DIPLOMADO EN BIG DATA PARA LA TOMA DE DECISIONES

Introducción al Modelamiento Estadístico

Parte 3: Métodos de Agrupación

3.1.- Introducción

Definición

`Clustering` o análisis de conglomerados es el proceso de particionar un conjunto de observaciones en subconjuntos.

Se busca que los objetos dentro de un mismo grupo sean lo más similares posible entre sí y lo más diferentes con los de otro.

Introducción

Gráficamente

(a) Original points.

(b) Two clusters.

(c) Four clusters.

(d) Six clusters.

Introducción

- Las agrupaciones son realizadas por algoritmos, por lo que es útil para detectar agrupamientos previamente desconocidos.
- Diferentes métodos de agrupamiento pueden llevar a diferentes grupos de los mismos datos.
- Un cluster o grupo de objetos pueden tratarse como pertenecientes a una clase implícita $\Longrightarrow$ clasificación automática.
- Aplicaciones: inteligencia de negocios, reconocimiento de imágenes, búsquedas Web.

Introducción

- En algunos contextos, esta técnica se conoce como *segmentación de datos.*
- También puede utilizarse para detectar *outliers:* fraudes bancarios, actividades criminales en comercio electrónico.

Introducción

Para que un algoritmo sea útil como herramienta necesita cumplir con los siguientes requisitos:

- Escalabilidad: útiles en grandes bases de datos (millones o billones de objetos).
- Capaz de manejar diferentes tipos de atributos: numéricos, binarios, nominales, ordinales o mezclas de ellos.
- Encontrar clusters con cualquier forma: muchos algoritmos se basan en distancias que sólo detectan clusters esféricos.
- Requisitos sobre conocimiento del contexto para determinar los parám de entrada.

Introducción

- Robustos frente a la presencia de datos “ruidosos”.
- Métodos incrementares y no sensibles al orden de llegada de los registros.
- Capaz de agrupar datos con alta dimensionalidad.
- Realizar clustering sujeto a ciertas restricciones.
- Interpretable, comprensible y útil.

Introducción

- Criterio de partición: hay métodos en los que no hay jerarquía entre los clusters y otros que separan los datos jerárquicamente.
- Separación de los clusters: separación de los objetos en grupos mutuamente excluyentes o no.
- Medidas de similaridad: algoritmos basados en distancias vs. algoritmos basados en densidades o métodos continuos.
- Búsqueda de clusters en subespacios de los datos.

3.2.- Métodos de Partición

Métodos de Partición en Número Fijo

- Es el método más simple y fundamental de análisis de conglomerados. Organiza los datos en varios grupos exclusivos, cuya cantidad es fijada de antemano.

- Dado un conjunto de datos, D, de n objetos y k el número de grupos a formar, el algoritmo distribuye los objetos de D en los grupos, $C_1, \ldots, C_k$, donde $C_i \subset D$ y $C_i \cap C_j = \emptyset$, para $1 \leq i, j \leq k$. De esta manera, cada partición representa un grupo o `cluster`.

Métodos de Partición en Número Fijo

- Los grupos se forman optimizando un criterio de particionamiento, como puede ser la función de disimilaridad basada en distancia, tal que los objetos dentro de un grupo son similares y diferentes a los de otros, en términos de las variables que los describen.

- Se utiliza una función para evaluar la calidad de la partición, que pretende lograr alta similaridad intra-cluster (dentro del grupo) y baja similaridad inter-clusters (entre grupos).

3.2.1.- K-MEANS

K-Means

- K-Means: Utiliza el centroide de un cluster, C_i, para representarlo.
- Conceptualmente, el centroide corresponde es el centro de la distribución del grupo.
- Este centroide puede definirse de varias maneras.
- Se calcula la distancia entre un objeto $\boldsymbol{p} \in C_i$ y el representante del grupo, $\boldsymbol{c}_i \Longrightarrow dist(\boldsymbol{p}, \boldsymbol{c}_i)$.

K-Means

- La calidad del cluster C_i se puede medir usando la *variación dentro del cluster*, que corresponde a la suma de los errores cuadráticos entre los objetos en C_i y el centroide $\boldsymbol{c}_i$ definida como:

$$E = \sum_{i=1}^{k} \sum_{\boldsymbol{p} \in C_i} dist(\boldsymbol{p}, \boldsymbol{c}_i)^2$$

- Esta función objetivo trata de asegurarse que los k grupos sean los más compactos y separados posible.

K-Means

Input:

- D: datos con d tuplas clasificadas.
- k: número de grupos o clusters.

Output: un conjunto de k grupos.

Paso 1: Seleccionar aleatoriamente k objetos de D como centros de grupos iniciales.

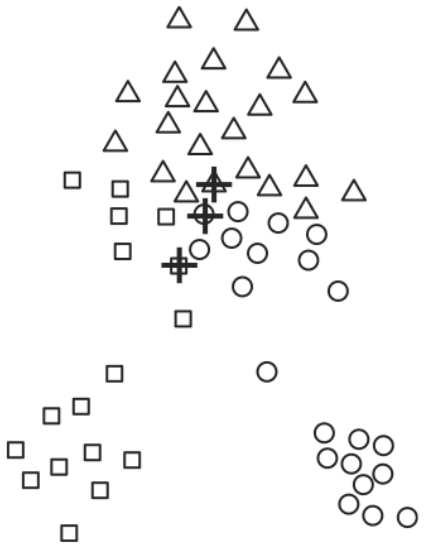
Paso 2: Asignar cada objeto al grupo al cual es más similar, basado en la distancia Euclidiana entre él y el promedio del grupo.

Paso 3: Actualizar las medias de los grupos y volver al paso 2.

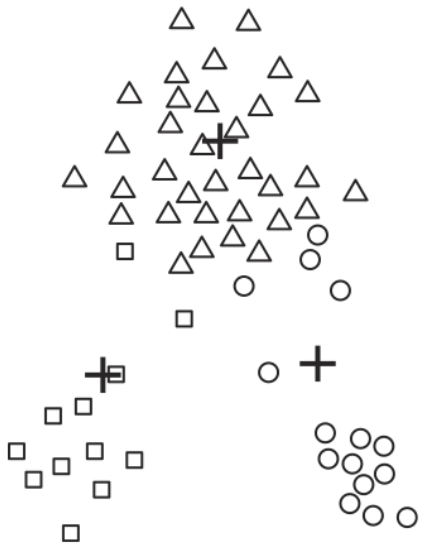
Paso 4: Se para cuando la asignación es estable.

K-Means

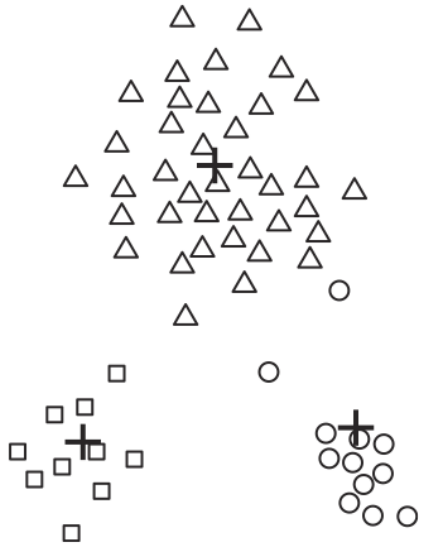
Gráficamente



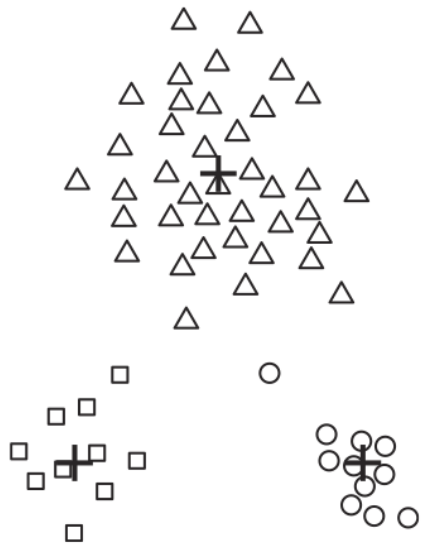
(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.



(d) Iteration 4.

K-Means: Ejemplo

- Utilicemos nuevamente la base de datos `iris` de `R`, pero esta vez ignoraremos la clasificación sobre la especie. y nos centraremos en las variables asociadas al sépalo.

- Función `kmeans` de R implemeta el algoritmo.

K-Means: Ejemplo

```
datos <- iris

X.dat <- datos[,1:2]

### Supongamos 8 grupos

X.kmeans <- kmeans(X.dat, 8)
X.kmeans
```

K-Means

- El algoritmo K-Means no está garantizado que converja a un óptimo global y usualmente termina en un óptimo local.
- Esto depende de la selección aleatoria inicial de los centros iniciales.
- Se recomienda realizar varias veces el algoritmo con diferentes centros de grupos iniciales.
- El método es relativamente escalable y eficiente en el procesamiento de grandes bases de datos.

K-Means

- Este método es útil sólo cuando es posible calcular la media de un grupo de objetos.
- En presencia de variables nominales $\Longrightarrow$ k-modas.
- k-medias + k-modas $\Longrightarrow$ datos con atributos mixtos.

K-Means: Desventajas

- Necesidad de especificar k anticipadamente.
  Propuesta: rango de valores de k y comparar resultados para determinar el mejor k.
- No sirve cuando se quieren encontrar grupos de forma no convexa o de tamaños muy diferentes.
- Sensible a observaciones extremas o outliers.
- No funciona bien con bases de datos muy grandes (no es muy escalable).

3.2.2.- K-MEDOIDS

K-Medoids

- Los medoides son objetos “representativos” de un conjunto de datos o un grupo cuya disimilaridad promedio a todos los objetos en el grupo, es mínima.
- Es un concepto similar a la media o centroide, pero los medoides siempre son miembros del conjunto de datos.
- Puede ocurrir que exista más de un medoide, como es el caso de la mediana.

K-Medoids

- Consideremos 6 puntos en un espacio unidimensional con los siguientes valores:

$$1, 2, 3, 8, 9, 10, 25$$

- K-Means es altamente sensible a la presencia de outliers.

K-Medoids

- En vez de tomar el valor promedio de los objetos de un grupo como punto de referencia, se pueden tomar los mismos objetos que lo conforman como representantes.
- Los restantes objetos son asignados de acuerdo a su similitud con los representantes.
- El método de partición se realiza minimizando la suma de las disimilitudes entre cada objeto $\boldsymbol{p}$ y su correspondiente representante.

K-Medoids

- Se utiliza el criterio de error absoluto, definido como:

$$E = \sum_{i=1}^{k} \sum_{\boldsymbol{p} \in C_i} dist(\boldsymbol{p}, \boldsymbol{o}_i)$$

donde $\boldsymbol{o}_i$ es el objeto representativo de C_i.

- Ésta es la base del método K-Medoids, que agrupa n objetos in k grupos minimizando el error absoluto definido recientemente.

- Partitioning around medoids (PAM): Implementación popular. Funció `pam` de la libería `cluster` implemeta el algoritmo.

K-Medoids: Ejemplo

```
head(iris)
iris.use<-iris[,1:4]

iris.pam<-pam(iris.use,3)
iris.pam

names(iris.pam)
table(iris.pam$clustering,iris[,5])

plot(iris.pam,ask=T)
iris.pam$silinfo$avg.width
```

K-Medoids: Ventajas y Desventajas

- Es más robusto que K-Means en presencia de outliers.
- Es más complejo y por lo tanto, computacionalmente más costoso.
- Al igual que K-Means, se debe especificar el valor de k previamente.
- No funciona bien con bases de datos muy grandes (no es muy escalable).

3.3.- Métodos Jerárquicos

Métodos Jerárquicos

- En algunas situaciones, queremos particionar los datos en grupos en diferentes niveles (jerarquía).
- Los métodos de agrupamiento jerárquicos, aglomeran los objetos en jerarquías o “árboles” de grupos.
- Es una herramienta útil de inspección y visualización.
- Ejemplo: organizar trabajadores en gerentes, ejecutivos, analistas y administrativos. Luego, los administrativos pueden dividirse en seniors, juniors y trainees.

Métodos Jerárquicos

- En algunos casos, los datos pueden llevar una estructura jerárquica subyacente que queremos descubrir. Por ejemplo, podemos descubrir la jerarquía de trabajadores en cuanto a sueldo.
- Estos métodos pueden ser aglomerativos o divisivos.
- En ambos casos, es crucial la agrupación o separación de objetos, ya que en la siguiente etapa se trabaja sobre lo realizado previamente. No se puede deshacer ni intercambiar nada después.

Métodos Jerárquicos: Aglomerativo vs Divisivo

La diferencia radica en si la descomposición se forma “agrupando” o “separando” (bottom-up / top-down).

Método agrupamiento jerárquico aglomerativo

Utiliza la estrategia “hacia arriba” (bottom-up.)

Parte con cada objeto siendo un grupo e iterativamente los agrupa en conglomerados de mayor tamaño, hasta que todos los objetos están en un solo grupo o se cumple cierto criterio.

El grupo singular se convierte en la raíz jerárquica. Para agrupar, se busca los 2 grupos que están más cercanos y los combina para formar uno nuevo. Como 2 grupos se juntan por interación, donde cada uno tiene al menos 1 objeto, este método requiere a lo más n iteraciones.

Métodos Jerárquicos: Aglomerativo vs Divisivo

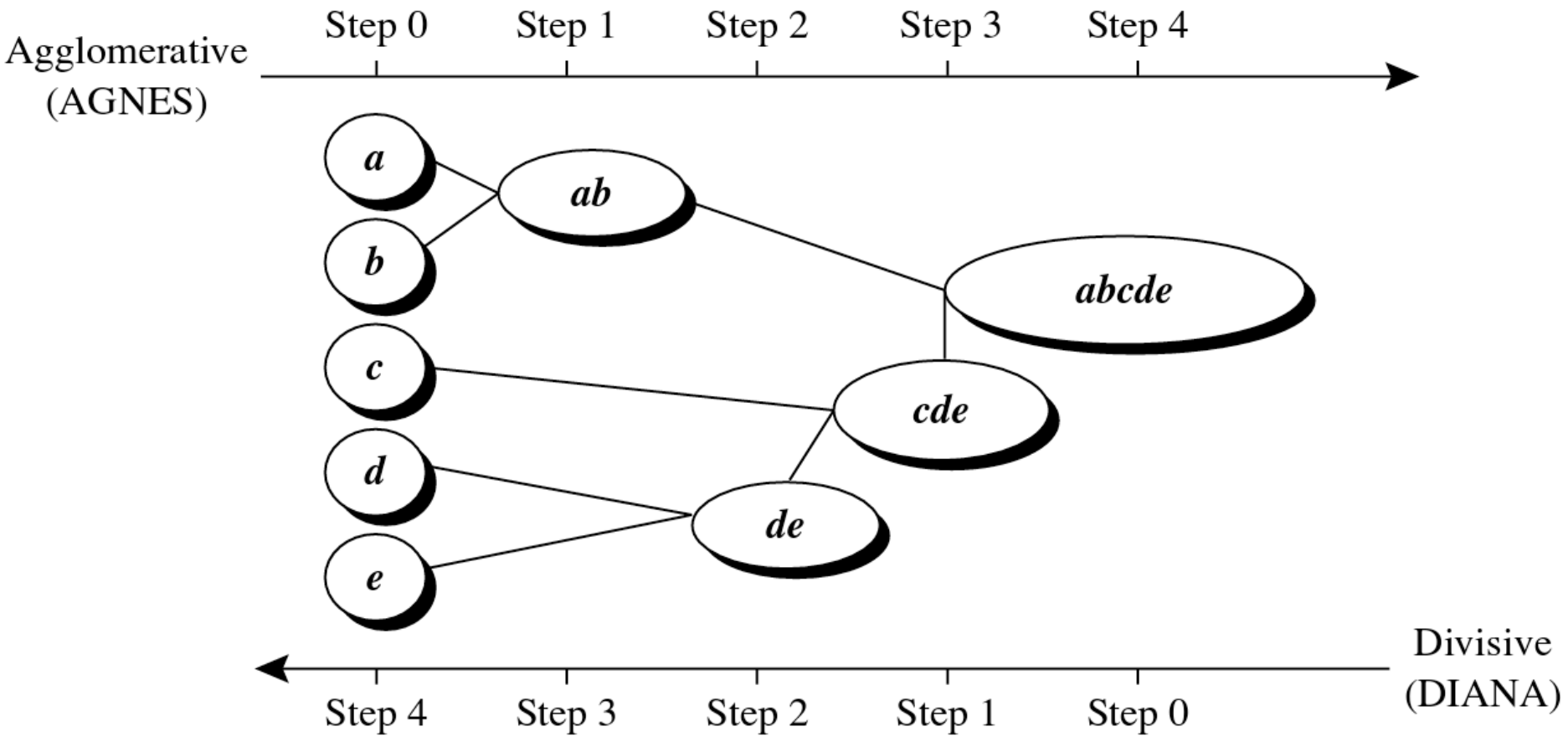
Método agrupamiento jerárquico divisivo

Utiliza la estrategia "hacia abajo" `(top-down.)`

Parte con todos los objetos en un grupo, que corresponde a la raíz jerárquica. Luego divide la raíz en varios subgrupos, y particiona recursivamente estos subgrupos en grupos más pequeños. El proceso continúa hasta que grupo en el nivel más bajo es suficientemente coherente, ya sea con un objeto o con varios que suficientemente parecidos.

Métodos Jerárquicos: Aglomerativo vs Divisivo

Gráficamente



Métodos Jerárquicos

- Función `(hclust)` implementa un método de conglomerados jerárqu aglomerativo.
- La librería `cluster` tiene la función `agnes` que hace exactamente lo mismo.
- La librería `cluster` tiene la función `diana` que implementa un método de conglomerados divisivo.

Métodos Jerárquicos: Ejemplo

```
##################
###  Cars example
##################

cars <- read.table('cars.csv',header=TRUE,sep=";")
head(cars)
names(cars)

# Estandarizacin de variables.
cars.use <- cars[,-c(1,2)]
medians <- apply(cars.use,2,median)
mads <- apply(cars.use,2,mad)
cars.use <- scale(cars.use,center=medians,scale=mads)
```

Métodos Jerárquicos: Ejemplo

```
# Matriz de distancias
cars.dist <- dist(cars.use)

# Hierarchical clustering
cars.hclust <- hclust(cars.dist)
cars.hclust

plot(cars.hclust,labels=cars$Car,main='Default from hclust')
```

3.4.- Otros Métodos

Otros Métodos

- Agrupación basada en modelos.
- Métodos Bayesianos noparamétricos.

3.5.- Lecturas Sugeridas

Lecturas Sugeridas

- Hastie, T., Tibshirani, R. and Friedman, J. 2001. The Elements of Statistical Learning : Data Mining, Inference, and Prediction. Springer.

- Mller, P., Quintana, F., Jara, A., and Hanson, T. E. 2015. Bayesian Nonparametric Data Analysis, Springer.