Minería de datos e Inteligencia de Negocios

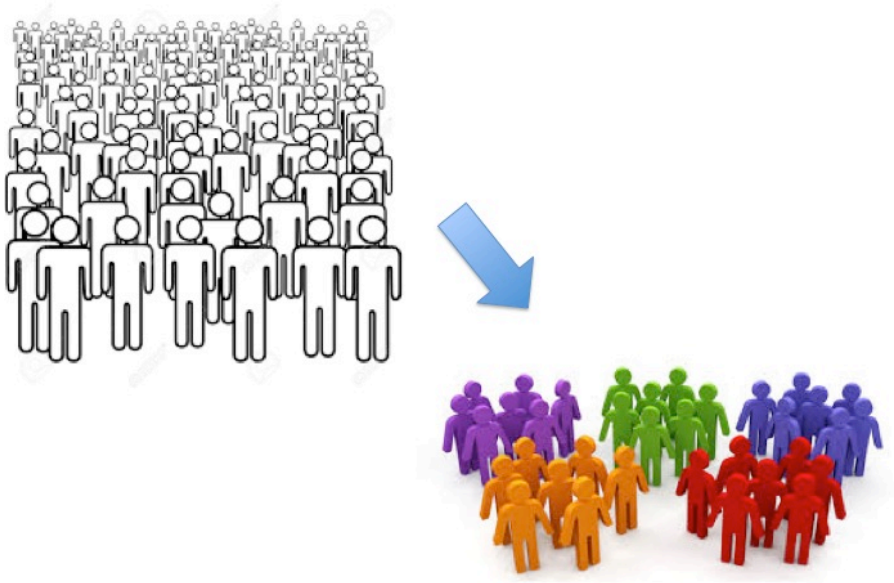
Clustering

Prof. Lorena Zúñiga S.

Marzo,2020

Clustering

- Encontrar posibles agrupaciones naturales en los datos
  - definidos o determinados por el algoritmo



Algunas aplicaciones

- Mercadeo − Segmentación de clientes
- Clustering de documentos
- Grupos de votantes
- *Web analytics* (sesiones)
- Pre-procesamiento para otros algoritmos predictivos
- NLP (Natural Language Processing)

Clustering
vs
Clasificación

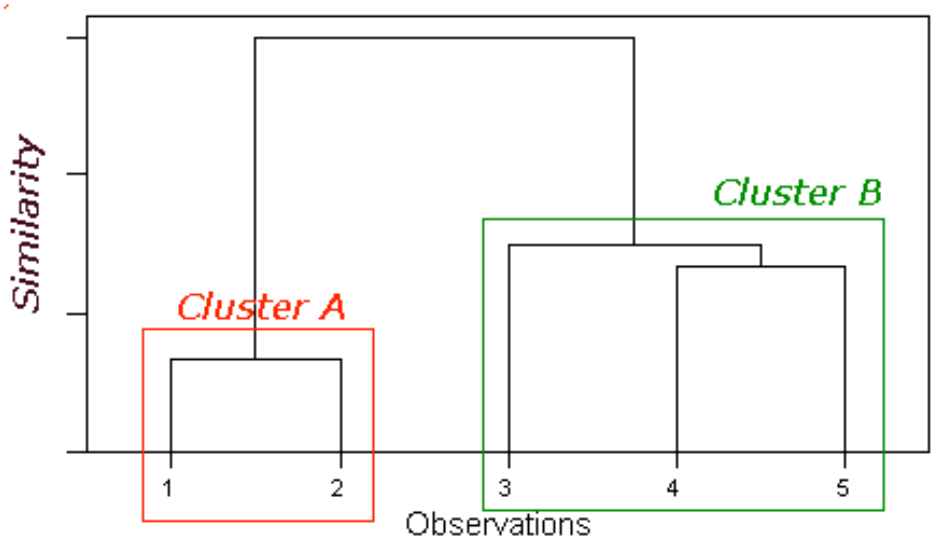
TIPOS DE TECNICAS DE CLUSTERING

Técnicas según el algoritmo

- **Clustering basado en prototipos**
- **Agrupación por densidad**
- **Clustering Jerárquico**

Técnicas según el algoritmo

- **Clustering jerárquico**
  - Bottom − up
  - Top - down



Técnicas según el algoritmo

- **Clustering basado en modelos**
  - Distribuciones de probabilidad
  - Mixture of Gaussians

Técnicas según el algoritmo

https://cran.r-project.org/web/views/Cluster.html

CRAN Task View: Cluster Analysis & Finite Mixture Models

Maintainer: Friedrich Leisch, Bettina Gruen
Contact: Bettina.Gruen at jku.at
Version: 2002-02-20
URL: https://CRAN.R-project.org/view=Cluster

This CRAN Task View contains a list of packages that can be used for finding groups in data
than one of the topics listed below, the section headings are mainly meant as quick starting p
and hence are part of every R installation), each package is listed only once.

Most of the packages listed in this CRAN Task View, but not all are distributed under the GP
distributed.

Hierarchical Clustering:

- Functions `hclust()` from package stats and `agnes()` from cluster are the primary func
  clustering. Faster alternatives to `hclust()` are provided by the packages fastcluster and
- Function `dendrogram()` from stats and associated methods can be used for improved v
- The dendextend package provides functions for easy visualization (coloring labels and
  heuristics for optimal branch rotations, and tree correlation measures with bootstrap an
- Package dynamicTreeCut contains methods for detection of clusters in hierarchical clu
- Package genie implements a fast hierarchical clustering algorithm with a linkage criter
  robustify the linkage method while retaining computational efficiency to allow for the
- Package idendr0 allows to interactively explore hierarchical clustering dendrograms ar

Tipos de clusters

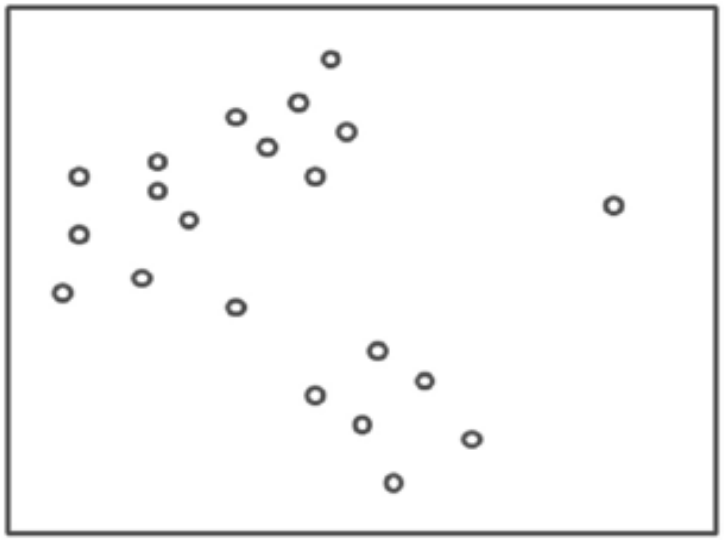
- Exclusivos
- Traslapados
- Jerárquicos
- Probabilísticos

Método de las k-medias

- Divide el espacio en k particiones
- Cada partición tiene un centroide o prototipo
- Objetivo: encontrar un prototipo para cada cluster
- Asigna cada punto al prototipo más cercano

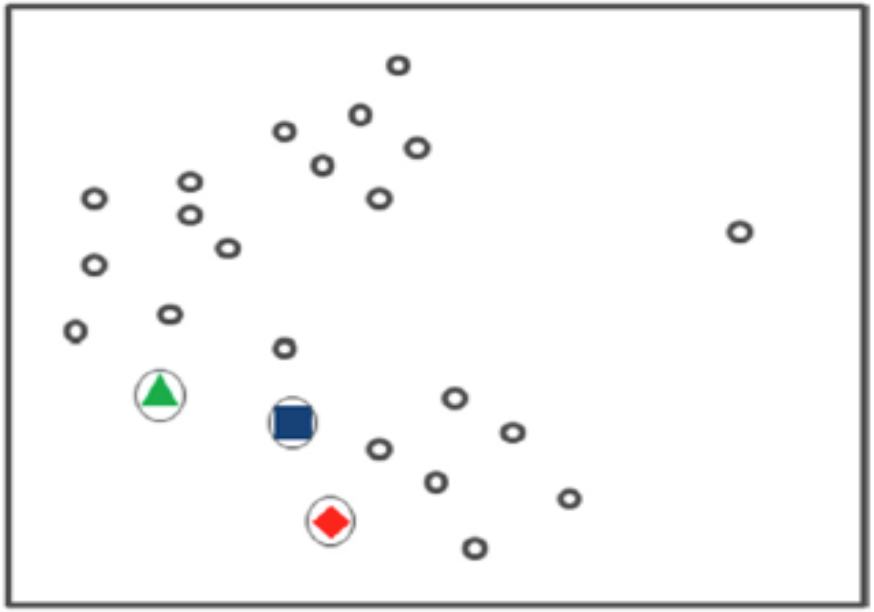
Ejemplo

Conjunto de datos de dos dimensiones



Ejemplo

- **1. Centroides iniciales**
  - se determinan aleatoriamente k centroides iniciales

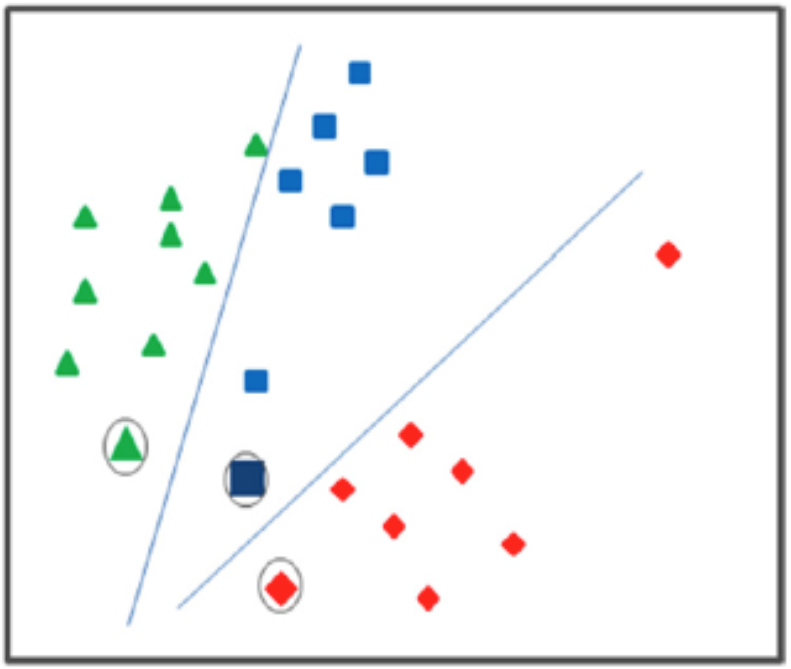


Método de las k-medias - Ejemplo

- **2. Asignación de puntos de datos**
  - Se asignan los puntos de datos al centroide más cercano
  - Determinar la cercanía con medidas de proximidad.
    - Ej: Euclidiana, Manhattan, etc.

Método de las k-medias - Ejemplo

- **2. Asignación de puntos de datos**

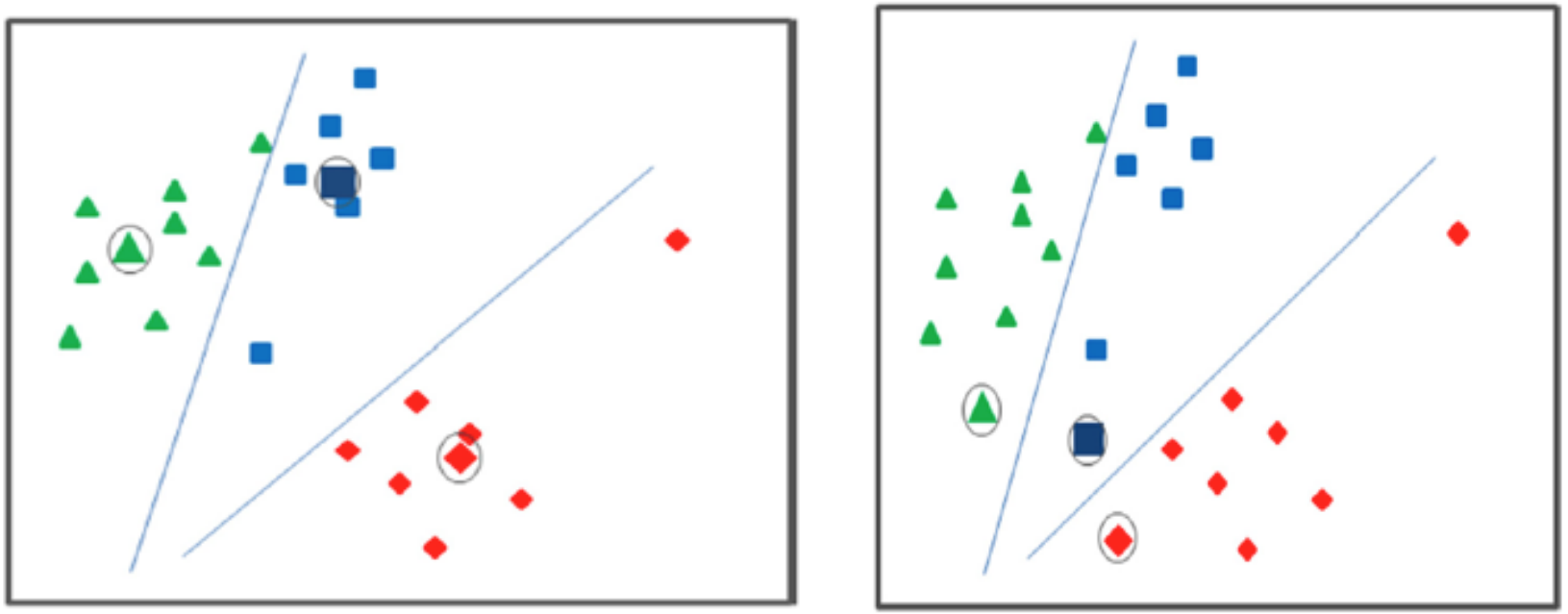


Método de las k-medias - Ejemplo

- **3. Calcular los nuevos centroides**
  - Para cada cluster calcular un nuevo centroide
  - El centroide con menor error es la nueva media del cluster

Ejemplo

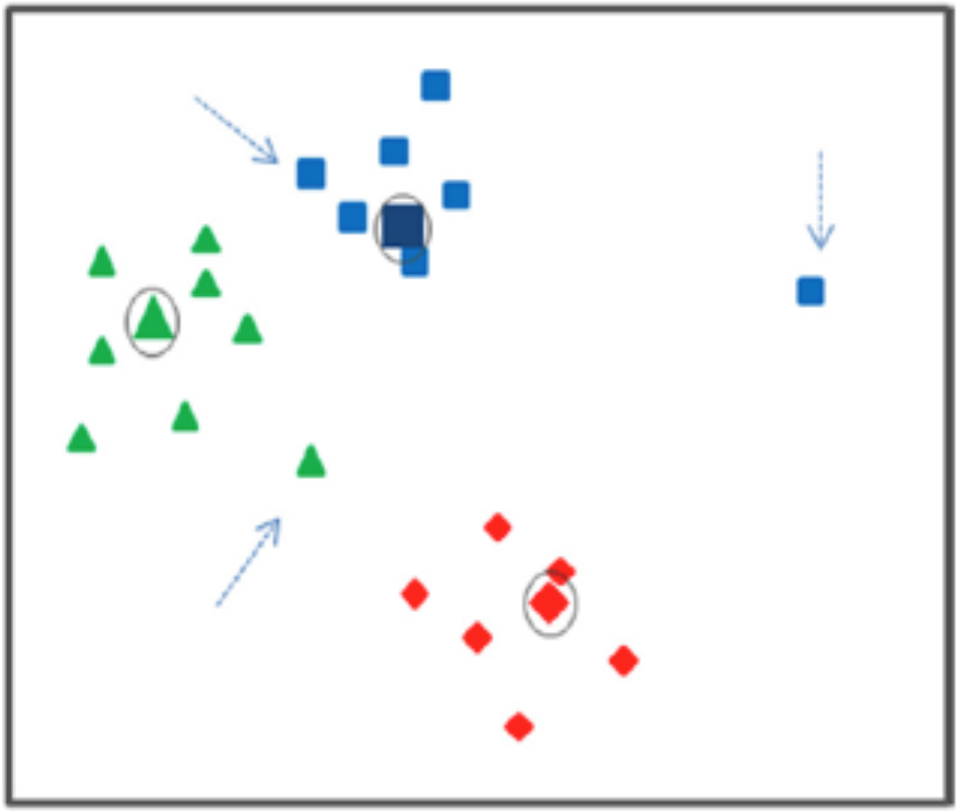
Nuevos centroides



Ejemplo

- **4. Repetir la asignación y calcular los nuevos centroides**
  - Después de identificar los nuevos centroides continúa la asignación de puntos
  - Repetir hasta asignar todos los puntos de datos
- **5. Finalizar**

Ejemplo



Algunas limitaciones

- La agrupación final depende de los centroides iniciales
- No existe una forma de estimar k
- Es sensible a valores atípicos
- Puede generar clusters vacíos

K-MEANS
EJEMPLO - R

K-medoids

- PAM (Partitioning Around Medoids)
- Algoritmo para encontrar los centroides

1. **Elegir k puntos de datos como puntos iniciales para los centroides de los clusters**
2. Calcular la distancia vs todos los puntos
3. Clasificar cada punto en el cluster donde esté más cerca del centroide
4. **Seleccionar un nuevo punto que minimice la suma de las distancias en ese cluster**
5. Repetir hasta que no cambien los centroides

KMEDOIDS
EJEMPLO - R

¿Cómo escoger el valor de k?

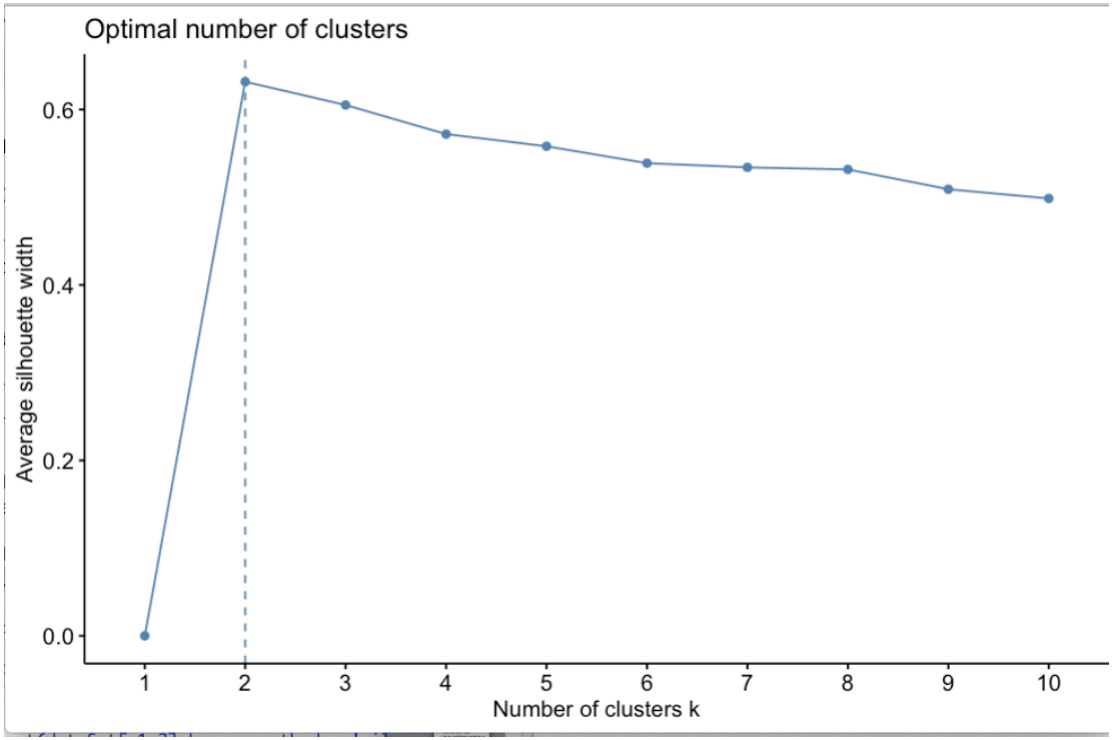
- Método del codo (WSS = Within Sum of Squares)
- Promedio Silhouette
- Estadística Gap

Silhouette score

- Cuantifica la calidad de los clusters generados por un algoritmo
- Calcula distancias promedio entre un punto P y todos los puntos en su cluster y en clusters cercanos

Silhouette score

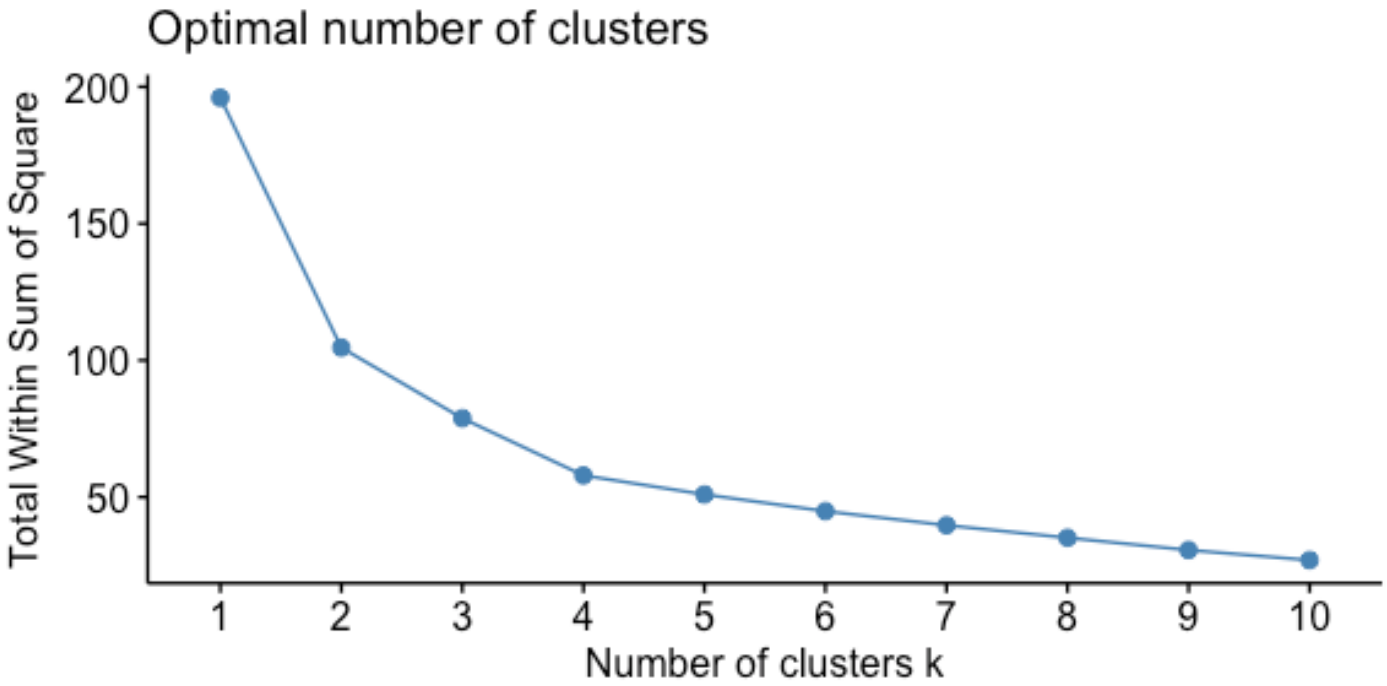
- Genera scores por punto, por cluster y finalmente para el dataset
- Valores entre -1 y 1



Método del codo (WSS)

- Suma de los cuadrados de las distancias de todos los puntos de un cluster
- Usar diferentes valores de k
- Graficar los valores de k vs la sumatoria (WSS)
- Identificar k (cuando WSS ya no disminuye drásticamente)

Método del codo (WSS)



Estadística Gap

- Compara el resultado WSS del conjunto de datos vs el WSS un conjunto de datos de referencia
  - Distribución uniforme de los datos
- Se elige el valor de Gap que sea mayor

EJEMPLO – R

SILHOUETTE SCORE

WSS

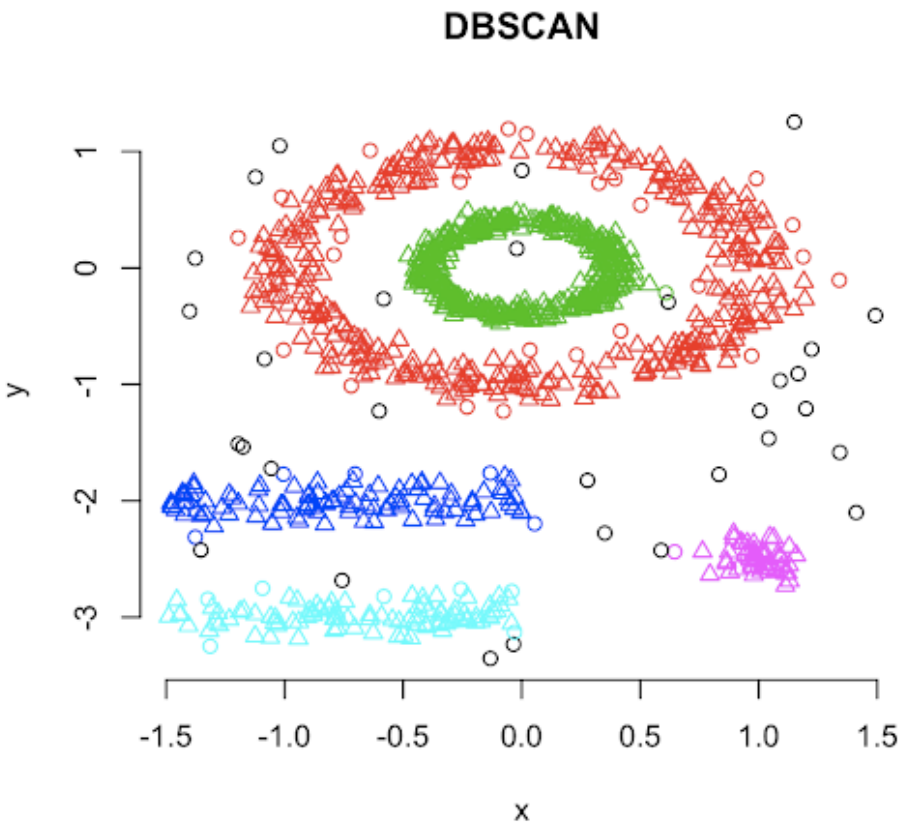
GAP

Density Based Spatial Clustering of Applications with Noise

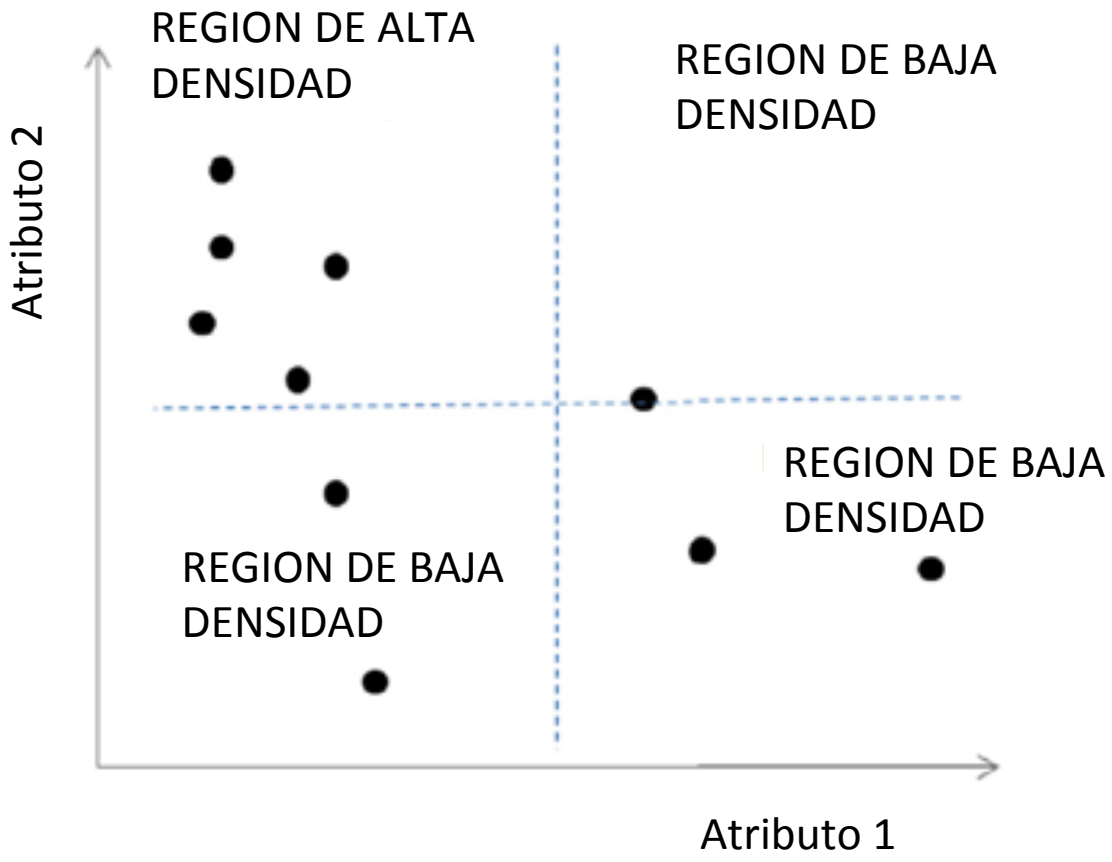
DBSCAN

DBSCAN

- Identifica clusters mediante la medición de la distribución de l densidad en un espacio de N dimensiones
- No hay que especifica # de clusters

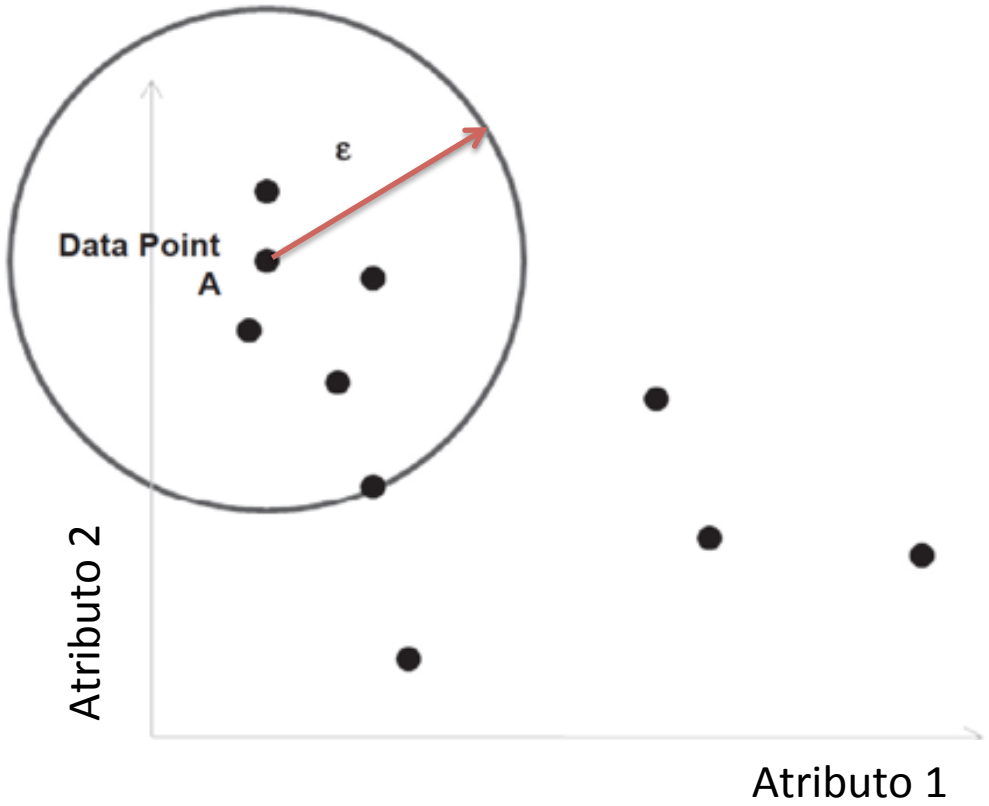


Densidad


REGION DE ALTA DENSIDAD
REGION DE BAJA DENSIDAD
REGION DE BAJA DENSIDAD
REGION DE BAJA DENSIDAD
Atributo 2
Atributo 1

Densidad

6 = # de puntos en el espacio circular de radio ε

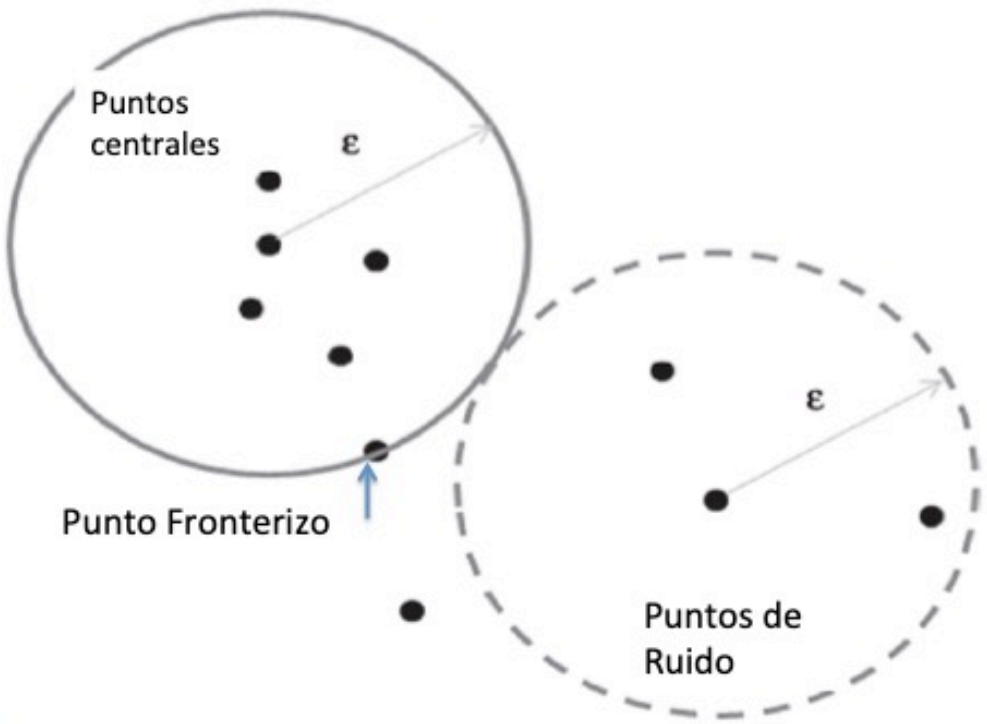


DBSCAN

- **Paso 1 – Definir el límite de densidad**
  - El radio (ε) es fijo y se usa en el cálculo de la densidad para todos los puntos
  - MinPoints = cantidad de puntos que definen si un sector es de alta o baja densidad

DBSCAN

- **Paso 2 - Clasificar puntos de datos**
  - Central points
  - Border points
  - Noise points



DBSCAN

Paso 3 - Agrupar puntos

- Se forman grupos de puntos centrales
- Dos puntos pertenecen al mismo cluster si la distancia entre ellos es **ε**
- *Noise points* no se asignan a ningún cluster

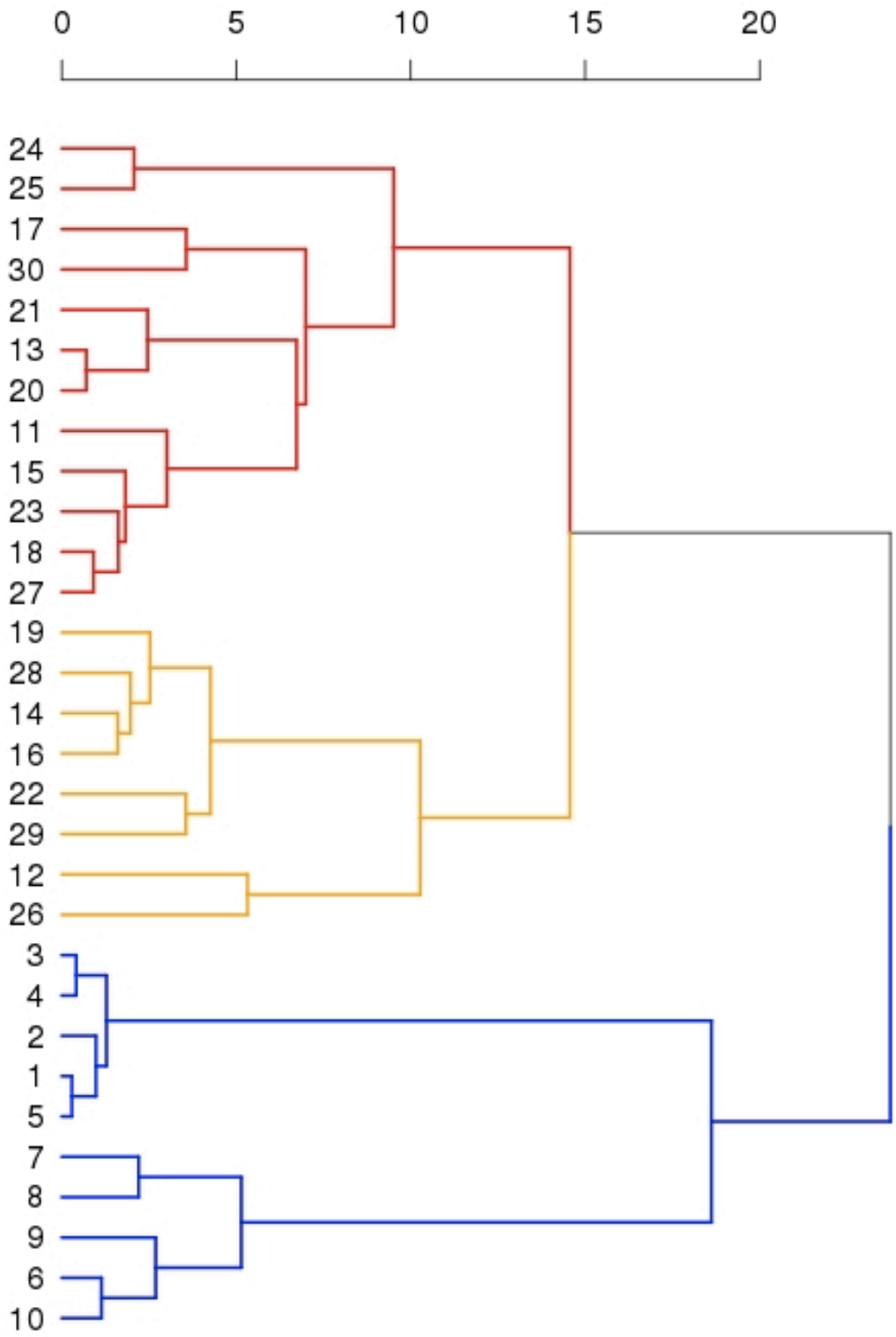
EJEMPLO – DBSCAN EN R

```
library(dbscan)
library(factoextra)


clusters<- dbscan(iris[,1:2],eps=0.3, minPts = 5)


fviz_cluster(clusters,data=iris[,1:2],geom='point',palette='set2',
             ellipse = F)
```

CLUSTERING JERARQUICO

Clustering jerárquico

- No se requiere un número previo de clusters
- Dos tipos de algoritmos:
  - Aglomerativos (bottom-up)
  - Divisivos (top-down)
- Representación mediante dendrograma


0
5
10
15
20
24
25
17
30
21
13
20
11
15
23
18
27
19
28
14
16
22
29
12
26
3
4
2
1
5
7
8
9
6
10

Algoritmos aglomerativos

1. Inicia con N clusters de 1 punto
2. Compara pares de puntos mediante medidas de distancia
3. Aplica un criterio (*linkage*) para combinar los puntos
4. Recalcula las distancias
5. Definir dónde cortar el árbol (dendrograma)

Algoritmos aglomerativos

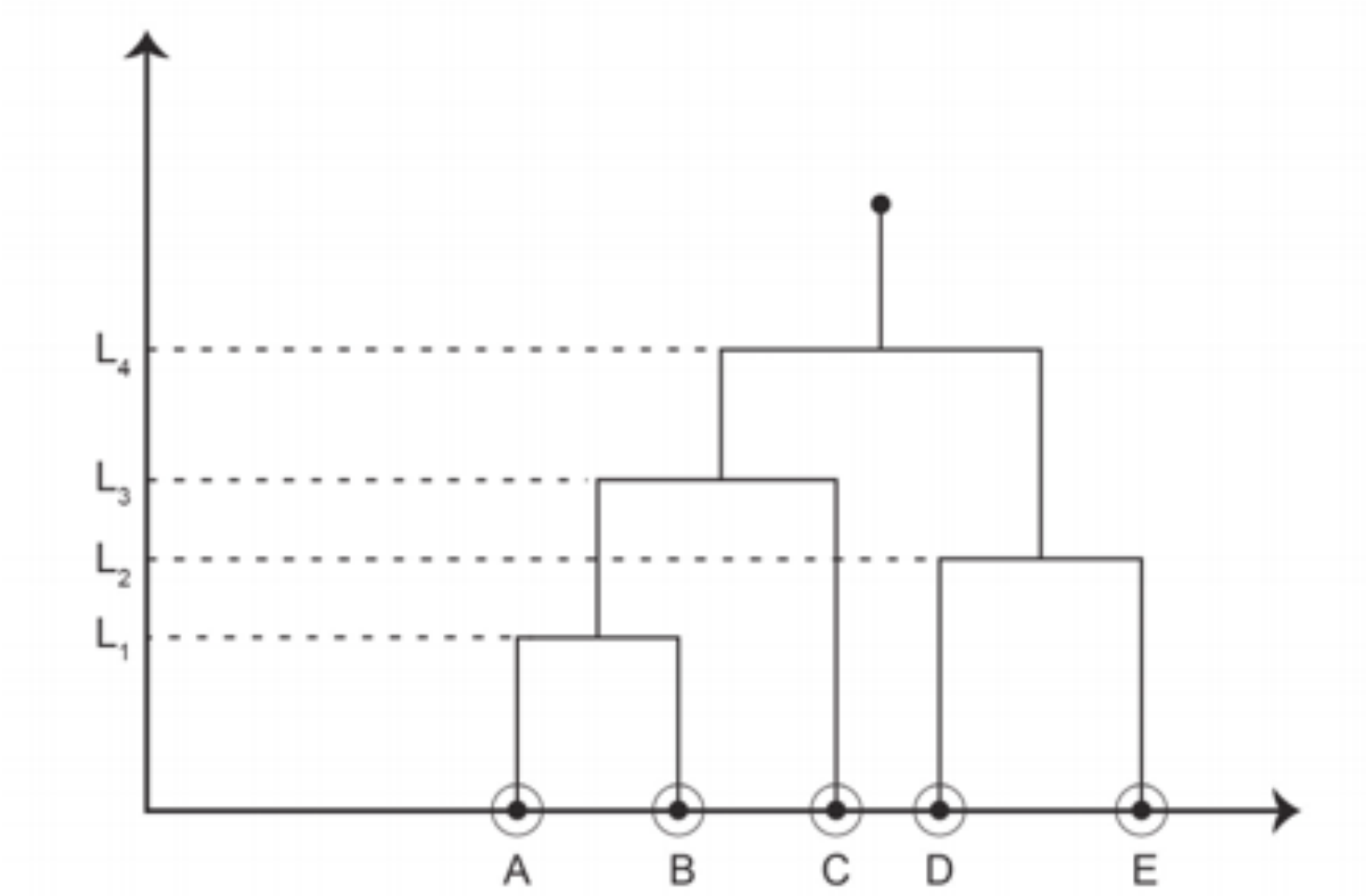
Linkage

- Máximo o Completo
- Mínimo o Single
- Promedio
- Centroide
- Método de Ward

Algoritmos aglomerativos

- Cada punto se toma como un cluster
- Calcular medidas de semejanza entre pares de clusters
- Unir los dos clusters más semejantes
- Repetir hasta que sólo quede un cluster

Algoritmos aglomerativos



Algoritmos divisivos

- Inicia con un cluster que contiene todos los puntos
- Calcular medidas de semejanza entre pares de clusters
- Separa los dos clusters menos semejantes
- Repetir hasta que sólo quede un cluster
- DIANA = **Di**visive **Ana**lysis

¿Preguntas?