazo, *Z* puede quedarse en *B*, o puede moverse al nuevo conglomerado representado por *M*.
La Figura 11 ilustra los cuatro casos.

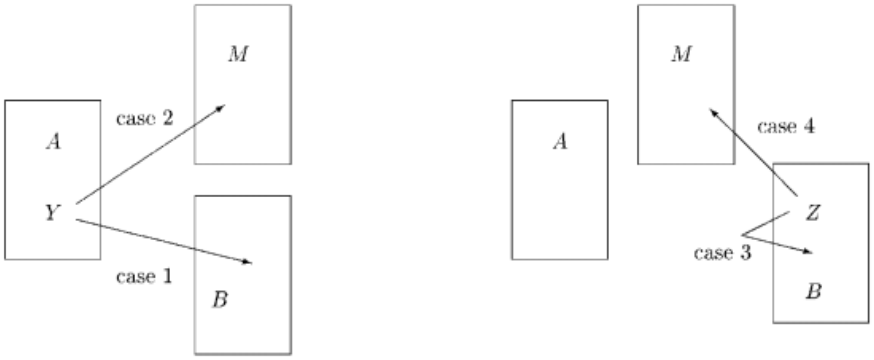

Figura 11: Cuatro casos para reemplazar A con M

Notación:

- O_m para denotar un medoide que va a ser reemplazado (ejemplo *A* en la Figura 11),
- *Op* para denotar el nuevo medoide a reemplazar O_m (ejemplo *M* en la Figura 11),
- O_j para denotar otros objetos no-medoides que pueden o no pueden necesitar ser movidos. (ejemplo *Y* y *Z* en la Figura 11), y
- $O_{j;2}$ para denotar el medoide más próximo a O_j sin *A* y *M* (ejemplo *B* en la Figura 11).

Ahora, para formalizar el efecto del intercambio entre O_m y O_p, PAM computa los costos C_{jmp} para todos los objetos no-medoides O_j. Dependiendo en cuál de los siguientes casos se encuentra O_j. Para este caso el C_{jmp} es definido de manera diferente,

Caso 1. Suponga que O_j actualmente pertenece al conglomerado representado por O_m. Incluso, sea O_j más similar a $O_{j;2}$ que a O_p, por ejemplo $d(O_j; O_p) \geq d(O_j; O_{j,2})$, donde $O_{j;2}$ es el segundo medoide más similar a O_j. Entonces, si O_m es reemplazado por O_p como medoide, O_j pertenecería al conglomerado representado por $O_{j;2}$ (como el Caso 1 en la Fig. 1). Por tanto, el costo del intercambio con respecto a O_j:

$$C_{jmp} = d(O_j; O_{j,2}), - d(O_j; O_{m)}.$$

Esta ecuación siempre resulta en un C_{jmp} no-negativo, indicando que se incurre en un costo no-negativo al reemplazar O_m por O_p.

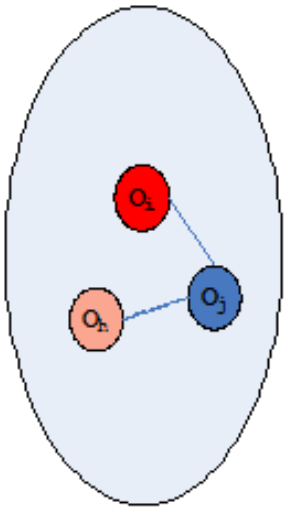

Figura 12: Esquema de partición del Algoritmo PAM

FUENTE: Jiawei Han, R. (2015)

Caso 2. O_j actualmente pertenece al conglomerado representado por O_m. Pero, esta vez, O_j es menos similar a $O_{j,2}$ que a O_p, e.j., $d(O_j; O_p) < d(O_j; O_{j,2})$. Entonces, si O_m es reemplazado por O_p, O_j pertenecería al conglomerado representado por O_p

Entonces, el costo para O_j está dado por:

$$C_{jmp} = d(O_j; O_p) - d(O_j; O_m)$$

A diferencia del anterior, el C_{jmp} puede ser positivo o negativo, dependiendo de la similitud de O_j con O_m u O_p.

Caso 3. Supóngase que O_j actualmente pertenece a un conglomerado diferente al representado por O_m. Sea $O_{j,2}$ el objeto más representativo de ese conglomerado. Mas aun, sea O_j más similar a $O_{j,2}$ que a O_p. Entonces, aun si O_m es reemplazado por O_p, O_j permanecería en el conglomerado representado por $O_{j,2}$. Por tanto, el costo es:

$$C_{jmp} = 0.$$

Caso 4. O_j actualmente pertenece al conglomerado representado por $O_{j,2}$. Pero, O_j es menos similar a $O_{j,2}$ que a O_p. Por tanto, el reemplazar a O_m con O_p ocasionaría que O_j salte del conglomerado $O_{j,2}$ al conglomerado de O_p. Entonces el costo es:

$$C_{jmp} = d(O_j; O_p) - d(O_j; O_{j,2}),$$

y siempre es negativo. Al combinar los cuatro casos explicados arriba, el costo total de reemplazar a O_m con O_p está dado por:

$$TC_{mp} = \sum_j C_{jmp}.$$

A continuación se define el algoritmo PAM.

Algoritmo PAM

1. Seleccionar *k* objetos representativos arbitrariamente.
2. Ingresar el TC_{mp} para todos los pares de objetos O_m; O_p donde O_m ya se encuentre seleccionado, y O_p no.
3. Seleccionar el par (O_m; O_p) que corresponde a $min_{Om;Op}\, TC_{mp}$.
4. Si el mínimo TC_{mp} es negativo, reemplace O_m con O_p, y vuelva al paso 2.
   De lo contrario, para cada objeto no-seleccionado, se debe encontrar el objeto representativo más similar.

Los resultados experimentales muestran que PAM trabaja satisfactoriamente para bases de datos pequeñas (como por ejemplo 100 objetos en 5 conglomerados).
Pero, no es eficiente el manejar bases de datos medias o grandes. Esto no es de sorprender si realizamos un análisis complejo en PAM. En los pasos 2 y 3, existen en conjunto k(n-k) pares de O_m, O_p.

Para cada par, el computar TC_{mp} requiere el análisis de *(n-k)* objetos no seleccionados. Por tanto, los pasos 2 y 3 combinados es de $O(k(n-k)^2$.
Y esta es solo la complejidad de una iteración. Es obvio que PAM es muy caro para gran cantidad de valores de n y k. Este análisis motiva el desarrollo de CLARA.

3.1.3. CLARA

Diseñado por Kaufman and Rousseeuw para manejar bases de dato grandes, CLARA (Clustering Large Applications , en español Aplicación de Conglomerados Grandes),

Este algoritmo se basa en muestreo. Es decir en vez de encontrar objetos representativos por toda la base de datos, CLARA toma una muestra de la base de datos, aplica PAM a la muestra, y encuentra los medoides de la muestra.

El punto es que si la muestra es obtenida de una forma suficientemente aleatoria, los medoides de la muestra se aproximaran a los medoides de la base de datos entera. Para hallar mejores aproximaciones, CLARA obtiene muestras múltiples y brinda la mayor conglomeración como producto. Así, por precisión, la calidad de una aglomeración es medida en base al promedio de disparidad de todos los objetos de toda la base de datos, y no solo para los objetos en las muestras. Los experimentos reportados indicaron que cinco muestras de tamaño 40 + 2k dieron resultados satisfactorios.

Algoritmo CLARA

1. Para $i = 1$ al 5, repita los siguientes pasos:
2. Extraiga una muestra de $40 + 2k$, objetos aleatoriamente de la base de datos total, y aplique el algoritmo PAM para encontrar k medoides de la muestra.
3. Para cada objeto O_j en toda la base de datos, determinar cuál de los k-medoides es el más similar a O_j.
4. Calcule el promedio de disparidad del conglomerado obtenido en el paso previo.
5. Si este valor es menor al mínimo actual, use este valor como el mínimo actual, y retenga los k-medoides encontrados en el paso 2 como la mejor base de medoides encontrados hasta ahora.
6. Retorne al Paso 1 para iniciar la iteración.
7. Complementariamente a PAM, CLARA se aplica satisfactoriamente para bases de data grandes (como por ejemplo 1000 objetos en 10 conglomerados).

Cada iteración de PAM es $O(k(n-k)^2$. Pero, para CLARA, al aplicar PAM solo a las muestras, cada iteración es de $O(k(n-k)^2 + k(n-k))$. Esto explica porque CLARA es más eficiente que PAM para mayores valores de n.

Seleccionar K representativo de objetos de manera arbitriaria

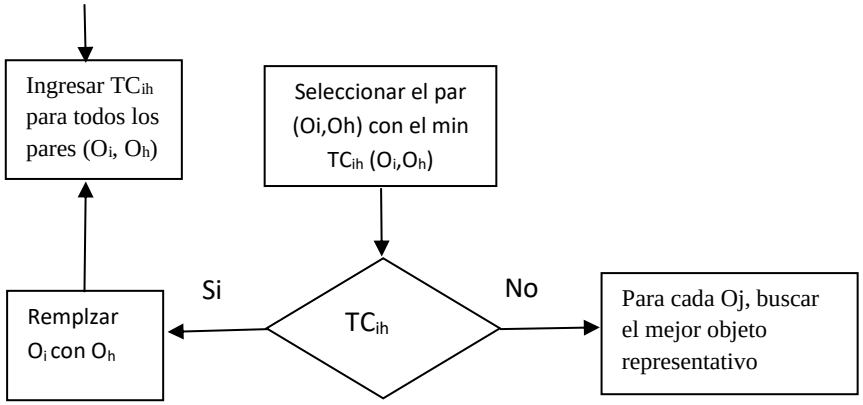

Figura 13: Diagrama de flujo del Algoritmo PAM

FUENTE: Jiawei Han, R. (2015)

3.1.4. Beneficios de PAM

- PAM encuentra los medoides en una muestra del conjunto de datos
- Si las muestras son suficientemente aleatorias, los medoides de la muestra se aproximan a los medoides del conjunto de datos.
- 5 muestras de tamaño 40 + 2k dan resultados satisfactorios
- Funciona bien para conjuntos de datos grandes (n = 1000, k = 10)

CAPITULO IV: ALGORITMO CLARANS

4.1. Definición:

CLARANS **(Clustering Large Applications based on RANdomized Search)** es un algoritmo de agrupamiento de individuos que poseen carácterísticas similares en base a un procedimiento en el cual se usan reglas de asociación basados en la cercanía (vecino mas cercano) a un medoide.

CLARANS (Clustering Large Applications based on RANdomized Search o en español Conglomeración de Grandes Aplicaciones basadas en Búsqueda Aleatoria).
Primero vamos se dará un marco grafico-teórico dentro del cual podamos comparar PAN y CLARA, y motivar el desarrollo de CLARANS. Luego, después de describir los detalles del algoritmo, vamos a presentar resultados que muestran como configurar CLARANS y que demuestran que CLARANS supera a CLARA y PAM en términos de eficiencia y efectividad.

4.2. Denotación:

- Un nodo en este gráfico, que lo denomina como $G_{n,k}$,
- En donde cada nodo está conformado por un conjunto de objetos, $\{O_{m_1},\ldots,O_{m_k}\}$, $O_{m_1},\ldots,O_{m_k} \in D$.
- **k** es el valor predefinido para elegir los k **medoides**.
- como resultado, los nodos en el gráfico son un conjunto de $\{\{O_{m_1},\ldots,O_{m_k}\} \mid O_{m_1},\ldots,O_{m_k} \in D\}$.
- Si dos nodos, $S_1 = \{O_{m_1},\ldots,O_{m_k}\}$, y $S_2 = \{O_{w_1},\ldots,O_{w_k}\}$ son vecinos, entonces $|S_1 \cap S_2| = k-1$

- Cada nodo en el $G_{n,k}$ representa un conjunto de medoides y el grupo relacionado con él.

El costo está relacionado con cada nodo, donde este costo es la distancia total entre cualquier objeto y el medóide representante de su grupo. La diferencia de costos de dos vecinos se puede calcular con la función de medida de costo introducida en el algoritmo PAM.

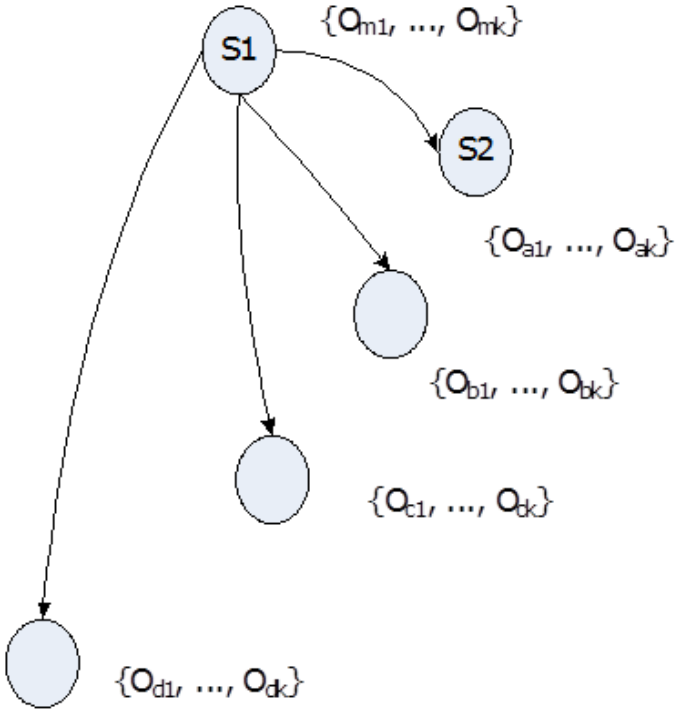

Figura 14: Esquema de partición del algoritmo CLARANS

FUENTE: Jiawei Han, R. (2015)

4.3. Explicación del proceso CLARANS

CLARANS tiene dos parámetros: el número máximo de vecinos examinados (maxNeighbor) y el número de mínimos locales obtenidos (numLocal).

Cuanto más alto es el valor de maxNeighbor, más cercano será CLARANS a PAM, y más largo es cada búsqueda de un mínimo local. Pero la calidad de tales mínimos locales es mayor y se necesitan menos mínimos locales.

El proceso que realiza este algoritmo es encontrar una muestra con una cierta aleatoriedad en cada paso de la búsqueda. El agrupamiento obtenido después de sustituirlo a un solo medoide se denomina el vecino del agrupamiento actual. Si en el camino el objeto (individuo) encuentra un mejor vecino, CLARANS lo mueve al nodo del vecino y el proceso comienza de nuevo; si ya no lo encuentra entonces el agrupamiento actual para y se produce un óptimo local (Cluster).

Si se encuentra en el grado óptimo local, CLARANS comienza con un nuevo nodo aleatoriamente seleccionado en búsqueda de un nuevo grado óptimo local [Ng & Han, 2002].

En la figura 13, un nodo está representado por un conjunto de k objetos seleccionados como mediodes (medianas). Dos nodos son vecinos si sus conjuntos difieren por un solo objeto. En cada iteración, CLARANS considera un conjunto de nodos vecinos aleatoriamente elegidos como candidatos de nuevas medianas.

Nos trasladaremos al nodo vecino si el vecino es una mejor opción para las madres. De lo contrario, permanece como un óptimo local. Todo el proceso se repite varias veces para encontrar mejor opción.

4.4. Paso a paso del Algoritmo CLARANS

- 1. Parámetros de ingreso *numlocal* y *maxneighbor*.
- Iniciar i en 1, y *mincost* como un numero grande.

  2. Establezca current a un nodo arbitrario en $G_{n,k}$.

  3. Establezca j como 1.

  4. Considere un vecino aleatorio S de current, y basado en 5, calcular el costo diferencial de dos nodos.

  5. Si S tiene un menor costo, establezca current a S, y regrese al paso 3.

  6. De lo contrario, incremente j en 1. Si j ≤ *maxneighbor*, vaya al paso 4.

  7. De lo contrario, cuando j > *maxneighbor*, compare el costo del current con *mincost*. Si el anterior es menor que *mincost*, establezca *mincost* al costo del current y seleccione *bestnode* a current.

  8. Incremente i en 1. Si i > *numlocal*, resulta en *bestnode* y halt. De lo contrario regrese al paso 2.

- Los pasos de 3 al 6 descritos anteriormente buscan nodos con costos menores progresivamente. Pero, si el nodo actual ya ha sido comparado con el máximo número de vecinos del nodos (especificado por *maxneighbr*) y resulta aun el costo menos, el nodo actual es declarado como el mínimo "local". Luego, en el paso 7, el costo de este mínimo local es comparado con el menor costo obtenido hasta entonces. Luego, el algoritmo CLARANS se repite para buscar otro mínimo local, hasta que el *numlocal* de ellos sea encontrado.
- Como se ha demostrado en la parte de arriba, CLARANS tiene dos parámetros: el máximo número de vecinos examinados (*maxneighbor*) y el número de mínima local obtenido (*minlocal*). A mayor valor de *maxneighbor,* CLARANS será más cercano a PAM, y mayor será el tiempo de cada búsqueda de una mínima local. Pero la calidad de tal mínima local será mayor y menor la cantidad de mínimas locales que deben ser encontradas. Como muchas aplicaciones de búsqueda aleatoria, nosotros nos basamos en experimentos para determinar los valores apropiados para estos tres parámetros.

A continuación en la figura 14 se muestra el diagrama de flujo del algoritmo Clarans.

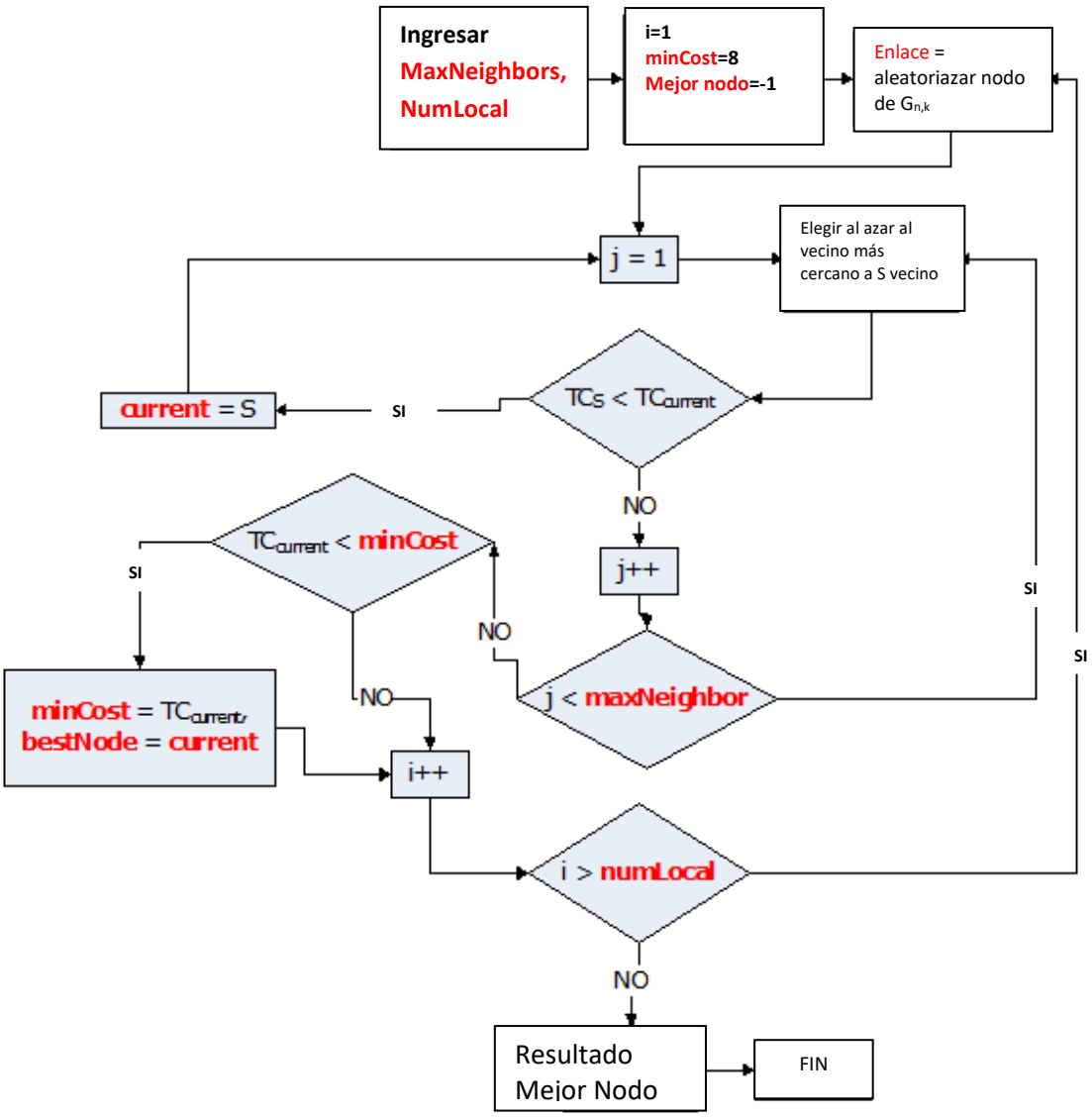

Figura 15: Diagrama de flujo del algoritmo CLARANS

FUENTE: Jiawei Han, R. (2002)

4.5. Beneficios de algoritmo CLARANS:

- El procedimiento de este algoritmo es predominantemente en dos dimensiones, sin embargo el algoritmo CLARANS funciona de la misma manera para conjuntos de datos de dimensiones superiores, debido a que CLARANS se basa en la búsqueda aleatoria y no utiliza ninguna estructura auxiliar y se ve menos afectado por el aumento de la dimensionalidad.
- CLARANS, es una técnica de búsqueda local, es decir no tienen ningún requisito sobre la naturaleza de la función de distancia.

- CLARANS es más general y soporta objetos poligonales. Una parte considerable de este documento se dedica a la manipulación eficaz de objetos poligonales.
- CLARANS es una técnica de clustering de memoria principal, mientras que muchas de las técnicas antes mencionadas están diseñadas para aplicaciones de clustering fuera del núcleo. (Vijaya R. Sagvekar, 2013).

CAPITULO IV: EJEMPLO DE LA METODOLOGÍA PARA EL ANALISIS CLUSTER USANDO EL ALGORITMO CLARANS

5.1. Ejemplo práctico aplicando la metodología

Se tiene un conjunto de notas de 5 alumnos, en donde se quiere agrupar a los alumnos según sus resultados.

Min costo 1

Mejor nodo = 0

datos originales

S0

i	Nota
1	10
2	12
3	15
4	18
5	18

Partición : Muestras con reemplazo

S1

i	Nota
1	10
1	10
2	12
5	18
4	18

S2

i	Nota
2	12
3	15
4	18
5	18
5	18

S3

i	Nota
2	12
2	12
3	15
5	18
1	18

S4

i	Nota
1	18
1	10
3	15
4	18
5	18

(medianas) **medoides** 15 12 18 15 18

$TC = \sum_j C$jmp, donde TC Costo total de reemplazo

$C_{jmp} = d(O_j, O_p) - d(O_j, O_{j,2})$

Diferencia Abs

	s0	s1	s2	s3	s4
s0	0				
s1	3	0			
s2	3	6	0		
s3	0	3	3	0	
s4	3	6	0	3	0

Nodo madre s4y s2

	S0∩S3	d	d
2	12	3	3
3	15	0	0
5	18	3	3
1	10	5	5

	S4∩S2	d	d
4	18	0	0
5	18	0	0

Nodo hijo

		dif
2	12	0
3	15	3
1	10	2
mediana	12	

		dif
4	18	0
5	18	0
mediana	18	

Tal como se puede observar se formaron 2 cluster el cual agrupa a los individuos 1,2 y 3 en un nodo hijo con un medoide 12 y en el otro nodo hijo 2 lo conforman los individuos 4 y 5 con un medoide 18.

Se crearon 2 nodos, En donde cada uno con respecto a su medoide tienen diferencias, Si la diferencia menor a 1 el proceso termina, tal como sucede en el nodo 2.
Con el nodo 1, volveremos a realizaremos el paso 3. Generando la iteración.

nodo hijo		S'1		S'2		S'3		S'4	
2	12	2	12	3	15	1	10	2	12
3	15	2	12	3	15	2	12	1	10
1	10	3	15	1	10	3	15	1	10
medoides	12		12		15		12		10

Diferencia Abs

	S'0	S'1	S'2	S'3	S'4
s0	0				
s1	0	0			
s2	3	3	0		
s3	0	0	3	0	
s4	2	2	5	2	0

	S0∩S
	1
2	12
3	15
mediana	13.5

	S3∩S
	1
1	10
2	12
3	15
median a	12

	S3∩S
	0
1	10
2	12
3	15
median a	12

Por la poca cantidad de datos el resultado será el mismo el siguiente cluster será conformado por los individuos 1, 2 y 3

Conclusión:

Realizando el algoritmo CLARANS, se tienen que el cluster 1 es conformado por los indivudos 1,2 y 3 y el cluster 2 por los individuos 4 y 5. Se puede apreciar como clarans aleatoriza cada proceso y evalúa el costo total de reemplazo moviendo al individuo entre un cluster a otro hasta obtener optimo local. El proceso culminó hasta que todos los individuos tengan un costo menor frente a su óptimo local, posterior a esto se

5.2. Ejemplo en R : PACIENTES CON DIABETES

Para este ejemplo se usaron las librerías dectools, qtcat del software estadístico R.

Se tiene un conjunto de datos entre la comparación de la comparación entre genotipos extraídos de un conjunto de pacientes con Diabetes. En donde se desean agrupar para clasificarlos y estudiarlos de manera particular. Se cuenta con una data de 1000 Genomas según las siguientes variables:

- SNP
- código
- Riesgo
- Frecuencia
- Locus
- Cromosoma
- Diabetes tipo 1

```
install.packages("devtools")
library(devtools)
library(qtcat)

gfile <- system.file("extdata/snpdata.csv", package =
"qtcat")
snp <- read.snpData(gfile, sep = ",")
clust <- clarans(snp, 3)
clust
$clusters

$medoids
loci260 loci392  loci66
```

```
    260     392      66

$objective
[1] 0.7215901

$all.objectives
 [1] 0.7300984 0.7215901 0.7265908 0.7231079 0.7353707
0.7282308 0.7338746 0.7354295 0.7229498 0.7235969

attr(,"class")
[1] "k-medoids"

(ver Anexos para los resultados de los individuos ubicados en
cada cluster)
```

https://rdrr.io/github/QTCAT/qtcat/man/clarans.html
https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781783982103/5/ch05lvl1sec54/clarans

5.3. Resultados de ejemplo usando Clarans en R.

En el siguiente ejemplo se puede apreciar que se tienen los siguientes medoides:

$medoids
loci260 loci392 loci66
260 392 66

Los medoides resultantess son

260, 392 y 66. En donde se agrupará a todos los individuos alrededor de estos valores.

El software R, mediante la función CLARANS, arrojó los resultados de:

- MEDOIDES:

$medoids
loci260 loci392 loci66
260 392 66

- OBJETIVOS

$objective
[1] 0.7215901

$all.objectives
[1] 0.7300984 0.7215901 0.7265908 0.7231079 0.7353707 0.7282308 0.7338746 0.7354295
0.7229498 0.7235969

- INDIVIDUOS AGRUPADOS EN CADA CLUSTER. Ver Anexo

Del total de individuos que se estudió, el 27% pertenecen al medoide 1 (vecinos cercanos al loci 260), seguido del 27.5% de individuos que se lograron agrupars al medoide 2 (loci 392) y por último el 45.6% de individuos se agruparon en el medoide 3, es decir de manera cercana al loci 66. Lo que CLARANS hace es tomar de un conjunto de datos, a ciertos referentes como por ejemplo individuos representativos de cada clúster y ubicarlos como el más representativo del conglomerado. A su vez CLARANS obtuvo un 72.15% de objetivo promedio en agrupamiento en cada proceso interno lo cual demuestra su efectividad al finalizar el proceso iterativo en cada nodo encontrado.

CONCLUSIONES

1. El algoritmo clarans extrajo a partir de una un base de datos, una muestra con cierta aleatoriedad, de donde se particionó ubicando a cada individuo dentro de un óptimo local en base al vecino más cercano a un medoide. Clarans demuestra en el ejemplo mostrado obtener un 72.15% de objetivo promedio en agrupamiento en cada proceso interno lo cual demuestra su efectividad al finalizar el proceso iterativo en cada nodo encontrado.

2. El algoritmo CLARANS agrupó individuos teniendo criterio ubicarlo a un medoide vecino más cercano, el cual si en el proceso el individuo encuentra un mejor vecino, CLARANS lo mueve al nodo del vecino y el proceso comienza de nuevo hasta que encuentre producir un óptimo local (Cluster).

3. El agrupamiento que genera CLARANS tiene como ventaja la aleatoriedad en cada paso de la búsqueda, tal como se vió en el ejemplo, el agrupamiento obtenido después de sustituir un solo medoide. El algoritmo también nos brinda un grado de objetivo, el cual es un porcentaje efectivo de agrupamiento. En donde CLARANS obtuvo un 72.15% de objetivo promedio en agrupamiento en cada proceso interno, demostrando su efectividad al finalizar el proceso iterativo en cada nodo encontrado.

REFERENCIAS BIBLIOGRÁFICAS

[1] Spatial Data Mining: Theory and Application (2016).
Springer, 308 páginas

[2] Charu C. Aggarwal & Chandan k. Reddy (Editors) (2014).
" Data Clustering: Algorithms and Applications"
CRC Press, 652 páginas

[3] L. Kaufman & P.J. Rousseeuw (2009).
" Finding Groups in Data: An Introduction to Cluster"
John Wiley & Sons, 342 páginas

[4] Ng R. T. y Han Jiawei (2009).
“CLARANS a method for clustering objects for spatial data mining”

[5] Raymond T. Ng, Jiawei Han Pavan Podila (2015).
"Efficient and Effective Clustering Methods for Spatial Data Mining"

[6] F B. Visauta Vinacua (1998).
“Análisis estadístico con SPSS para Windows, volumen II: Estadísitca multivariante”. McGraw-Hill, 2003 - 360 páginas

ANEXOS

1.-LIBRERIAS

install.packages("devtools")

library(devtools)

install_github("QTCAT/qtcat")

snp	Un conjunto de objetos de una matriz snpMatrix.
k	Un número entero positive en el que se especifica el npumero de cluster que serán formados
maxNeigbours	un entero positivo que especifica el número máximo de búsquedas aleatorias
nLocal	un entero positivo que especifica el número de ejecuciones de optimización.
mc.cores	un entero positivo para la cantidad de núcleos para computación paralela. Ver mclapply para más detalles.

2.-PROGRAMACION DE LA FUNCIÓN CLARANS EN EL SOFTWARE R

A continuación se detalla el código en R de la función CLARANS

```
clarans
function(snp, k, maxNeigbours = 100, nLocal = 10, mc.cores = 1)
{
  stopifnot(is(snp, "snpMatrix"))
  if (missing(k))
    stop("'k' must be specifid")
  if (k < 2L)
    stop("'k' must be at least two")
  # cluster optimisation by clarans in parallel
  clarans.i <- function(i, snp, k, maxNeigbours) {
    # cluster optimisation by clarans
    out <- corClarans(snp@snpData, k, maxNeigbours)
    out
  }
  out.nLocal <- mclapply(1L:nLocal, clarans.i,
                         snp, k, maxNeigbours,
                         mc.cores = mc.cores)
  opt.func <- function(i, snp) {snp[[i]][[3L]]}
  all.objectives <- sapply(1:nLocal, opt.func, out.nLocal)
  out.opt <- out.nLocal[[which.min(all.objectives)]]
  clusters <- out.opt[[1L]]
  names(clusters) <- colnames(snp)
  medoids <- out.opt[[2L]] + 1
  names(medoids) <- colnames(snp)[medoids]

  # output
  out <- list(clusters = clusters,
              medoids = medoids,
              objective = out.opt[[3L]],
              all.objectives = all.objectives)
  class(out) <- "k-medoids"
  out
}
```

3.-RESULTADOS DE AGRUPAMIENTO DE INDIVIDUOS

Se agruparon individuos en base a 3 medoides, que posteeriormente resultaron los Cluster.

```
 loci1  loci2  loci3  loci4  loci5  loci6  loci7  loci8  loci9 loci10 loci11 loci12 loci13 loci14 loci15
loci16 loci17 loci18 loci19 loci20
    3      2      3      3      3      3      1      3      1      3      1      3      3      3      3      3      1      3      3
3
 loci21 loci22 loci23 loci24 loci25 loci26 loci27 loci28 loci29 loci30 loci31 loci32 loci33 loci34
loci35 loci36 loci37 loci38 loci39 loci40
    3      3      3      2      3      3      3      3      3      3      3      2      1      3      3      3      3      3      3
3
 loci41 loci42 loci43 loci44 loci45 loci46 loci47 loci48 loci49 loci50 loci51 loci52 loci53 loci54
loci55 loci56 loci57 loci58 loci59 loci60
    1      3      3      3      2      2      3      3      2      1      3      3      3      3      3      2      3      2      2
3
 loci61 loci62 loci63 loci64 loci65 loci66 loci67 loci68 loci69 loci70 loci71 loci72 loci73 loci74
loci75 loci76 loci77 loci78 loci79 loci80
    3      3      3      3      3      3      3      3      3      3      3      3      3      3      2      3      3      3      3
3
 loci81 loci82 loci83 loci84 loci85 loci86 loci87 loci88 loci89 loci90 loci91 loci92 loci93 loci94
loci95 loci96 loci97 loci98 loci99 loci100
    1      3      3      1      1      2      3      3      3      3      3      3      2      3      3      2      2      2      1
2
loci101 loci102 loci103 loci104 loci105 loci106 loci107 loci108 loci109 loci110 loci111 loci112 loci113
loci114 loci115 loci116 loci117 loci118 loci119 loci120
    2      2      3      3      3      3      2      3      3      3      3      3      3      3      3      3      3      3      3
3
loci121 loci122 loci123 loci124 loci125 loci126 loci127 loci128 loci129 loci130 loci131 loci132 loci133
loci134 loci135 loci136 loci137 loci138 loci139 loci140
    1      3      3      3      3      3      3      3      2      3      3      3      3      3      1      3      3      3      3
3
loci141 loci142 loci143 loci144 loci145 loci146 loci147 loci148 loci149 loci150 loci151 loci152 loci153
loci154 loci155 loci156 loci157 loci158 loci159 loci160
    1      3      3      3      3      1      3      3      3      3      2      2      1      3      1      3      3      3      1
3
loci161 loci162 loci163 loci164 loci165 loci166 loci167 loci168 loci169 loci170 loci171 loci172 loci173
loci174 loci175 loci176 loci177 loci178 loci179 loci180
    3      2      3      3      3      3      2      3      3      1      3      3      3      1      3      3      3      1      2
3
loci181 loci182 loci183 loci184 loci185 loci186 loci187 loci188 loci189 loci190 loci191 loci192 loci193
loci194 loci195 loci196 loci197 loci198 loci199 loci200
    3      3      3      2      3      3      1      3      3      1      1      3      3      1      3      3      2      2      3
3
loci201 loci202 loci203 loci204 loci205 loci206 loci207 loci208 loci209 loci210 loci211 loci212 loci213
loci214 loci215 loci216 loci217 loci218 loci219 loci220
    3      3      2      1      2      3      3      3      2      2      2      2      2      2      3      1      2      2      1
2
loci221 loci222 loci223 loci224 loci225 loci226 loci227 loci228 loci229 loci230 loci231 loci232 loci233
loci234 loci235 loci236 loci237 loci238 loci239 loci240
    3      2      2      1      2      2      1      3      2      3      3      2      1      1      2      3      1      2      2
1
loci241 loci242 loci243 loci244 loci245 loci246 loci247 loci248 loci249 loci250 loci251 loci252 loci253
loci254 loci255 loci256 loci257 loci258 loci259 loci260
    1      2      2      2      1      1      1      1      2      2      2      1      1      1      1      3      1      2      1
1
```

loci261 loci262 loci263 loci264 loci265 loci266 loci267 loci268 loci269 loci270 loci271 loci272 loci273 loci274 loci275 loci276 loci277 loci278 loci279 loci280
3 1 2 1 1 1 1 1 2 1 1 1 3 3 1 1 2 1 1 1
loci281 loci282 loci283 loci284 loci285 loci286 loci287 loci288 loci289 loci290 loci291 loci292 loci293 loci294 loci295 loci296 loci297 loci298 loci299 loci300
1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 2 1
loci301 loci302 loci303 loci304 loci305 loci306 loci307 loci308 loci309 loci310 loci311 loci312 loci313 loci314 loci315 loci316 loci317 loci318 loci319 loci320
1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2 1 2 3 2
loci321 loci322 loci323 loci324 loci325 loci326 loci327 loci328 loci329 loci330 loci331 loci332 loci333 loci334 loci335 loci336 loci337 loci338 loci339 loci340
2 1 1 2 3 1 1 3 3 2 1 3 2 2 1 1 2 1 1 2
loci341 loci342 loci343 loci344 loci345 loci346 loci347 loci348 loci349 loci350 loci351 loci352 loci353 loci354 loci355 loci356 loci357 loci358 loci359 loci360
2 2 3 3 2 1 1 3 3 1 1 3 2 1 1 1 3 2 3 3
loci361 loci362 loci363 loci364 loci365 loci366 loci367 loci368 loci369 loci370 loci371 loci372 loci373 loci374 loci375 loci376 loci377 loci378 loci379 loci380
3 2 2 2 2 2 2 2 3 2 2 2 2 3 2 2 3 2 2 2
loci381 loci382 loci383 loci384 loci385 loci386 loci387 loci388 loci389 loci390 loci391 loci392 loci393 loci394 loci395 loci396 loci397
2 2 2 2 2 2 2 2 2 3 2 2 3 2 1 2 2

4.-CLARA VS OTROS ALGORITMOS

En el gráfico 8 se puede observar que :

- CLARANS supera a PAM y CLARA en términos de tiempo de ejecución y calidad de agrupamiento
- (n_2) para cada iteración

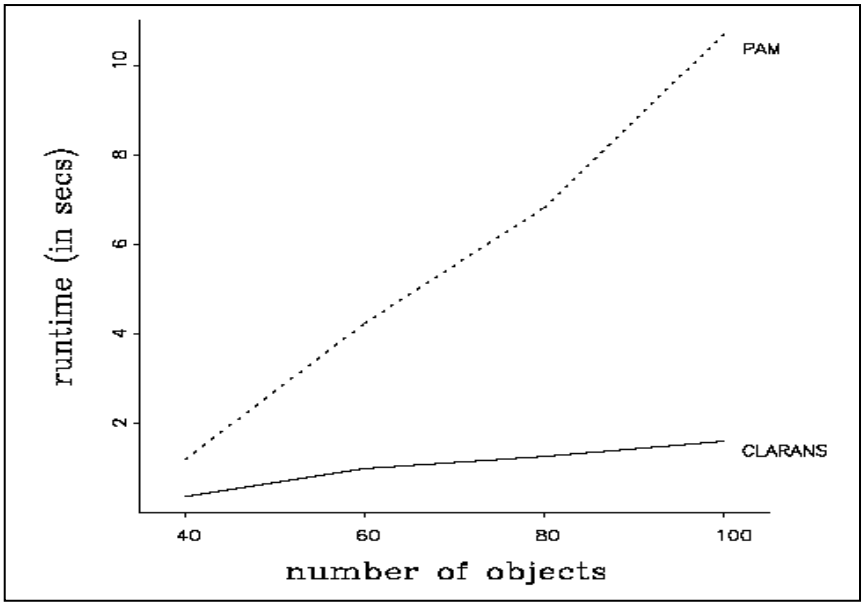

Figura 16: Clarans vs PAM en tiempo de ejecución

FUENTE: Jiawei Han, R. (2015)

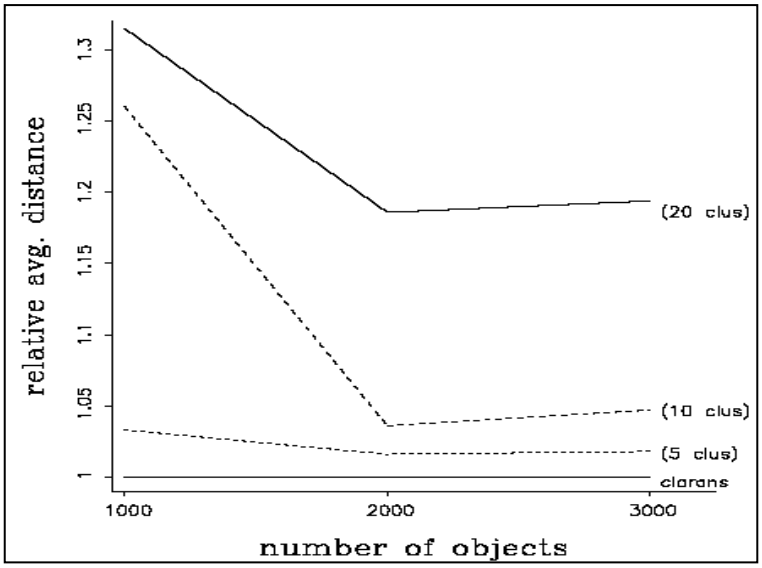

Figura 17: CLARA VS CLARANS

FUENTE: Jiawei Han, R. (2015)