Aprendizaje no supervisado (Clustering)

2do cuatrimestre - 2025

Juan E. Kamienkowski

Distintos tipos de Clustering (Tan, Steinbach & Kumar "Introduction to Data Mining" https://www-users.cs.umn.edu/~kumar001/dmbook/index.php#chapters)

- Clustering por prototipo (k-means / PAM o k-medoids)
- Clustering jerárquico
- Clustering por densidad (DBSCAN)
- Clustering difuso

Librerías de Python para calcular principalmente scikit-learn y para graficar hay muchas herramientas en yellowbrick, y también herramientas como t-SNE, MDS, nMDS, o UMAP. Recientemente, también apareció clusteval. Por ejemplo,

```
>>> from yellowbrick.cluster import SilhouetteVisualizer
>>> from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
>>> from sklearn_extra.cluster import KMedoids
>>> import skfuzz
```

Medidas de Similaridad, Disimilaridad, Proximidad, Distancias

(http://www.iiisci.org/journal/CV$/sci/pdfs/GS315JG.pdf)

- Distancias. Ej. Métricas de Minkowski: Manhattan (L1), Euclídea (L2), … distancia de Mahalanobis
- Ángulos. Ej. distancia coseno
- Binarias. Ej. Coeficiente de coincidencias, Coeficiente de Jaccard
- Multiestado
- Mixtas

Métodos de validación de clusters

- No supervisada o Interna.
  - Tendencia al clustering (Hopkins)
  - Matriz de similaridad
  - Silhuette
  - SSE / SSB
  - Coeficiente de correlación cofenético (Jerárquico)
  - Bootstraping (Jerárquico)
  - Partición de un cluster jerárquico.
- Supervisada o Externa.
  - Clasificación
    - Entropía
    - Pureza
    - Precisión
    - Recall
    - F
  - Similaridad
    - Jaccard
    - Rand
    - van Dongen

Distintos tipos de Clustering (Tan, Steinbach & Kumar "Introduction to Data Mining" https://www-users.cs.umn.edu/~kumar001/dmbook/index.php#chapters)

- Particiones vs Jerárquico (anidado)
  - Particiones: k-means, PAM
  - Jerárquico: Aglomerativo
- Exclusivo (cada punto pertenece a un cluster) vs Superpuesto (cada punto puede pertenecer a más de una cluster) vs Difuso (todos los puntos pertenecen a todas las cluster con cierta probabilidad)
  - Superpuesto: DBSCAN
  - Difuso: Clustering difuso
- Completo (todos los puntos están asignados a algún cluster) vs Parcial (hay puntos no asignados)
  - Parcial: DBSCAN

Clustering por prototipos (k-means, algoritmo de Lloyd)

- Seleccionar K.

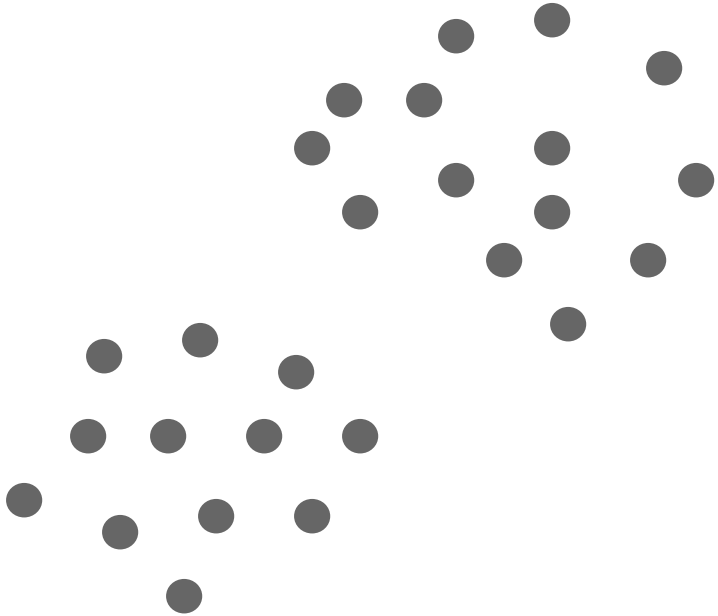
  Seleccionar K puntos como centroides iniciales.

  Repetir:

  Asignar cada punto a uno de los K clusters.

  Recomputar los centroides de cada clusters.

  Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

- Seleccionar K = 2.

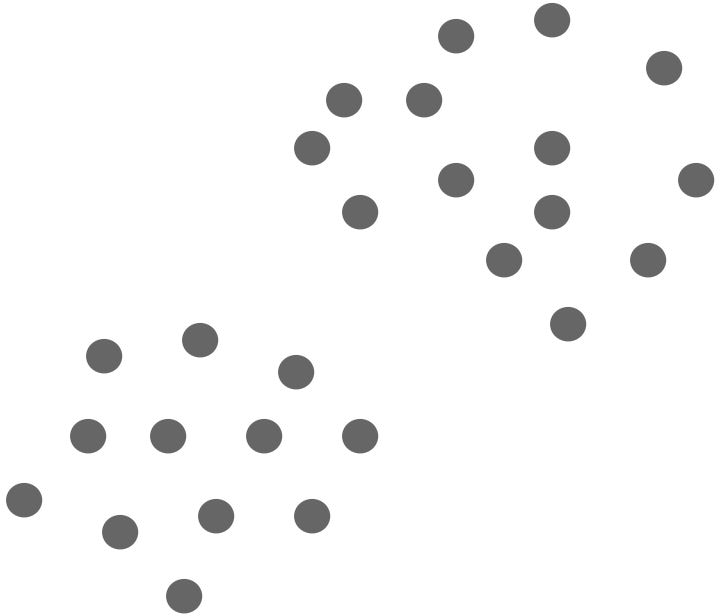
Seleccionar K puntos como centroides iniciales.

Repetir:

Asignar cada punto a uno de los K clusters.

Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

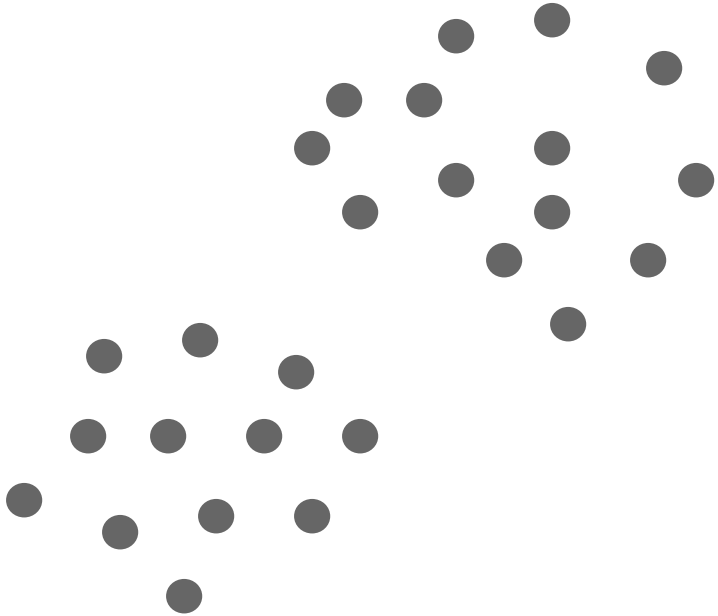
Seleccionar K puntos como centroides iniciales.

Repetir:

Asignar cada punto a uno de los K clusters.

Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

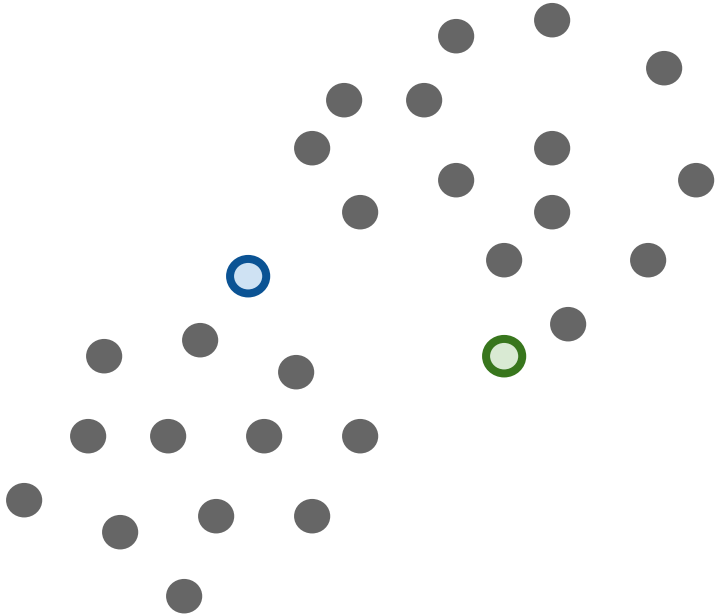
Seleccionar K puntos como centroides iniciales.

Repetir:

Asignar cada punto a uno de los K clusters.

Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

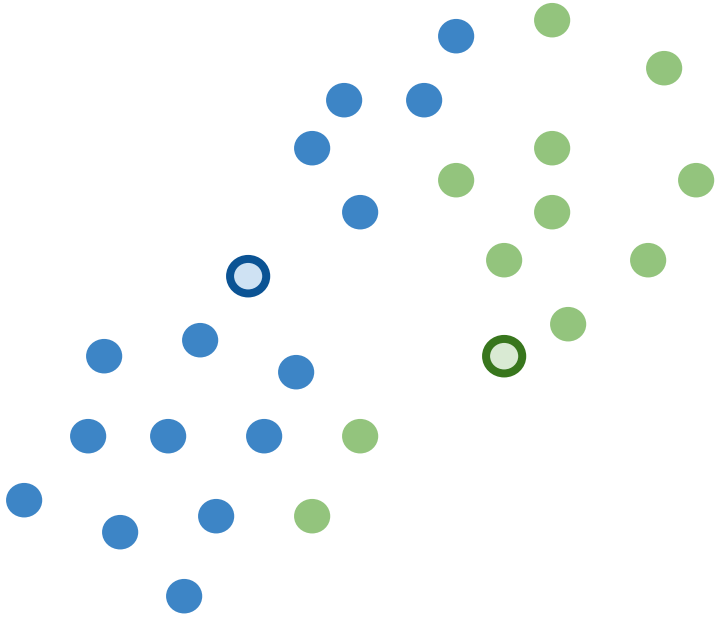
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

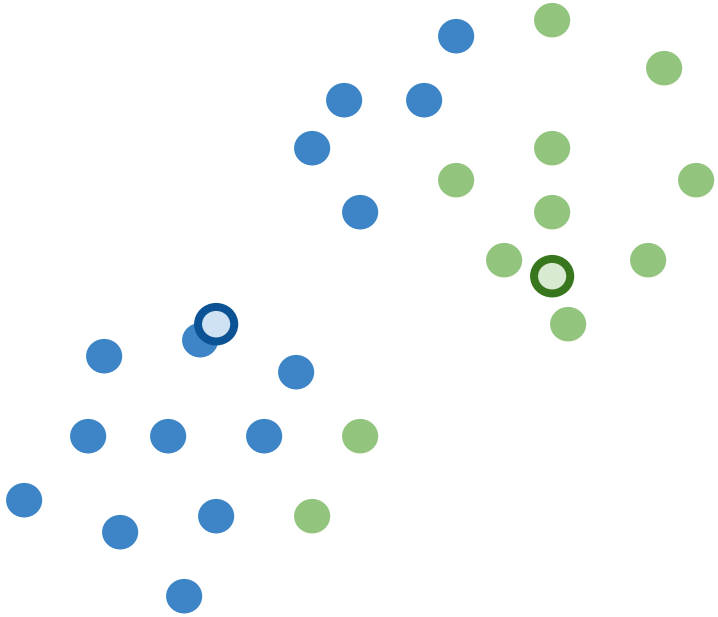
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

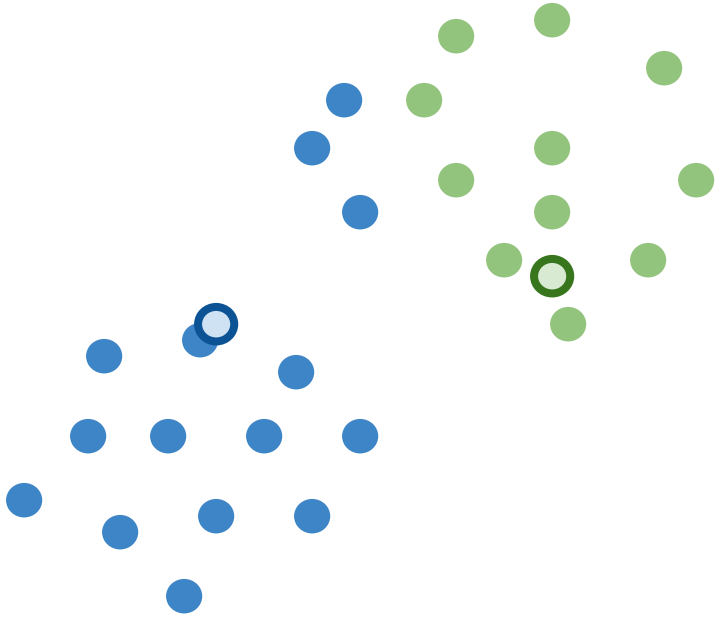
Seleccionar K.

Seleccionar K puntos como centroides iniciales.

Repetir:

- Asignar cada punto a uno de los K clusters.
- Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

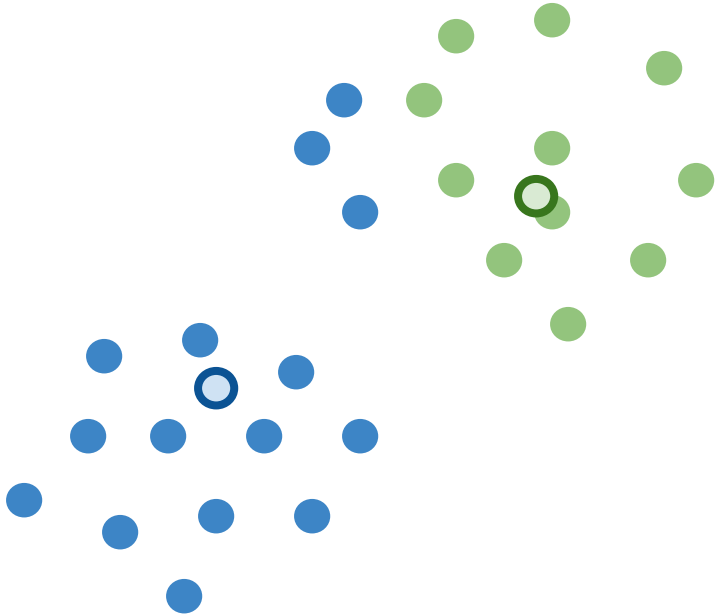
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

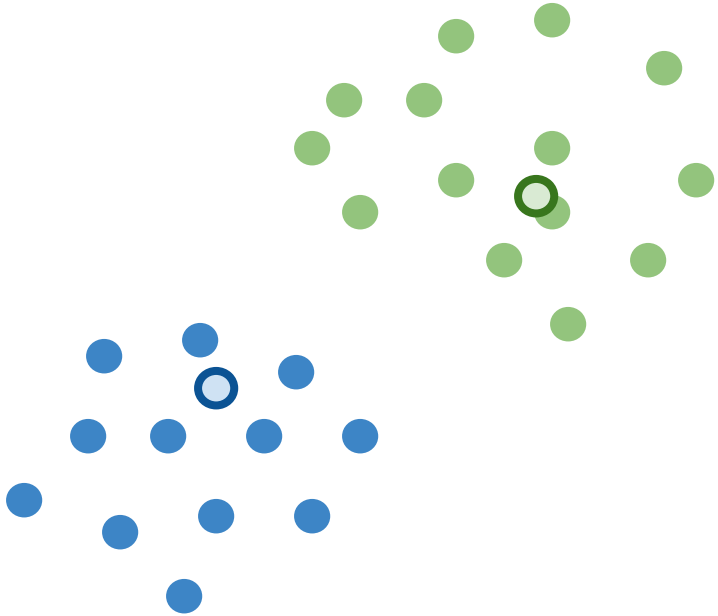
Seleccionar K.

Seleccionar K puntos como centroides iniciales.

Repetir:

- Asignar cada punto a uno de los K clusters.
- Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

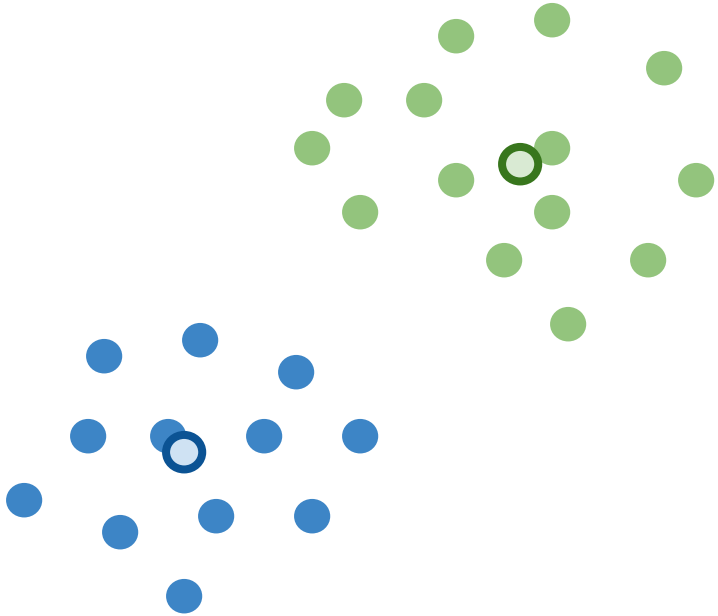
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

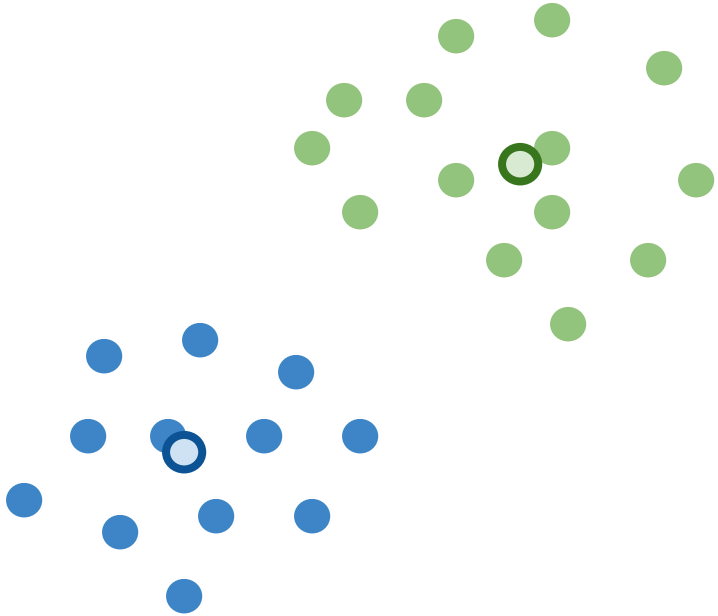
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

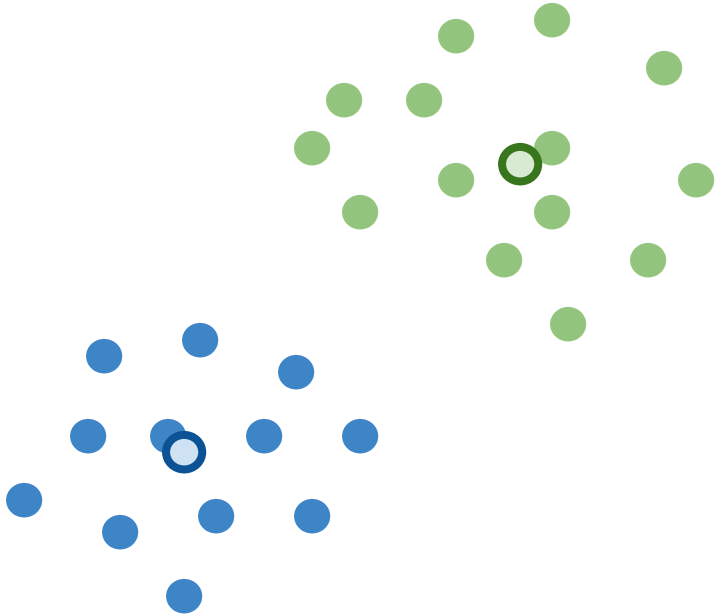
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



Clustering por prototipos (k-means)

Seleccionar K.

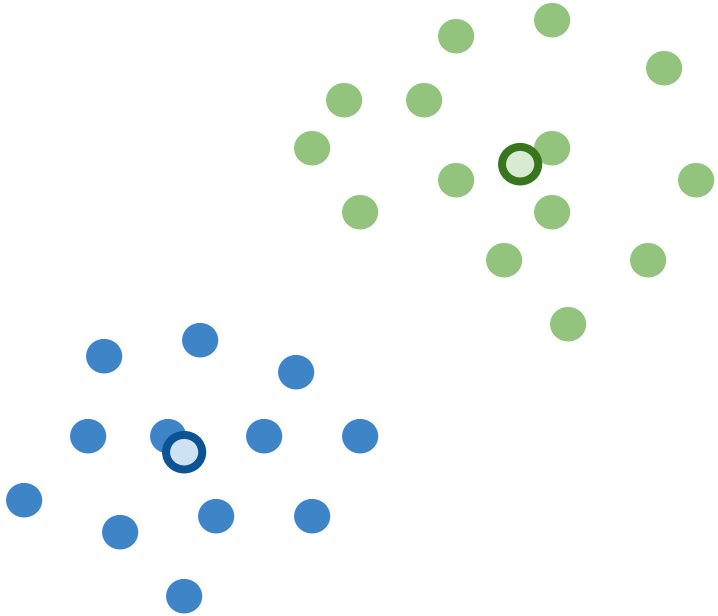
Seleccionar K puntos como centroides iniciales.

Repetir:

    Asignar cada punto a uno de los K clusters.

    Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d^2(x - x_{c_i}) = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - x_{c_i}\|^2$$

Se pueden definir distintas *Funciones objetivo*, como minimizar el SSE (Sum of Squared Errors, Suma de Errores Cuadráticos).

Clustering por prototipos (k-means)

Seleccionar K.

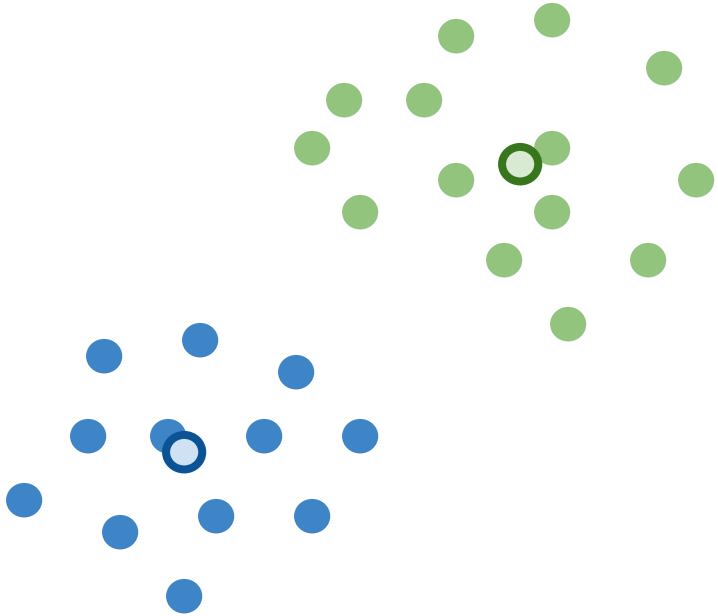
Seleccionar K puntos como centroides iniciales.

Repetir:

Asignar cada punto a uno de los K clusters.

Recomputar los centroides de cada clusters.

Hasta que: los centroides no cambien



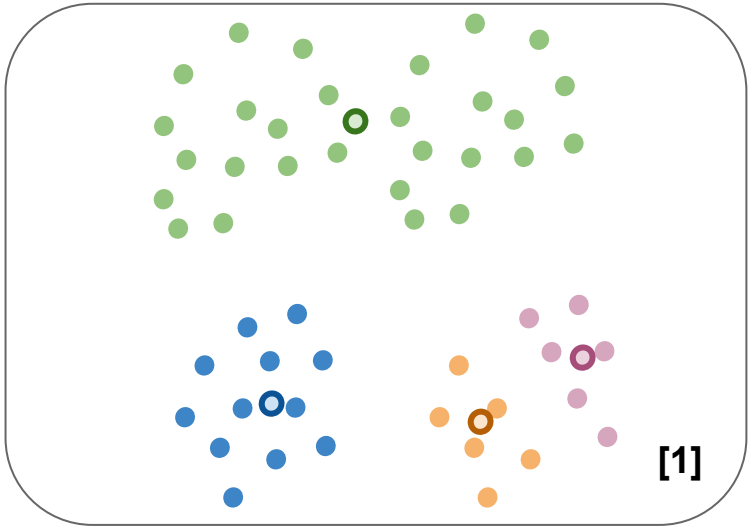
Complejidad

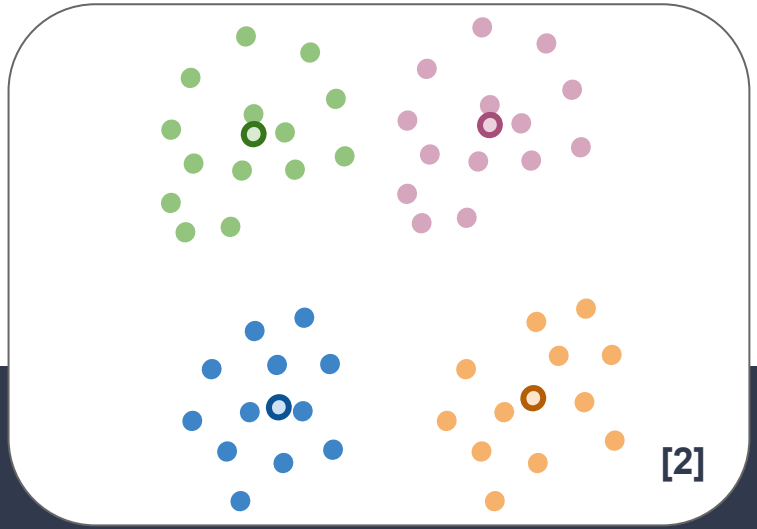
n = número de puntos, d = número de atributos,
k = número de clusters, m = número iteraciones

Si se fijan los parámetros n, k, d, m =>

O(n * k * d * m) ~ O(n * k * ...)

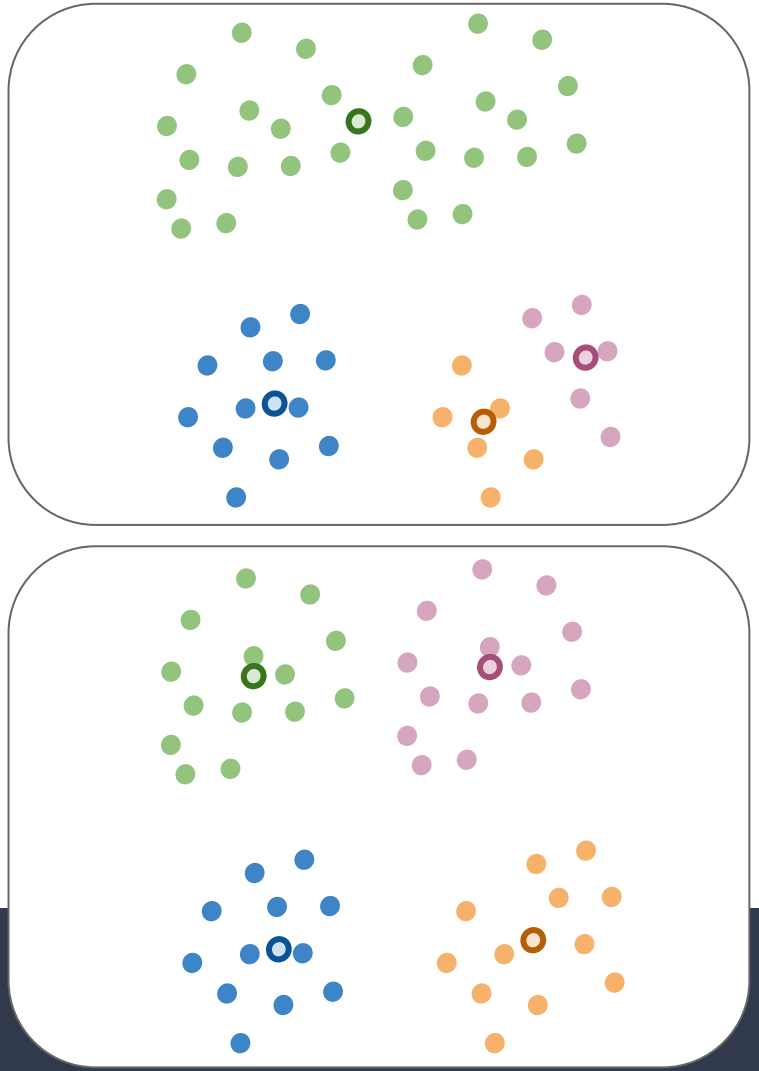
Clustering por prototipos (k-means) > Inicialización



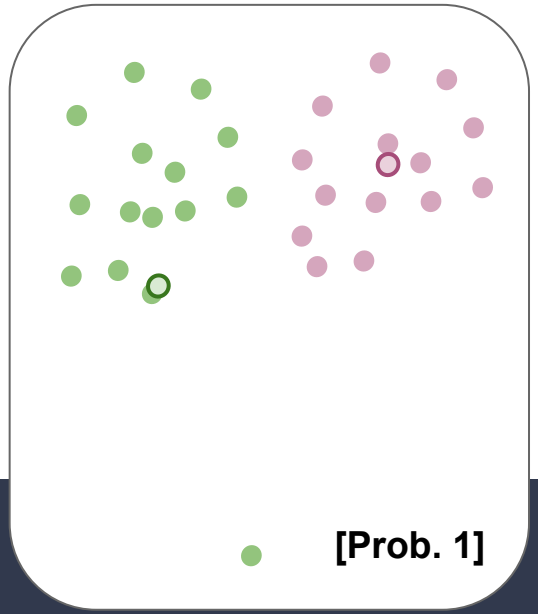


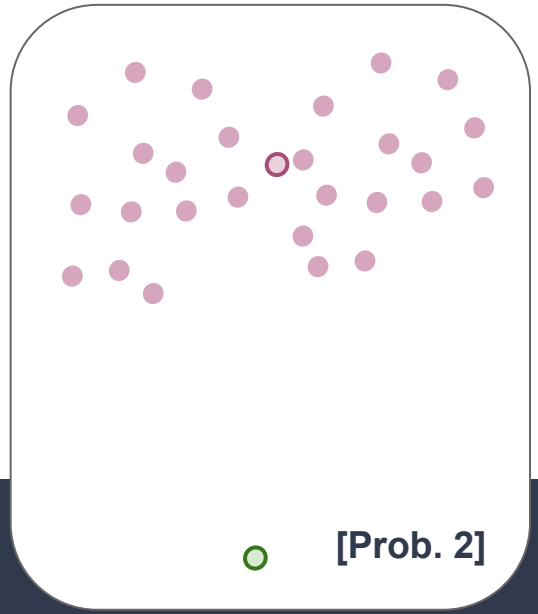
Clustering por prototipos (k-means) > Inicialización

- Elegir al azar y repetir muchas veces.
- Partir de otro clustering:
  - un clustering jerárquico.
  - un k-means con k más bajo: *Bisecting k-means*.
- Elegir un punto al azar, y luego ir eligiendo puntos alejados.
  - *k-means++*:
    Esos nuevos puntos se eligen con probabilidad ~ ‘distancia a los centros anteriores’ ^2



Clustering por prototipos (k-means) > Outliers

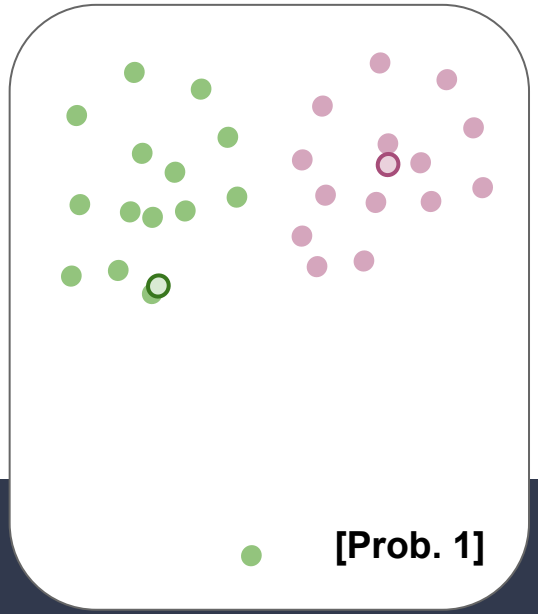


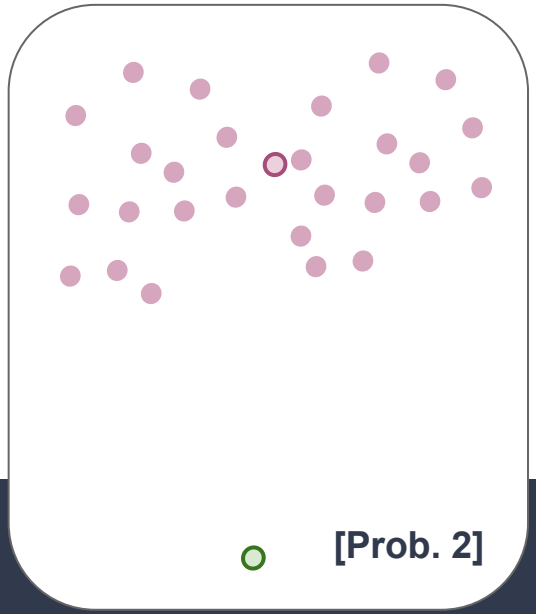


Clustering por prototipos (k-means) > Outliers

Elementos que más aportan al error > Clase de validación / Silhouette / SSE

- Eliminar outliers en el preprocesamiento.
- Eliminar outliers en el postprocesamiento [Problema 1]
- *k-means++* u otro método de inicialización “inteligente” [Problema 2]
- ***k-medoids*** **[Problema 1]**





Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

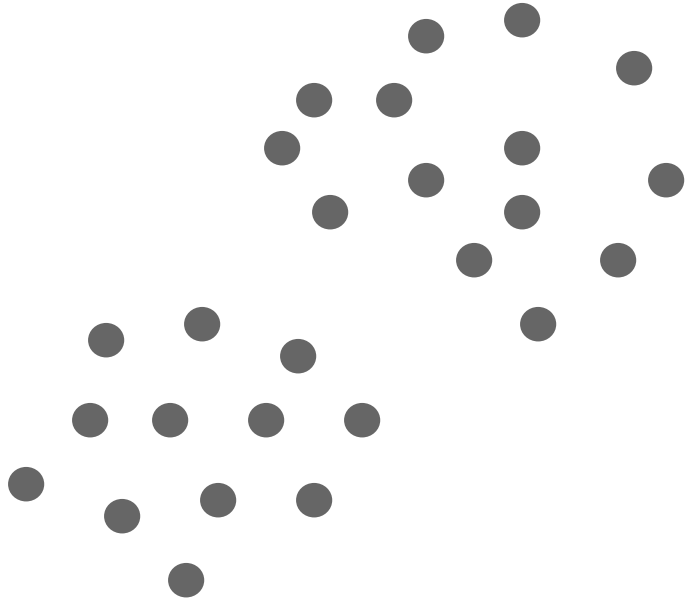
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

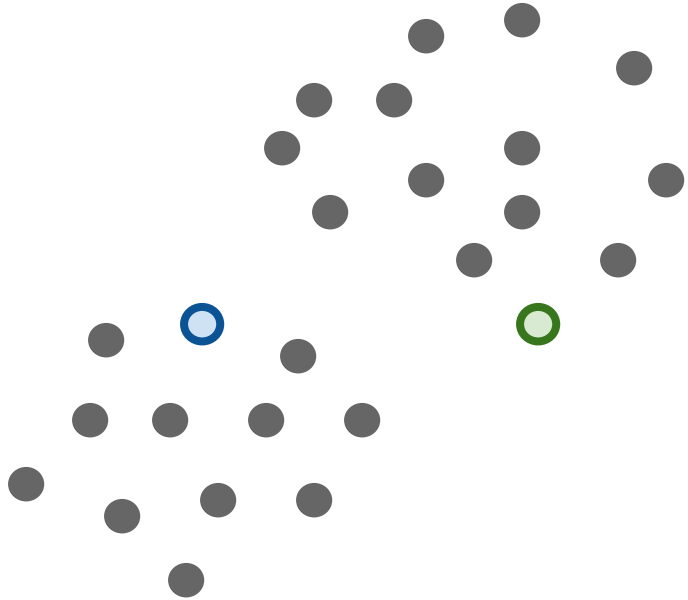
Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de construcción (Build phase)

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

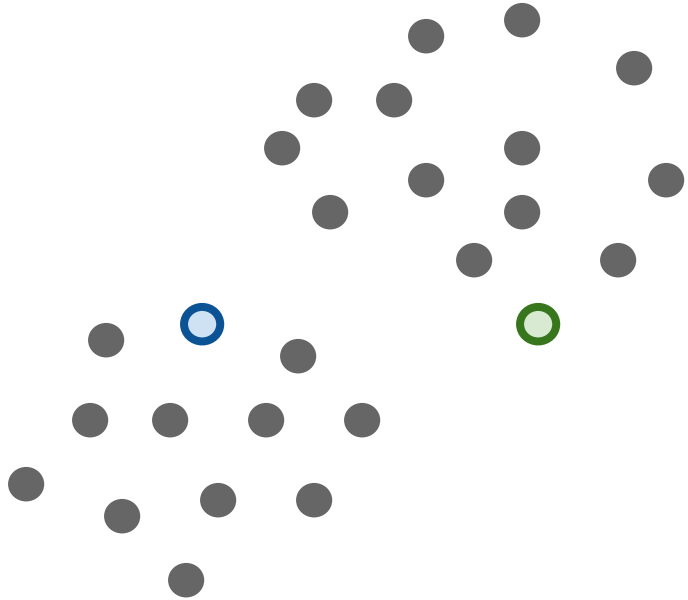
Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de construcción (Build phase)

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

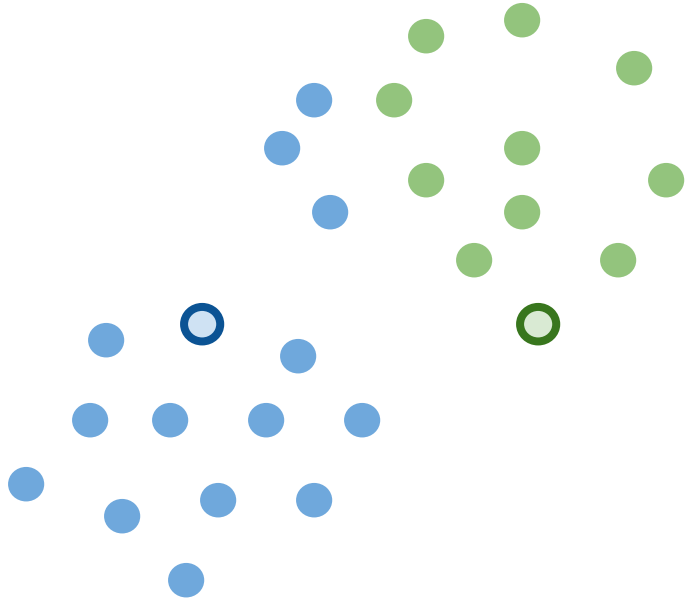
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

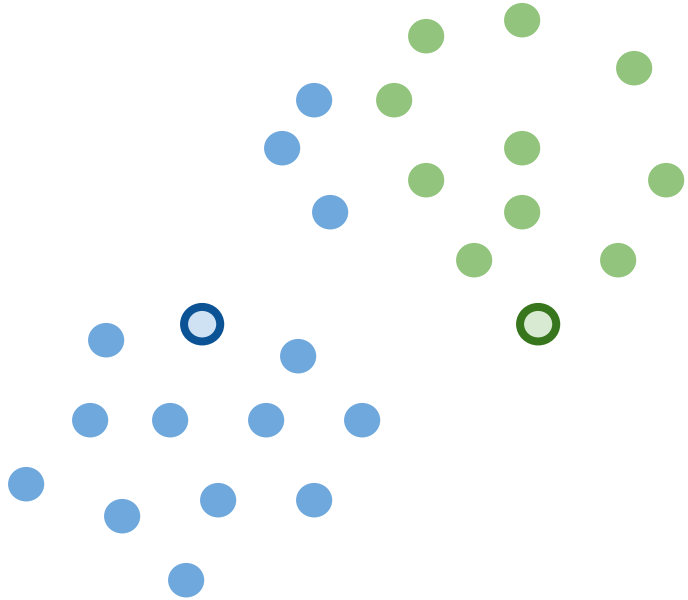
Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de construcción (Build phase)

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

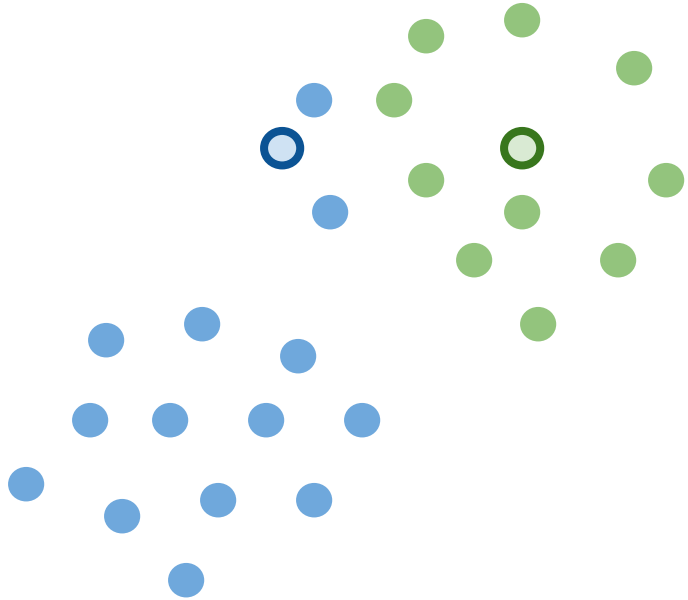
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

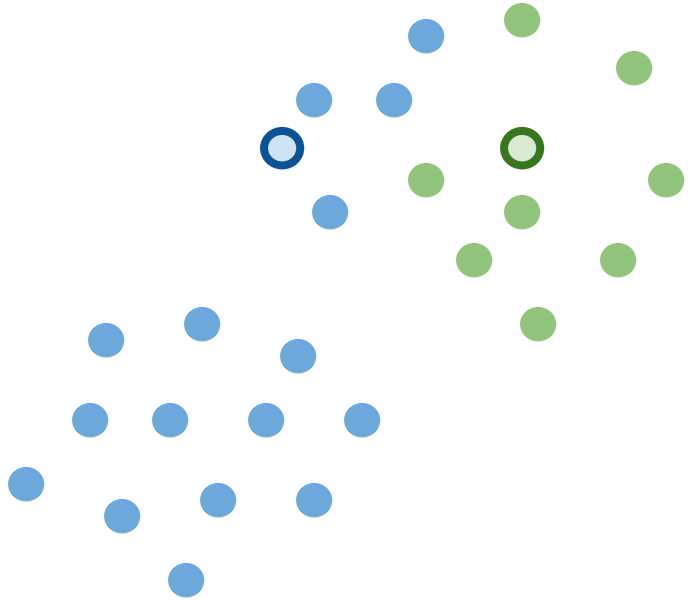
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

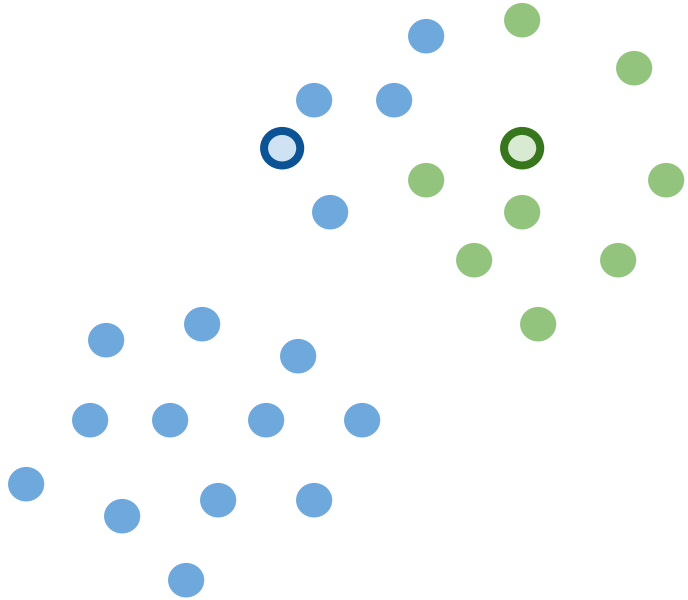
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

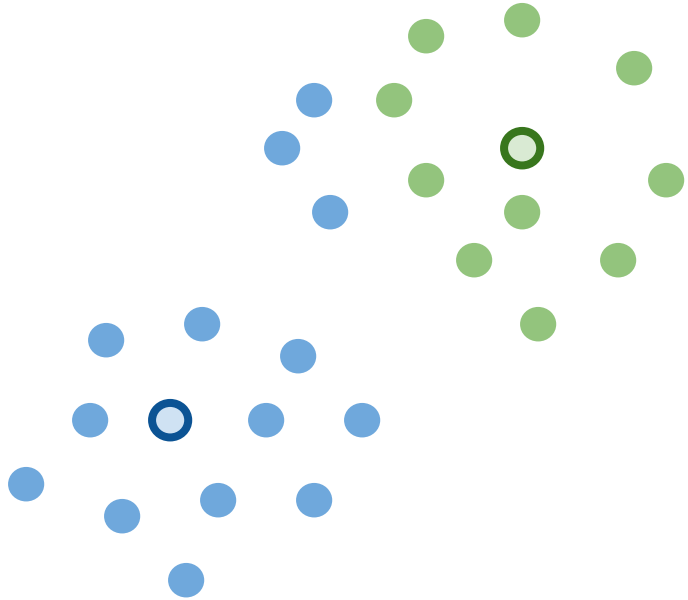
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

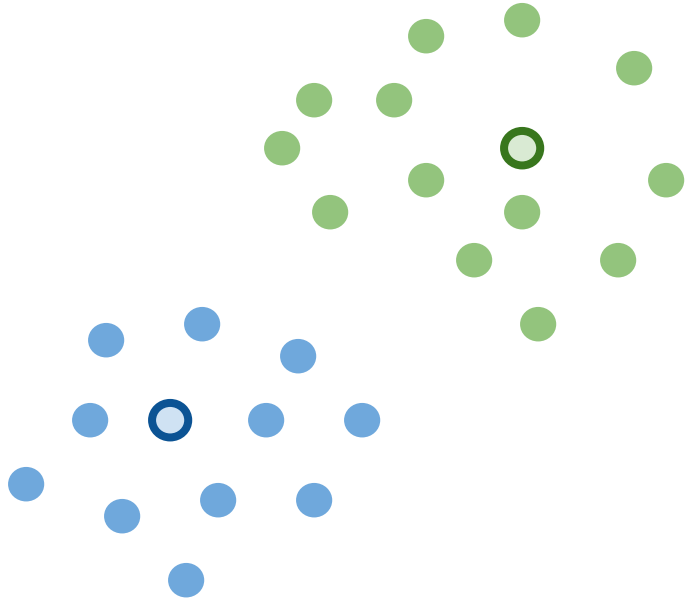
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

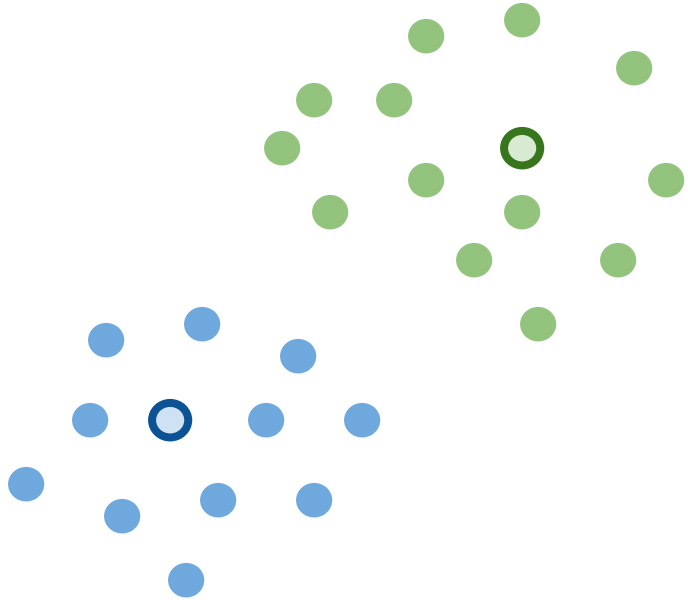
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

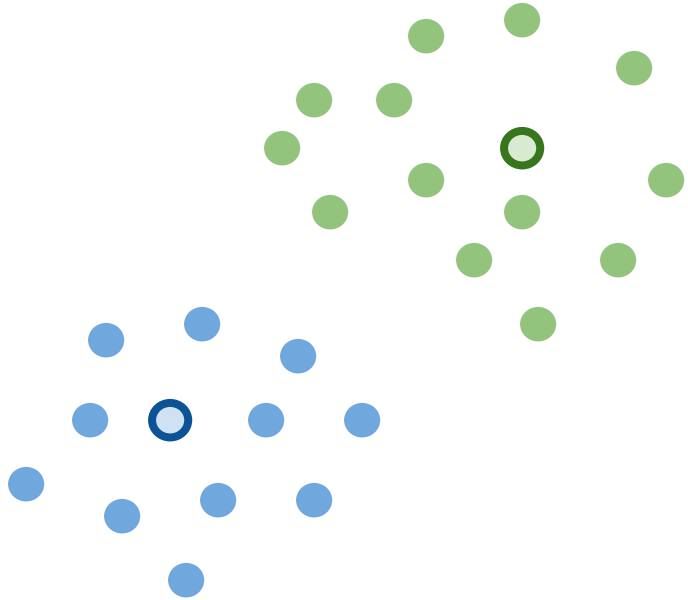
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

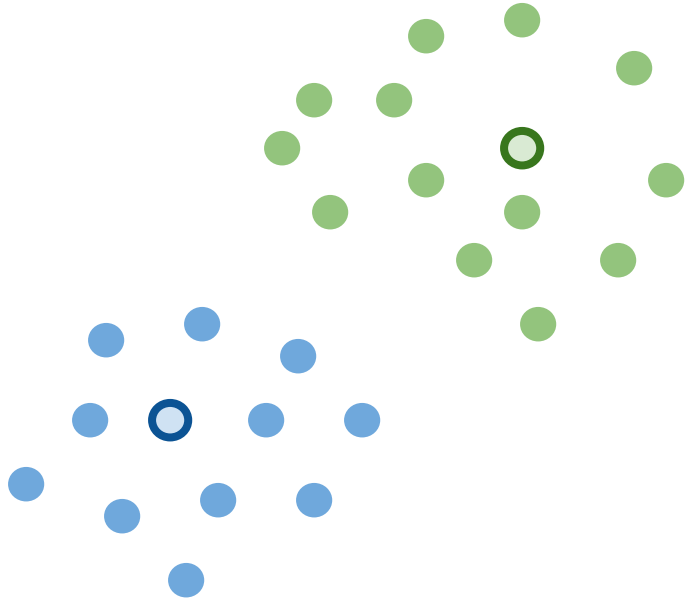
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

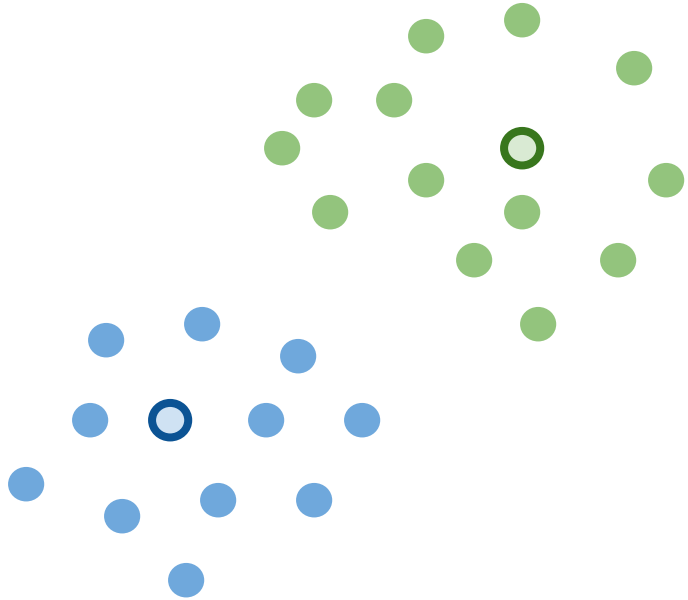
Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de construcción (Build phase)

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

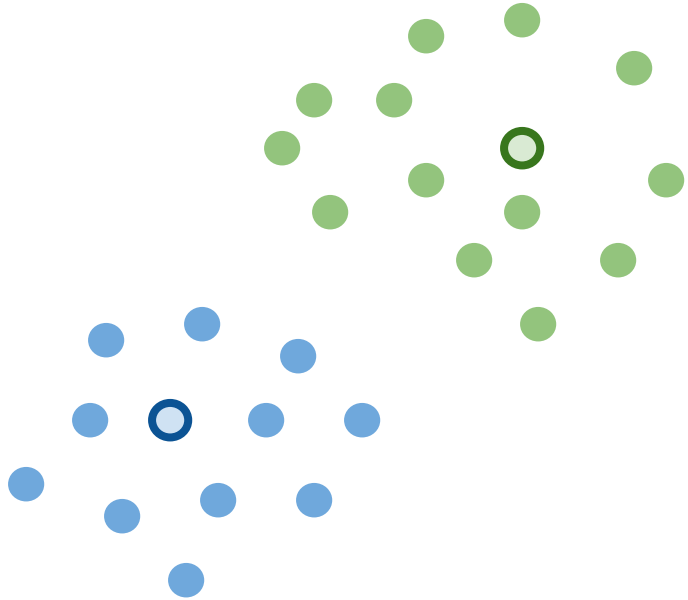
Etapa de construcción (Build phase)

Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)



Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de distancias.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las distancias entre los miembros y el medoide; calcular el total entre clusters.

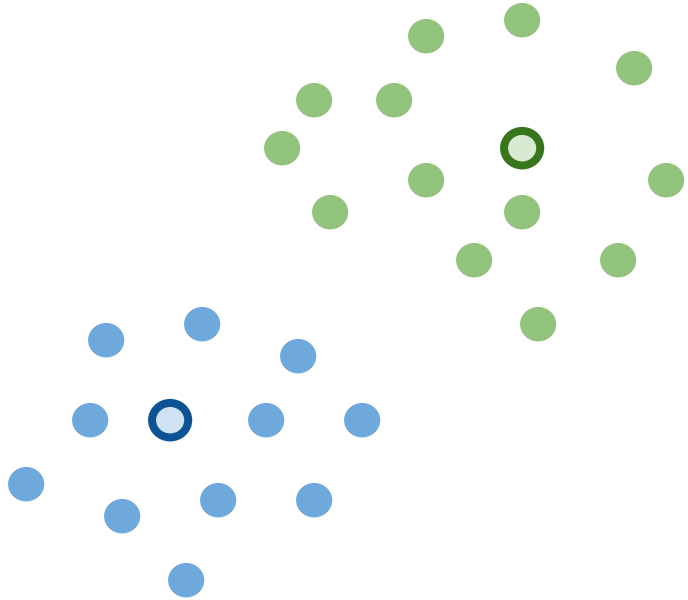
Repetir:

5. Intercambiar el medoide dentro del cluster.
6. Recomputar (3) de cada cluster.
7. Si (4) disminuye, reemplazar el medoide.

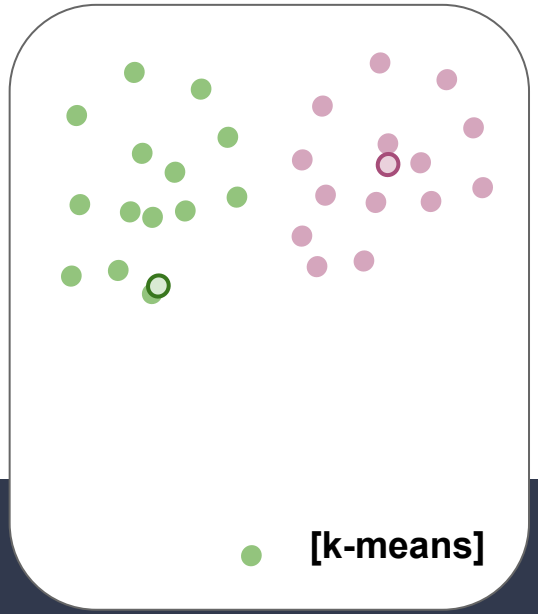
Hasta que: los medoides no cambien

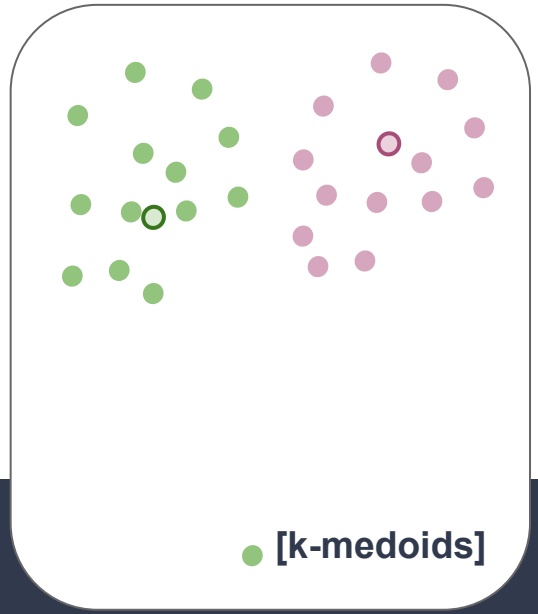
Etapa de construcción (Build phase)

Etapa de intercambio (Swap phase)



Clustering por prototipos (k-medoids) > Outliers





Clustering por prototipos k-means vs k-medoids

k-means

Típicamente la distancia es la *Euclidea*, y la función objetivo:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d^2(x - x_{c_i}) = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - x_{c_i}\|^2$$

Complejidad

n = número de puntos, d = número de atributos,
k = número de clusters, m = número iteraciones

Si se fijan los parámetros n, k, d, m =>

O(n * k * d * m) ~ O(n * k * ...)

k-medoids

Menos sensible a outliers

Más flexible para otras distancias

Complejidad

n = número de puntos, d = número de atributos,
k = número de clusters, m = número iteraciones

Si se fijan los parámetros n, k, d, m =>

O(k * (n - k)² * ...)

No escala bien para *n* grandes.

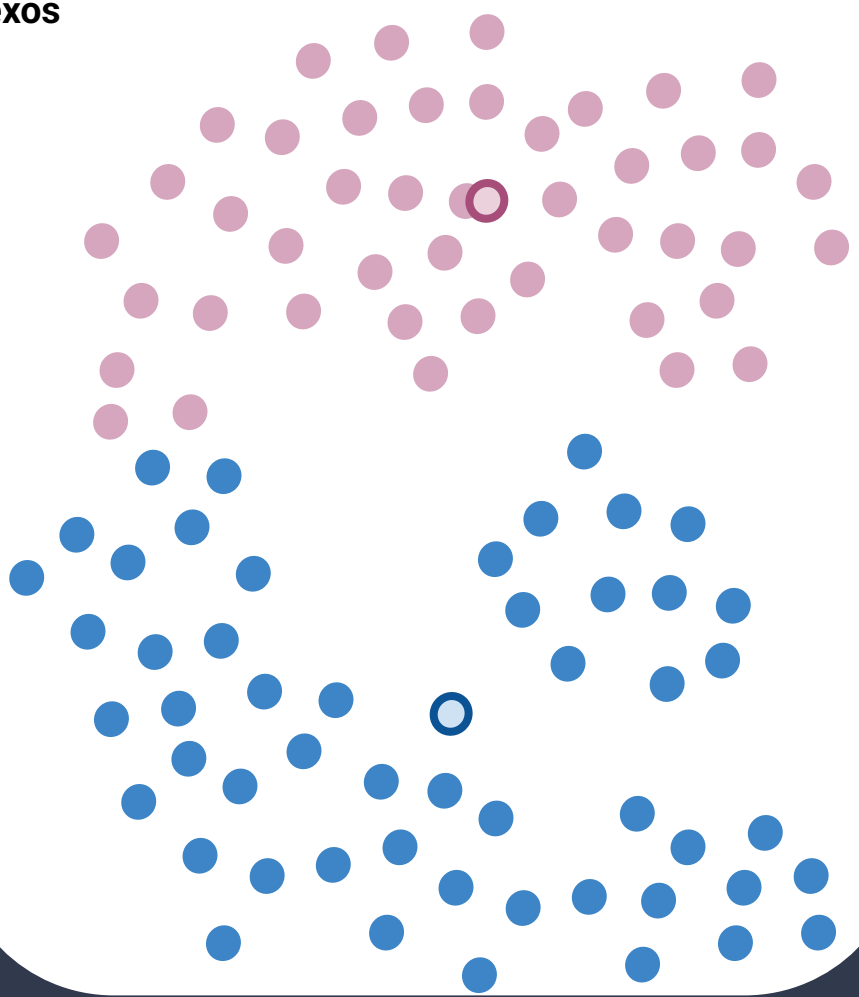
Se puede pre-computar la matriz de distancias, pero dependiendo del *n* puede ser muy demandante en memoria.

Clustering por prototipos (PAM, Partition around Medoids)

- Implementación en scikit-learn: https://scikit-learn-extra.readthedocs.io/en/latest/generated/sklearn_extra.cluster.KMedoids.html#sklearn_extra.cluster.KMedoids

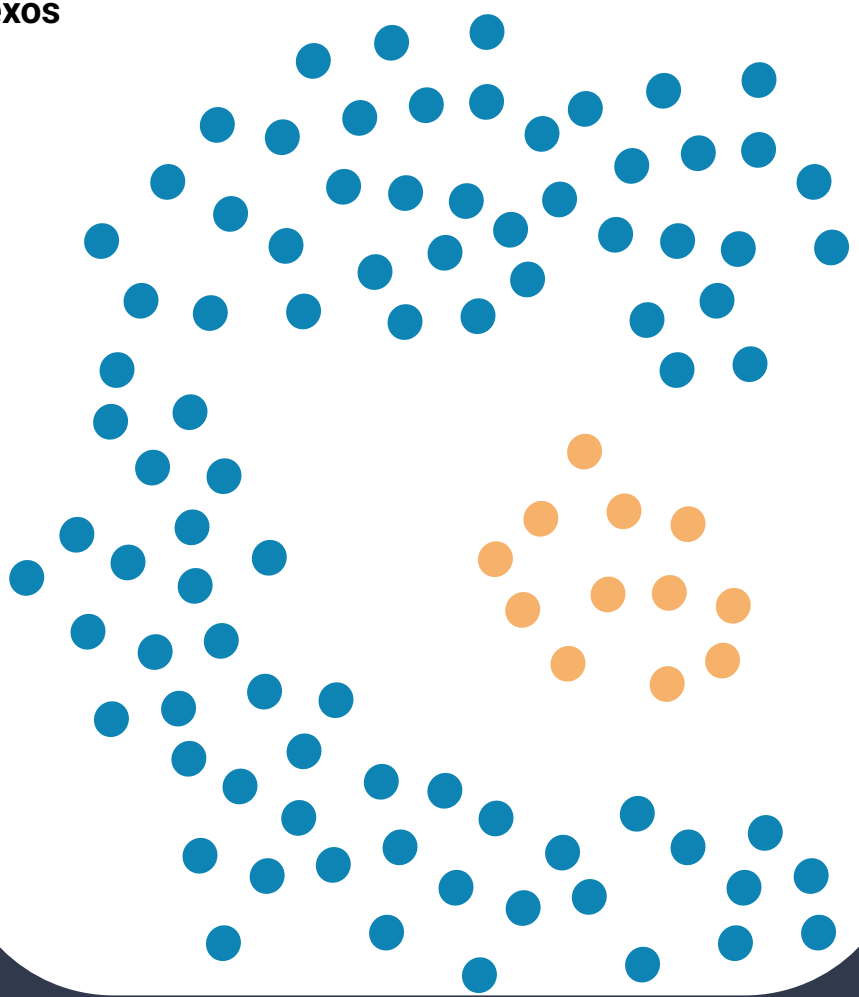
- Implementación a mano: https://github.com/salspaugh/machine_learning/blob/master/clustering/kmedoids.py
  (Seguro se le puede pasar la matriz de disimilaridad que uno quiera)

Clustering por prototipos (k-means) > Clusters no convexos



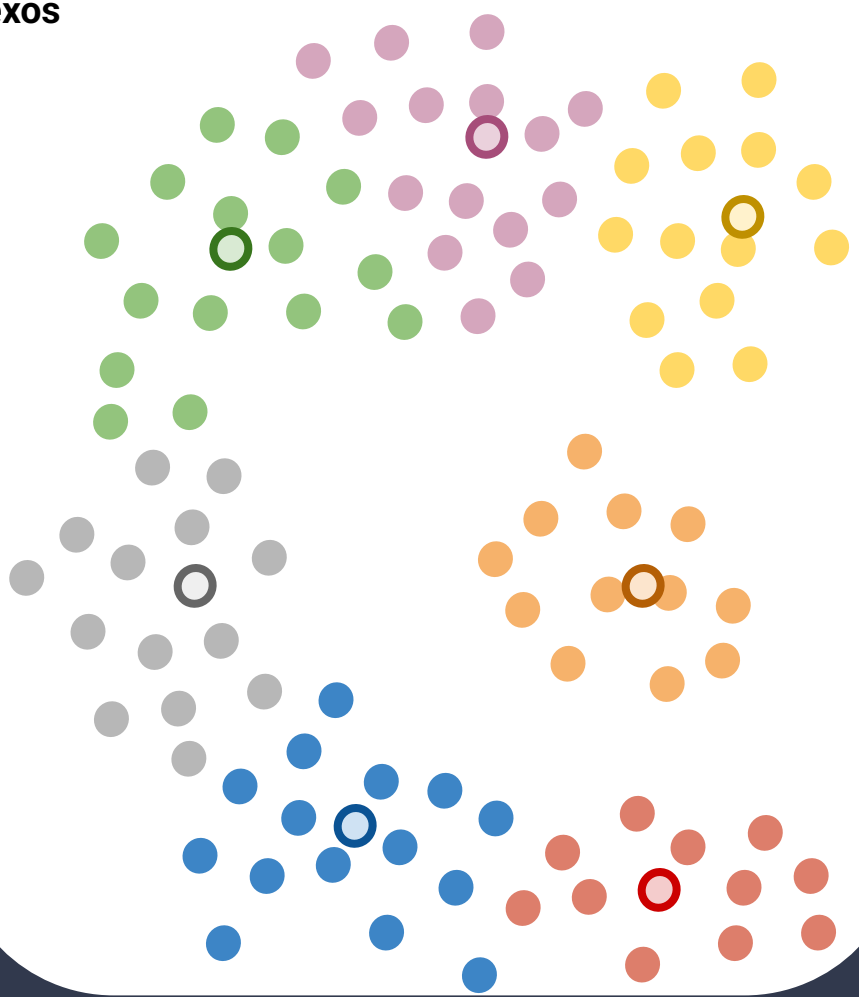
Clustering por prototipos (k-means) > Clusters no convexos

- Siempre, primero, ***VISUALIZACIÓN***
- Aumentar el *k*
- Otro método de clustering



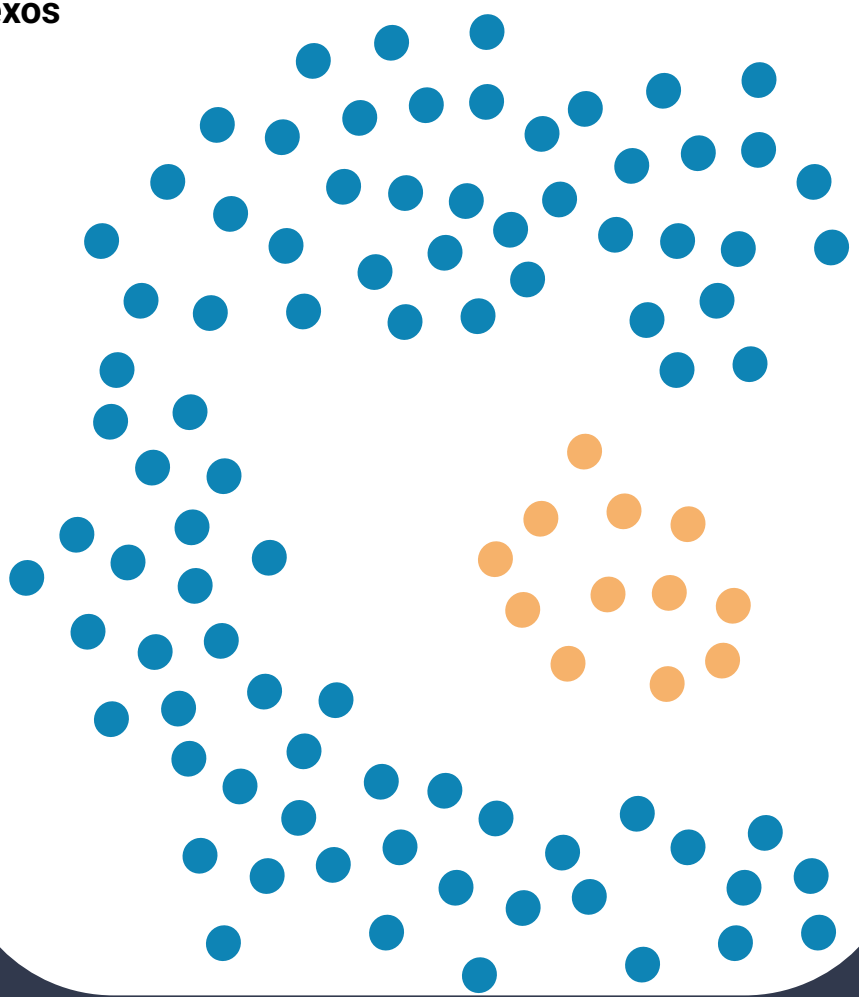
Clustering por prototipos (k-means) > Clusters no convexos

- Siempre, primero, ***VISUALIZACIÓN***
- Aumentar el *k*
- Otro método de clustering



Clustering por prototipos (k-means) > Clusters no convexos

- Siempre, primero, ***VISUALIZACIÓN***
- Aumentar el *k*
- Otro método de clustering



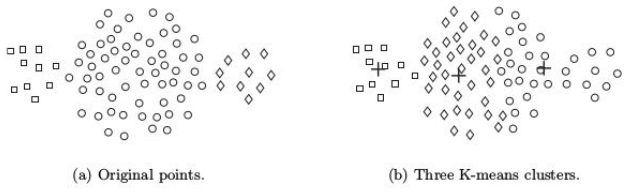
Clustering por prototipos (k-means) > Inicialización

Clustering por prototipos (k-means) > Outliers

Clustering por prototipos (k-means) > Clusters no convexos

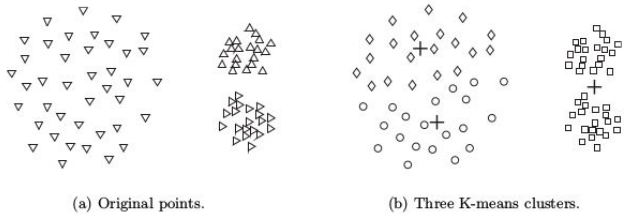
Clustering por prototipos (k-means) > Clusters con distinta densidad

Clustering por prototipos (k-means) > Clusters de distinto tamaño



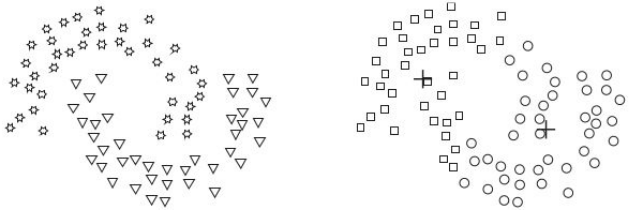
(a) Original points. (b) Three K-means clusters.

Figure 8.9. K-means with clusters of different size.



(a) Original points. (b) Three K-means clusters.

Figure 8.10. K-means with clusters of different density.



(a) Original points. (b) Two K-means clusters.

Figure 8.11. K-means with non-globular clusters.

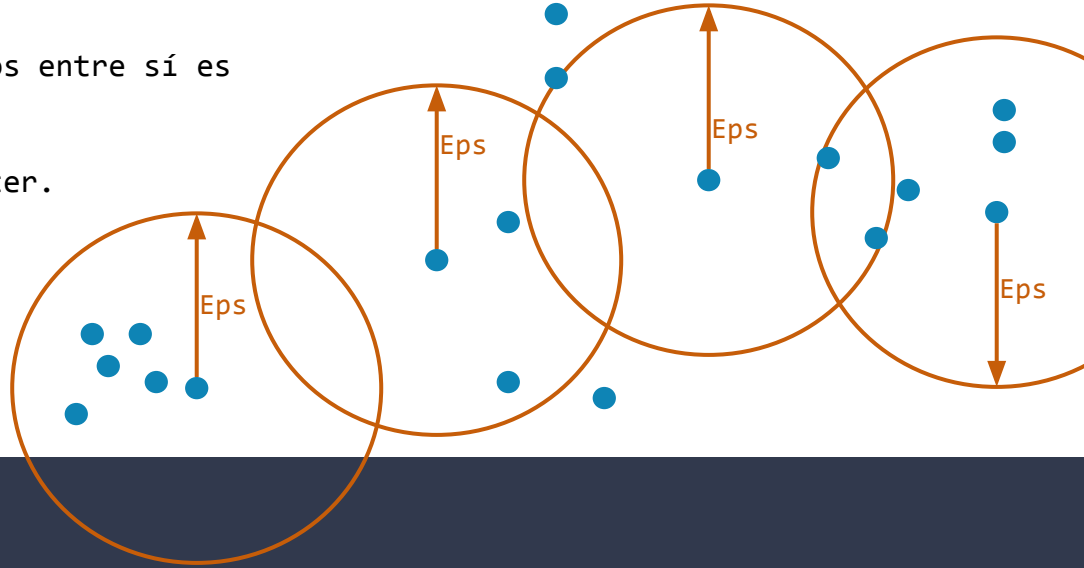
Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

Semilla = Core

Borde = Border

Ruido = Noise

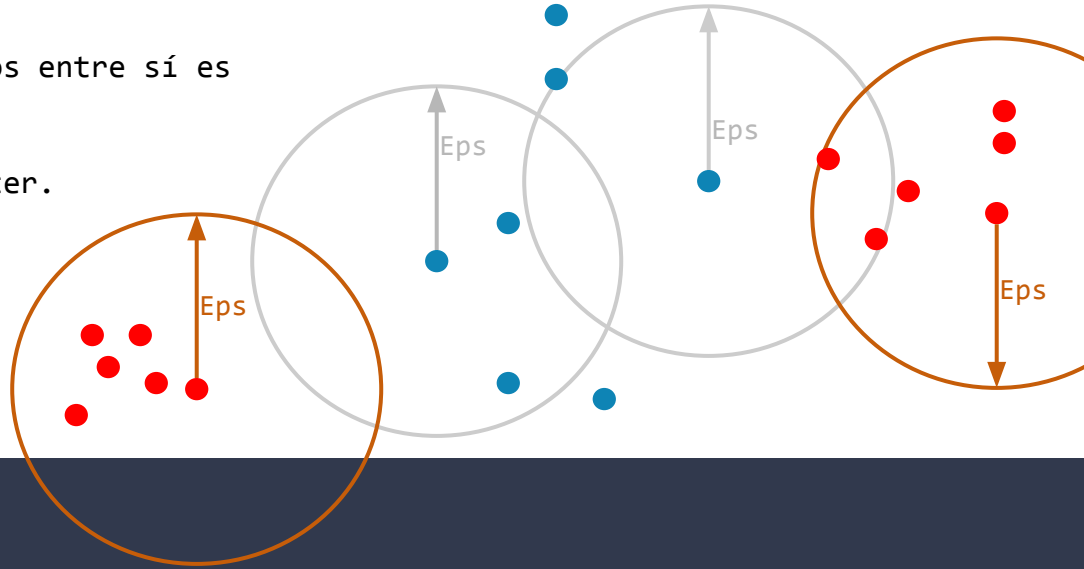


Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

Es **semilla** si tiene al menos MinPts dentro de un radio **Eps**.

Ej. Eps, MinPts = 4

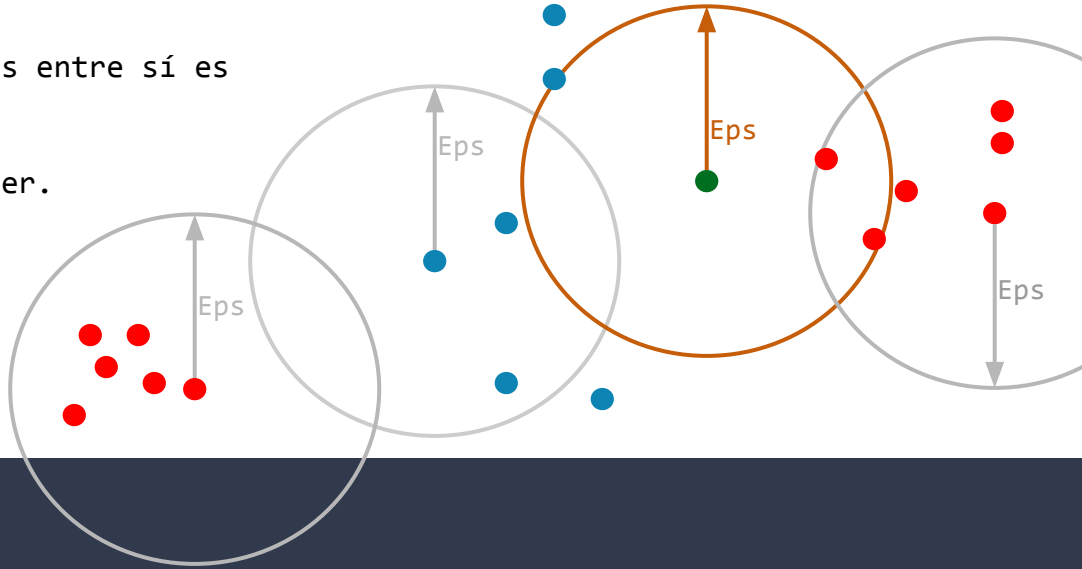


Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

Es **semilla** si tiene al menos MinPts dentro de un radio **Eps**.
Es **borde** si tiene menos de MinPts dentro de un radio **Eps**, y está conectado con puntos semilla.

Ej. Eps, MinPts = 4

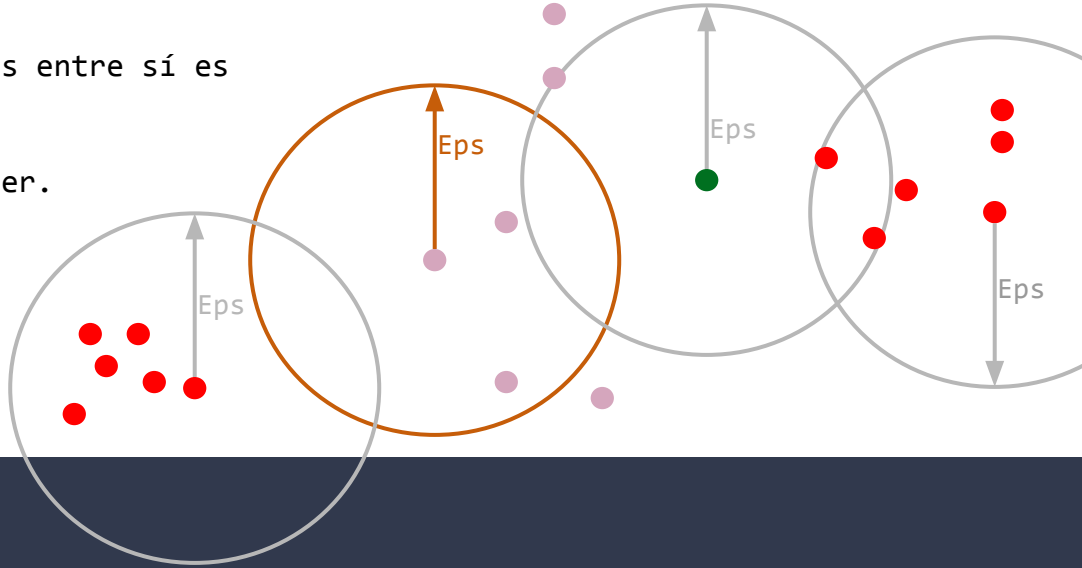


Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

Es **semilla** si tiene al menos **MinPts** dentro de un radio **Eps**.
Es **borde** si tiene menos de **MinPts** dentro de un radio **Eps**, y está conectado con puntos semilla.
Es **ruido** si tiene menos de **MinPts** dentro de un radio **Eps**, y no está conectado con puntos semilla.

Ej. Eps, MinPts = 4

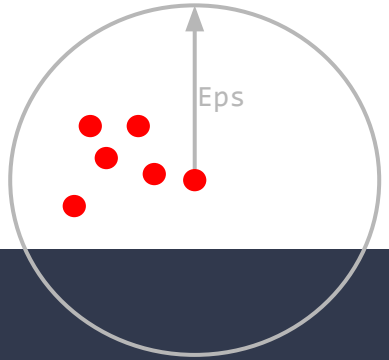


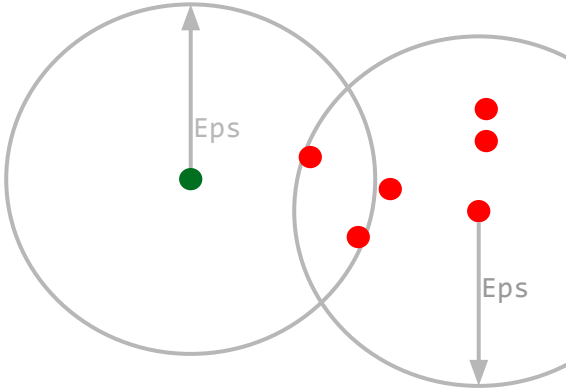
Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

Es **semilla** si tiene al menos **MinPts** dentro de un radio **Eps**.
Es **borde** si tiene menos de **MinPts** dentro de un radio **Eps**, y está conectado con puntos semilla.
Es **ruido** si tiene menos de **MinPts** dentro de un radio **Eps**, y no está conectado con puntos semilla.

Ej. Eps, MinPts = 4



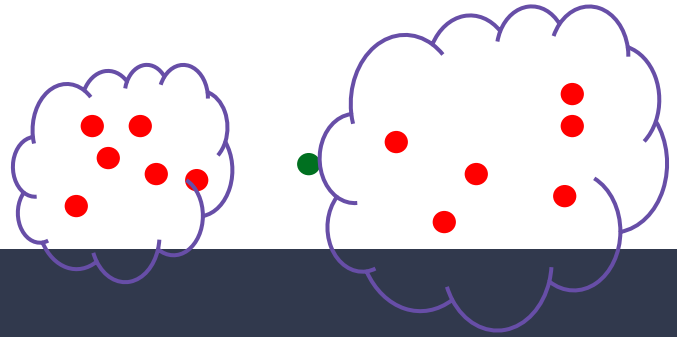


Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar los elementos Borde a algún cluster.

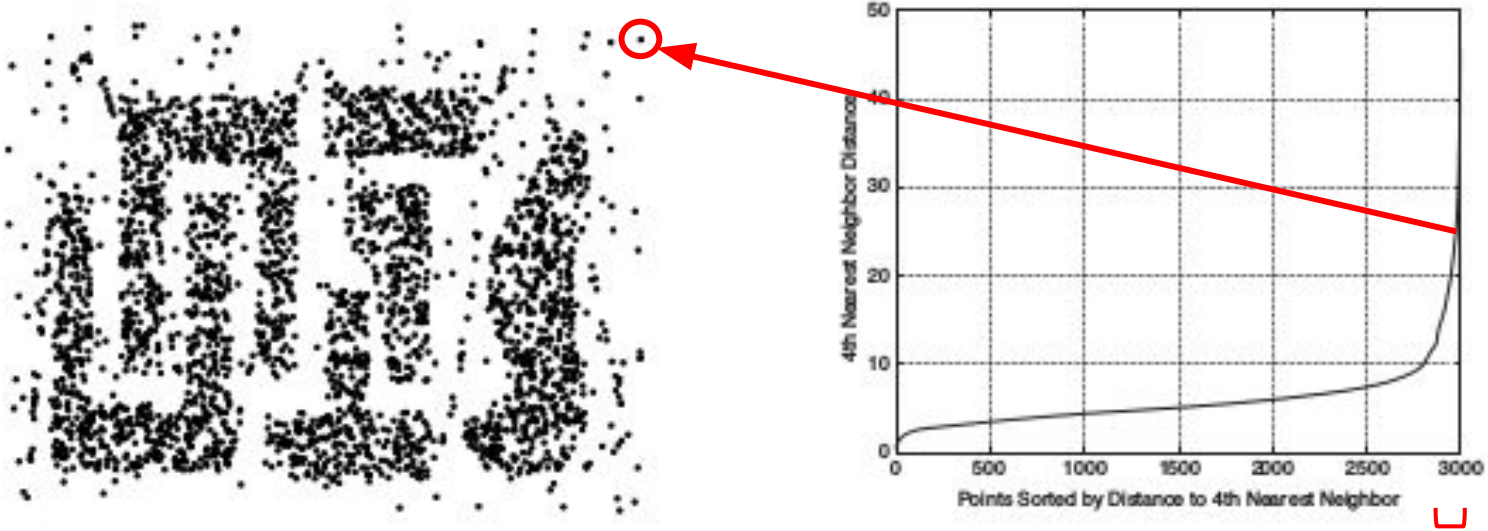
Es **semilla** si tiene al menos **MinPts** dentro de un radio **Eps**.
Es **borde** si tiene menos de **MinPts** dentro de un radio **Eps**, y está conectado con puntos semilla.
Es **ruido** si tiene menos de **MinPts** dentro de un radio **Eps**, y no está conectado con puntos semilla.

Ej. Eps, MinPts = 4



Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.

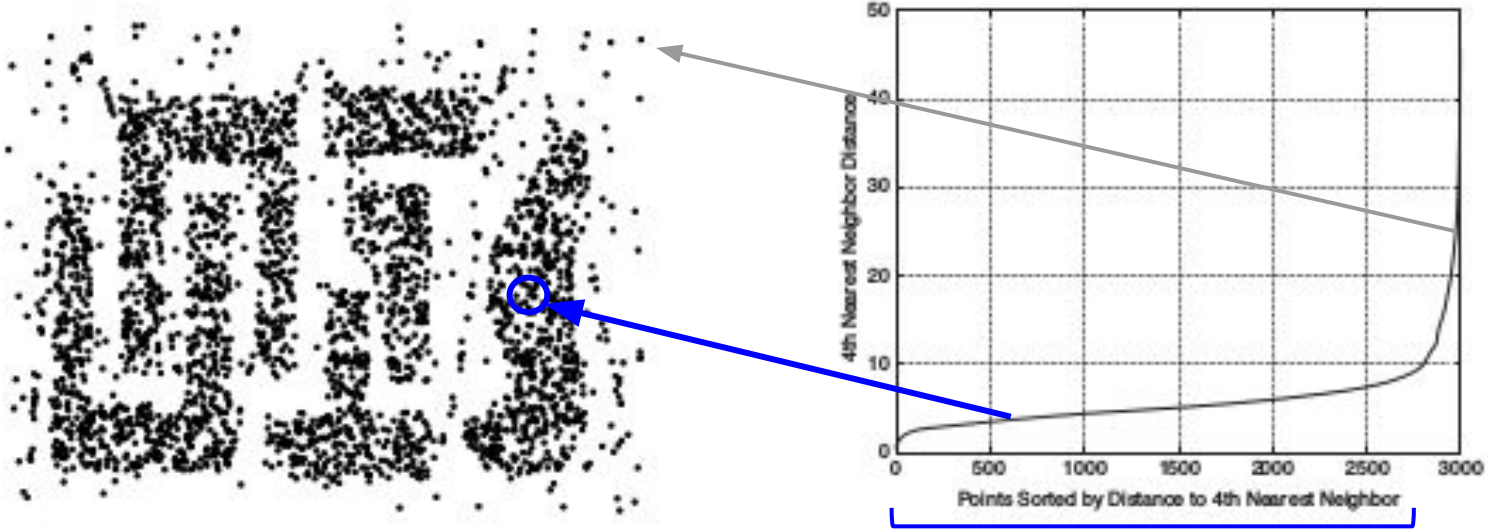


Ruido = Noise

(Pocos) Puntos que están lejos del resto

Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.

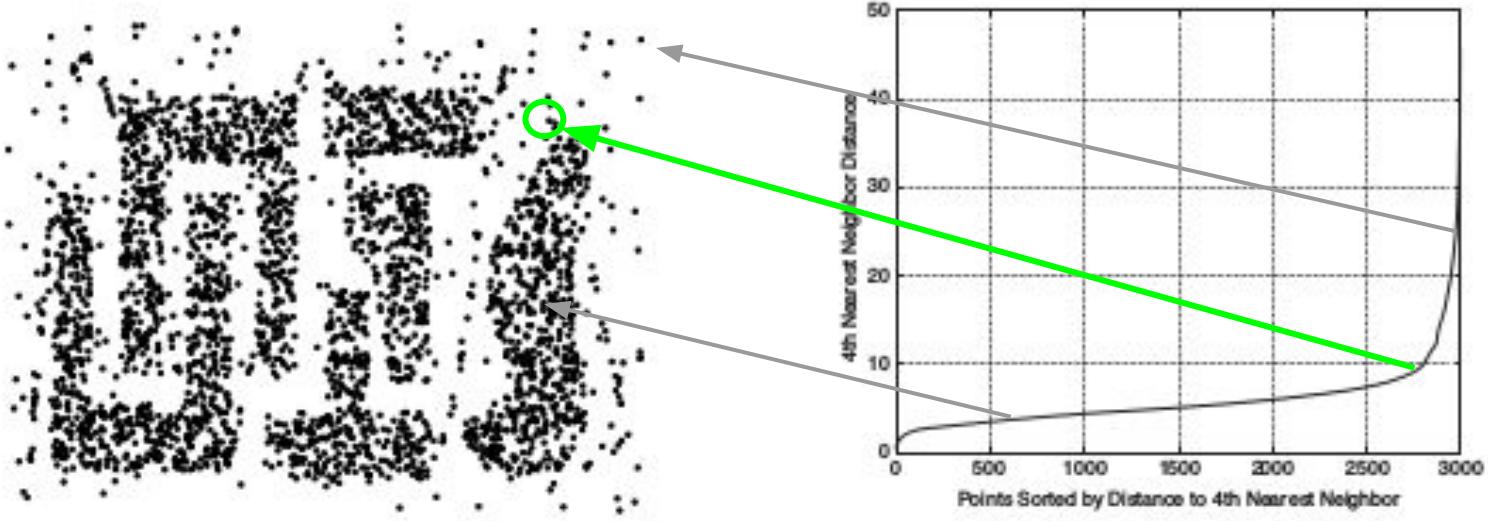


Semilla = Core

(Muchos) Puntos que están cerca de otros

Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.

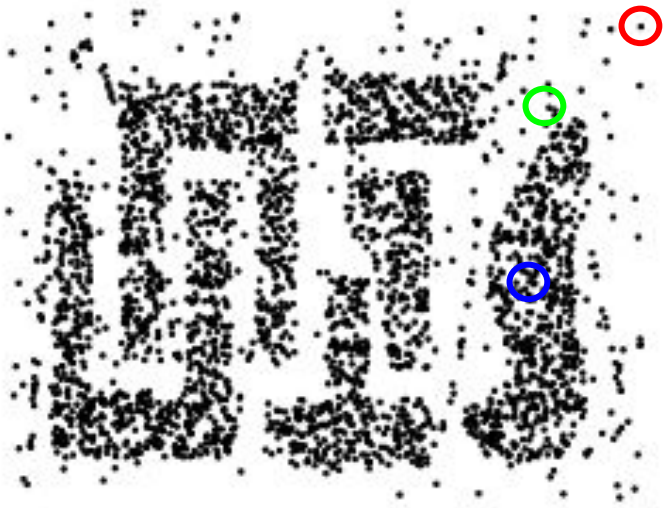


Borde = Border

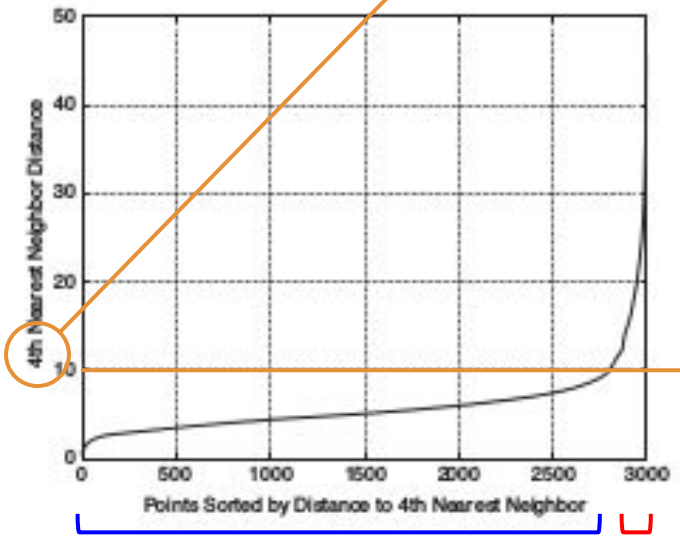
(Algunos) Puntos que están en el límite

Clustering por densidad (DBSCAN)

0. Elegir valores para los parámetros *Eps* y *MinPts*.



MinPts = 4
(este es un valor típico, se pueden explorar varios)

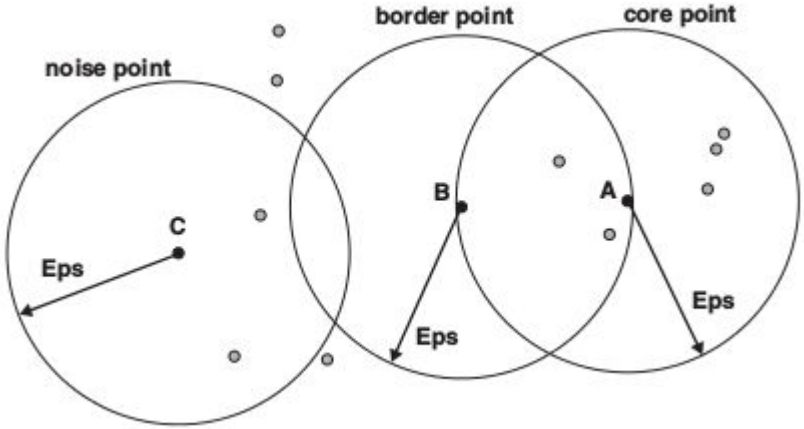


Eps = 10

Clustering por densidad (DBSCAN)

- Puede identificar clusters con formas no esféricas.
- Permite un clustering parcial (eliminando elementos que no pertenecen a ningún cluster).

- Puede tener problemas para identificar clusters con densidades muy distintas (porque se elige un único *Eps*).



Clustering por densidad (DBSCAN)

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

class `sklearn.cluster.` **DBSCAN** (*eps=0.5, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None*) [source]

Perform DBSCAN clustering from vector array or distance matrix.

DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density.

Read more in the User Guide.

Parameters:

eps : *float, optional*

The maximum distance between two samples for one to be considered as in the neighborhood of the other. This is not a maximum bound on the distances of points within a cluster. This is the most important DBSCAN parameter to choose appropriately for your data set and distance function.

min_samples : *int, optional*

The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself.

metric : *string, or callable*

The metric to use when calculating distance between instances in a feature array. If metric is a string or callable, it must be one of the options allowed by `sklearn.metrics.pairwise_distances` for its metric parameter. If metric is "precomputed", X is assumed to be a distance matrix and must be square. X may be a sparse matrix, in which case only "nonzero" elements may be considered neighbors for DBSCAN.

New in version 0.17: metric *precomputed* to accept precomputed sparse matrix.

Eps

MinPts

Se pueden usar varias medidas de similaridad, y se le puede pasar una precomputada.

Clustering jerárquico

- Aglomerativo: Se parte de clusters individuales (*singleton*, hojas) y se van uniendo los más cercanos.
- Divisivo: Se parte de un sólo cluster (*root*, raíz) y se van separando hasta quedarse sólo con los clusters individuales.

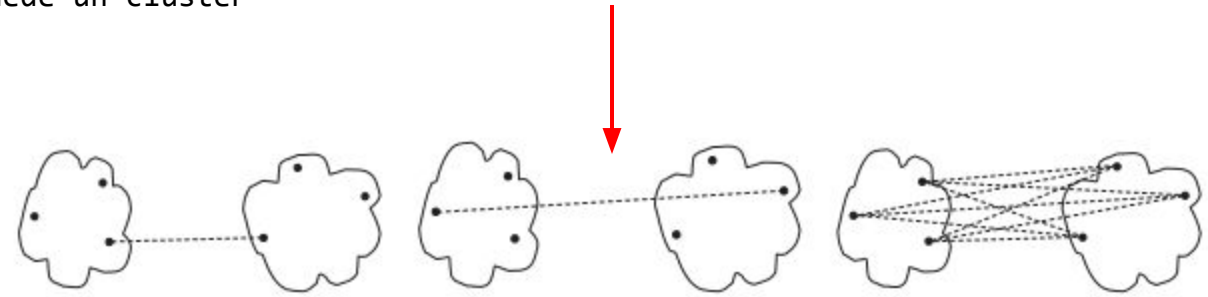
Clustering jerárquico aglomerativo

0. Computar la matriz de similaridad.

Repetir:

1. Juntar los dos más cercanos.
2. Actualizar la matriz de similaridad utilizando el nuevo cluster.

Hasta que: Sólo quede un cluster



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

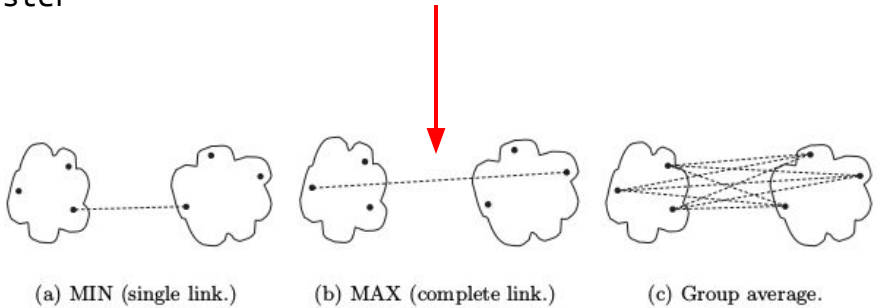
Clustering jerárquico aglomerativo

0. Computar la matriz de similaridad.

Repetir:

1. Juntar los dos más cercanos.
2. Actualizar la matriz de similaridad utilizando el nuevo cluster.

Hasta que: Sólo quede un cluster



Minimizar el incremento de SSE (Ward)

Distancia entre centroides

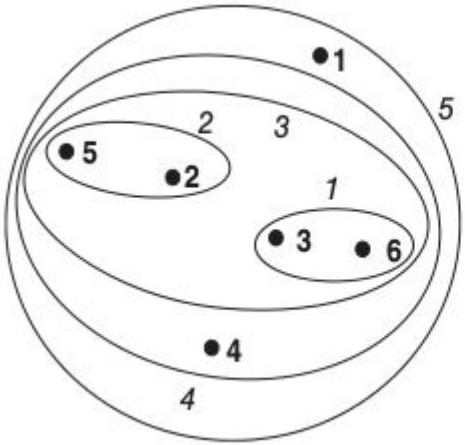
Clustering jerárquico aglomerativo

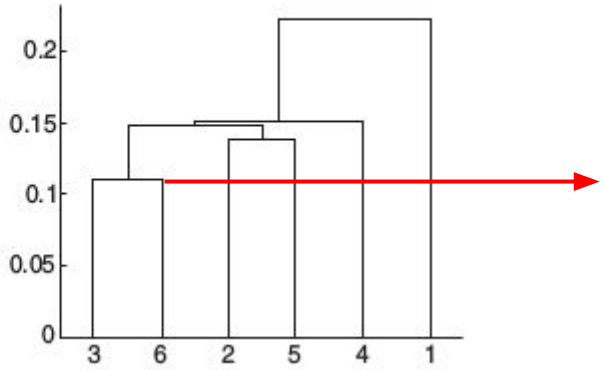
0. Computar la matriz de similaridad.

Repetir:

1. Juntar los dos más cercanos.
2. Actualizar la matriz de similaridad utilizando el nuevo cluster.

Hasta que: Sólo quede un cluster





Distancia entre elementos

Clustering jerárquico aglomerativo

- Se pueden usar nociones de proximidad arbitrarias.

- Minimiza propiedades locales, no globales.
  Por lo que la solución no necesariamente es el mínimo global. Muchas veces el resultado final se utiliza como inicialización de *k-means* u otro por partición, acomodando los resultados al mínimo global con una buena inicialización.

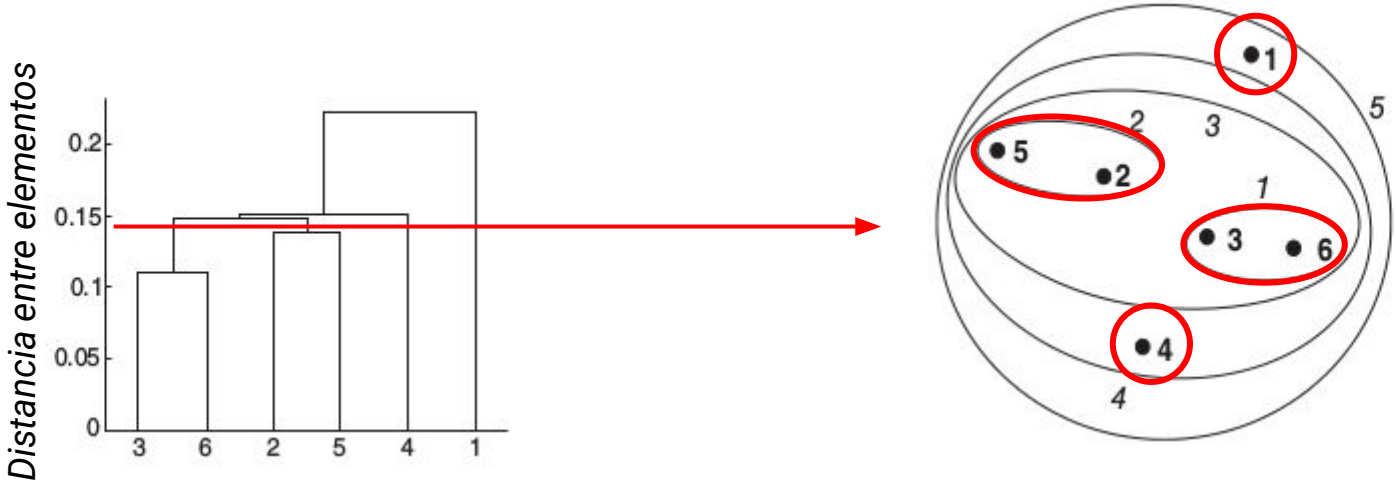
- Aporta una noción de jerarquía además de grupos.

- Para generar los grupos es necesario establecer un criterio de corte.

- Complejidad: $O(N^3)$ es peor para datasets grandes.

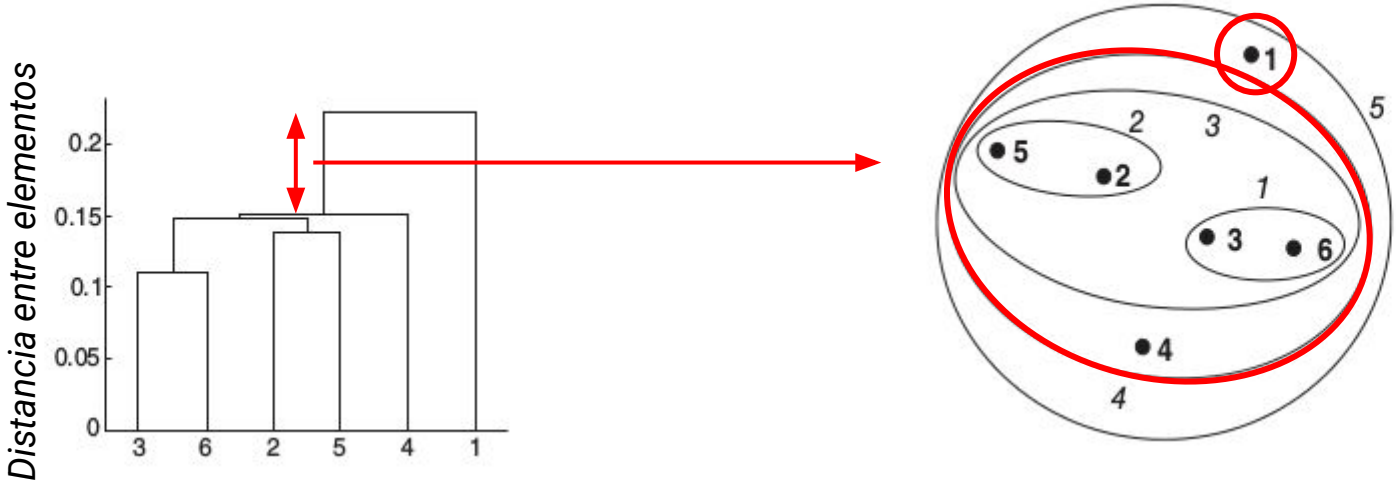
Clustering jerárquico aglomerativo

- Para generar los grupos es necesario establecer un criterio de corte.



Clustering jerárquico aglomerativo

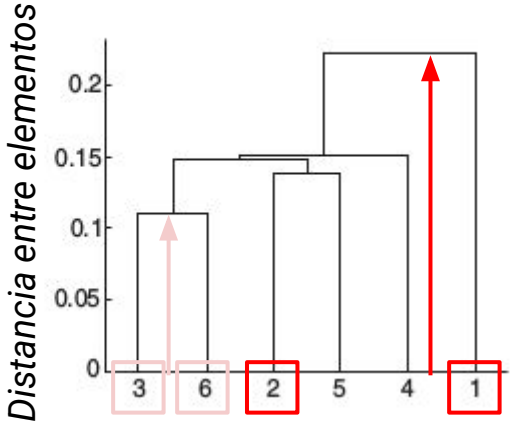
- Para generar los grupos es necesario establecer un criterio de corte.

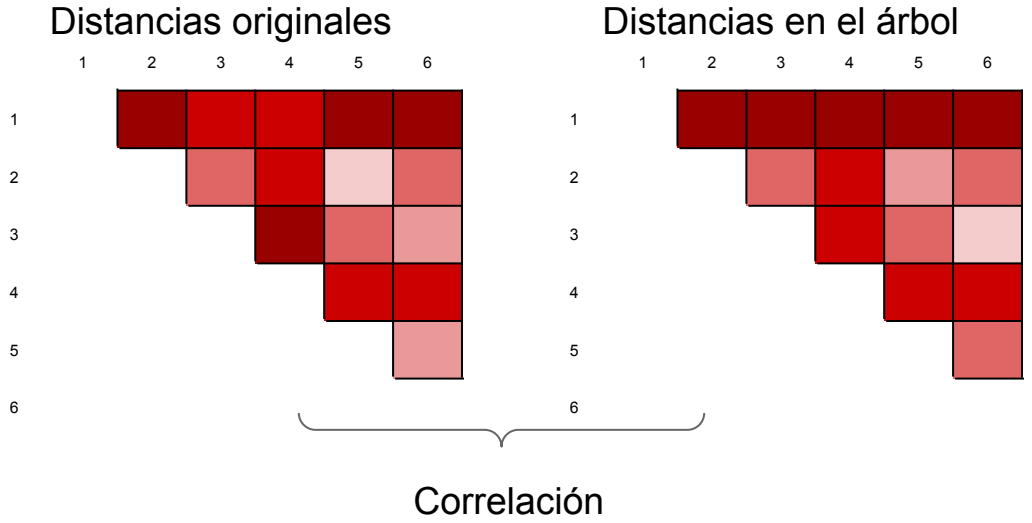


Clustering jerárquico aglomerativo

- Para generar los grupos es necesario establecer un criterio de corte.

  (validar. el Coeficiente de Correlación Cofenético, CoPhenetic Correlation Coefficient, CPCC).





Clustering jerárquico aglomerativo

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

class `sklearn.cluster.` **`AgglomerativeClustering`** (*n_clusters=2, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', pooling_func='deprecated', distance_threshold=None*)

[source]

Agglomerative Clustering

Recursively merges the pair of clusters that minimally increases a given linkage distance.

Read more in the User Guide.

Parameters:

n_clusters : *int or None, optional (default=2)*

The number of clusters to find. It must be `None` if `distance_threshold` is not `None`.

affinity : *string or callable, default: "euclidean"*

Metric used to compute the linkage. Can be "euclidean", "l1", "l2", "manhattan", "cosine", or "precomputed". If linkage is "ward", only "euclidean" is accepted. If "precomputed", a distance matrix (instead of a similarity matrix) is needed as input for the fit method.

Se pueden usar varias medidas de similaridad, y se le puede pasar una precomputada.

Clustering difuso o c-means

https://pythonhosted.org/scikit-fuzzy/

Cada elemento no asignado exclusivamente a un cluster, si no que existe una probabilidad de pertenecer a cada uno de los clusters.

Pesos (Función) de membresía,

$$w_{ij}(z_{ij}) \in [0, 1] \qquad \begin{matrix} i=1...m \text{ registros} \\ j=1...k \text{ clusters} \end{matrix}$$

Para cada elemento x_i

$$\sum_{j=1}^{k} w_{ij} = 1$$

Para cada cluster C_j

$$0 < \sum_{i=1}^{m} w_{ij} < m$$

Clustering difuso o c-means

Algoritmo.

0. Inicializo w_{ij}
1. **Repite**
   2. Calculo/Actualizo c_k a partir de los pesos y los centroides

   3. Calculo/Actualizo w_{ij}

4. **hasta** que los centroides no cambien o SSE<ε
   si no, vuelve a 1.

$$c_j = \frac{\sum_{i=1}^{m} w_{ij}^{p} x_i}{\sum_{i=1}^{m} w_{ij}^{p}}$$

Clustering difuso o c-means

Algoritmo.

0. Inicializo w_{ij}
1. **Repite**
   2. Calculo/Actualizo c_k a partir de los pesos y los centroides
   3. Calculo/Actualizo w_{ij}
4. **hasta** que los centroides no cambien o SSE<ε
   si no, vuelve a 1.

$$c_j = \frac{\sum_{i=1}^{m} w_{ij}^{p} x_i}{\sum_{i=1}^{m} w_{ij}^{p}}$$

$$w_{ij} = \frac{(1/\mathrm{dist}(x_i, c_j)^2)^{\frac{1}{p-1}}}{\sum_{q=1}^{k} (1/\mathrm{dist}(x_i, c_q)^2)^{\frac{1}{p-1}}}$$

Clustering difuso o c-means

Algoritmo.

0. Inicializo w_{ij}
1. **Repite**
   2. Calculo/Actualizo c_k a partir de los pesos y los centroides
   3. Calculo/Actualizo w_{ij}
4. **hasta** que los centroides no cambien o SSE<ε
   si no, vuelve a 1.

$$c_j = \frac{\sum_{i=1}^{m} w_{ij}^p x_i}{\sum_{i=1}^{m} w_{ij}^p}$$

$$w_{ij} = \frac{(1/\text{dist}(x_i, c_j)^2)^{\frac{1}{p-1}}}{\sum_{q=1}^{k} (1/\text{dist}(x_i, c_q)^2)^{\frac{1}{p-1}}}$$

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^p \text{dist}(x_i, c_j)^2$$

Clustering difuso o c-means

Algoritmo.

0. Inicializo w_{ij}
1. **Repite**
   2. Calculo/Actualizo c_k a partir de los pesos y los centroides
   3. Calculo/Actualizo w_{ij}
4. **hasta** que los centroides no cambien o SSE<ε
   si no, vuelve a 1.

$$c_j = \frac{\sum_{i=1}^{m} w_{ij}^{p} x_i}{\sum_{i=1}^{m} w_{ij}^{p}}$$

p=2

$$w_{ij} = \frac{(1/\mathrm{dist}(x_i, c_j)^2)^{\frac{1}{p-1}}}{\sum_{q=1}^{k} (1/\mathrm{dist}(x_i, c_q)^2)^{\frac{1}{p-1}}}$$

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p} \mathrm{dist}(x_i, c_j)^2$$

Clustering difuso o c-means

https://pythonhosted.org/scikit-fuzzy/

- Cada elemento no asignado exclusivamente a un cluster, si no que existe una probabilidad de pertenecer a cada uno de los clusters.

- Permite asignar varios clusters a un elemento (p.ej. cuando un individuo tiene distintos roles en una organización, o un gen participa de distintas funciones), también permite separar objetos en una imagen con bordes difusos (no asignarlos a ningún cluster).

- Tiene desventajas similares a k-means, además de que es mucho más demandante computacionalmente.

Próxima clase

- Medidas de Distancias / Proximidad / Similitud / Disimilaridad / ...
- Métodos de validación interna (no supervisado)
- Métodos de validación externa (supervisado)

Distancias

Medidas de Proximidad, Disimilaridad, Similaridad, Distancias
(http://www.iiisci.org/journal/CV$/sci/pdfs/GS315JG.pdf)

- Distancias
  - métricas de Minkowski: Manhattan (L1), Euclídea (L2), …
  - distancia Canberra
  - distancia de Mahalanobis
- Ángulos
  - distancia coseno
  - Correlación de Pearson
- Variables binarias:
  - Coeficiente de coincidencias
  - Coeficiente de Jaccard
- Multiestado
- Mixtos
- ...

Matrices de Disimilaridad
Representan relaciones entre los N elementos del conjunto, entonces son matrices de N x N.

Las matrices de disimilaridad (**D**) deben cumplir con varias propiedades:

- La disimilaridad de dos objetos idénticos (o consigo mismo) es cero, **d(a,a) = 0**. Y en general, las disimilaridades solo pueden ser mayores o iguales que cero, **d(a,b) ≥ 0**.

- Los objetos no idénticos pueden ser distinguibles o no.
  Si **a ≠ b** entonces **d(a,b) ≥ 0**.
  Si **a = b** entonces **d(a,b) = 0**.

- Las medidas de disimilaridad son simétricas: **d(a, b) = d(b, a)**.

Matrices de Disimilaridad

Representan relaciones entre los N elementos del conjunto, entonces son matrices de N x N. Las matrices de disimilaridad (**D**) deben cumplir con varias propiedades:

- Si **a ≠ b** entonces **d(a,b) ≥ 0**.
- Si **a = b** entonces **d(a,b) = 0** (o si son el mismo objeto entonces **d(a,a) = 0**)
- **d(a, b) = d(b, a)**.
    - Si además se cumple la desigualdad triangular:

      d(a,b) ≤ d(a,c) + d(b,c)

      entonces tenemos una **matriz de disimilaridad métrica**.
    - Cuando la medida de disimilitud no cumple con la desigualdad triangular, pero satisface la desigualdad ultramétrica:

      d(a, b) ≤ max{d(a, c), d(c, b)}

      entonces tenemos una **matriz de disimilaridad ultramétrica**. Este es el tipo de distancias que ocurren al representar gráficamente un cluster jerárquico con un dendrograma.

Matrices de Similaridad

Una **matriz de similaridad** (S) es aquella donde s(a,a) = 1. En algunos casos se puede transformar una matriz de disimilaridad (D) en una de similaridad (S):

- Si el dominio de la similaridad es [0, 1]:

$$d(a, b) = 1 - s(a,b)$$

- Si el dominio de S es [-1, 1] y s = -1 se corresponde con la mayor distancia normalizada:

$$d(a, b) = 1 - (s(a, b) + 1)/2$$

Matrices de Proximidad, Similaridad, Disimilaridad y Distancia y Afinidad ...
¡Ojo! ¡La diferencia entre estos términos depende muchas veces del dominio!

En general,

una **matriz de disimilaridad métrica** ~ una **matriz de distancia**

una **matriz de afinidad** ~ una **matriz de similaridad**, con la particularidad de que

afinidad(a,b) ~ exp(- d(a,b)^2) ⇨ si d(a,b) = 0, afinidad(a,b) = 1
si d(a,b) >> 1, afinidad(a,b) ~ 0

Matrices de Similaridad y Disimilaridad para variables BINARIAS

Matrices de Similaridad y Disimilaridad para variables BINARIAS

Supongamos que tenemos dos objetos para los cuales se registran los valores que toman diferentes variables binarias (Verdadero/Falso, 0/1, +/-. etc.):

	var.0	var.1	var.2	var.3	var.4	var.5	var.6	var.7
item.x	1	1	1	1	0	0	0	1
item.y	1	1	0	0	1	0	0	0

Podemos definir estas cantidades:

- **a**: cantidad de variables con **1** tanto para **x** como para **y**
- **b**: cantidad de variables con **1** para **x** y **0** para **y**
- **c**: cantidad de variables con **0** para **x** y **1** para **y**
- **d**: cantidad de variables con **0** tanto para **x** como para **y**

p = a + b + c + d

Matrices de Similaridad y Disimilaridad para variables BINARIAS

Supongamos que tenemos dos objetos para los cuales se registran los valores que toman diferentes variables binarias (Verdadero/Falso, 0/1, +/-. etc.):

	var.0	var.1	var.2	var.3	var.4	var.5	var.6	var.7
item.x	1	1	1	1	0	0	0	1
item.y	1	1	0	0	1	0	0	0

Podemos definir estas cantidades:

a: cantidad de variables con **1** tanto para **x** como para **y** → **a = 2**
b: cantidad de variables con **1** para **x** y **0** para **y** → **b = 3**
c: cantidad de variables con **0** para **x** y **1** para **y** → **c = 1**
d: cantidad de variables con **0** tanto para **x** como para **y** → **d = 2**

p = a + b + c + d → **p = 8**

Matrices de Similaridad y Disimilaridad para variables BINARIAS

	var.0	var.1	var.2	var.3	var.4	var.5	var.6	var.7
item.x	1	1	1	1	0	0	0	1
item.y	1	1	0	0	1	0	0	0

Podemos definir estas cantidades:

a: x ~ 1 & y ~ 1 → **a = 2**
b: x ~ 1 & y ~ 0 → **b = 3**
c: x ~ 0 & y ~ 1 → **c = 1**
d: x ~ 0 & y ~ 0 → **d = 2**
p = a + b + c + d → **p = 8**

Para calcular la (di)similaridad entre **x** e **y** hay diferentes opciones. Dos de las más conocidas son:

➔ Coeficiente de coincidencias

(similaridad) **s(x,y) = (a + d)/p** (está acotada en [0,1]... ¡probar los límites!)
(disimilaridad) **d(x,y) = (b + c)/p** (está acotada en [0,1]... ¡probar los límites!)

➔ Coeficiente de Jaccard

(similaridad) **s(x,y) = a / (a + b +c)** (está acotada en [0,1]... ¡probar los límites!)
(disimilaridad) **d(x,y) = (b + c) / (a + b + c)** (está acotada en [0,1]... ¡probar los límites!)

Matrices de Similaridad y Disimilaridad para variables BINARIAS

	var.0	var.1	var.2	var.3	var.4	var.5	var.6	var.7
item.x	1	1	1	1	0	0	0	1
item.y	1	1	0	0	1	0	0	0

Podemos definir estas cantidades:

a: x ~ 1 & y ~ 1 → **a = 2**
b: x ~ 1 & y ~ 0 → **b = 3**
c: x ~ 0 & y ~ 1 → **c = 1**
d: x ~ 0 & y ~ 0 → **d = 2**
p = a + b + c + d → **p = 8**

El **coeficiente de coincidencias** le asigna importancia a aquellos casos a la coincidencia de valores **1-1**, **V-V**, etc. y también a aquellos donde la coincidencia es **0-0**, **F-F**, etc.

El **coeficiente de Jaccard**, en cambio, sólo considera en el numerador aquellos casos donde la coincidencia es **1-1** o **V-V**.

Dependiendo del problema y dominio de aplicación, una medida puede ser más adecuada que otra.

Matrices de Similaridad y Disimilaridad para variables BINARIAS

	var.0	var.1	var.2	var.3	var.4	var.5	var.6	var.7
item.x	1	1	1	1	0	0	0	1
item.y	1	1	0	0	1	0	0	0

Podemos definir estas cantidades:

a: x ~ 1 & y ~ 1 → **a = 2**
b: x ~ 1 & y ~ 0 → **b = 3**
c: x ~ 0 & y ~ 1 → **c = 1**
d: x ~ 0 & y ~ 0 → **d = 2**
p = a + b + c + d → **p = 8**

¡Existe un gran número de medidas de (di)similaridad desarrolladas para variables binarias! Por ejemplo, *Choi, S. S., Cha, S. H., & Tappert, C. C. (2010). A survey of binary similarity and distance measures. Journal of Systemics, Cybernetics and Informatics, 8(1), 43-48.*
http://www.iiisci.org/journal/CV$/sci/pdfs/GS315JG.pdf

Medidas para variables CONTINUAS

Medidas para variables cuantitativas (continuas)

Métricas de Minkowski:

$$d(x,y) = \left(\sum_{k=1}^{p} w_k^{\lambda} |x_k - y_k|^{\lambda} \right)^{1/\lambda}$$

Distancia Euclídea (*λ=2*):

$$d(x,y) = \sqrt{\sum_{k=1}^{p} |x_k - y_k|^2}$$

Distancia Manhattan (*λ=1*):

$$d(x,y) = \sum_{k=1}^{p} |x_k - y_k|$$

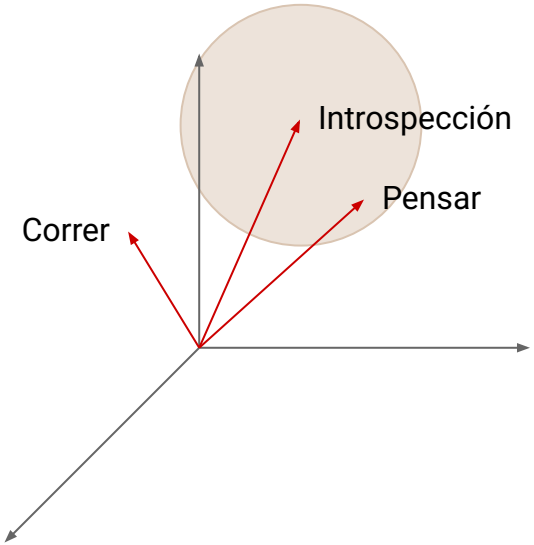
Los valores w_k son pesos que suelen aplicarse para que las variables estén estandarizadas u acotadas en [0,1].

Otras distancias menos utilizadas son la de Canberra o Mahalanobis.

Medidas para comparar los ángulos de los vectores

A veces los valores precisos que toman las variables no son tan importantes en cuanto a cómo afectarán a las distancias entre objetos. El interés, en cambio, se enfoca en la comparación de las direcciones de los vectores que definen a cada objeto en el espacio multidimensional de las variables.

El coeficiente de correlación de Pearson y la similitud coseno son dos medidas de similitud que sirven para comparar la separación angular entre objetos.



Medidas para comparar los ángulos de los vectores

Correlación de Pearson para dos objetos x e y:

$$S(x,y) = \frac{\sum_{k=1}^{p}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{p}(x_k - \bar{x})^2 \sum_{k=1}^{p}(y_k - \bar{y})^2}}$$

El coeficiente está acotado en el intervalo [-1,1] y está centrado, es decir que es invariante a desplazamientos.

SPOILER: Ideas parecidas vamos a usar para construir las distancias en el grafo de conexiones entre regiones cerebrales en el TP de la segunda parte de la materia.

Medidas para comparar los ángulos de los vectores

Similitud coseno:

$$s(x,y) = \cos(\phi) = \frac{a \,.\, b}{\|a\| \, \|b\|} = \frac{\sum_{k=1}^{p} x_k y_k}{\sqrt{\sum_{k=1}^{p} x_k^2 \sum_{k=1}^{p} y_k^2}}$$

donde $a \cdot b$ es el producto interno del vector a y el vector b, y $\|a\|$ o $\|b\|$ es la norma cuadrada o norma 2 de los vectores a y b respectivamente.

NOTA: Un ejemplo muy de moda últimamente son las dirección en el espacio semántico de las palabras (a partir de los *word embeddings*).

Medidas para variables MULTIESTADO

Medidas para variables categóricas multiestado y para variables ordinales
Consideramos variables categóricas multiestado a aquellas que presentan dos o más estados o categorías (más que binarias). Si estás categorías presentan algún tipo de ordenamiento, la variable es categórica ordinal, o simplemente ordinal.

Variables categóricas multiestado:

- Si el número de estados es chico una solución sencilla es separar la variable multiestado original en varias variables binarias, una para cada categoría (one-hot encoding) [problema: muchos estados=muchas variables!!].

Medidas para variables categóricas multiestado y para variables ordinales

Consideramos variables categóricas multiestado a aquellas que presentan dos o más estados o categorías (más que binarias). Si estás categorías presentan algún tipo de ordenamiento, la variable es categórica ordinal, o simplemente ordinal.

Variables categóricas multiestado:

- Si el número de estados es chico una solución sencilla es separar la variable multiestado original en varias variables binarias, una para cada categoría (one-hot encoding) [problema: muchos estados=muchas variables!!].
- Otra alternativa es calcular un criterio de coincidencias:

$$S(x,y) = \frac{1}{r}\sum_{l=1}^{r} S_{xyl}$$

Donde la similaridad entre *x* e *y* son objetos multidimensionales de dimensión *r*.

$$S_{xyl}=0 \qquad si\ x_l \neq y_l$$
$$S_{xyl}=w \qquad si\ x_l = y_l$$

w es típicamente 1, aunque si se quieren premiar las coincidencias se puede aumentar.

Medidas para variables ordinales

Se puede considerar que la disimilaridad será proporcional a la cantidad de “saltos” entre estados. Por ejemplo, suponiendo que se tienen 3 estados: A < B < C, y que la distancia entre dos estados cualquiera es 1 ⇨ podemos decir que A=0, B=1, C=2. La distancia entre dos estados *x* e *y* es:

$$d(x, y) = |x - y|^r$$

donde típicamente $r=1$. Así $d(A,B) = d(B,C) = 1$ y $d(A,C)=2$. Luego, estas medidas suelen ser normalizadas al rango [0, 1], en este caso:

$$d(x, y) = \frac{|x - y|^r}{2}$$

Para muchas variables ordinales, esta medida puede generalizarse a:

$$d(x, y) = \frac{\sum_{k=1}^{p} d_k(x, y)}{p}$$

Medidas para comparar variables de tipos mixtos

En común tener que calcular matrices de distancia a partir de un conjunto de variables con variables de diferentes tipos, binarias, categóricas, continuas. Una forma simple de combinarlas es el *coeficiente de similaridad de Gower*:

$$S(x,y) = \frac{\sum_{k=1}^{p} S_{xyk}\delta_{xyk}}{\sum_{k=1}^{p} \delta_{xyk}}$$

donde S_{xyk} es la similaridad entre x e y para la variable k (en la métrica que corresponda), y δ_{xyk} toma valores 0 o 1 según esta comparación esté presente o no.

El coeficiente de disimilaridad se toma como $1\text{-}S(x,y)$, y muchas veces la noción de distancia se extiende a $d(x,y)^2 = 1\text{-}S(x,y)$.

Esta distancia en el fondo se calcula tomando el promedio entre las distancias de todas las variables calculadas en cada caso como corresponda.

Medidas para comparar variables de tipos mixtos

En común tener que calcular matrices de distancia a partir de un conjunto de variables con variables de diferentes tipos, binarias, categóricas, continuas. Una forma simple de combinarlas es el *coeficiente de similaridad de Gower*:

$$S(x,y) = \frac{\sum_{k=1}^{p} S_{xyk}\delta_{xyk}}{\sum_{k=1}^{p} \delta_{xyk}}$$

donde S_{xyk} es la similaridad entre x e y para la variable k (en la métrica que corresponda), y δ_{xyk} toma valores 0 o 1 según esta comparación esté presente o no.

Típicamente,

S_{xyk} (continua) → Manhattan
S_{xyk} (binaria) → Dice (DSC = 2*a / (2*a + b +c))

Python

sklearn: Todas métricas sobre datos continuos
https://scikit-learn.org/stable/modules/metrics.html#metrics
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html

scipy: También tiene sobre binarios como Jaccard
https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html#scipy.spatial.distance.pdist

gower: En ninguno de los dos hay una implementación para datos mixtos o categóricos como Gower.
https://pypi.org/project/gower/

Validación

Métodos de validación: ¿Por qué es importante validar?

- Siempre vamos a encontrar grupos PERO ¿Son buenos grupos?
- Determinar el mejor número de clusters (k).
- Comparar métodos de Clustering (sobre los mismos datos).

Métodos de validación: ¿Por qué es importante validar? Validación Externa vs Validación Interna

- Siempre vamos a encontrar grupos PERO ¿Son buenos grupos?
  - Antes de empezar a buscar es importante verificar si existe una ***tendencia al clustering***.
- Determinar el mejor número de clusters (k).
- Comparar métodos de Clustering (sobre los mismos datos).

Métodos de validación: ¿Por qué es importante validar? Validación Externa vs Validación Interna

- Siempre vamos a encontrar grupos PERO ¿Son buenos grupos?
  - Antes de empezar a buscar es importante verificar si existe una ***tendencia al clustering***.

- Determinar el mejor número de clusters (k).
  - Determinar cuán buenos son los grupos (***externa***/***interna***).
  - Determinar cuán buena es la separación entre ellos (***externa***/***interna***).
  - Constatar el número de clusters con las etiquetas *a priori* (si existen) (***externa***).

- Comparar métodos de Clustering (sobre los mismos datos).
  - Determinar cuán buenos son los grupos (***externa***/***interna***).
  - Determinar cuán buena es la separación entre ellos (***externa***/***interna***).
  - Constatar la pertenencia a los grupos con las etiquetas *a priori* (si existen) (***externa***).

Métodos de validación: ¿Por qué es importante validar? Validación Externa vs Validación Interna

- Siempre vamos a encontrar grupos PERO ¿Son buenos grupos?
  - Antes de empezar a buscar es importante verificar si existe una ***tendencia al clustering***.

- Determinar el mejor número de clusters (k).
  - Determinar cuán buenos son los grupos (***externa***/***interna***).
  - Determinar cuán buena es la separación entre ellos (***externa***/***interna***).
  - Constatar el número de clusters con las etiquetas *a priori* (si existen) (***externa***).

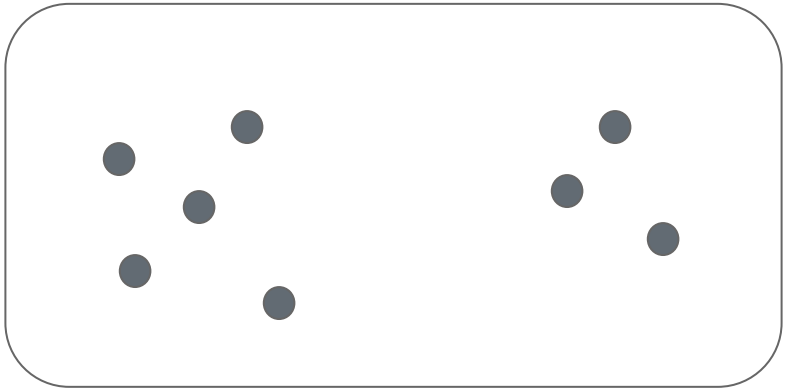
- Comparar métodos de Clustering (sobre los mismos datos).
  - Determinar cuán buenos son los grupos (***externa***/***interna***).
  - Determinar cuán buena es la separación entre ellos (***externa***/***interna***).
  - Constatar la pertenencia a los grupos con las etiquetas *a priori* (si existen) (***externa***).

Existen diferentes medidas de validación y no existe un criterio único para determinar cuál es la mejor.

No hay una medida única que se pueda usar para todos los métodos de clustering.

Tendencia al Clustering

Estadístico (coeficiente) de Hopkins



elementos del dataset

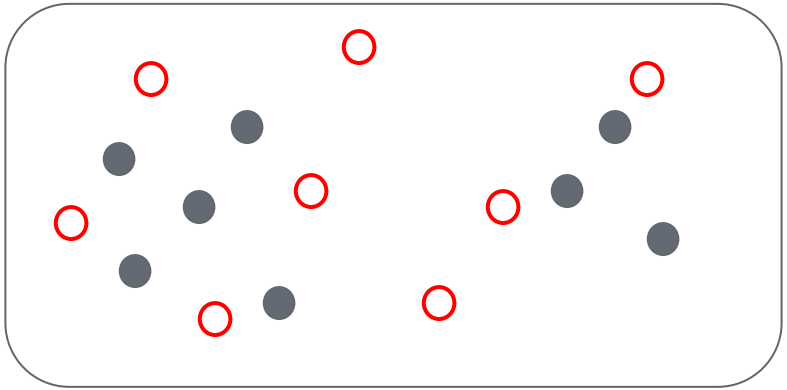
Tendencia al Clustering

Estadístico (coeficiente) de Hopkins

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

w_i = distancia de un elemento i al azar al vecino más cercano

u_i = distancia de un punto i agregado al azar al vecino más cercano



elementos del dataset

elementos agregados al azar

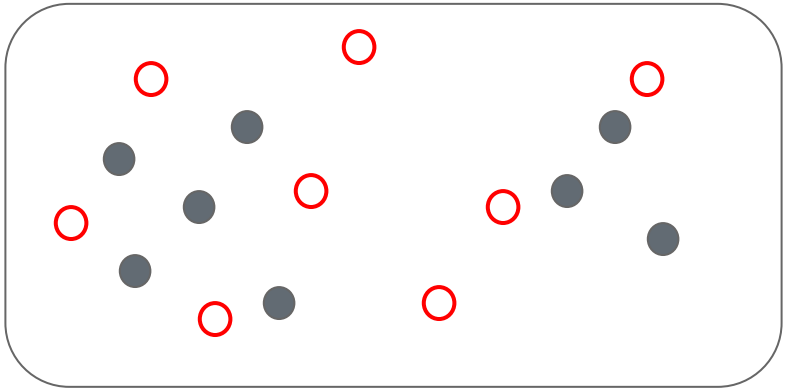
Tendencia al Clustering

Estadístico (coeficiente) de Hopkins

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

w_i = distancia de un elemento i al azar al vecino más cercano

u_i = distancia de un punto i agregado al azar al vecino más cercano



H ~ 0 : Clusters!
H ~ 0.5 : Distrib. homogenea

elementos del dataset

elementos agregados al azar

Validación externa

1. Matriz de confusión

Clusters	0	1	2	3	4
Labels					
a	1	25	1	0	13
b	7	7	2	0	24
c	34	0	2	0	4
d	0	0	39	0	1
e	0	0	0	40	0

Wu, J., Xiong, H., & Chen, J. (2009, June). Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 877-886).
https://maths-people.anu.edu.au/~johnm/courses/mathdm/2009/talks/justin-paper.pdf

Validación externa

***2. Medida Normalizada de van Dongen**: Medida mejorada de la pureza (que mide cuánto se aleja de tener sólo tengo un elemento por fila|columna)*

Clusters	0	1	2	3	4
Labels					
a	1	25	1	0	13
b	7	7	2	0	24
c	34	0	2	0	4
d	0	0	39	0	1
e	0	0	0	40	0

$$VD_n = \frac{(2n - \sum_i \max_j n_{ij} - \sum_j \max_i n_{ij})}{(2n - \max_i n_{i.} - \max_j n_{.j})}$$

Van Dongen, S. (2000). Performance criteria for graph clustering and Markov cluster experiments. In *National research institute for mathematics and computer science*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.9783&rep=rep1&type=pdf

Validación externa

***2. Medida Normalizada de van Dongen**: Medida mejorada de la pureza (que mide cuánto se aleja de tener sólo tengo un elemento por fila|columna)*

Clusters	0	1	2	3	4
Labels					
a	1	25	1	0	13
b	7	7	2	0	24
c	34	0	2	0	4
d	0	0	39	0	1
e	0	0	0	40	0

2*200 - (25 + 24 + 34 + 39 + 40) - (25 + 24 + 34 + 39 + 40)

$$VD_n = \frac{(2n - \sum_i \max_j n_{ij} - \sum_j \max_i n_{ij})}{(2n - \max_i n_{i.} - \max_j n_{.j})}$$

2*200 - 40 - 40

Van Dongen, S. (2000). Performance criteria for graph clustering and Markov cluster experiments. In *National research institute for mathematics and computer science*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.9783&rep=rep1&type=pdf

Validación externa

***2. Medida Normalizada de van Dongen**: Medida mejorada de la pureza (que mide cuánto se aleja de tener sólo tengo un elemento por fila|columna)*

Clusters	0	1	2	3	4
Labels					
a	1	25	1	0	13
b	7	7	2	0	24
c	34	0	2	0	4
d	0	0	39	0	1
e	0	0	0	40	0

$$VD_n = \frac{(2n - \sum_i \max_j n_{ij} - \sum_j \max_i n_{ij})}{(2n - \max_i n_{i.} - \max_j n_{.j})}$$

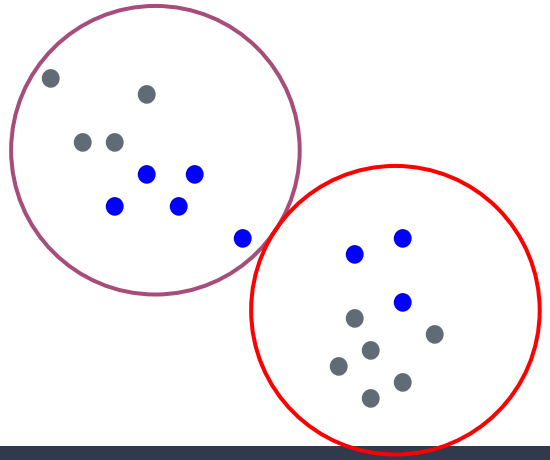
(mejor) 0 ≤ VDn ≤ 1 (peor)

Van Dongen, S. (2000). Performance criteria for graph clustering and Markov cluster experiments. In *National research institute for mathematics and computer science*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.9783&rep=rep1&type=pdf

Validación externa

***3. Índice Rand e Índice Rand Ajustado o Normalizado (Adjusted Rand Index, ARI)**:*

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$



a = número de pares de elementos que aparecen juntos en un clúster y además pertenecen a la misma clase.

b = número de pares de elementos que pertenecen a clases diferentes y además están en clústeres diferentes.

c = número de pares de elementos que comparten la clase, pero se ubican en diferentes clústeres.

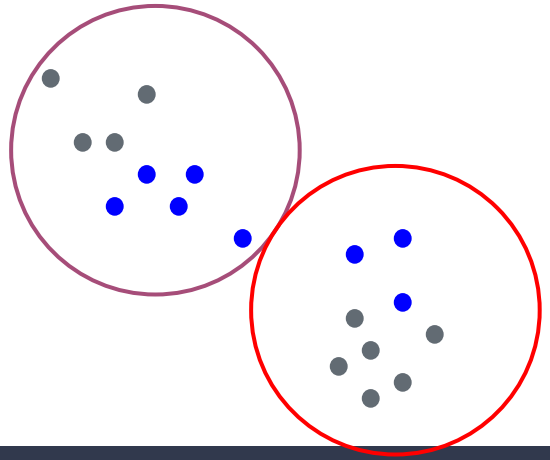
d = número de elementos que pertenecen a clases diferentes, sin embargo se agrupan en el mismo clúster.

n = número total de elementos.

Validación externa

***3. Índice Rand e Índice Rand Ajustado o Normalizado (Adjusted Rand Index, ARI)**:*

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$



a = 6 + 10 + 3 + 15 = 34

b = 30 + 12 = 42

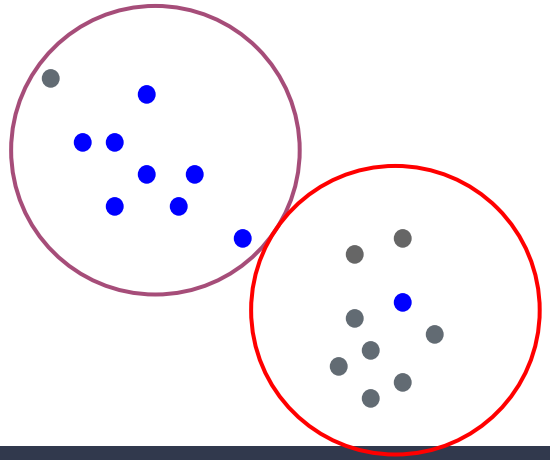
n = 153

R = 76 / 153 ~ 0.5

Validación externa

***3. Índice Rand e Índice Rand Ajustado o Normalizado (Adjusted Rand Index, ARI)**:*

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$



a = 28 + 28 = 56

b = 64 + 1 = 65

n = 153

R = 111 / 153 ~ 0.73

Validación externa

3. Índice Rand e Índice Rand Ajustado o Normalizado (Adjusted Rand Index, ARI):

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

a = número de pares de elementos que aparecen juntos en un clúster y además pertenecen a la misma clase.

b = número de pares de elementos que pertenecen a clases diferentes y además están en clústeres diferentes.

c = número de pares de elementos que comparten la clase, pero se ubican en diferentes clústeres.

d = número de elementos que pertenecen a clases diferentes, sin embargo se agrupan en el mismo clúster.

n = número total de elementos.

Validación externa

3. Índice Rand e Índice Rand Ajustado o Normalizado (Adjusted Rand Index, ARI):

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

ARI = (R - E(R)) / (max(R) - E(R))

E(R) = valor esperado de R si se distribuyen al azar.

max(R) = valor máximo posible de R para los datos.

Validación Interna: Criterios general: Cohesión y Separación

- ***Cohesión***: es una medida de las proximidad de los miembros de un clúster entre sí o con respecto al prototipo.
- ***Separación***: es la proximidad entre miembros de diferentes clústeres o entre prototipos de grupos y el prototipo general.

Suma de los Errores al Cuadrado (dentro de un cluster) (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

Suma de Cuadrados de Separación (SSB)

$$SSB = \frac{1}{2K} \sum_{i=1}^{K} \sum_{j=1}^{K} \frac{m}{K} \text{dist}(c_i, c_j)^2$$

c_i = centroide o medoide (prototipo)

Criterios general: Cohesión y Separación

- ***Cohesión***: es una medida de las proximidades de los miembros de un clúster con respecto al prototipo.

- ***Separación***: es la proximidad entre miembros de diferentes clústeres o entre prototipos de grupos y el prototipo general.

c_i = centroide o medoide (prototipo)

Suma Total de los Errores al Cuadrado (TSE)

$$TSE = \text{SSE} + \text{SSB}$$

Validación Interna

1. Coeficiente de Silhoutte:

$$s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

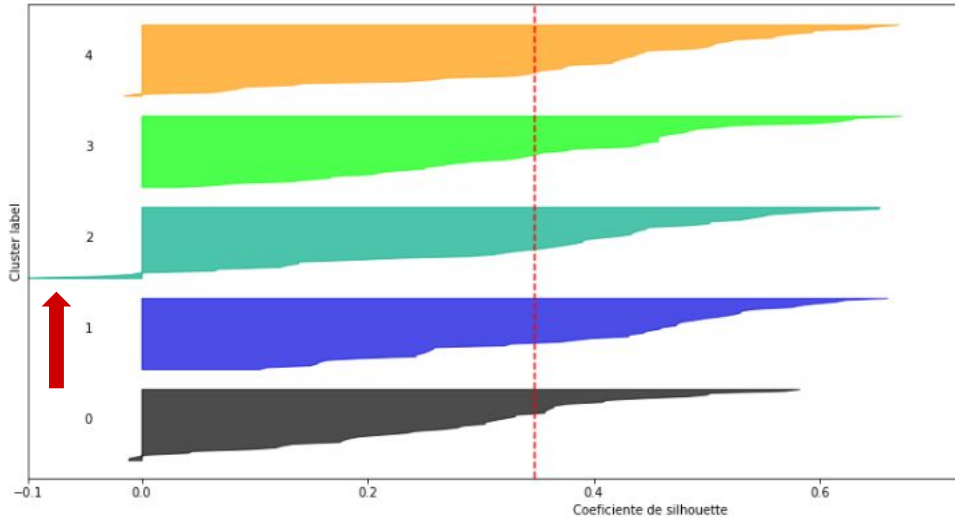
1. Para cada elemento ***i*** se calcula su distancia promedio a todos los otros elemento de su clúster (***a_i***).
2. Para el elemento ***i*** y todos los otros clústeres que no lo contienen, se calcula las distancias promedio a todos los elementos de cada clúster.

   Se buscar el mínimo de esas distancias promedio a cada clúster (***b_i***).
3. Se calcula el coeficiente Silhouette (***s_i***) del elemento i.
4. Luego se puede calcular el promedio para cada cluster o el promedio global.

Validación Interna

***1. Coeficiente de Silhoutte**:*

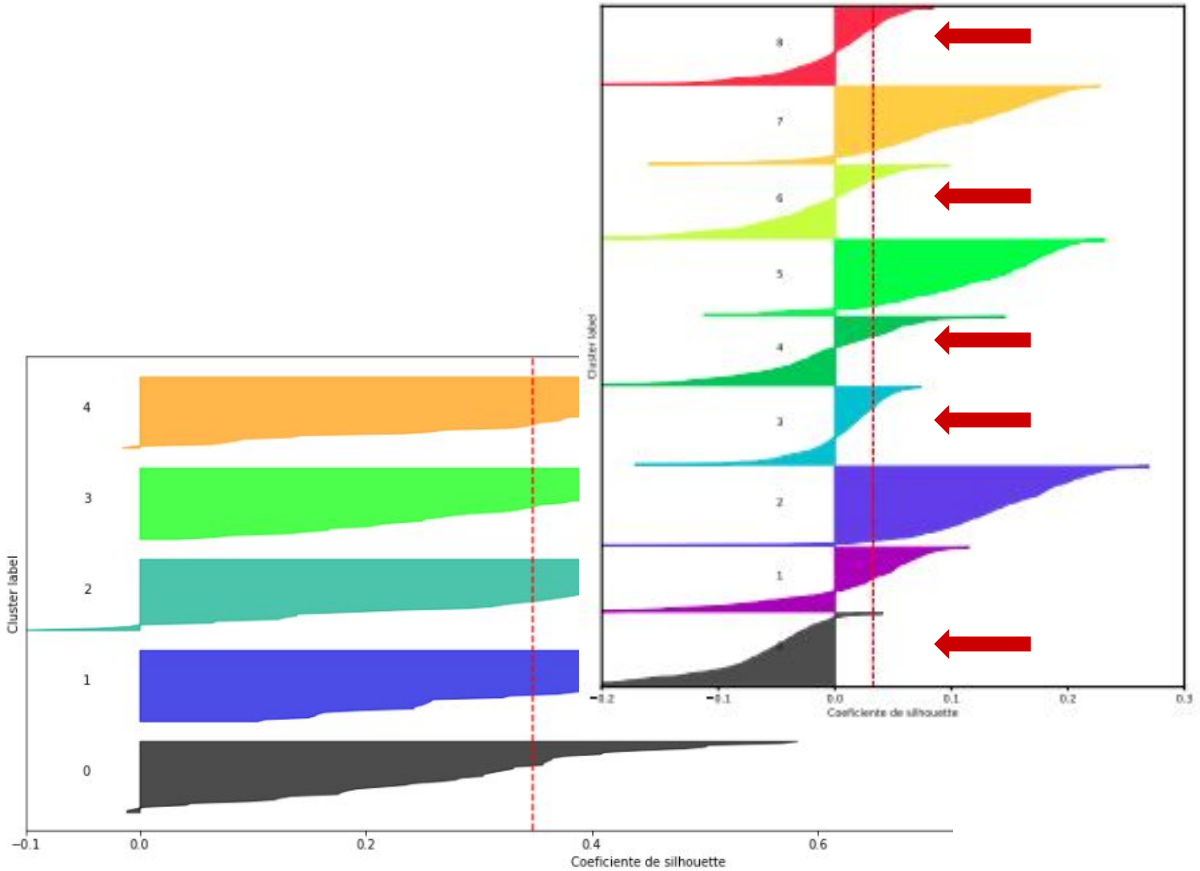
Puede ocurrir que algunos clusters tengan peor coeficiente o algunos elementos dentro del cluster. Esto puede ser indicativo de que quizás es mejor cambiar el valor de k.



Validación Interna

1. Coeficiente de Silhoutte:

Puede ocurrir que algunos clusters tengan peor coeficiente o algunos elementos dentro del cluster. Esto puede ser indicativo de que quizás es mejor cambiar el valor de k.

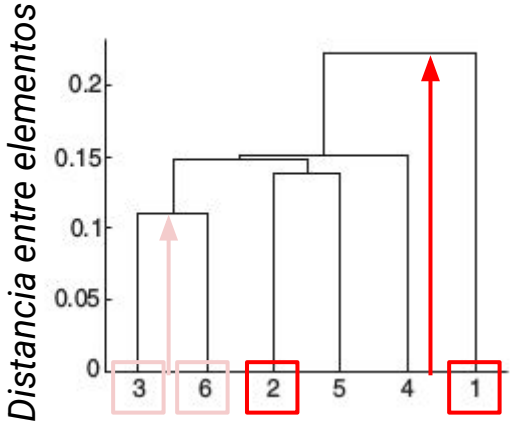


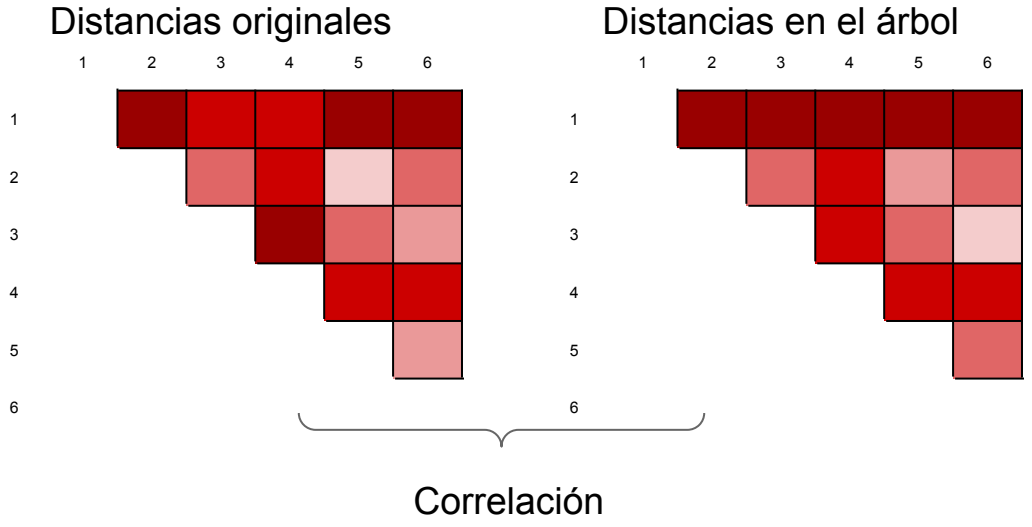
Validación Interna

2. (Jerárquico) Partición del árbol por distancia o por número de clusters: y luego se pueden aplicar las medidas de validación interna o externa igual que otros métodos.

Validación Interna

3. (Jerárquico) Coeficiente de Correlación Cofenético (CoPhenetic Correlation Coefficient, CPCC): Mide la correlación entre la matriz de distancia que dio origen al agrupamiento y los distancias extraídas del árbol (Altura del nodo que une por primera vez dos elementos).





Validación Interna

4. Bootstrapping: Sirve para evaluar la estabilidad de los clusters, y así determinar cuáles son “reales” y cuáles no. Vamos a volver a estos métodos más adelante.