Modelos de Soporte No Supervisado

Partitioning Clustering:

Clustering Large Applications (CLARA)

CLARA

- CLARA (Clustering Large Applications, Kaufman and Rousseeuw (1990)) es una extensión a los métodos k-medoids
- Permite tratar datos que contienen un gran número de objetos (big data)
- Optimizando el tiempo de cómputo y problema de almacenamiento de RAM
- Aplicando muestreo

CLARA

- En lugar de encontrar medoids para todo el conjunto de datos, CLARA considera una muestra de los datos con tamaño fijo y aplica el algoritmo PAM para generar un conjunto óptimo de medoids para la muestra.

- La calidad de los medoides resultantes se mide por la disimilaridad promedio entre cada objeto en el conjunto de datos completo y el medoid de su clúster, construyendo así la función de costo.

- CLARA repite los procesos de muestreo y agrupación un número de veces especificado para minimizar el sesgo de muestreo.

- Los resultados finales de la agrupación corresponden al conjunto de medoids con un costo mínimo.

CLARA: Algoritmo

1. Divida aleatoriamente los conjuntos de datos en múltiples subconjuntos con tamaño fijo
2. Calcule el algoritmo PAM en cada subconjunto y elija k-medoides. Asignar cada observación de los datos completos al medoid más cercano.
3. Calcule la media (o la suma) de las disimilaridades de las observaciones respecto a su medoid más cercano. Esto se usa como una medida de la bondad de la agrupación.
4. Conservar el conjunto de sub-datos para el cual la media (o suma) es mínima. Se realiza un análisis adicional en la partición final.
5. Es importante tener en cuenta que cada conjunto de subdatos está obligado a contener los medoids obtenidos de los mejores

subdatos establecidos hasta entonces.

Aplicación en R

- Generamos un conjunto de datos de prueba
- set.seed**(1234)**
- # Generamos 500 objetos, divididos en 2 clusters.
- **df <-** rbind**(**cbind**(**rnorm**(200,0,8),**rnorm**(200,0,8)),**
  cbind**(**rnorm**(300,50,8),** rnorm**(300,50,8)))**
- # Definimos un nombre para filas y columnas
- colnames**(df) <-** c**("x", "y")**

Aplicación en R

- rownames**(df) <-** paste0**("S", 1:**nrow**(df))**
- # Inspeccionamos los datos generados
- head**(df, nrow = 6)**

Aplicación en R

- Usamos la función *clara*
- clara**(x, k, metric = "euclidean", stand = FALSE, samples = 5, pamLike = TRUE)**

Donde:

x: una matriz de datos numéricos, cada fila corresponde a una observación y cada columna corresponde a una variable. Se permiten NAs.

k: la cantidad de clusters.

Métrica: las métricas de distancia que se utilizarán. Las opciones disponibles son "euclidianas" y "Manhattan".

Aplicación en R

- Usamos la función *clara*
- clara**(x, k, metric = "euclidean", stand = FALSE, samples = 5, pamLike = TRUE)**

Donde:

Stand: valor lógico; si es verdadero, las variables (columnas) en x están estandarizadas antes de calcular las disimilaridades. Se recomienda estandarizar variables antes de la agrupación.

Muestras: número de muestras que se extraerán del conjunto de datos. El valor predeterminado es 5, pero se recomienda un valor mucho mayor.

PamLike: Indica si el mismo algoritmo en la función pam () debería ser usado. Esto debería ser siempre verdadero.

Aplicación en R

Se utilizarán las librerías:

- library**(cluster)**
- library**(factoextra)**

Determinamos el número de clusters a construir:

- fviz_nbclust**(df, clara, method = "silhouette")+** theme_classic**()**

Aplicación en R

- # Ejecutamos el algoritmo CLARA
- clara.res <- clara(df, 2, samples = 50, pamLike = TRUE)
- # Objetos generados como resultado del algoritmo
- print(clara.res)

Aplicación en R

El resultado de la función clara () incluye los siguientes componentes:

- Medoids: objetos que representan clusters
- Agrupamiento: un vector que contiene el número de clúster de cada objeto
- Muestra: etiquetas o números de casos de las observaciones en la mejor muestra, la muestra utilizada por el algoritmo clara para la partición final.

Aplicación en R

Para agregar la clasificación a los datos originales, usamos:

- dd <- cbind(df, cluster = clara.res$cluster)
- head(dd, n = 4)
- # Medoids
- clara.res$medoids
- # Clustering
- head**(clara.res$clustering, 10)**

Aplicación en R

Para visualizar los clusters generados, usamos:

- fviz_cluster**(clara.res, palette =** c**("#00AFBB", "#FC4E07"),** # definimos la paleta de colores

  ellipse.type = "t", # Dibujamos una elipse

  geom = "point", pointsize = 1,

  ggtheme = theme_classic**()**

)

Conclusiones

- El algoritmo CLARA (Clustering Large Applications) es una extensión del PAM
- (Partitioning Around Medoids) método de agrupamiento para grandes conjuntos de datos.
- Tenía la intención de reducir el tiempo de cálculo en el caso de un gran conjunto de datos.
- Como casi todos los algoritmos de particionamiento, se requiere que el usuario especifique el apropiado número de clusters que se producirán
- Esto puede ser estimado usando la función fviz_nbclust [en el paquete factoextra R].
- La función R clara () [paquete de clúster] se puede usar para calcular el algoritmo CLARA.
- El formato simplificado es clara (x, k, pamLike = TRUE), donde "x" es la información y k es la cantidad de clusters que se generarán
- Después de calcular CLARA, la función R fviz_cluster () [factoextra package] puede ser utilizada para visualizar los resultados. El formato es fviz_cluster (clara.res), donde clara.res es los resultados de CLARA.