Análisis de clusters (Aprendizaje no supervisado)

Tambien conocido como segmentación de datos

Consiste en agrupar una colección de objetos, de manera que aquellos en el mismo grupo o cluster tengan una mayor relación o cercanía, que aquellos en distintos clusters.

Análisis de clusters (Aprendizaje no supervisado)

También conocido como **segmentación de datos**

Consiste en agrupar una colección de objetos, de manera que aquellos en el mismo grupo o **cluster** tengan una mayor relación o cercanía, que aquellos en distintos clusters.

Dado:
- Un conjunto X de objetos $X = \{X_1, \dots, X_N\}$
- Distancias entre objetos

Salida: Una partición C de X, la cual consiste en K subconjuntos de X

Notación: $k = C(i)$ quiere decir: la partición asigna la i-ésima observación al k-ésimo cluster

Medida para la distancia

Distancia basada en atributos

Definimos en primer lugar la distancia con respecto al j-ésimo atributo:

$$d_j(x_{ij}, x_{i'j})$$

se puede medir de diferentes formas.

La más común: $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$

variable numérica

Distancias dependiendo del tipo de variables

Cuantitativas:

la distancia natural es una función monótona y creciente de la diferencia absoluta:

$$d(x_i, x_i') = l(|x_i - x_i'|)$$

ejemplo

$l(x) = x^2 \longrightarrow$ Mayor énfasis en distancias mayores

$l(x) = id$

Distancias dependiendo del tipo de variables

Ordinales (cualitativas con un orden):

Ej: Malo, Regular, Bueno $(M=3)$

Se reemplazan los valores originales con $\frac{i-1/2}{M}$ $i=1,\dots,M$

Categóricas (sin orden):

$$d_{rr'} = d_{r'r}$$
$$d_{rr} = 0$$
$$d_{rr'} \geq 0 \quad \text{usualmente} \quad d_{rr'} = 1$$

Matriz de similaridad

Las distancias entre los objetos se puede representar mediante una matriz: matriz de similaridad

$$M = (d_{ij}) \qquad d_{ij} = d(x_i, x_j)$$

por lo general simétrica y con diagonal nula, pero no necesariamente.

Distancia general

Combinemos ahora las distancias basadas en atributos, para definir una distancia general entre dos objetos X_i, X_j

$$D(X_i, X_{i'}) = \sum_{j=1}^{p} w_j \, d_j(X_{ij}, X_{i'j})$$

$$\text{con} \quad \sum_{j=1}^{p} w_j = 1.$$

Observación: Asignar valores iguales de w_j para todos los atributos no necesariamente conllevará a que todos los atributos tengan la misma influencia en la distancia general $\longrightarrow$

La influencia del j-ésimo atributo en la distancia general depende de su contribución a la distancia promedio sobre todos los pares de observaciones en el conjunto de datos:

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} D(x_i, x_{i'})$$

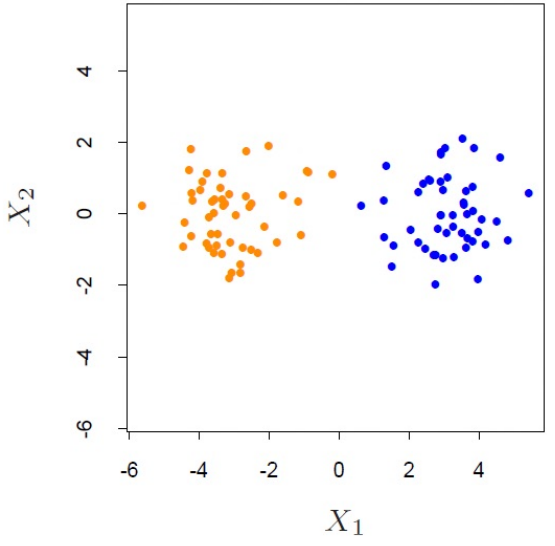
$$= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{i'j})$$

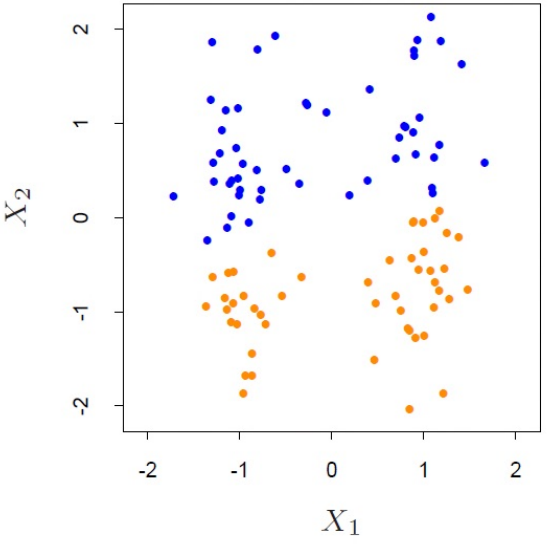
$$= \sum_{j=1}^{p} w_j \underbrace{\frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_j(x_{ij}, x_{i'j})}_{\bar{d}_j}$$

$\bar{d}_j$ → distancia promedio en el j-ésimo atributo

Si queremos que cada atributo tenga la misma influencia en la distancia general, asignamos $w_j = \frac{1}{\bar{d}_j}$.

The elements of statistical learning





"Especificar una distancia apropiada es mucho más importante para obtener éxito en segmentación, que la selección del algoritmo de clustering"

The elements of statistical learning

Clustering visto como optimización

Se puede pensar en una función de costo o función a optimizar en Clustering, basada en las distancias

La función de pérdida natural definida para una partición C sería

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

] distancias "intra cluster"

$$T = \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_{ii'} = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \left(\sum_{C(i')=k} d_{ii'} + \sum_{C(i') \neq k} d_{ii'} \right)$$

$d(x_i, x_{i'})$

$$T = W(C) + B(C), \text{ con } B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i') \neq k} d_{ii'}$$

] distancias entre (diferentes) clusters inter

Minimizar $W(C)$ es equivalente a maximizar $B(C)$

A priori, para minimizar $W(C)$ tendríamos que hacerlo sobre todas las particiones posibles

de particiones

$$S(N,K) = \frac{1}{K!}\sum_{k=1}^{K}(-1)^{K-k}\binom{K}{k}k^N$$

$$S(19,4) \simeq 10^{10}$$

Se necesitan algoritmos en los que en cada intento de cluster mejore la métrica, de manera que no tengamos que optimizar de forma exhaustiva!

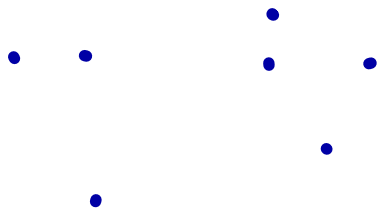
Algunos métodos de clustering

1. Métodos aglomerativos. Son aquellos que inician con una partición en la que cada objeto es su propio cluster. Luego se van uniendo pares de clusters progresivamente

   "método boton-up"

Single Linkage Clustering

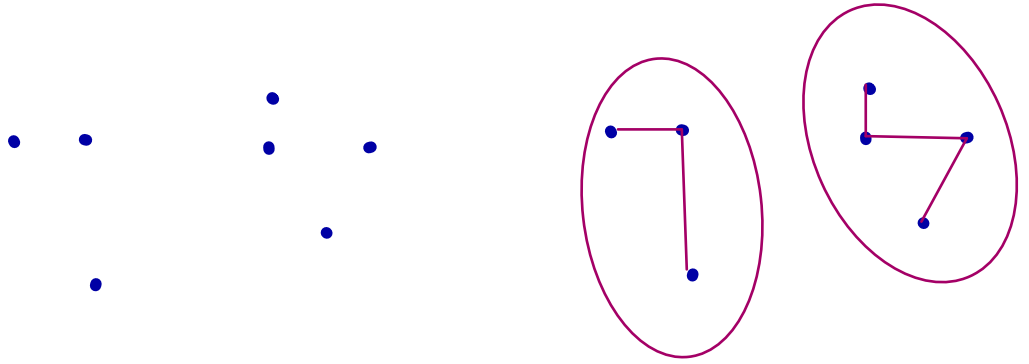
- Considerar cada objeto como un cluster (N objetos)
- Definir la distancia entre dos clusters como la distancia entre los dos puntos más cercanos (uno en cada cluster)
- Unir los dos clusters más cercanos
- Repetir $N-K$ veces para crear K clusters



Single Linkage Clustering

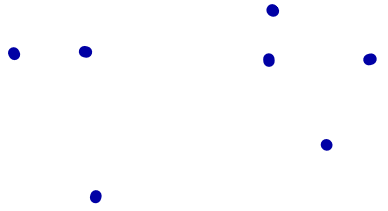
- Considerar cada objeto como un cluster (N objetos)
- Definir la distancia entre dos clusters como la distancia entre los dos puntos más cercanos (uno en cada cluster)
- Unir los dos clusters más cercanos

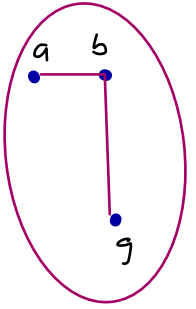
  Repetir $N-K$ veces para crear K clusters

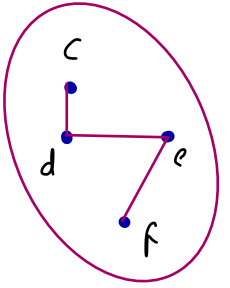


Single Linkage Clustering

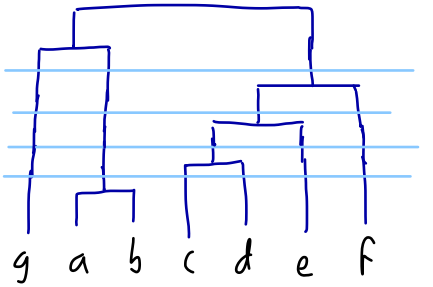
- Considerar cada objeto como un cluster (N objetos)
- Definir la distancia entre dos clusters como la distancia entre los dos puntos más cercanos (uno en cada cluster)
- Unir los dos clusters más cercanos
- Repetir N − K veces para crear K clusters

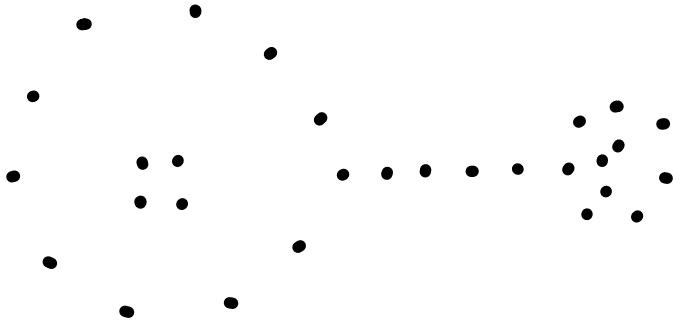






Dendograma (herárquico)





¿ qué partición obtendríamos con $K=2$?

Qué cambia en otros métodos aglomerativos?

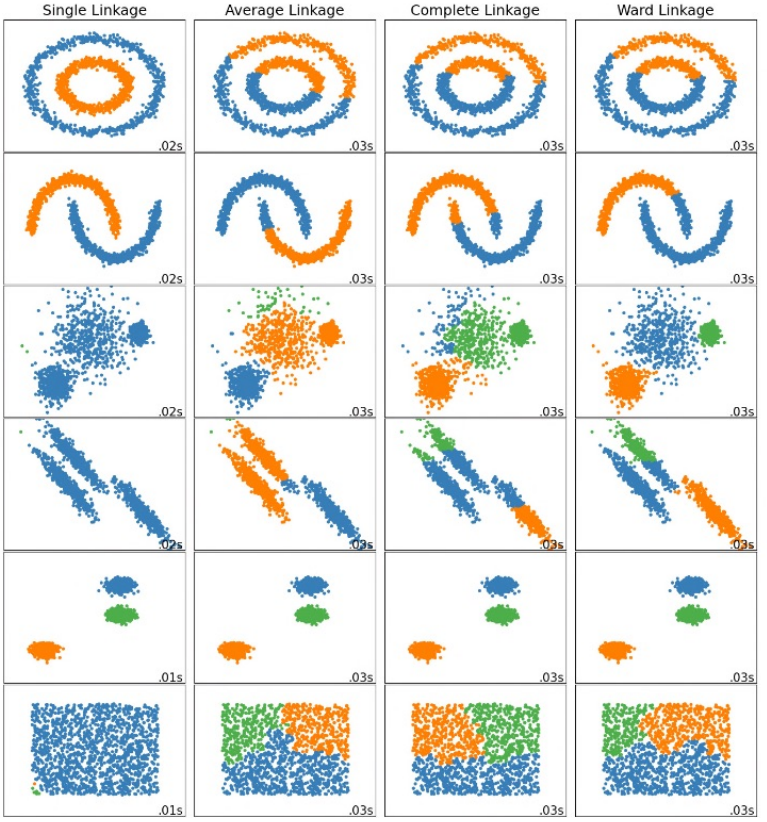
La forma en que se define la distancia entre clústers

Ward: Minimiza la suma de cuadrados de las diferencias en todos los clusters

Máximum or complete linkage: Minimiza la distancia máxima entre pares de clusters

Average Linkage: Minimiza el promedio de las distancias entre todas las observaciones de pares de clusters

Single Linkage: Minimiza la distancia entre las observaciones más cercanas de pares de clusters



Sklearn documentation

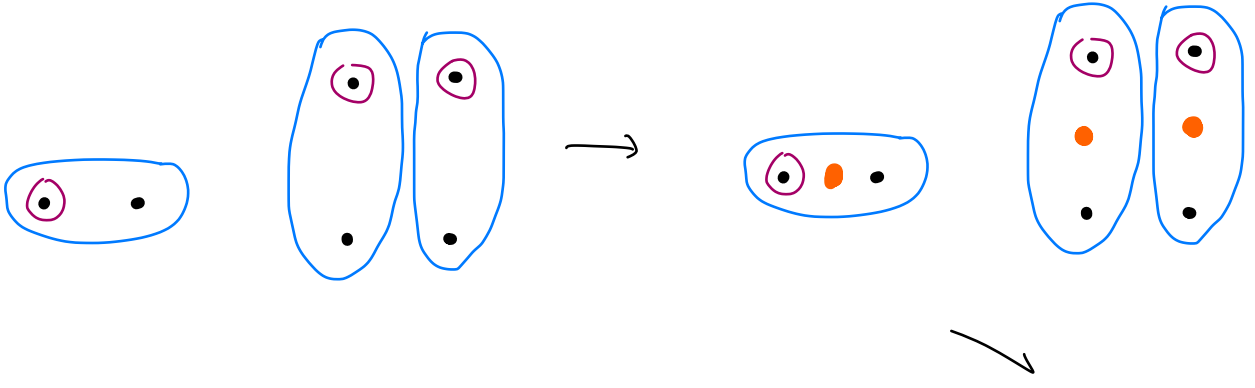
2. K-means ⟶ Procedimiento "top-down"

- Elegir K centros aleatoriamente
- Cada centro reclama los puntos más cercanos
- Se recalculan los centros promediando los puntos en cada cluster
- Repetir hasta converger

K=3 ?

2. K-means ⟶ Procedimiento "top-down"

- Elegir K centros aleatoriamente
- Cada centro reclama los puntos más cercanos
- Se recalculan los centros promediando los puntos en cada cluster
- Repetir hasta converger



En conclusión, en K-means se mueven iterativamente los centros para minimizar la varianza total dentro de los clusters

Convergencia?

En el espacio euclideano (distancia euclideana)

$P^t(X)$. Partición del conjunto X (en la iteración t)

C_i^t : Conjunto de puntos en el i-ésimo cluster

$$center_i^t = \sum_{y \in C_i^t} y \Big/ |C_i|$$

Centroide !

¡ En cada iteración la suma de cuadrados sólo puede disminuir !

Luego Converge

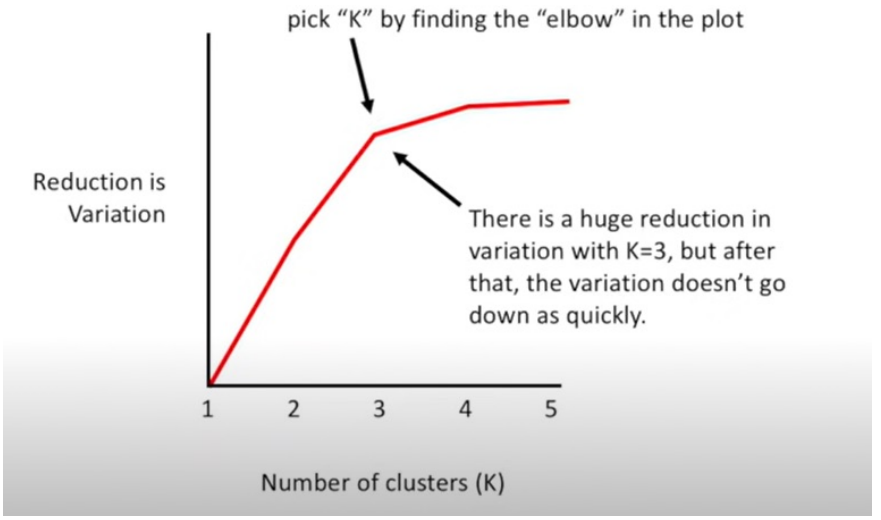
Algoritmo

$center_i^0$

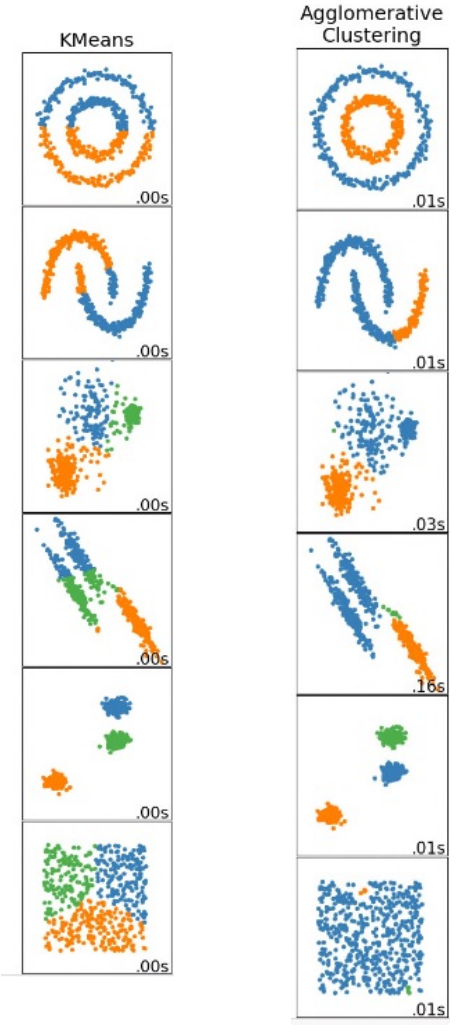
$$P^t(x) = \underset{i}{\operatorname{argmin}} \| X - center_i^{t-1} \|^2$$

$$center_i^t = \sum_{y \in C_i^t} y \Big/ |C_i|$$

¿Cómo escoger el mejor K?



Josh Starmer StatQuest: K-means


KMeans
Agglomerative Clustering
.00s
.01s
.00s
.01s
.00s
.03s
.00s
.16s
.00s
.01s
.00s
.01s