Modelos de Soporte No Supervisado

Determinar el número óptimo de clusters

Determinar el número óptimo de clusters

- El número óptimo de clusters es de cierta forma algo subjetivo ya que depende del método utilizado para medir similitudes o de los parámetros utilizados para la partición.

- A continuación se mencionarán diferentes métodos para determinar el número óptimo de clusters para k-medias, k-medoids (PAM) y clustering jerárquico.

Determinar el número óptimo de clusters

Se tienen dos tipos de métodos:

1. Métodos directos: consisten en optimizar un criterio. Ej: Elbow and silhouette methods

1. Métodos de prueba estadística: consisten en comparar contra una hipótesis nula. Ej: gap statistic.

Determinar el número óptimo de clusters

Elbow method

La idea básica de los métodos de partición, es definir clusters de modo que la variación total dentro del cluster (WSS) se minimice.

El WSS total mide que tan compacto es el agrupamiento y lo que se quiere es que sea lo más pequeño posible.

El método de Elbow analiza el WSS total como una función del número de clusters:

Se debe de elegir una cantidad de clusters de tal modo que agregar otro cluster no mejore el WSS total.

Determinar el número óptimo de clusters

Elbow method

El número óptimo de clusters puede ser definido de la siguiente forma:

1. Calcular el algoritmo de agrupamiento para diferentes valores de k. Por ejemplo, variando k de 1 a 10 clusters.

2. Para cada k, se calcula la suma total de cuadrados dentro del grupo (WSS).

3. Se traza la curva de WSS de acuerdo con el número de clusters.

4. La ubicación de un cambio en curva en la gráfica generalmente se considera como un indicador del número apropiado de clusters

Determinar el número óptimo de clusters

Average silhouette method

Este método mide la calidad de una agrupación, es decir, determina que tan bien se encuentra cada objeto dentro de su cluster.

Un promedio alto indica una buena agrupación.

Calcula el “average silhouette” de las observaciones para diferentes valores de k.

El número óptimo de clusters es el que maximiza el promedio sobre un rango de valores posibles para k

Determinar el número óptimo de clusters

Average silhouette method

El número óptimo de clusters puede ser definido de la siguiente forma:

1. Calcular algoritmo de agrupamiento para diferentes valores de k. Por ejemplo, variando k de 1 a 10 clusters.
2. Para cada k, se calcula el “average silhouette” de las observaciones (avg.sil).
3. Se traza la curva de avg.sil según el número de clusters k.
4. La ubicación del máximo se considera como el número apropiado de clusters.

Determinar el número óptimo de clusters

Gap statistic method

Este método compara el total de la variación dentro del cluster para las diferentes k con sus valores esperados bajo la distribución de referencia de los datos.

La estimación de los clusters óptimos será el valor que maximice la gap statistic.

Esto significa que la estructura de agrupamiento está muy lejos de la distribución aleatoria uniforme.

Determinar el número óptimo de clusters

Gap statistic method

Algoritmo:

1. Agrupar los datos observados, variando la cantidad de clusters de k = 1, ..., kmax, y calcular el total de la variación dentro del cluster “Wk”.

2. Generar conjuntos de datos de referencia B con una distribución uniforme aleatoria. Agrupar cada uno de esos conjuntos de datos de referencia con un número variable de custers k = 1, ..., kmax y calcular el total de la variación dentro del cluster “Wkb”.

Determinar el número óptimo de clusters

Gap statistic method

3. Calcular la "Gap statistic" estimado como la desviación del valor Wk observado y el valor esperado Wkb bajo la hipótesis nula:

$$Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W_{kb}^{*}) - log(W_k).$$

Calcular la desviación estándar de las estadísticas.

4. Elegir la cantidad optima de clusters como el valor más pequeño de k de modo que la gap statistic esté dentro de una desviación estándar de la "gap" en k + 1:

$$Gap(k) \geq Gap(k+1) - s_{k+1}.$$

NOTA: al usar B = 500, se obtienen resultados bastante precisos.

Determinar el número óptimo de clusters

La desventaja del método *"Elbow y Average Silhouette"* es que estos solo miden una característica de agrupamiento global.

El método más sofisticado es el *"Gap Statistic"* ya que éste proporciona un procedimiento estadístico lo que lo hace formalizar los otros métodos.

Determinar el número óptimo de clusters

Calcular el número de clusters usando R

Hay dos funciones en R que sirven para determinar el número óptimo de clusters:

1. **fviz_nbclust ()** [Está en el paquete factoextra R]: se puede usar para los métodos vistos anteriormente, además de que es aplicable para cualquier método de agrupamiento [K-means, K-medoids (PAM), CLARA, HCUT].

La función hcut () solo está disponible en el paquete factoextra y hace cálculos de manera jerárquica.

2. **NbClust ()** [Está el paquete NbClust R]: Proporciona 30 índices para determinar el número de clusters, propone el mejor esquema de agrupación de los resultados obtenidos variando las combinaciones de número de clusters, medidas de distancia y métodos de agrupamiento.

Determinar el número óptimo de clusters

Necesitamos los paquetes

factoextra

NbClust

```
pkgs <- c("factoextra", "NbClust")
install.packages(pkgs)
```

Determinar el número óptimo de clusters

Preparación de los datos

Se necesita estandarizar los datos para hacer variables comparables.

```
df <- scale(USArrests)
head(df)
```

```
##               Murder   Assault   UrbanPop         Rape
## Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
## Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
## Arizona    0.07163341 1.4788032  0.9989801  1.042878388
## Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
## California 0.27826823 1.2628144  1.7589234  2.067820292
## Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```

Determinar el número óptimo de clusters

Función fviz_nbclust()

La estructura de la función es la siguiente:

fviz_nbclust(x, FUNcluster, method = c("silhouette", "wss", "gap_stat"))

• x: matriz numérica o estructura de datos

• FUNcluster: Función de partición. Los valores permitidos incluyen kmedias, pam, clara y hcut (clustering jerárquico).

• método: el método que se utilizará para determinar el número óptimo de clusters ("silhouette", "wss“ y "gap_stat“)

Determinar el número óptimo de clusters

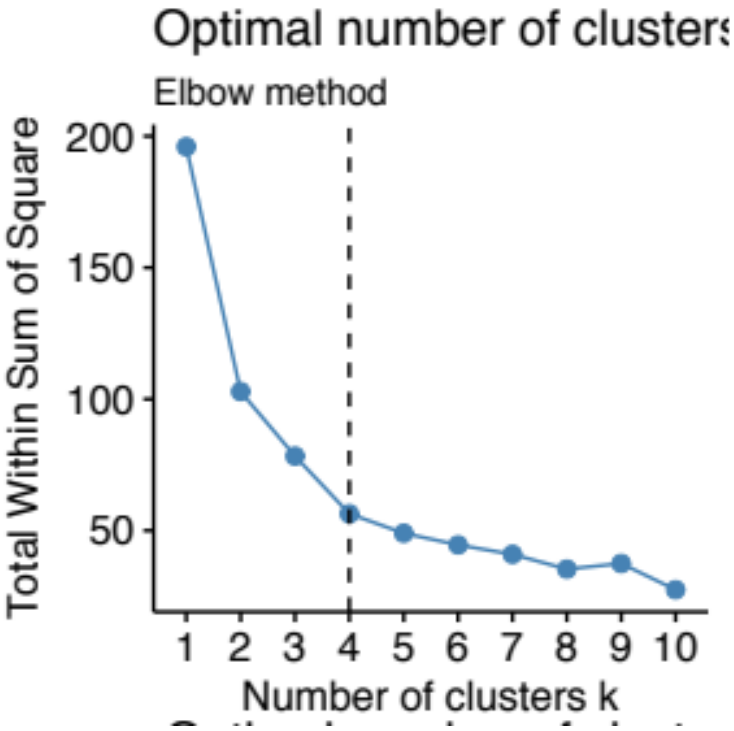
Función fviz_nbclust()

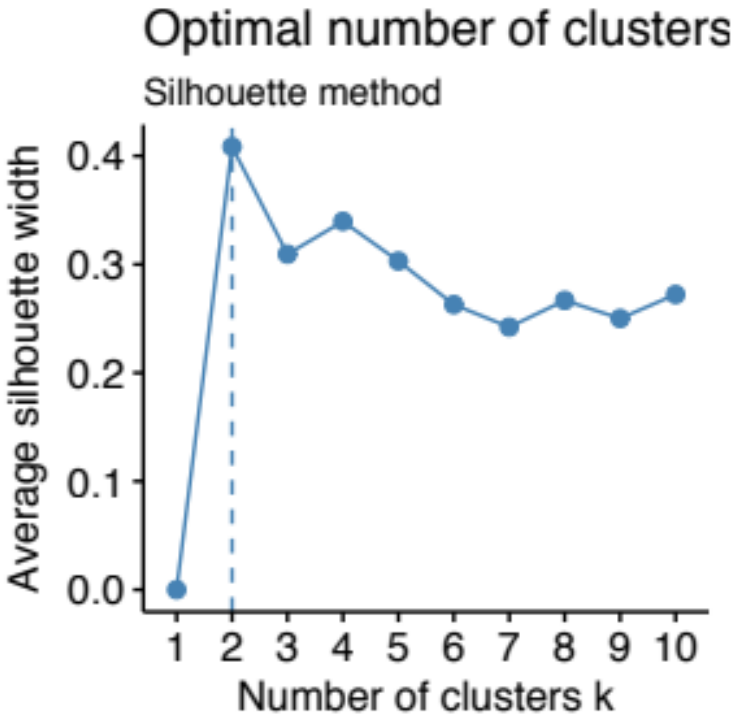
Código en R para los diferentes métodos:

```
# Elbow method
fviz_nbclust(df, kmeans, method = "wss") +geom_vline(xintercept = 4, linetype = 2)+ labs(subtitle = "Elbow method")

# Silhouette method
fviz_nbclust(df, kmeans, method = "silhouette")+labs(subtitle = "Silhouette method")

# Gap statistic
set.seed(123)
fviz_nbclust(df, kmeans, nstart = 25, method = "gap_stat", nboot = 50)+labs(subtitle = "Gap statistic method")
```

Determinar el número óptimo de clusters

Número óptimo de clusters dependiendo de los distintos métodos



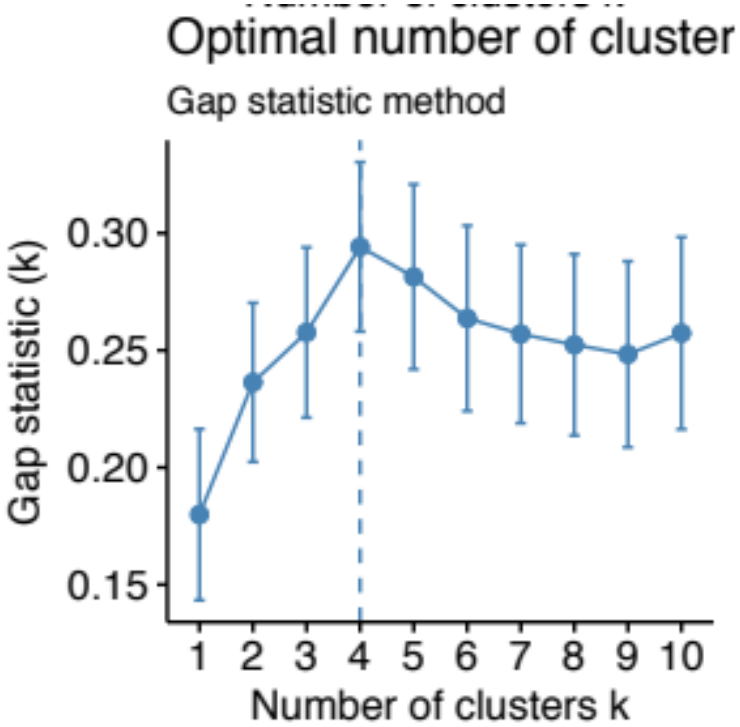


Elbow Method: 4 clusters

Silhouette method: 2 clusters

Determinar el número óptimo de clusters

Número óptimo de clusters dependiendo de los distintos métodos



Gap statistic: 4 clusters

Determinar el número óptimo de clusters

Función NbClust()

La estructura de la función es la siguiente:

NbClust(data = NULL, diss = NULL, distance = "euclidean", min.nc = 2, max.nc = 15, method = NULL)

- ***data:*** matriz de datos
- ***diss:*** matriz de disimilitud para ser utilizada. Por defecto es NULL.
- ***distance:*** la medida de distancia que se utilizará para calcular la matriz de no similitud. Sus posibles valores incluyen "euclidean", "manhattan" o "NULL".
- ***min.nc, max.nc:*** número mínimo y máximo de clusters.
- ***method:*** Se refiere al el método de análisis de cluster que se utilizará. Este puede ser "ward.D", "ward.D2", "Single", "complete", "average", "kmeans" entre otros.

Para k-medias se usa el método "kmeans" mientras que en la agrupación jerárquica, se debe usar "ward.D", "ward.D2", "single", "complete" o "average".

Determinar el número óptimo de clusters

Función NbClust()

El código en R que calcula el NbClust () para k-medias es el siguiente:

```
library("NbClust")
nb <- NbClust(df, distance = "euclidean", min.nc = 2,
max.nc = 10, method = "kmeans")
```

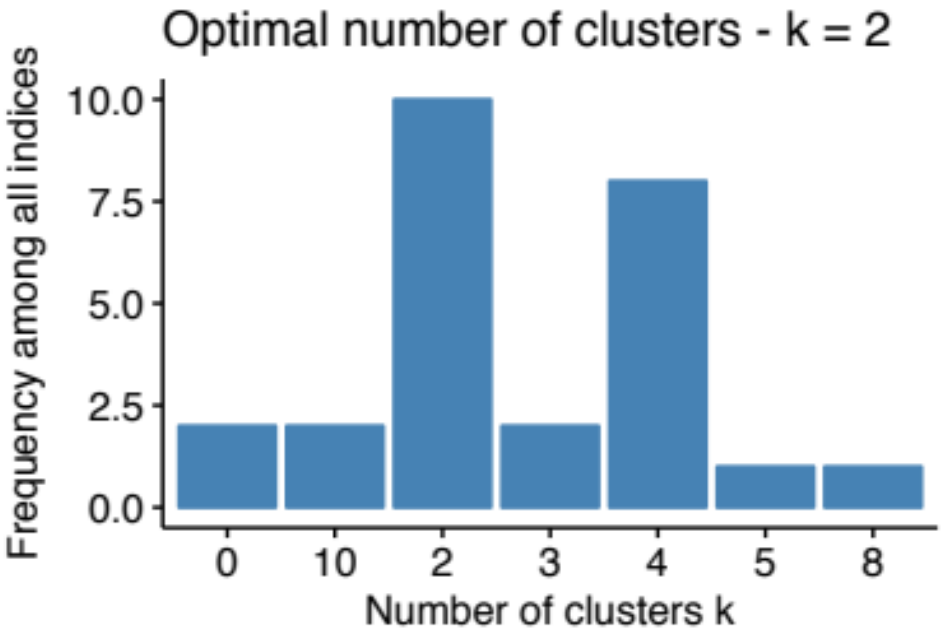
Determinar el número óptimo de clusters

El resultado de NbClust usando la función fviz_nbclust () es el siguiente:

```
library("factoextra")
fviz_nbclust(nb)
## Among all indices:
## ===================
## * 2 proposed 0 as the best number of clusters
## * 10 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 8 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 1 proposed 8 as the best number of clusters
## * 2 proposed 10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is 2 .
```

Determinar el número óptimo de clusters

Función NbClust()



- 2 índices propusieron 0 como el mejor número de clusters
- 10 índices propusieron 2 como el mejor número de clusters.
- 2 índices propusieron 3 como el mejor número de clusters.
- 8 índices propusieron 4 como el mejor número de clusters.

Por lo tanto, el mejor número de clusters es 2.

Resumen

Se revisaron diferentes métodos para elegir el número óptimo de Clusters en un conjunto de datos. Estos métodos fueron *“elbow, silhouette y gap statistic”*.

Se mostró como calcular esos métodos utilizando la función de R **fviz_nbclust ()**

Además del paquete NbClust () que puede ser utilizado para calcular de manera simultanea muchos otros índices y métodos para determinar la cantidad optima de clusters.

Después de elegir el número optimo de clusters k, el siguiente paso es aplicar el método para generar los clusters elegido.