Clustering I

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. Por ejemplo,

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

Para la próxima...

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
    - Entropia
    - 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)

  k-means Aglomerativo
  PAM

  DBSCAN

- 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)

  Clustering difuso

- Completo (todos los puntos están asignados a algún cluster) vs Parcial (hay puntos no asignados)

  DBSCAN

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

Clase métodos de validación

Inicialización

Clustering por prototipos (k-means) : Problemas

- Problemas de inicialización.
  - Hacer muchas corridas con inicializaciones al azar.
  - Elegir puntos alejados.
  - *Bisecting k-means*.
- Problemas de outliers.
  - Preprocesamiento.
  - Postprocesamiento.

Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de disimilitud.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las disimilitudes 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. Si disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)

Los medoides son elementos (o ítems) del conjunto.

Se puede partir de una matriz de disimilaridad arbitraria.

Clustering por prototipos (PAM, Partition around Medoids)

0. Seleccionar K.
1. Seleccionar K ítems como medoides iniciales.
2. Calcular la matriz de disimilitud.
3. Asignar cada ítem a uno de los K clusters.
4. Computar la sumatoria de las disimilitudes 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. Si disminuye, reemplazar el medoide.

Hasta que: los medoides no cambien

Etapa de intercambio (Swap phase)

k-medias -> minimizar la suma del cuadrado de las distancias euclídeas entre los miembros de un cluster y su respectiva media.
PAM -> minimizar la suma de disimilitudes entre los miembros del cluster y su medoide.
Esta diferencia en cuanto a la función a optimizar hace que PAM tienda a ser más robusto que k-medias en conjuntos de datos con outliers.

Más flexible en cuanto al tipo de datos y las medidas de distancia que se pueden aplicar (**PAM** parte de la matriz de disimilaridad).

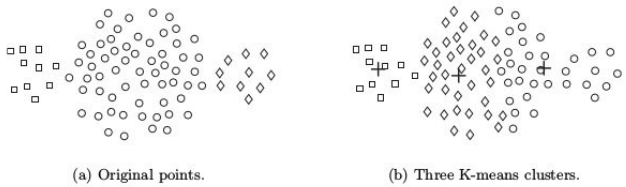
Clustering por prototipos (PAM, Partition around Medoids)

- 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)

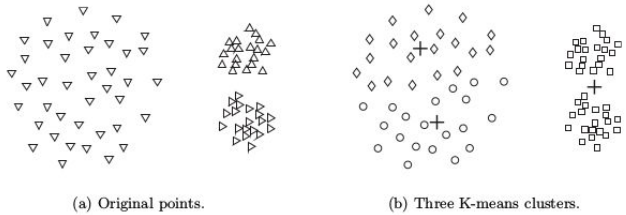
Clustering por prototipos (k-means) : Problemas

- Problemas de inicialización.
- Problemas de outliers.
- Problemas intrínsecos al enfoque
  - Clusters de diferentes tamaños.
  - Clusters no esféricos.
  - Clusters con diferentes densidades.



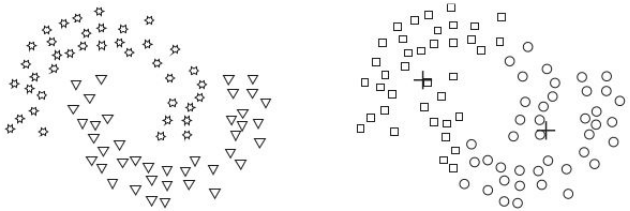
(a) Original points. (b) Three K-means clusters.

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.

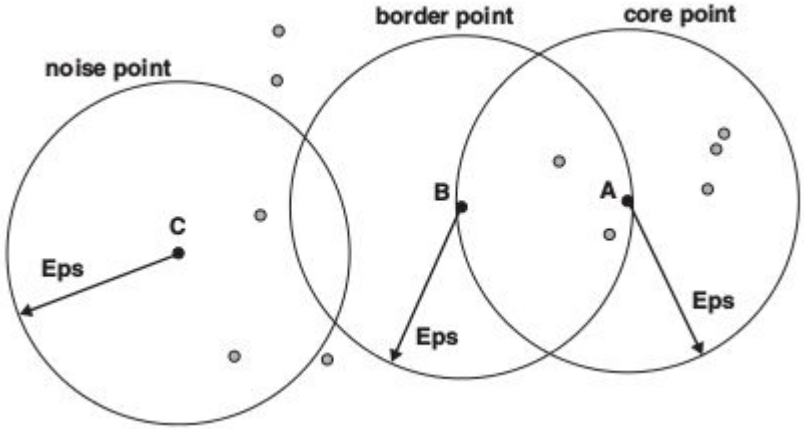


(a) Original points. (b) Two K-means clusters.

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.



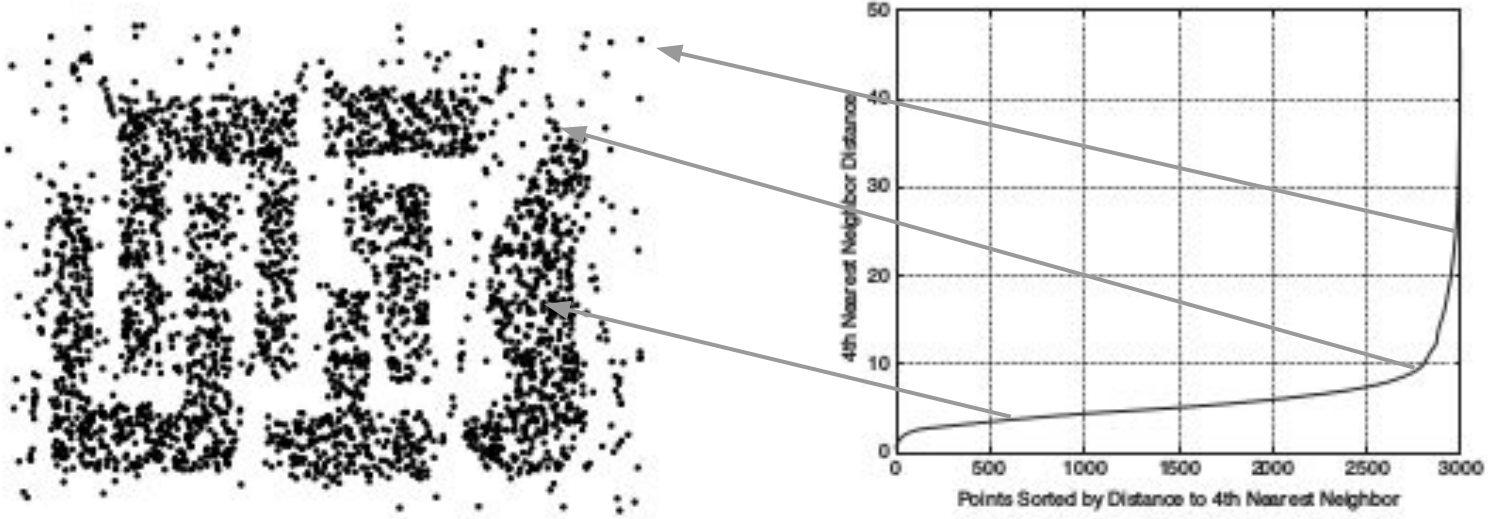
Semilla = Core

Borde = Border

Ruido = Noise

Clustering por densidad (DBSCAN)

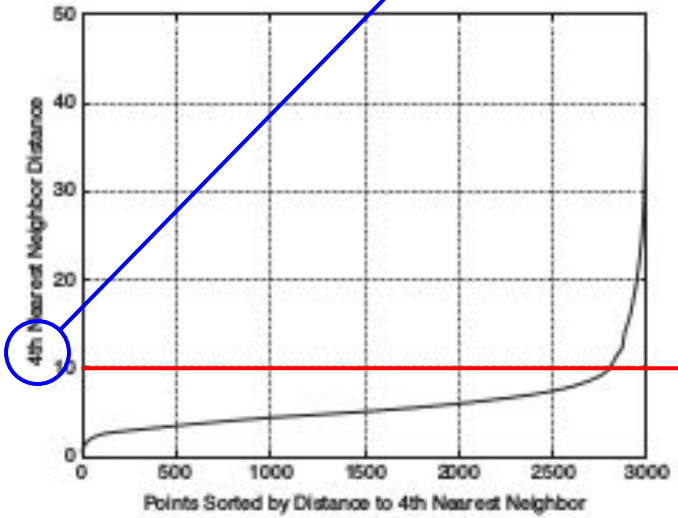
0. Elegir valores para los parámetros *Eps* y *MinPts*.



Clustering por densidad (DBSCAN)

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

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

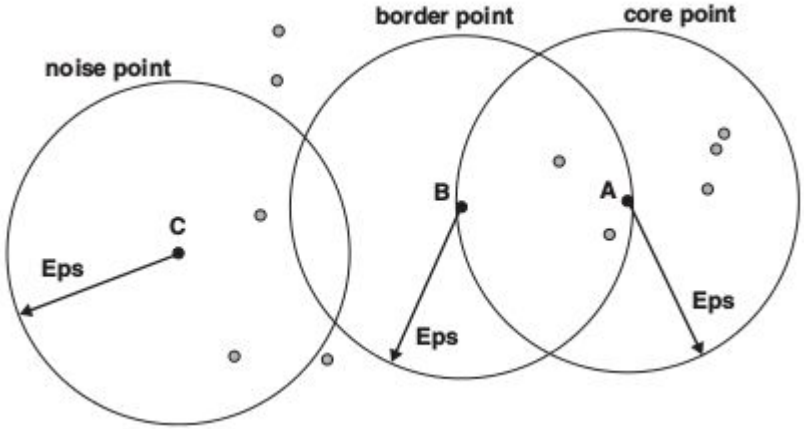


Eps = 10

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 (raíz) y se van separando hasta quedarse sólo con los clusters individuales.

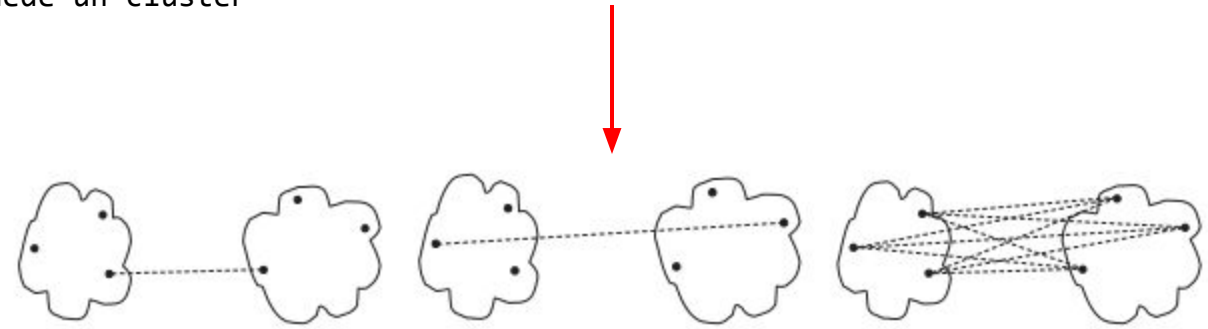
Clustering jerárquico aglomerativo

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.

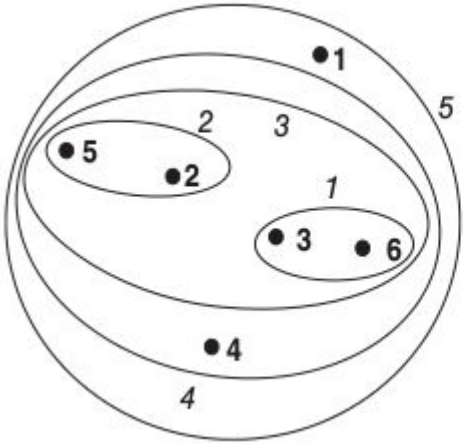
Clustering jerárquico aglomerativo

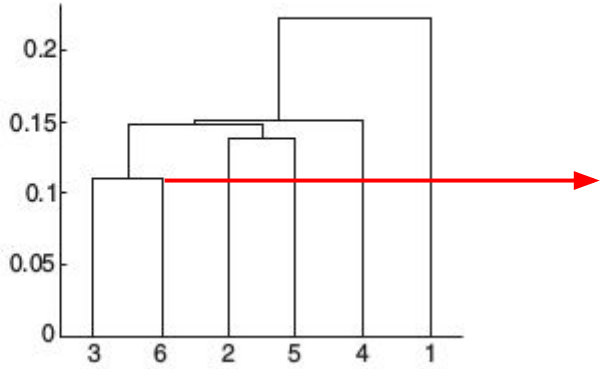
0. Computar la matriz de similaridad.

Repetir:

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

Hasta que: Sólo quede un cluster





Distancia entre elementos

Clustering jerárquico aglomerativo

- 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.

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.