

# Clustering 1

A dark blue diagonal gradient bar that starts from the bottom-left corner and extends towards the top-right corner, covering the lower half of the slide.

**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](#). 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


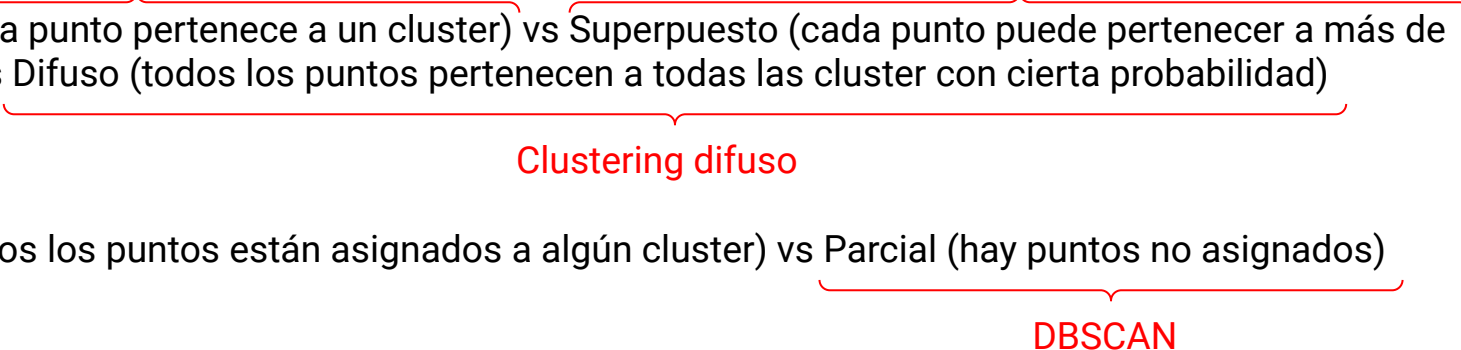

([http://www.iiisci.org/journal/CV\\$/sci/pdfs/GS315JG.pdf](http://www.iiisci.org/journal/CV$/sci/pdfs/GS315JG.pdf))

- 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
  - Silhouette
  - SSE / SSB
  - Coeficiente de correlación cofenético (Jerárquico)
  - Bootstrapping (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)  
  
k-means  
PAM  
Aglomerativo  
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  
DBSCAN
- Completo (todos los puntos están asignados a algún cluster) vs Parcial (hay puntos no asignados)  
  
DBSCAN

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

► Seleccionar K.

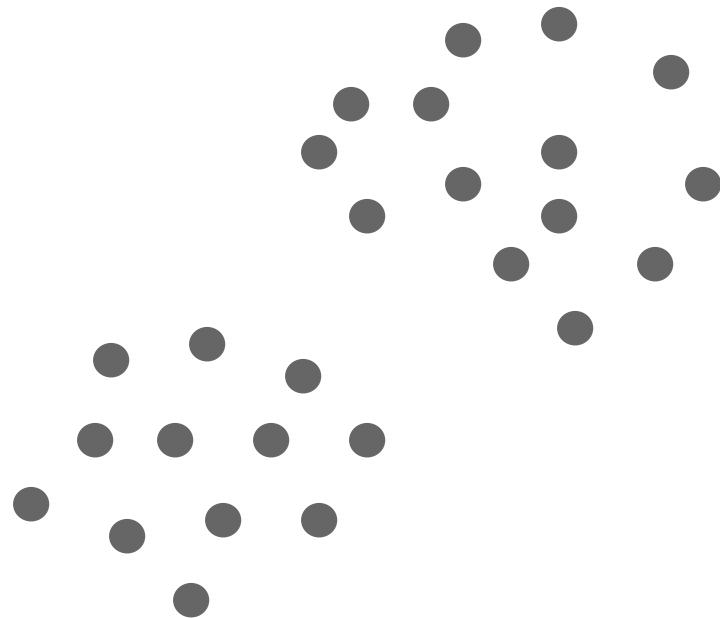
Seleccionar K puntos como centroides iniciales.

**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  $K = 2$ .

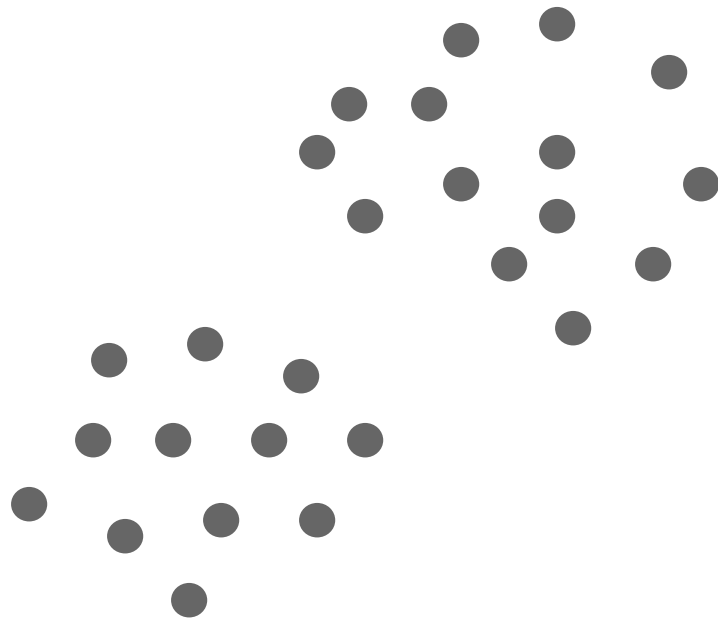
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

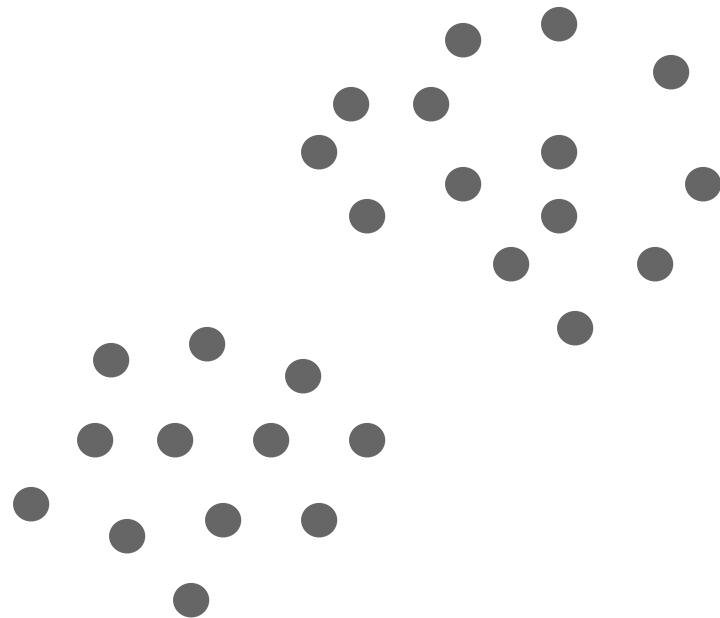
► Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

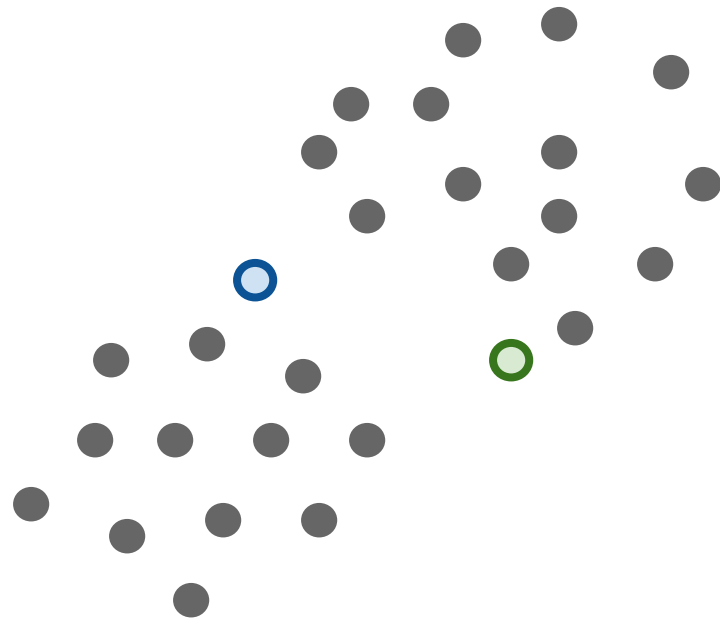
► Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

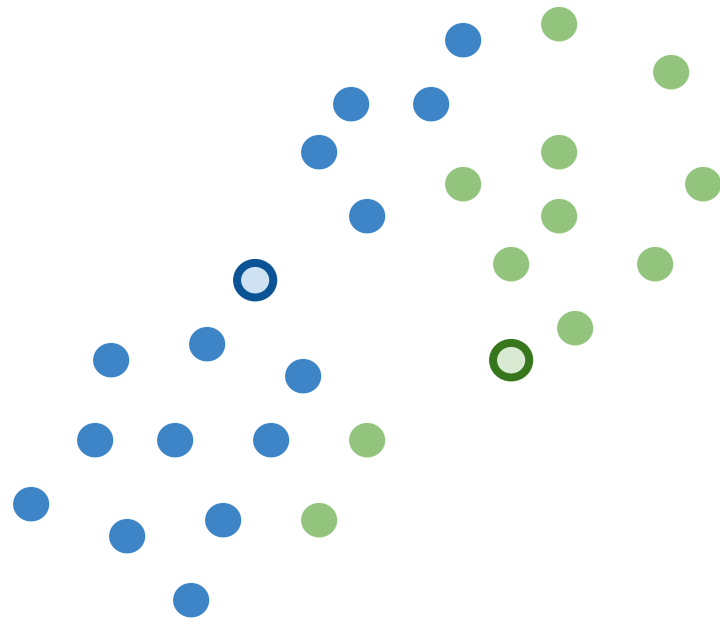
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

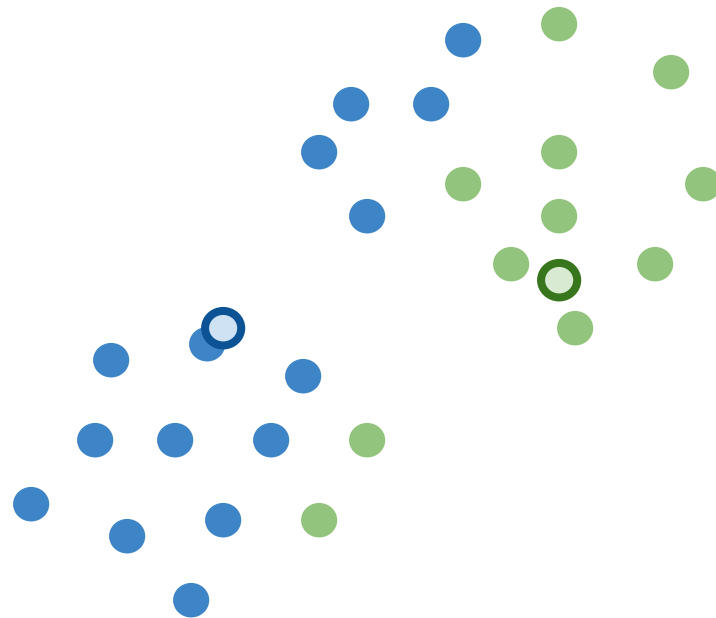
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

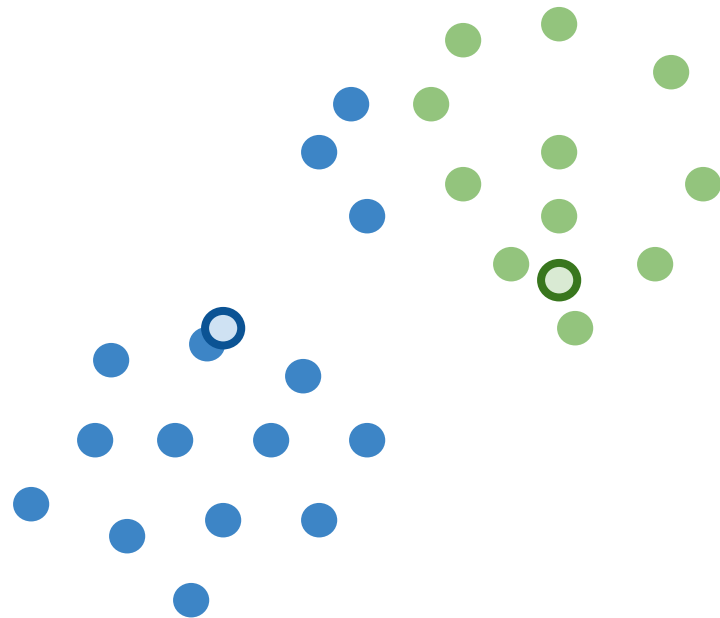
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

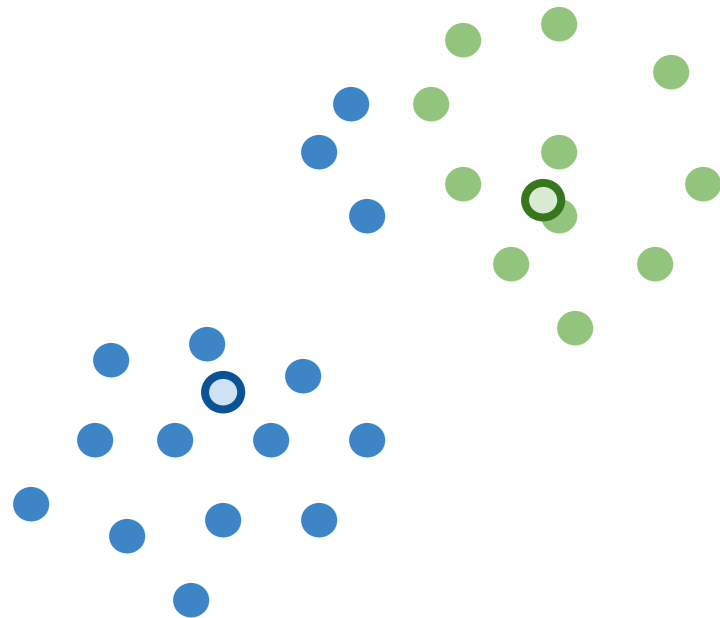
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

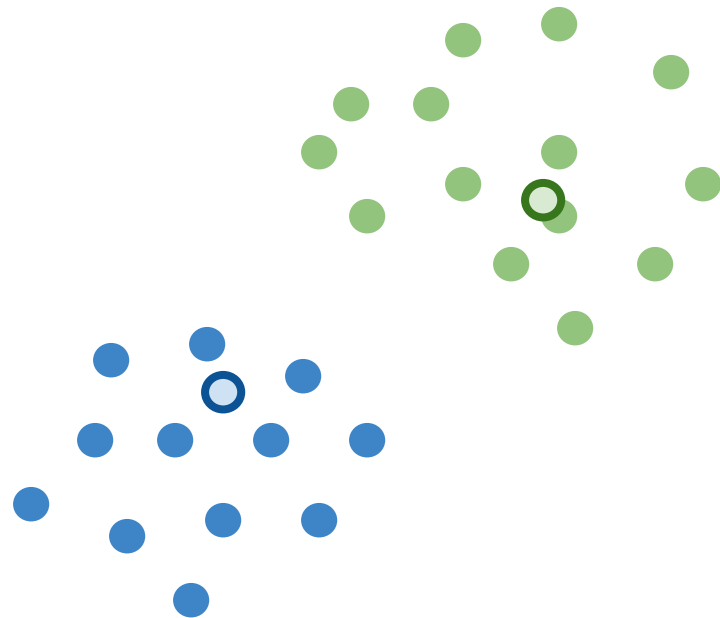
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

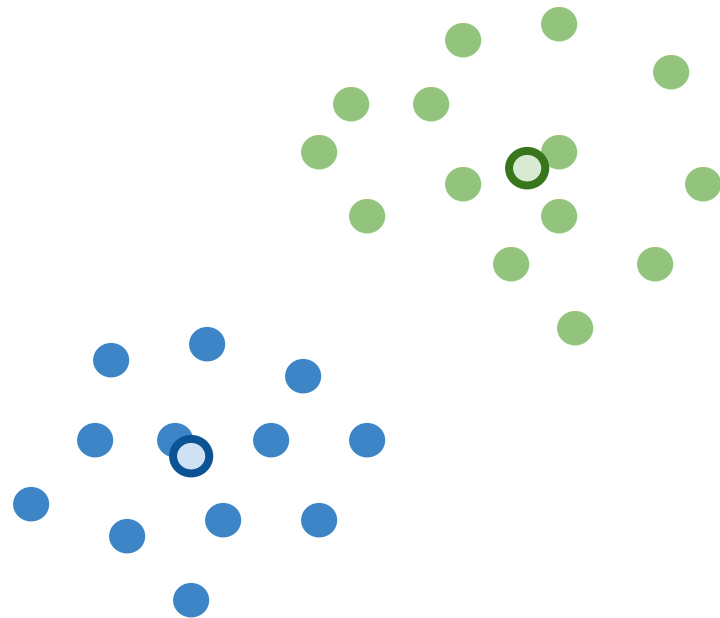
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

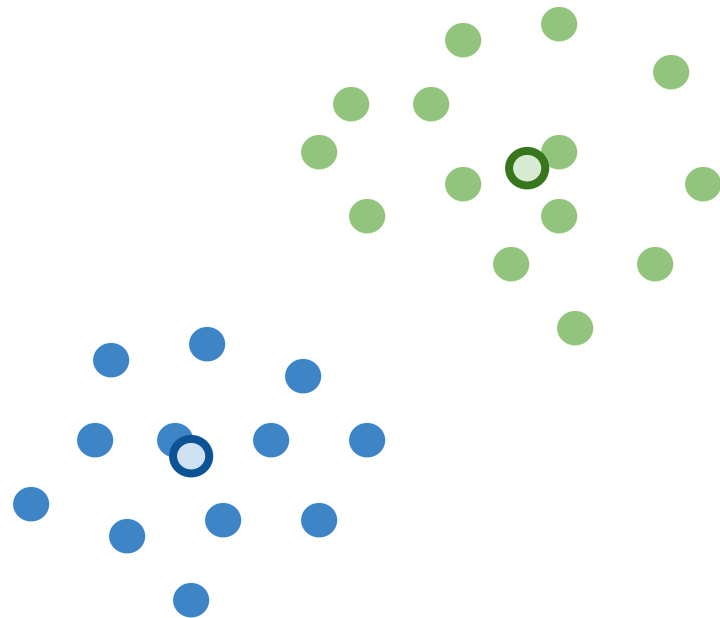
Seleccionar  $K$  puntos como centroides iniciales.

**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  $K$ .

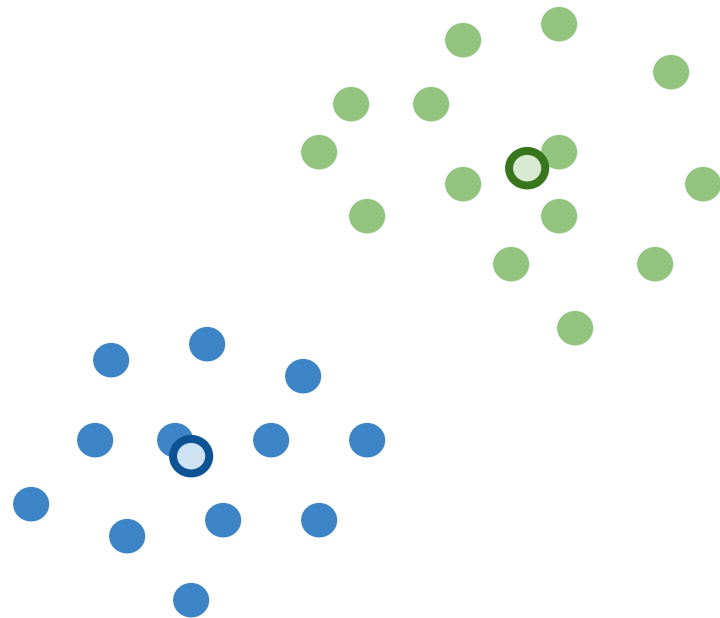
Seleccionar  $K$  puntos como centroides iniciales.

**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.

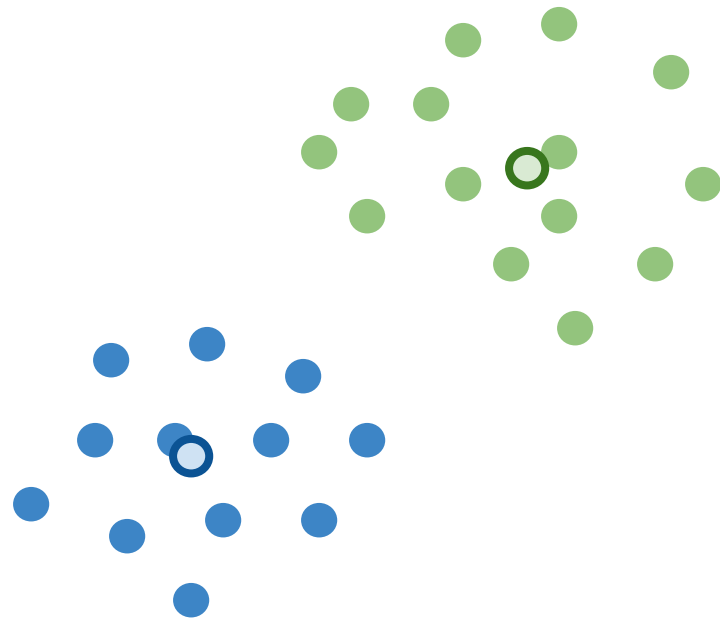
**Repetir:**

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.

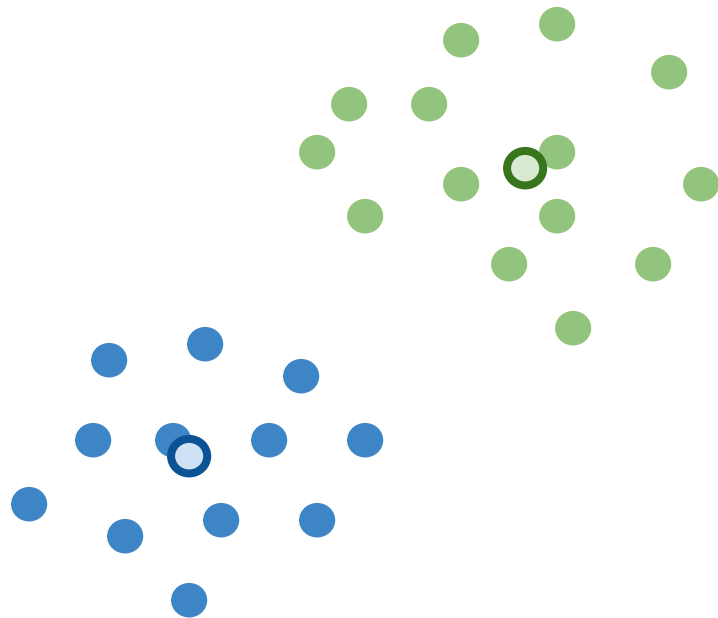
► **Hasta que:** los centroides no cambien

### Complejidad

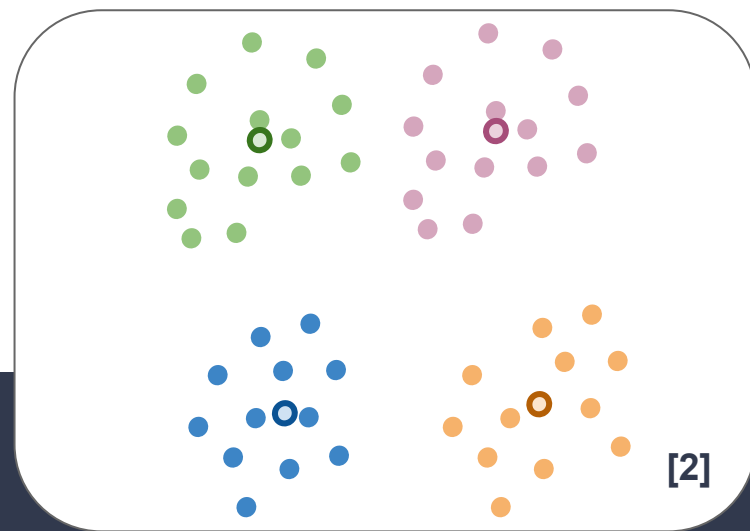
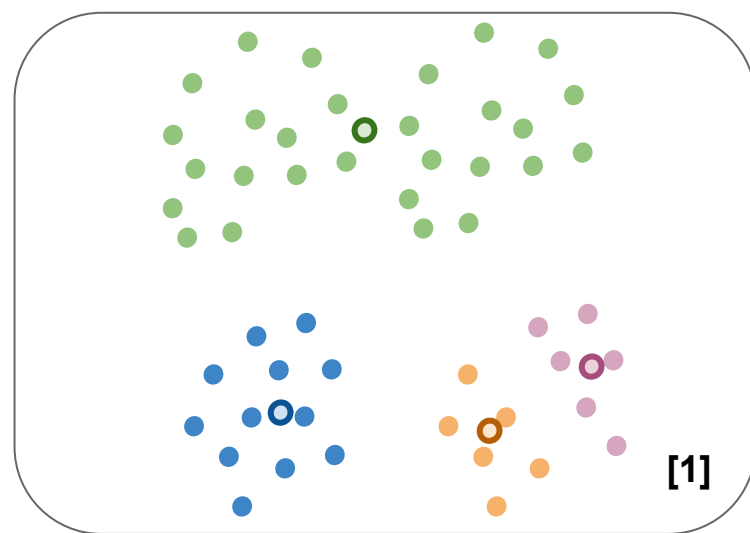
$n$  = número de puntos,  $d$  = número de atributos,  
 $k$  = número de clusters,  $m$  = número iteraciones

Si se fijan los parámetros  $n$ ,  $k$ ,  $d$ ,  $m \Rightarrow$

$$O(n * k * d * m) \sim O(n * k * \dots)$$

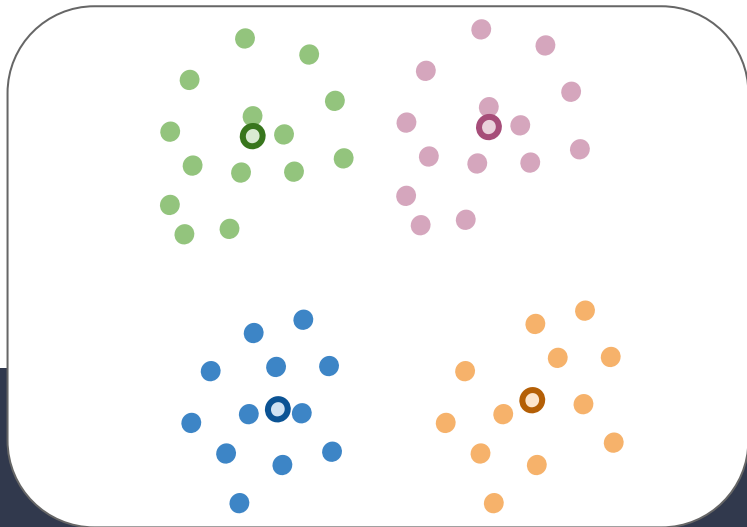
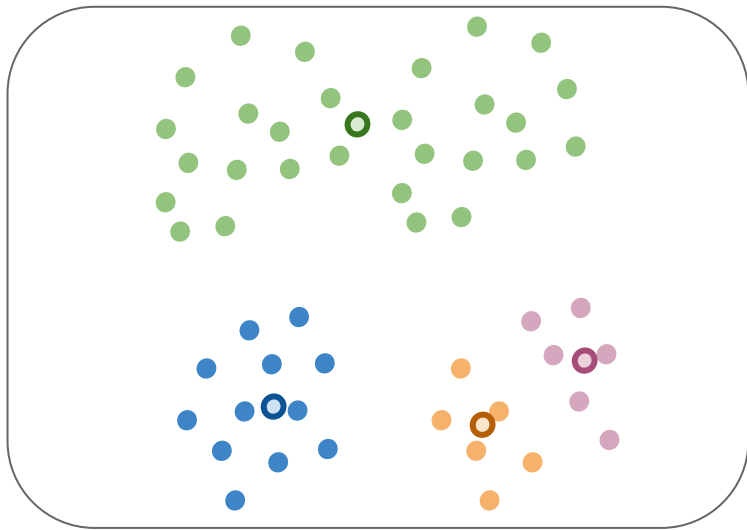


## Clustering por prototipos (k-means) > Inicialización



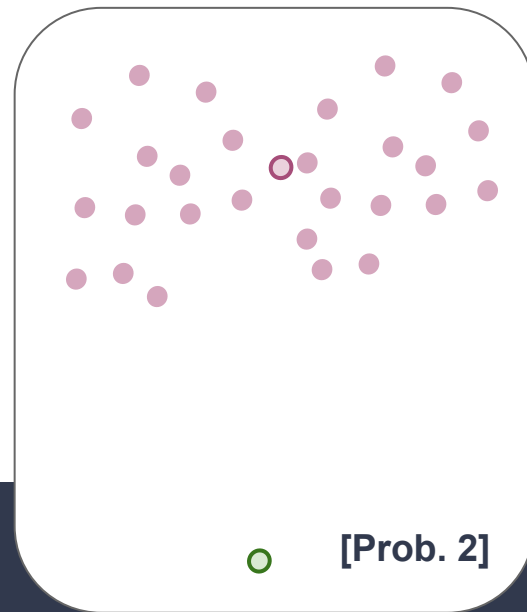
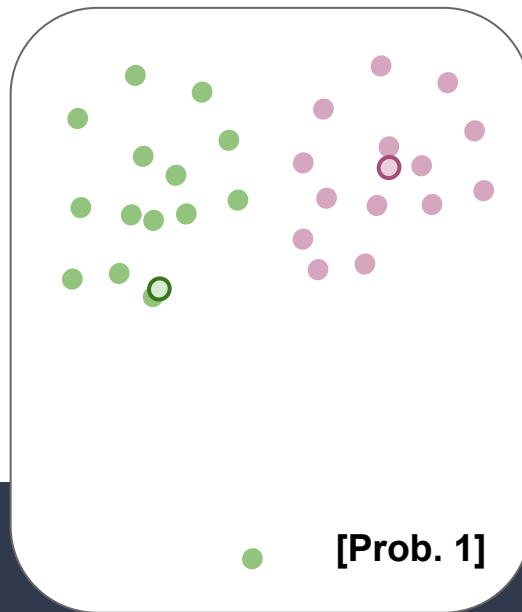
## Clustering por prototipos (k-means) > Inicialización

- 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  $\sim$  'distancia a los centros anteriores'  $^2$



## Clustering por prototipos (k-means) > Inicialización

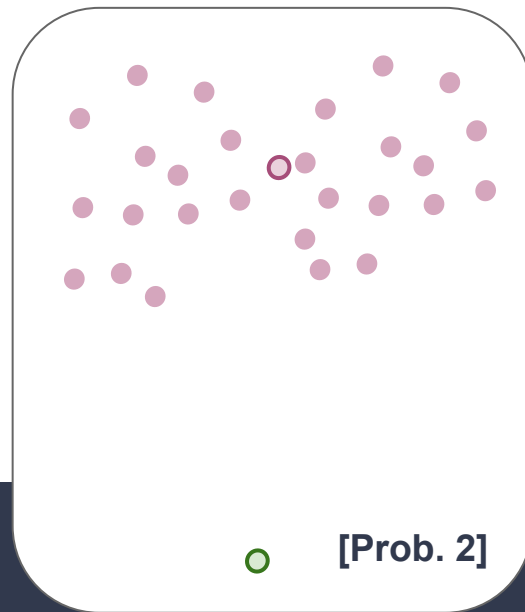
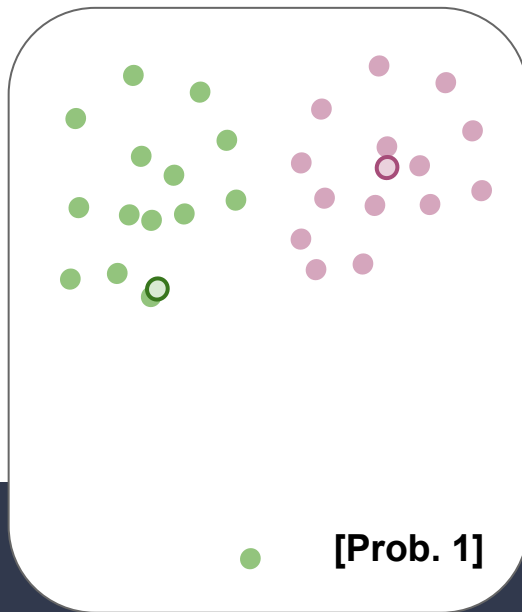
## Clustering por prototipos (k-means) > Outliers



## Clustering por prototipos (k-means) > Outliers

- 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]

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



## 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.

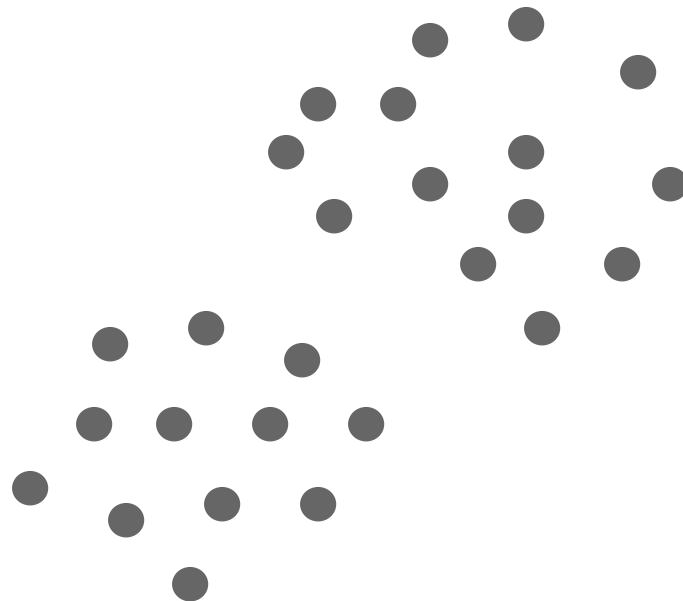
**Repetir:**

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.

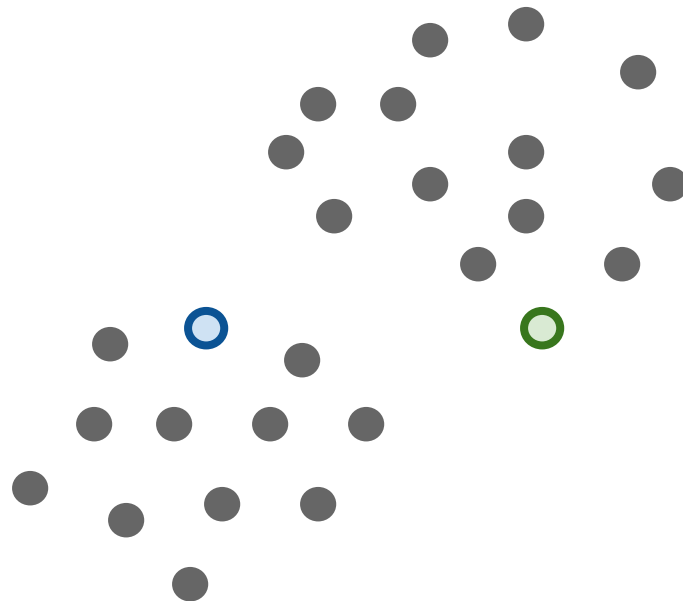
**Repetir:**

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.

**Repetir:**

5. Intercambiar el medoide dentro del cluster.

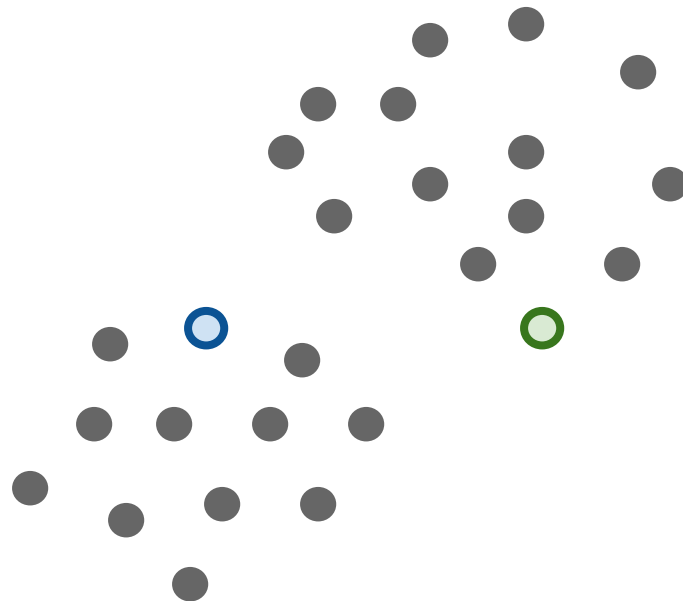
6. Recomputar (3) de cada cluster.

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.

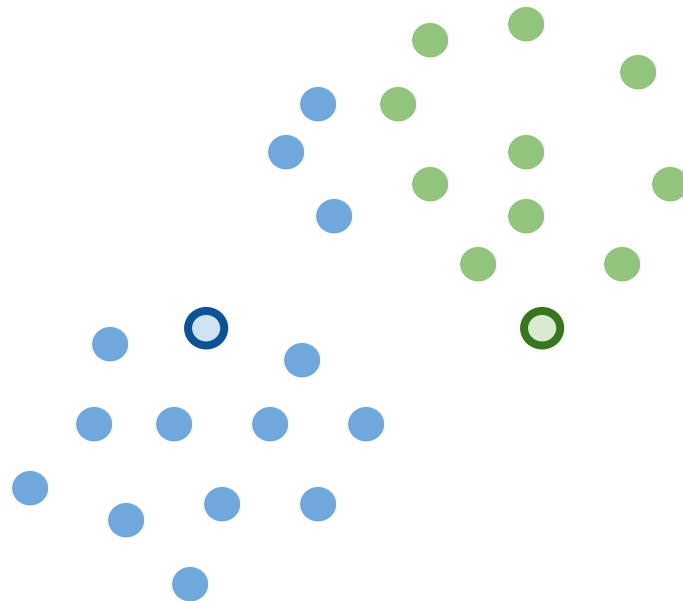
6. Recomputar (3) de cada cluster.

7. Si (4) disminuye, reemplazar el medoide.

**Hasta que:** los medoides no cambien

Etapas de construcción  
(Build phase)

Etapas 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.

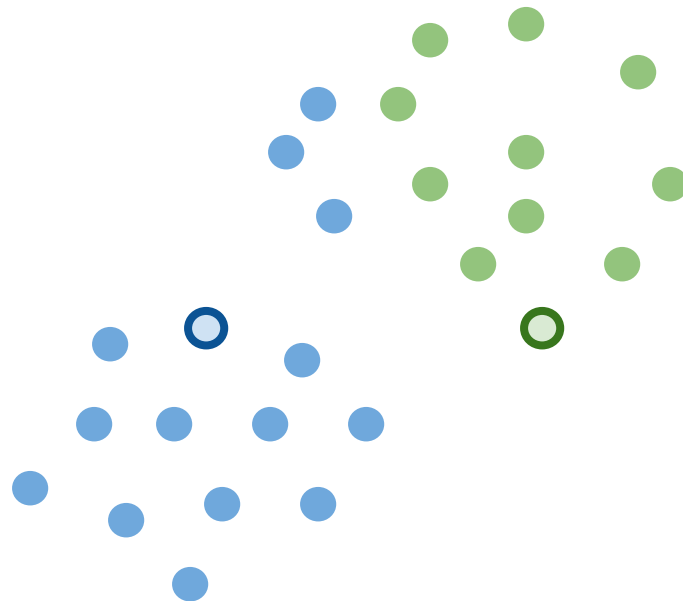
6. Recomputar (3) de cada cluster.

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.

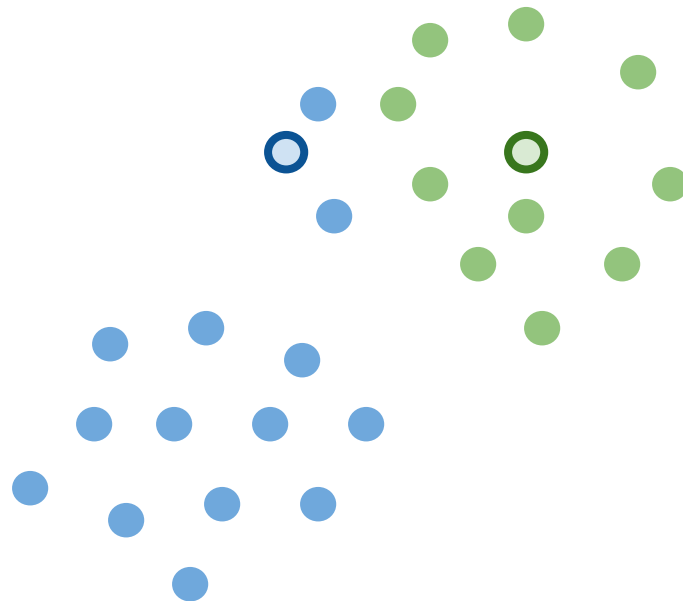
**Repetir:**

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

Etapas de construcción  
(Build phase)

Etapas 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.

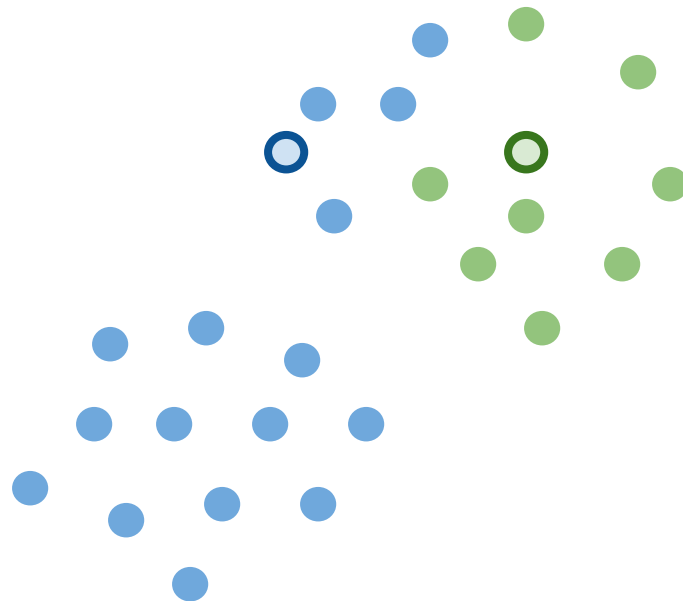
### Repetir:

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

Etapas de construcción  
(Build phase)

Etapas 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.

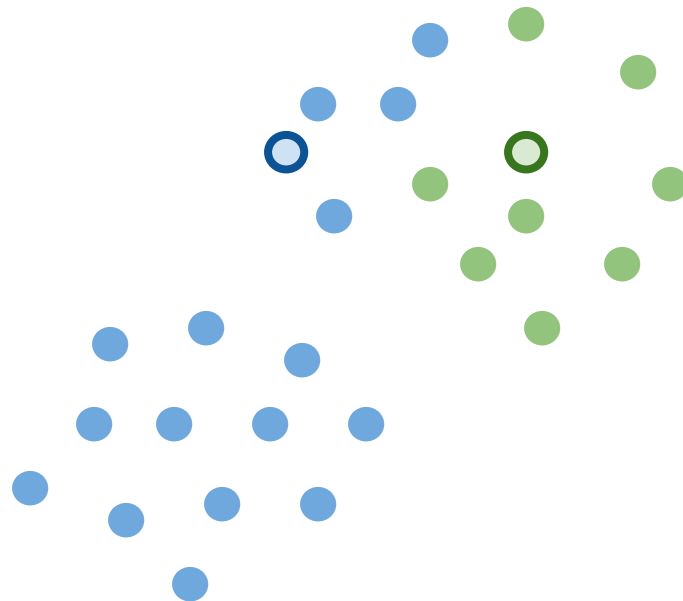
**Repetir:**

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

Etapas de construcción  
(Build phase)

Etapas 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.

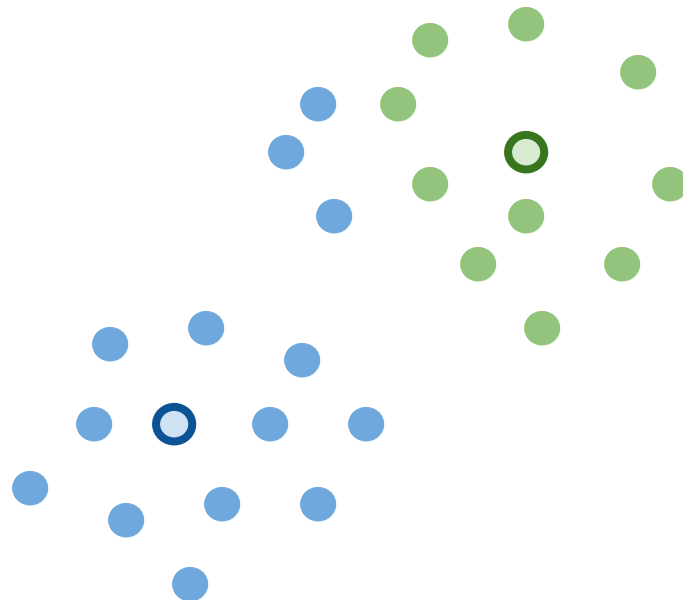
**Repetir:**

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.

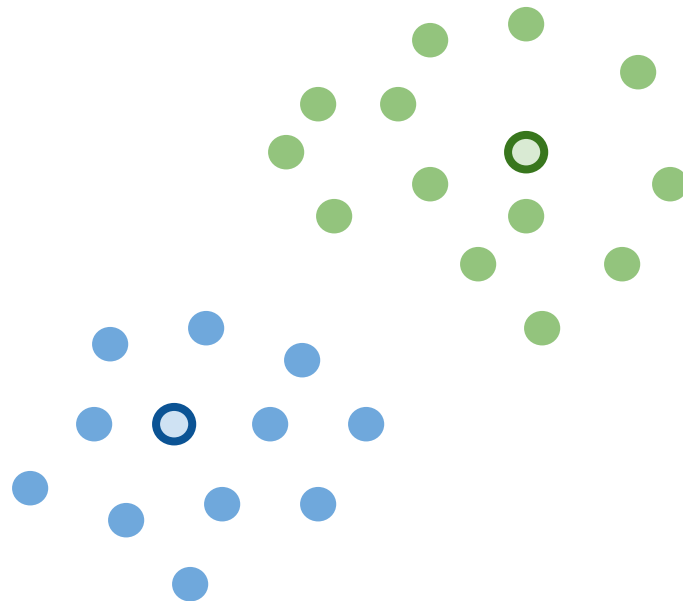
**Repetir:**

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.

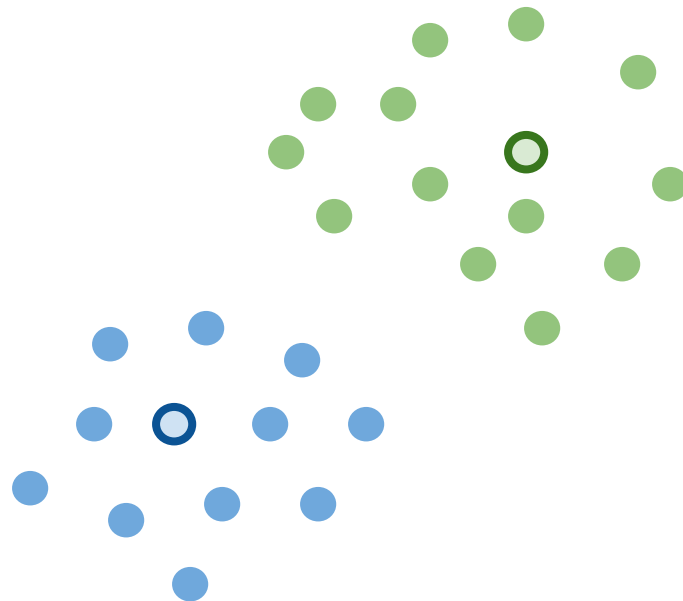
**Repetir:**

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.

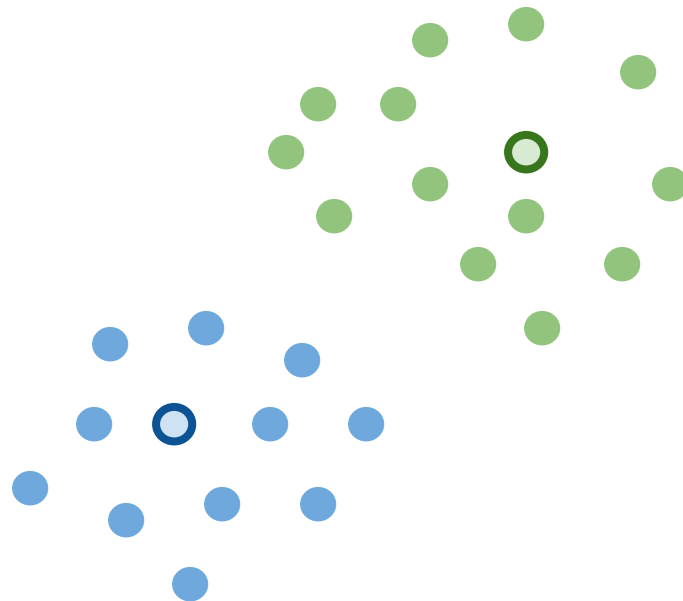
**Repetir:**

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.

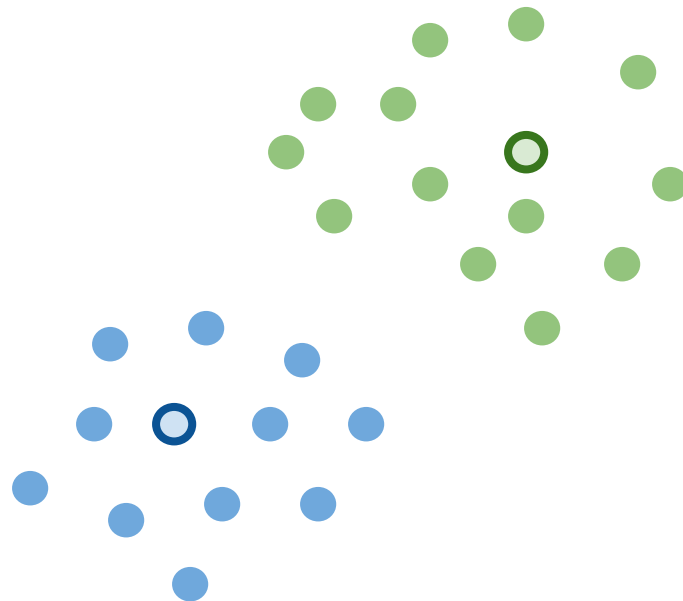
**Repetir:**

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.

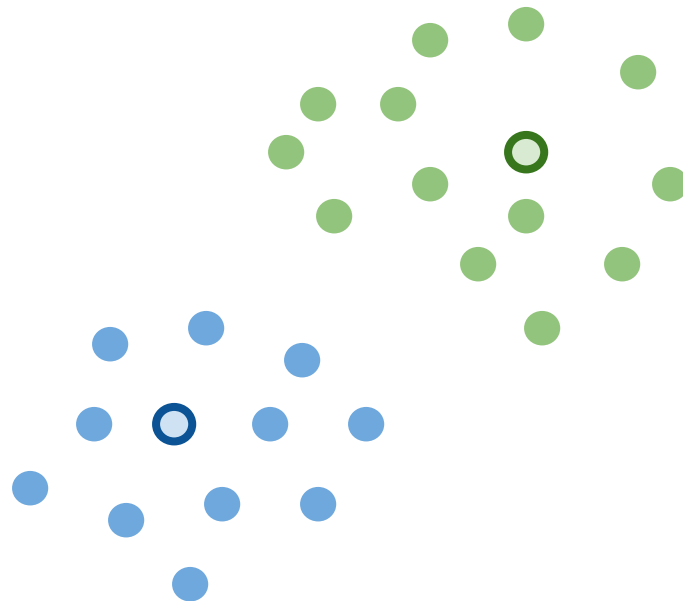
**Repetir:**

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.

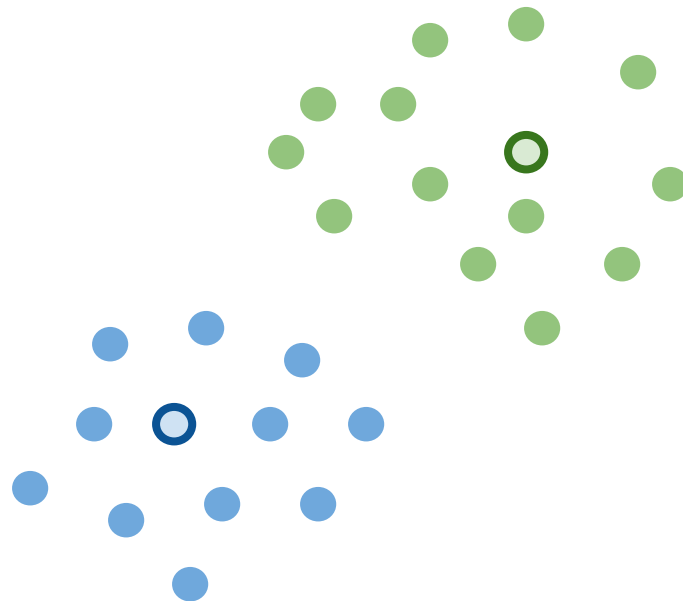
**Repetir:**

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.

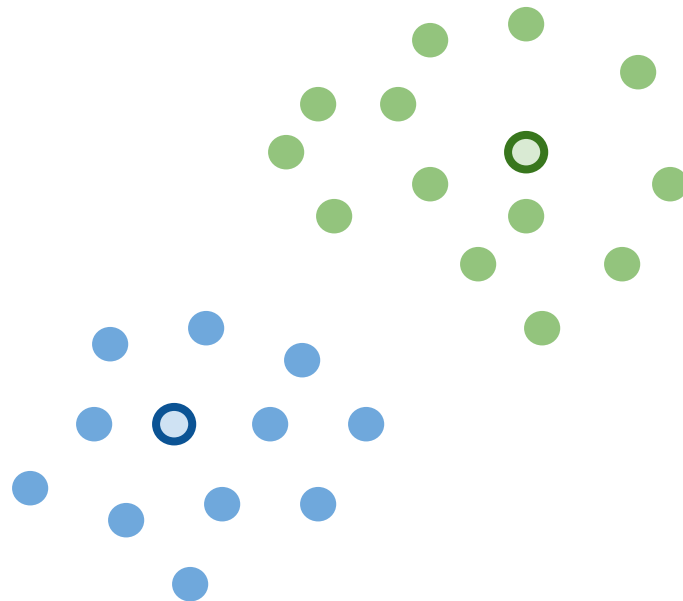
**Repetir:**

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

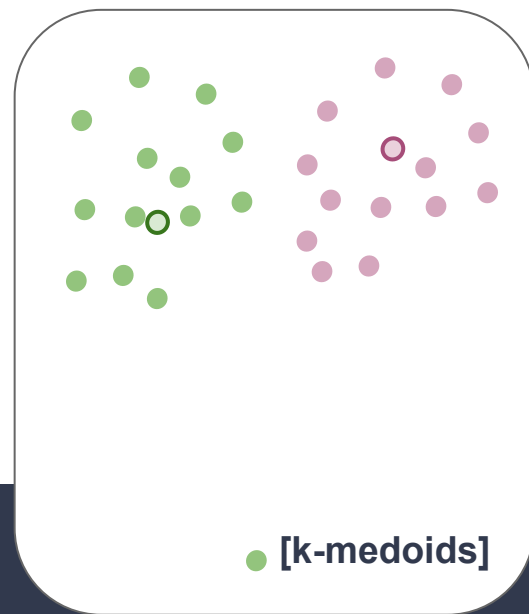
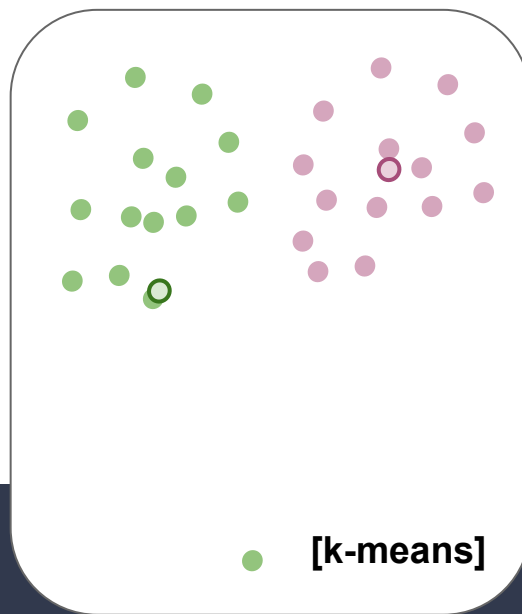
**Hasta que:** los medoides no cambien

Etapa de construcción  
(Build phase)

Etapa de intercambio  
(Swap phase)



## Clustering por prototipos (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  $n$ ,  $k$ ,  $d$ ,  $m \Rightarrow$

$$O(n * k * d * m) \sim O(n * k * \dots)$$

## *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  $n$ ,  $k$ ,  $d$ ,  $m \Rightarrow$

$$O(k * (n - k)^2 * \dots)$$

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:

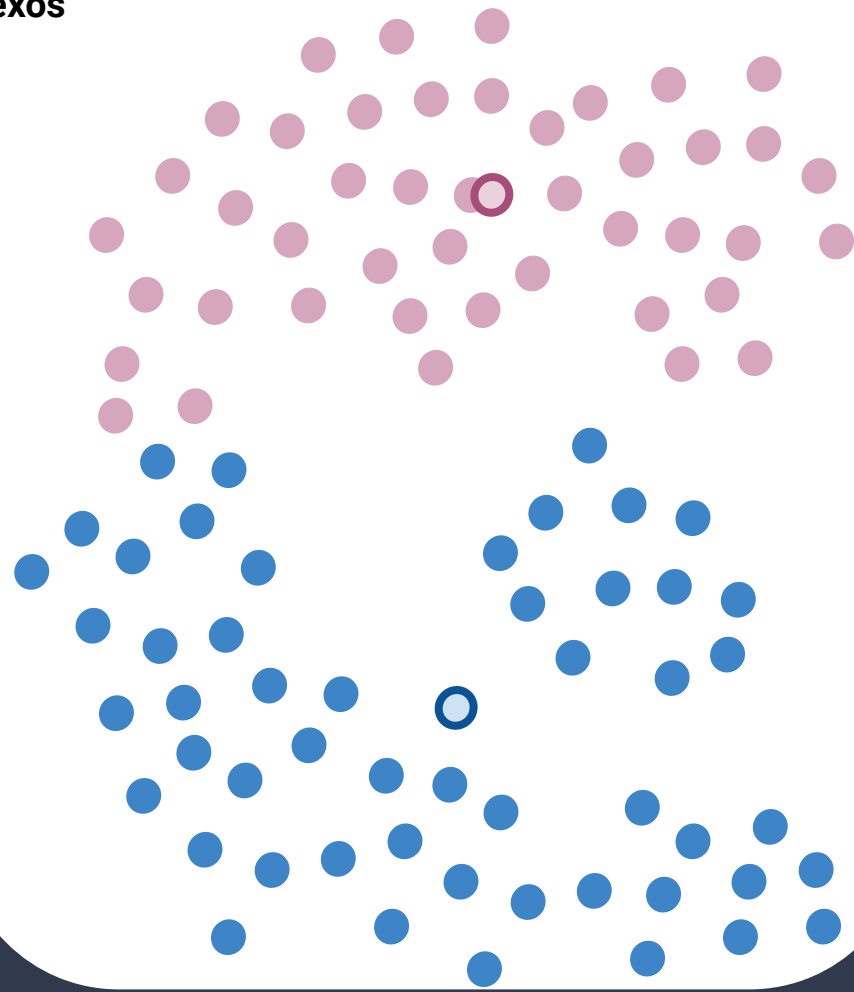
[https://scikit-learn-extra.readthedocs.io/en/latest/generated/sklearn\\_extra.cluster.KMedoids.html#sklearn\\_extra.cluster.KMedoids](https://scikit-learn-extra.readthedocs.io/en/latest/generated/sklearn_extra.cluster.KMedoids.html#sklearn_extra.cluster.KMedoids)

- Implementación a mano:

[https://github.com/salspaugh/machine\\_learning/blob/master/clustering/kmedoids.py](https://github.com/salspaugh/machine_learning/blob/master/clustering/kmedoids.py)

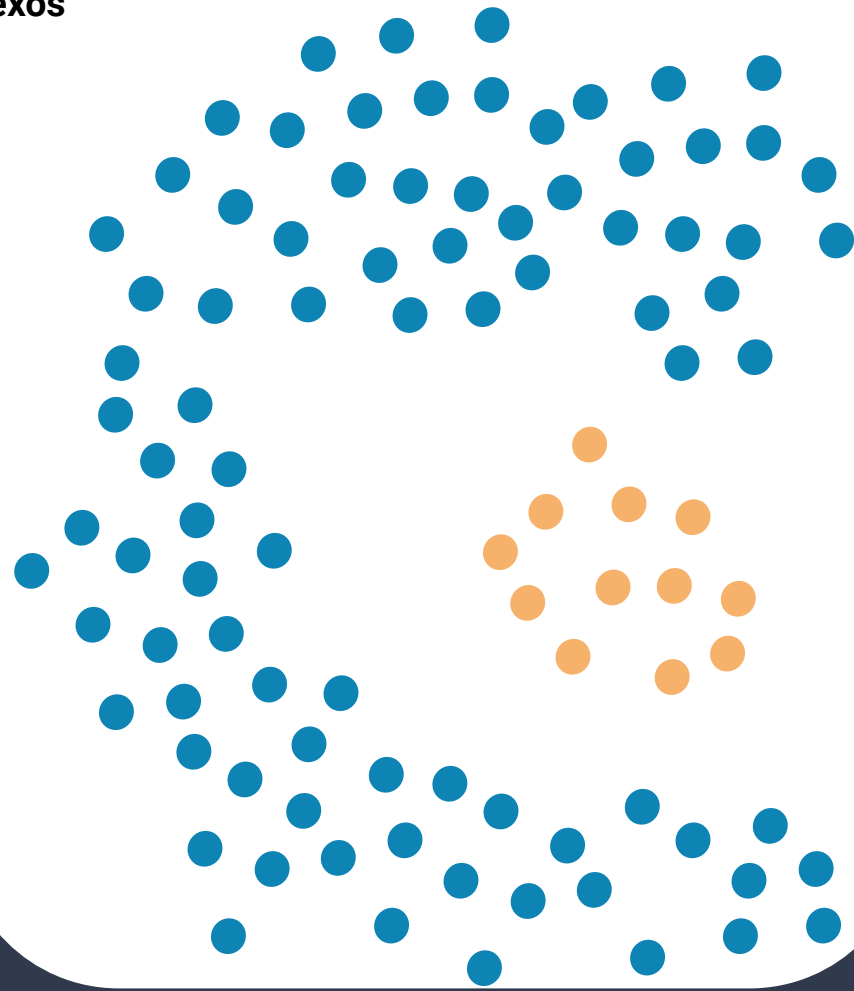
(Seguro se le puede pasar la matriz de disimilaridad que uno quiera)

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



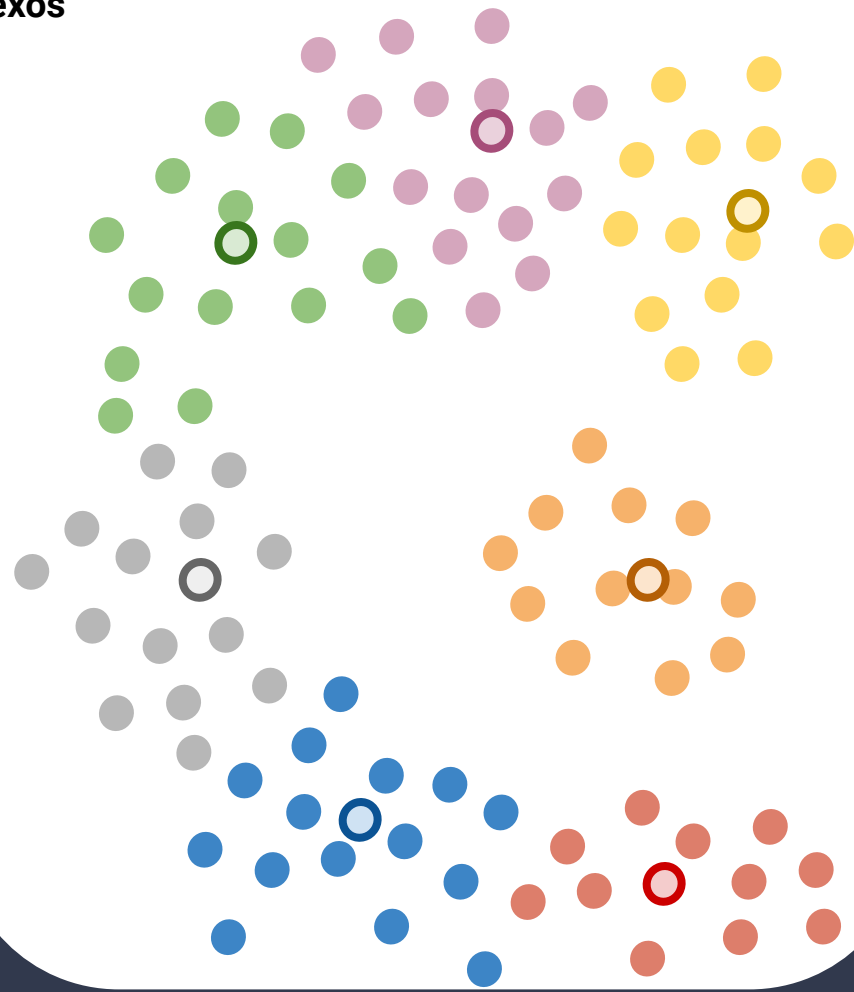
## Clustering por prototipos (k-means) > Clusters no convexos

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



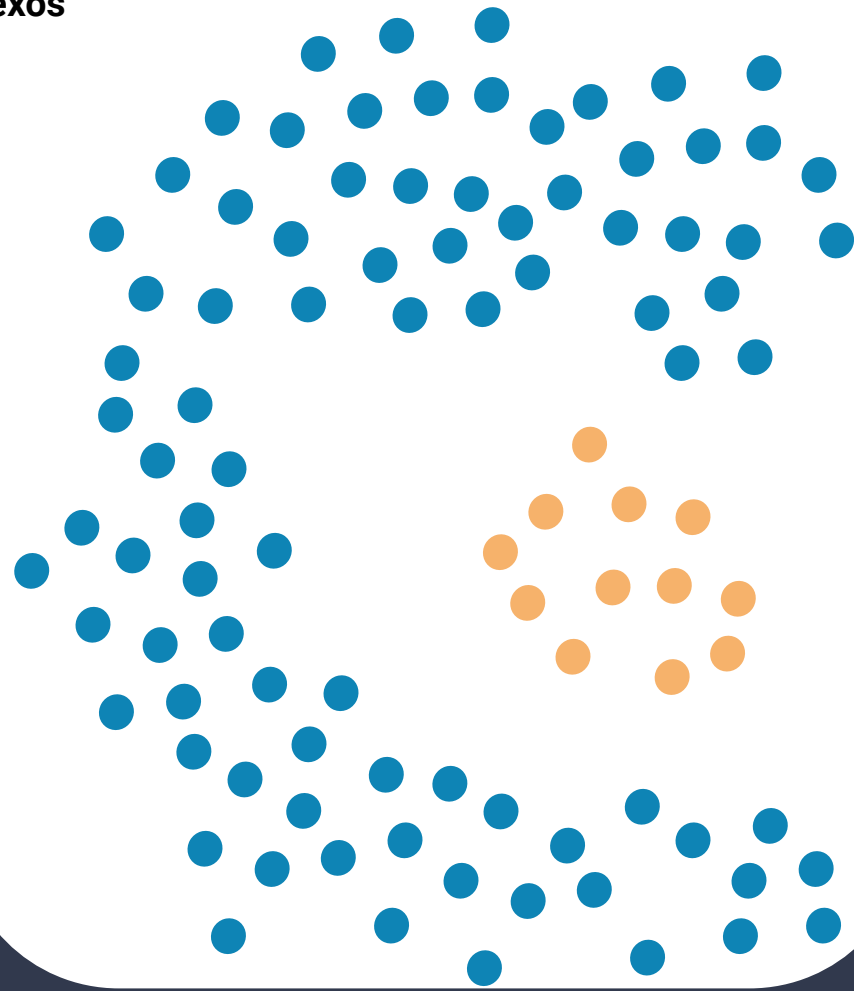
## Clustering por prototipos (k-means) > Clusters no convexos

- 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



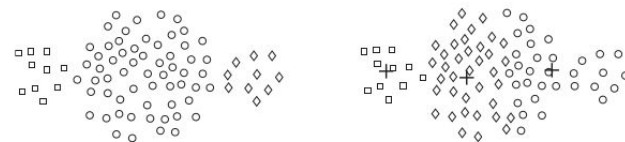
Clustering por prototipos (k-means) > Inicialización

Clustering por prototipos (k-means) > Outliers

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

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

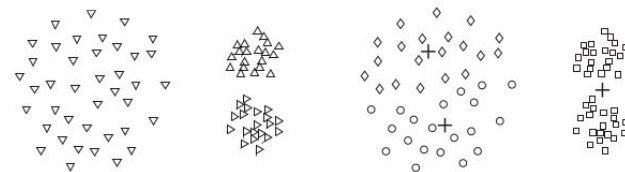
Clustering por prototipos (k-means) > Clusters de distinto tamaño



(a) Original points.

(b) Three K-means clusters.

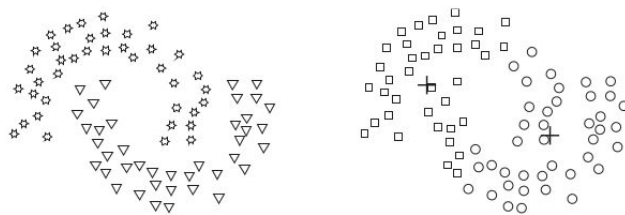
Figure 8.9. K-means with clusters of different size.



(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.





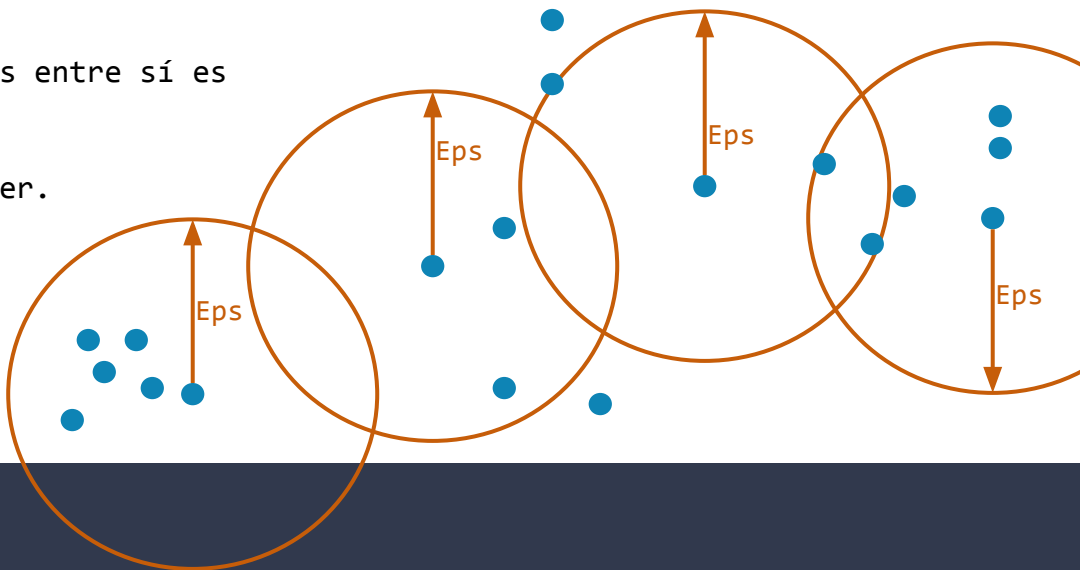
## Clustering por densidad (DBSCAN)

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



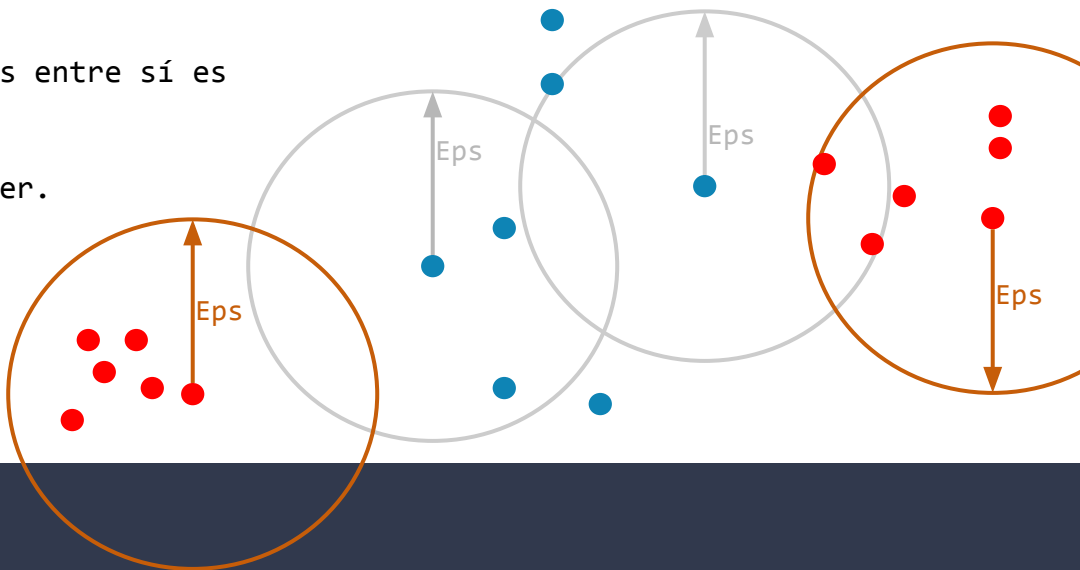


## Clustering por densidad (DBSCAN)

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



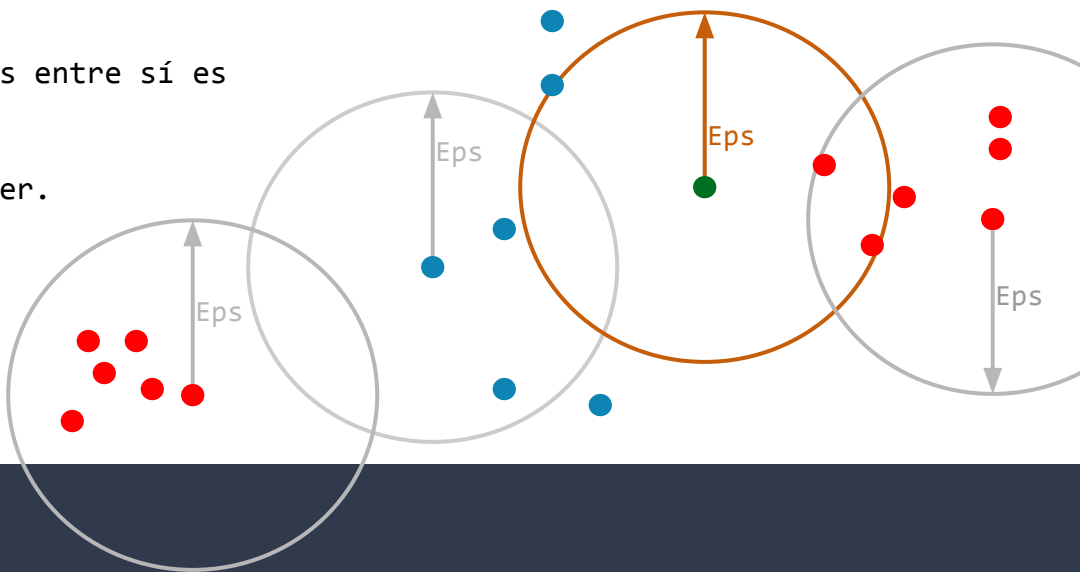
## Clustering por densidad (DBSCAN)

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)

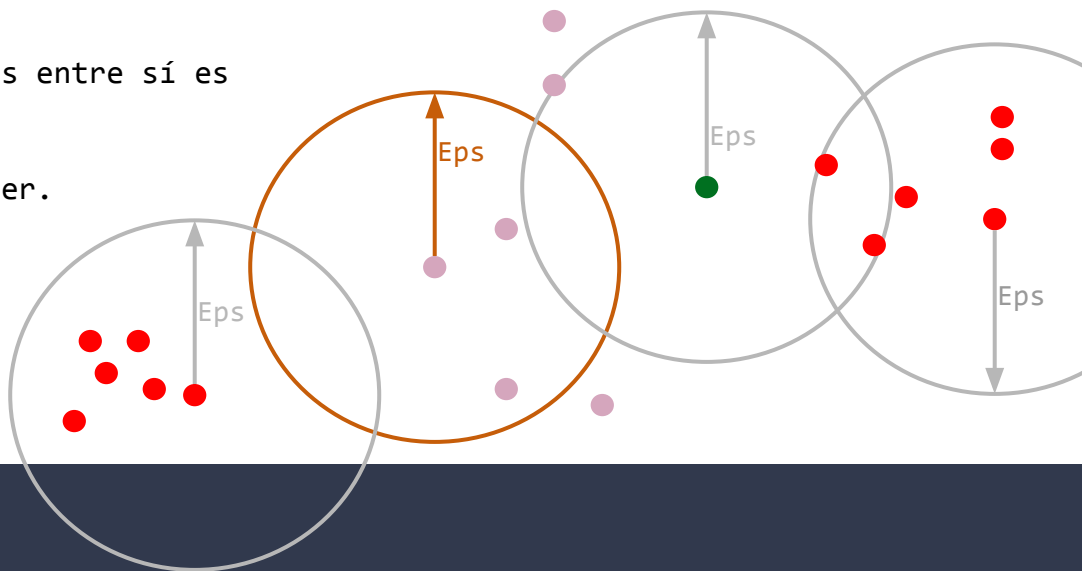
0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar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

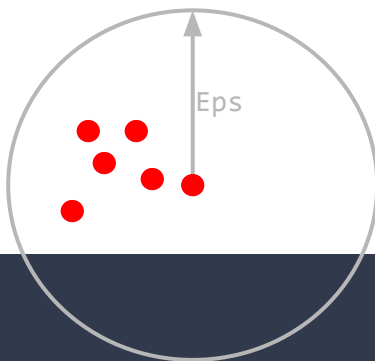
Es **ruido** si tiene menos de **MinPts** dentro de un radio **Eps**, y no está conectado con puntos semilla.

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.

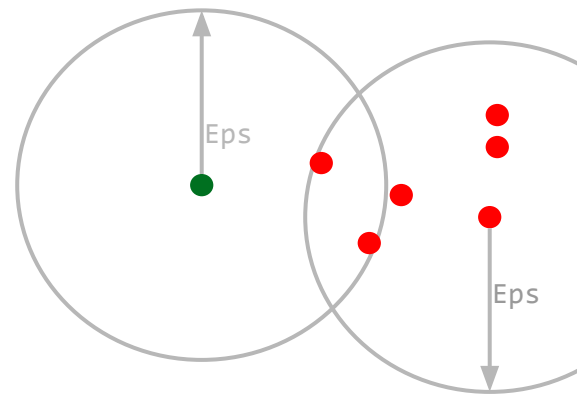


Es **semilla** si tiene al menos **MinPts** dentro de un radio **Eps**.

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)

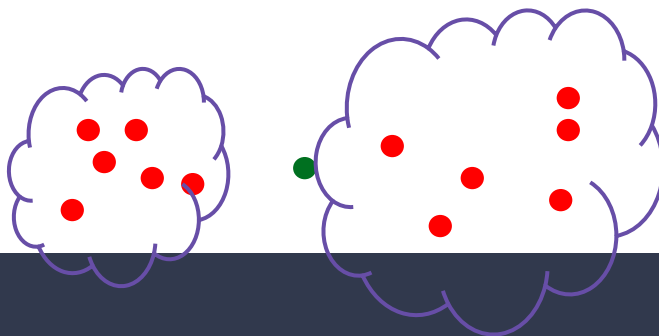
0. Elegir valores para los parámetros *Eps* y *MinPts*.
1. Identificar todos los elementos como Semilla, Borde o Ruido.
2. Eliminar Ruido.
3. Unir todos los elementos Semilla que se encuentren a menos de distancia *Eps*.
4. Cada grupo de elementos Semilla conectados entre sí es un cluster.
5. Asignar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

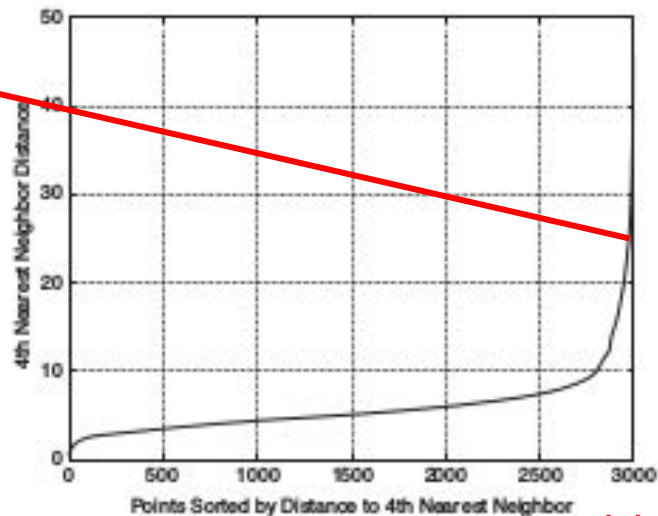
Es **ruido** si tiene menos de **MinPts** dentro de un radio **Eps**, y no está conectado con puntos semilla.

Ej. *Eps*, *MinPts* = 4



## Clustering por densidad (DBSCAN)

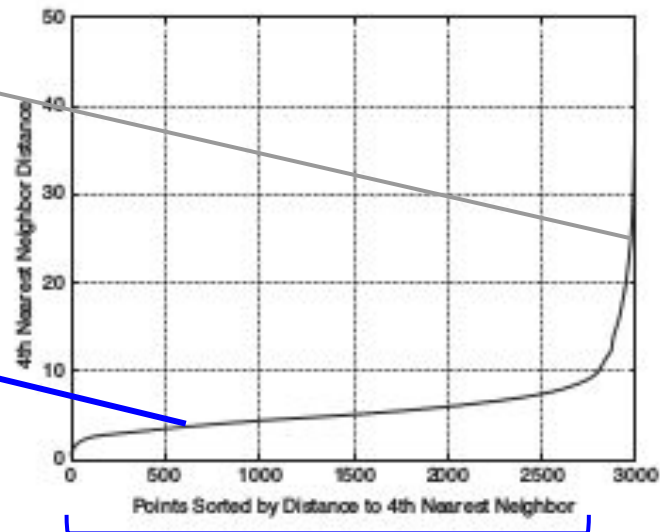
0. Elegir valores para los parámetros *Eps* y *MinPts*.



Ruido = Noise  
(Pocos) Puntos que  
están lejos del resto

## Clustering por densidad (DBSCAN)

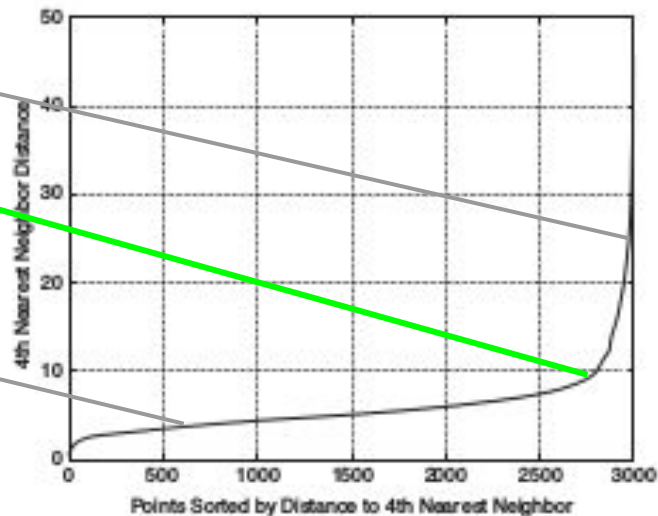
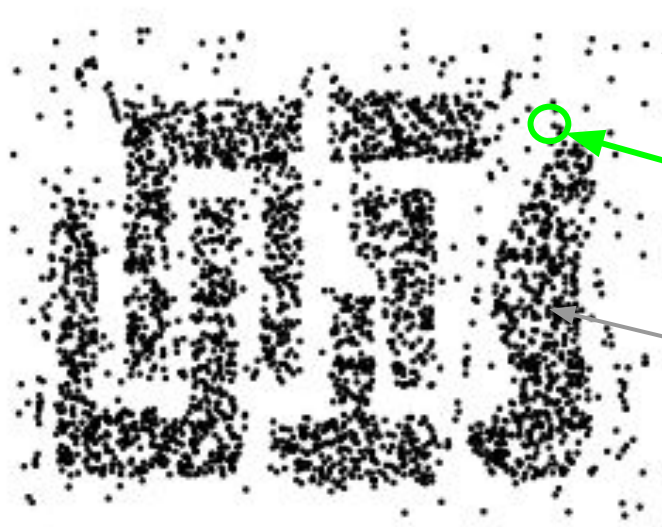
0. Elegir valores para los parámetros *Eps* y *MinPts*.



Semilla = Core  
(Muchos) Puntos que  
están cerca de otros

## Clustering por densidad (DBSCAN)

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

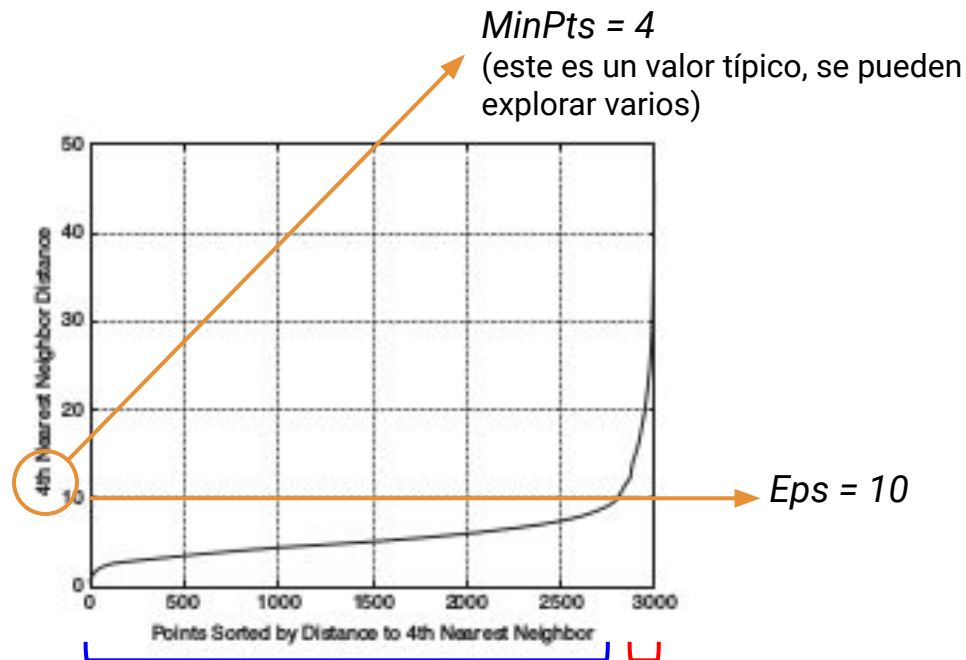


Borde = Border  
(Algunos) Puntos que  
están en el límite



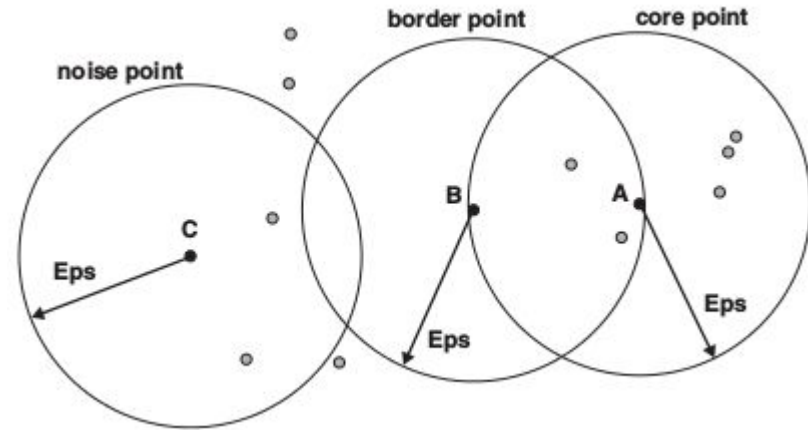
## Clustering por densidad (DBSCAN)

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



## 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.

*Eps*

**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.

*MinPts*

**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.

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

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

## 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.

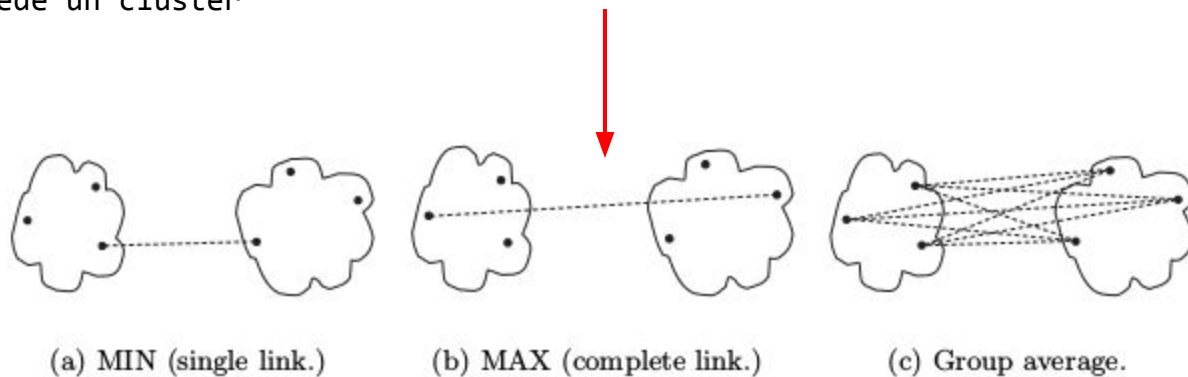
## Clustering jerárquico aglomerativo

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



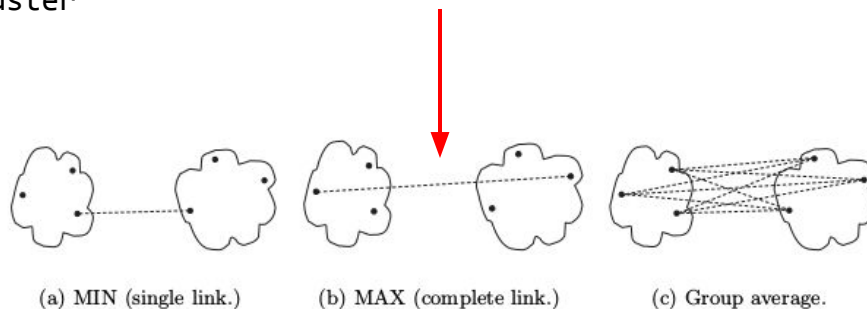
## Clustering jerárquico aglomerativo

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

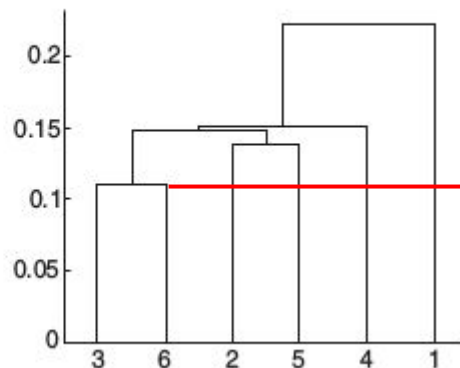
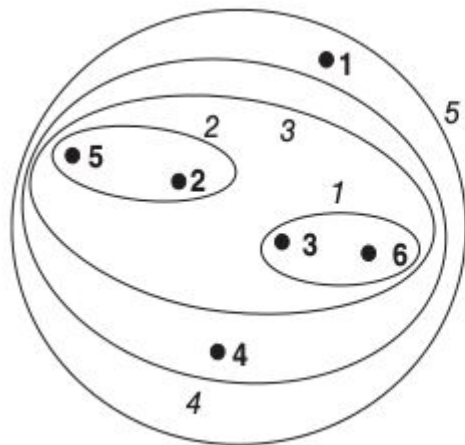
## Clustering jerárquico aglomerativo

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*

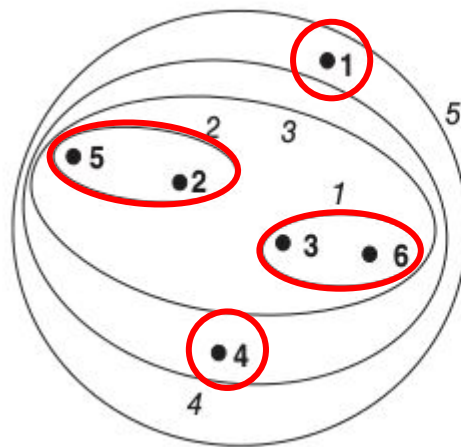
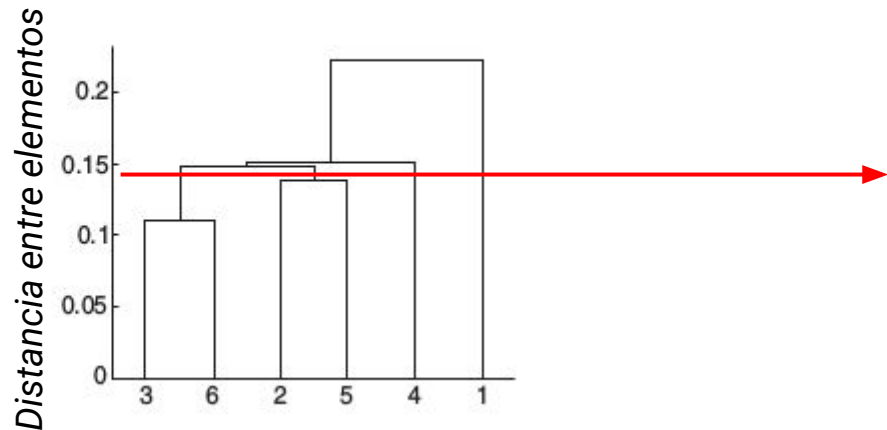
## Clustering jerárquico aglomerativo

- 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.



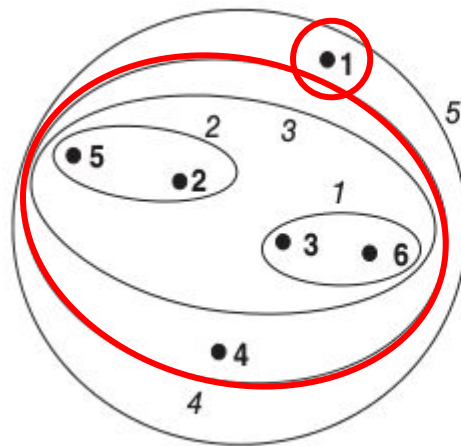
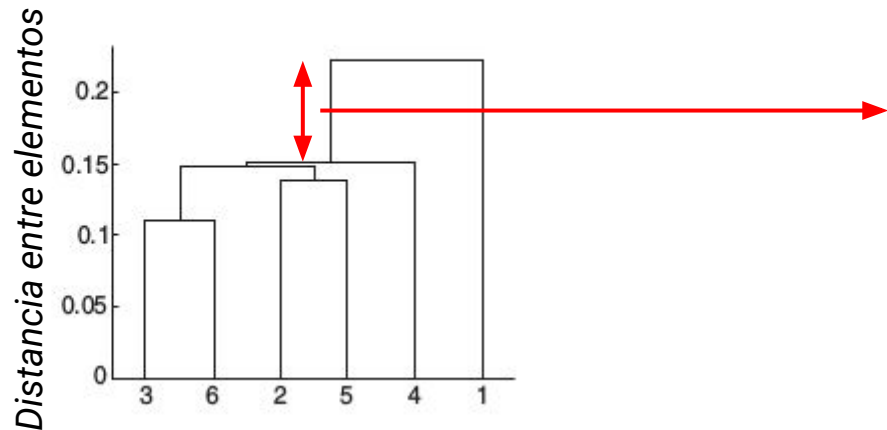
## Clustering jerárquico aglomerativo

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



## Clustering jerárquico aglomerativo

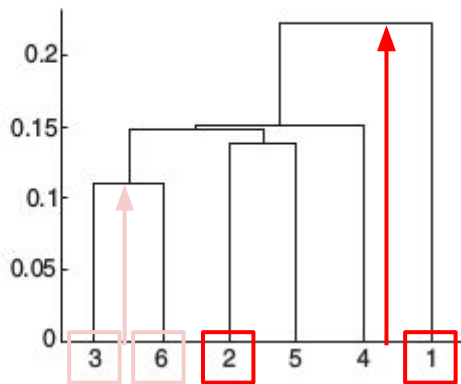
- Para generar los grupos es necesario establecer un criterio de corte.



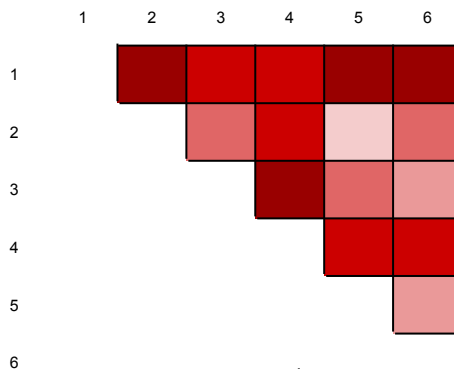
## Clustering jerárquico aglomerativo

- Para generar los grupos es necesario establecer un criterio de corte.  
(validar. el Coeficiente de Correlación Cofenético, CoPhenetic Correlation Coefficient, CPCC).

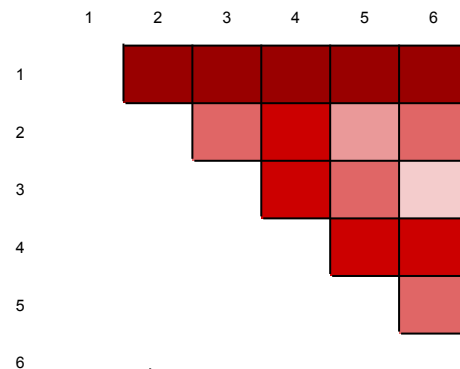
Distancia entre elementos



Distancias originales



Distancias en el árbol



Correlación

## 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,**

$$w_{ij} (z_{ij}) \in [0, 1] \quad \begin{array}{l} i=1\dots m \text{ registros} \\ j=1\dots k \text{ clusters} \end{array}$$

**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. **Repita**
  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  $SSE < \epsilon$   
**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  $SSE < \epsilon$   
si no, vuelve a 1.

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

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

## 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  $SSE < \epsilon$   
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  $SSE < \epsilon$   
si no, vuelve a 1.

$$c_j = \frac{\sum_{i=1}^m w_{ij}^p x_i}{\sum_{i=1}^m w_{ij}^p} \quad p=2$$

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

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

## Clustering difuso o c-means

<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)

