# Clustering

Jocelyn Dunstan Escudero

jdunstan@uc.cl

Departamento de Ciencia de la Computación
& Instituto de Matemática Computacional
Pontificia Universidad Católica de Chile

Santiago, Chile


22 de octubre de 2025

# Objetivos

- Comprender clustering como el acto de organizar objetos similares en grupos dentro de un algoritmo de aprendizaje automático
- Conocer los tipos de clustering.

# Clustering

# Idea general

- En el espacio de los predictores, la idea es calcular las distancias (normalmente euclidianas) y agrupar las observaciones similares.

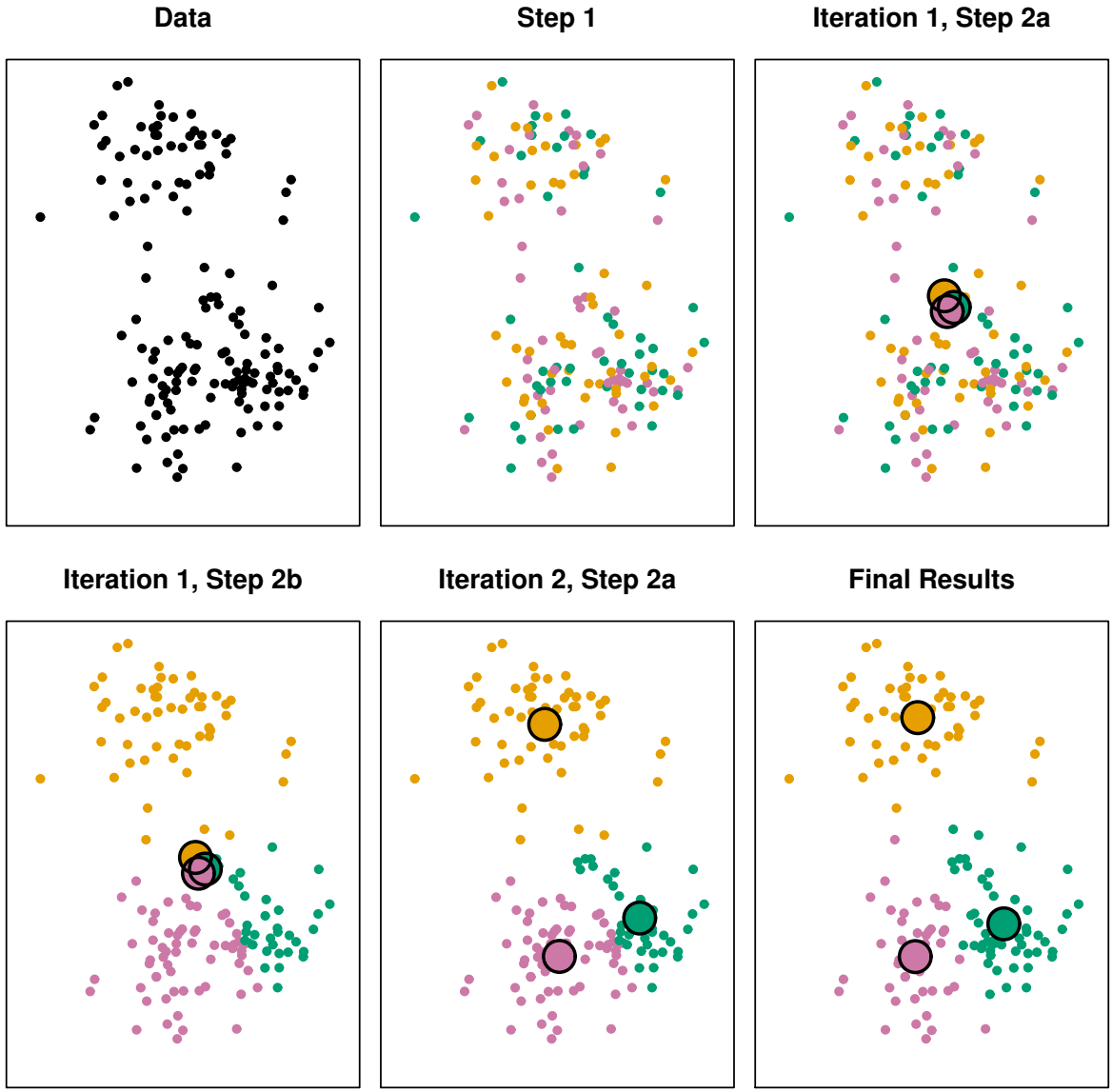
# Idea general

- En el espacio de los predictores, la idea es calcular las distancias (normalmente euclidianas) y agrupar las observaciones similares.
- Esto en la industria del marketing: quieres encontrar clientes similares y enviarles publicidad personalizada (por ejemplo, basada en los vídeos de Youtube que ven)

# Idea general

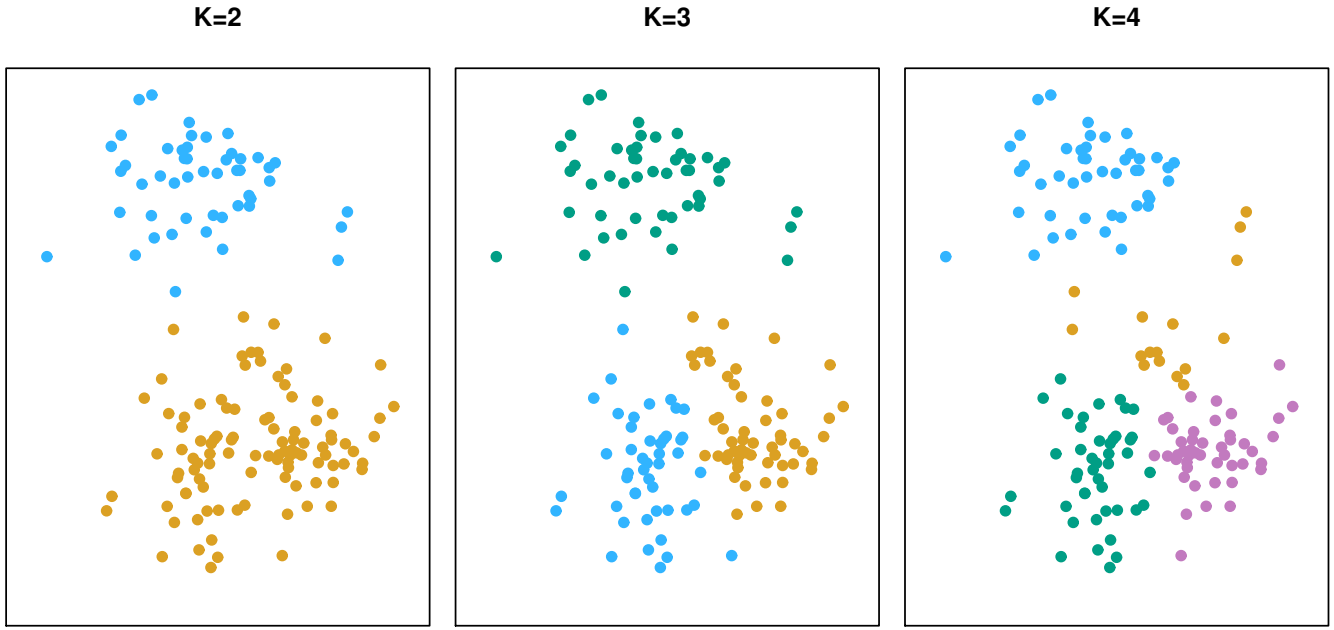
- En el espacio de los predictores, la idea es calcular las distancias (normalmente euclidianas) y agrupar las observaciones similares.
- Esto en la industria del marketing: quieres encontrar clientes similares y enviarles publicidad personalizada (por ejemplo, basada en los vídeos de Youtube que ven)
- Veremos tres tipos: k-means, en el que se especifica el número de clusters, jerárquico, en el que podemos construir diferentes clusters y luego decidir cuántos queremos, y DBSCAN que explota la idea que un cluster son regiones de alta densidad.

# k-means




https://www.youtube.com/watch?v=5I3Ei69I40s

# Diferente elección de k




https://www.statlearning.com/

# K-means

- Input: dataset con atributos numéricos. Parámetro: número de clusters K.

# K-means

- Input: dataset con atributos numéricos. Parámetro: número de clusters K.
- Se asignan k clases a los datos de manera aleatoria y con eso se calculan centroides

# K-means

- Input: dataset con atributos numéricos. Parámetro: número de clusters K.
- Se asignan k clases a los datos de manera aleatoria y con eso se calculan centroides
- Iterativamente se asignan clases y se recalculan centroides:
  1. **Asigno** la clase del centroide más cercano
  2. **Recalculo** los centroides promediando la posición de los puntos de esa clase.

# K-means

- Input: dataset con atributos numéricos. Parámetro: número de clusters K.
- Se asignan k clases a los datos de manera aleatoria y con eso se calculan centroides
- Iterativamente se asignan clases y se recalculan centroides:
  1. **Asigno** la clase del centroide más cercano
  2. **Recalculo** los centroides promediando la posición de los puntos de esa clase.
- Itero hasta converger, es decir, que la posición de los centroides no cambia.

Basado en el curso de F. Bravo y B. Poblete

# Cálculo de Centroides

Si tenemos los siguientes tres vectores:

$$\vec{x_1} = [6, 4, 5], \vec{x_2} = [4, 5, 1], \vec{x_3} = [2, -3, 5]$$

El centroide de estos vectores será:

$$c(\vec{x_1}, \vec{x_2}, \vec{x_3}) = [(6+4+2)/3, (4+5-3)/3, (3+1+5)/3] = [4, 2, 3]$$

Basado en el curso de F. Bravo y B. Poblete

# Detalles de k-means

- La distancia de los puntos al centroide se mide con alguna distancia (Euclideana usualmente)

# Detalles de k-means

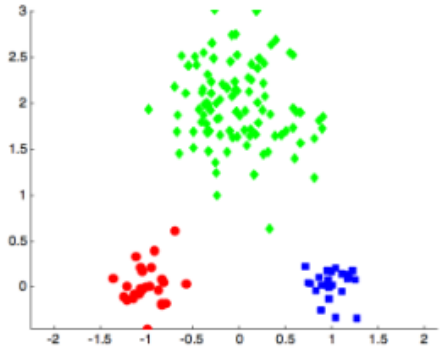
- La distancia de los puntos al centroide se mide con alguna distancia (Euclideana usualmente)
- K-means converge para distancias usuales

# Detalles de k-means

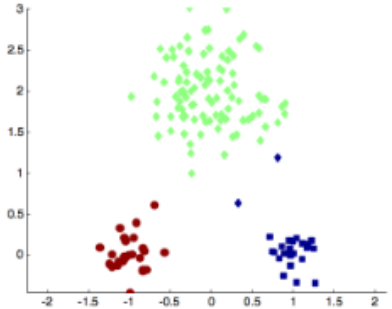
- La distancia de los puntos al centroide se mide con alguna distancia (Euclideana usualmente)
- K-means converge para distancias usuales
- La convergencia ocurre en general con pocas iteraciones. Si n es el número de puntos, K centros, I iteraciones y d dimensiones, la complejidad es O(n·K·I·d)

Basado en el curso de F. Bravo y B. Poblete

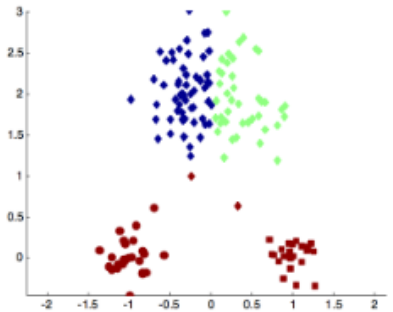
# K-means podría no encontrar clusters óptimos



*Puntos originales*



*Clustering óptimo*



*Clustering sub-optimal*

Basado en el curso de F. Bravo y B. Poblete

# Fortalezas y limitaciones de k-means

**Fortalezas:**

- Es escalable considerando la complejidad
- Funciona bien para clusters esféricos

# Fortalezas y limitaciones de k-means

**Fortalezas:**

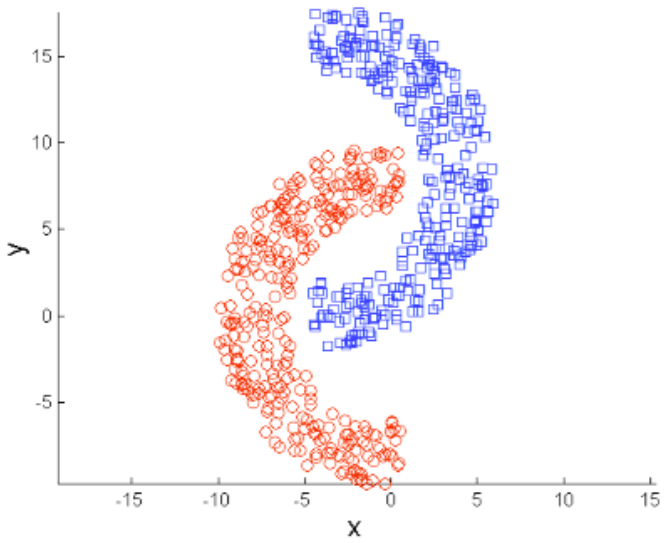
- Es escalable considerando la complejidad
- Funciona bien para clusters esféricos

**Limitaciones:**

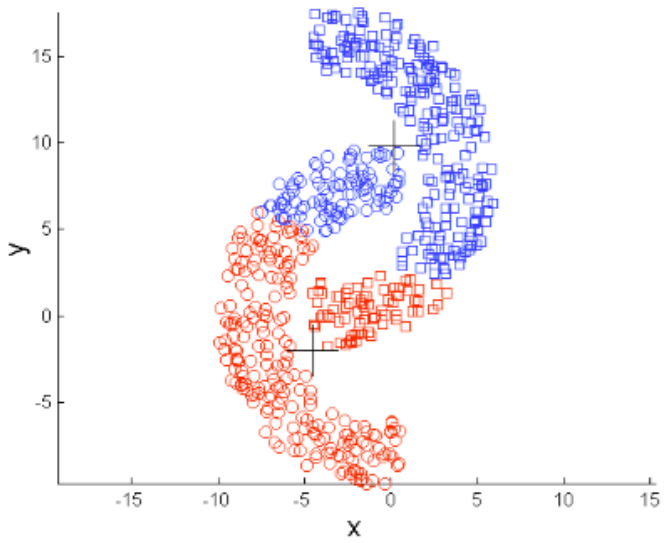
- No funcionan bien para clusters de formas complejas o tamaños muy distintos.
- Trata de dividir clusters cuando detecta densidades muy altas.

https://realpython.com/k-means-clustering-python/

# Ejemplo clásico de cluster no esférico




*Puntos originales*




*K-means (dos clusters)*

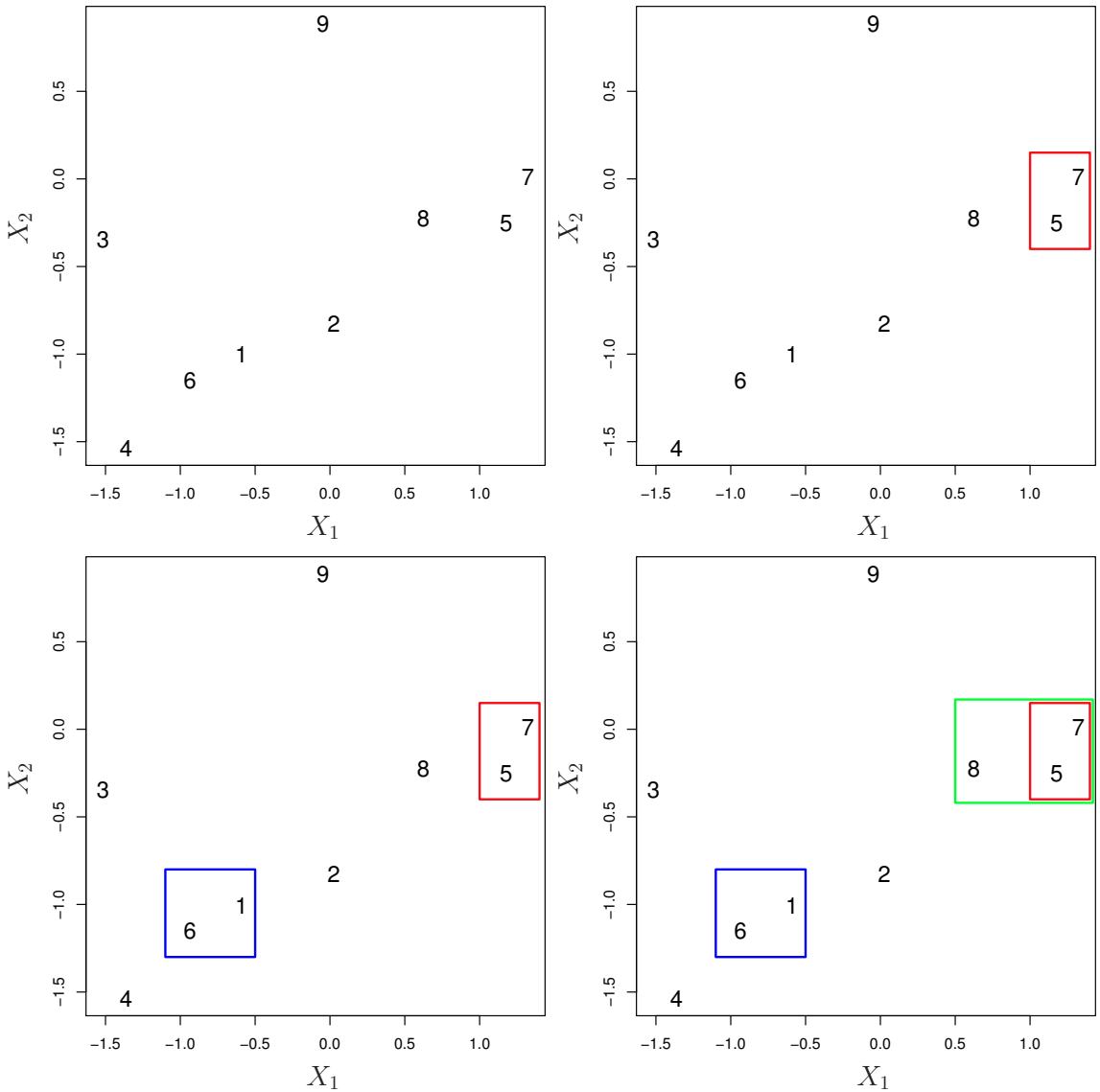
Basado en el curso de F. Bravo y B. Poblete

# Jerárquico

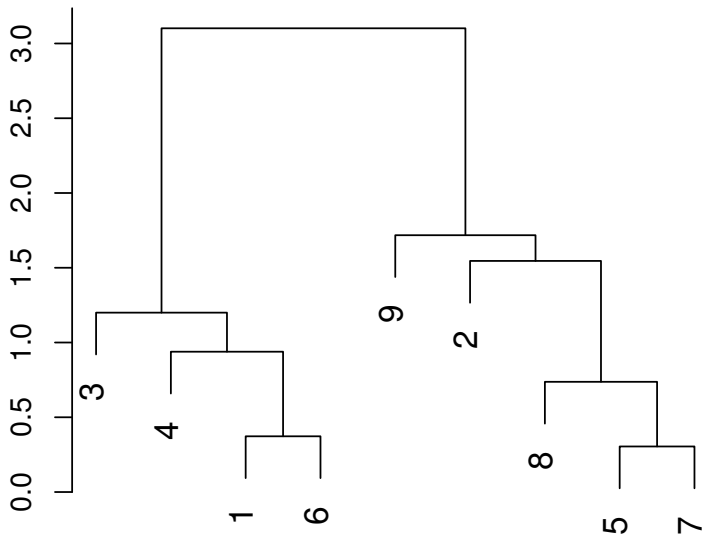
# Idea central del cluster jerárquico

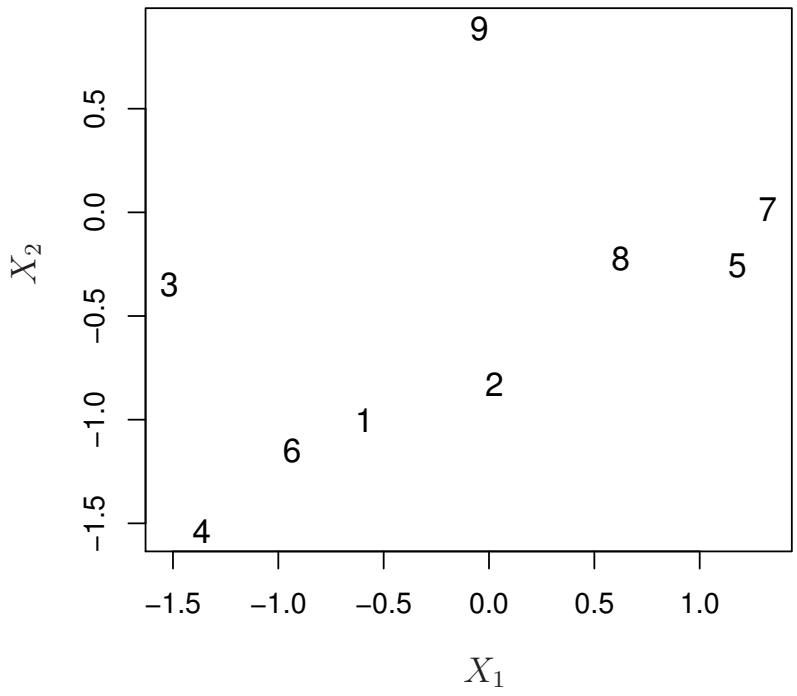
- Puede ser aglomerativo o divisivo.
- En el **aglomerativo** cada punto es un cluster y en cada paso se mezclan par de clusters hasta que queda un solo cluster (o k clusters)
- En **divisivo** partimos de un solo cluster y vamos dividiendo de dos en dos hasta que cada cluster tenga un solo punto (o k cluster).
- En este caso también necesitamos medir distancia entre puntos.

# Jerárquico aglomerativo



# Dendograma



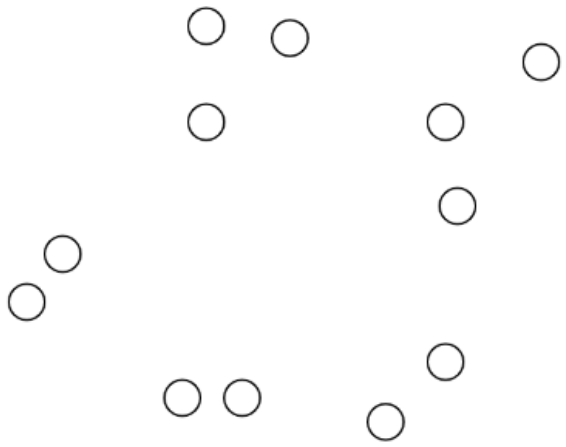

https://www.statlearning.com/

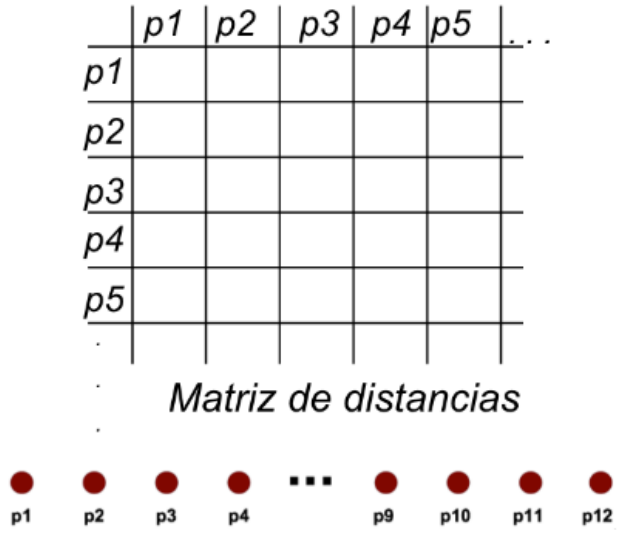
# Clustering jerárquico aglomerativo

- Parto con cada punto un cluster.
- Calculo la matriz de distancias.
- Fusiono par de clusters cercanos
- Actualizo la matriz de distancia
- Me detengo hasta tener un solo cluster (o k clusters, con k un parámetro).

Basado en el curso de F. Bravo y B. Poblete

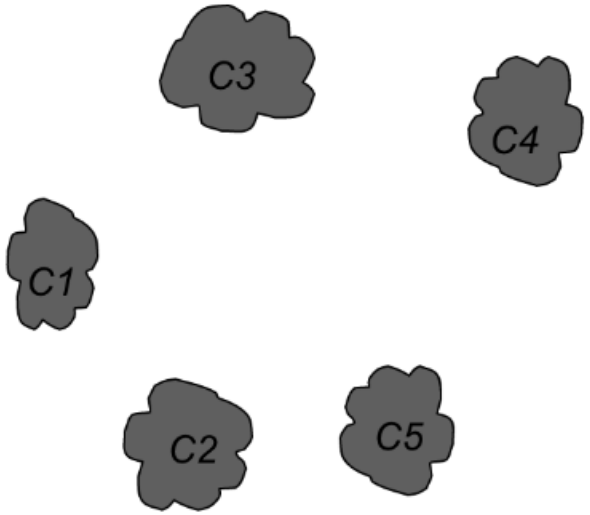
# Puntos individuales y matriz de distancia





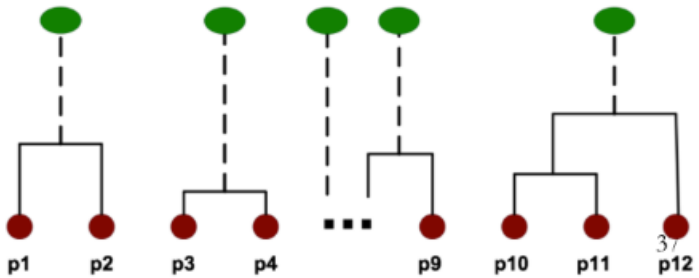

Basado en el curso de F. Bravo y B. Poblete

# Después de un par de iteraciones



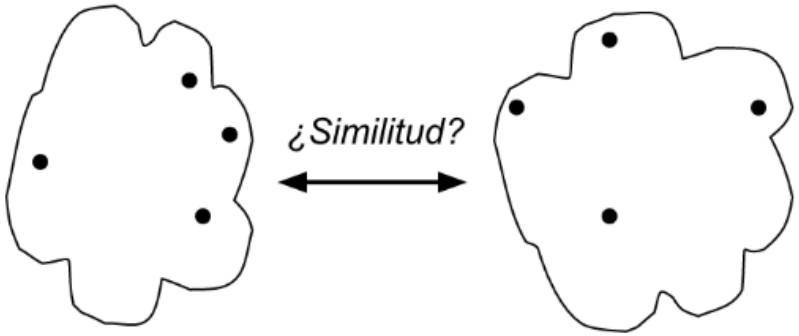

| | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|
| C1 | | | | | |
| C2 | | | | | |
| C3 | | | | | |
| C4 | | | | | |
| C5 | | | | | |

Matriz de distancias




Basado en el curso de F. Bravo y B. Poblete

# ¿Cómo definir la distancia entre clusters?




- MIN (single link)
- MAX (complete link)
- Promedio del grupo
- Distancia entre centroides

| | p1 | p2 | p3 | p4 | p5 | . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

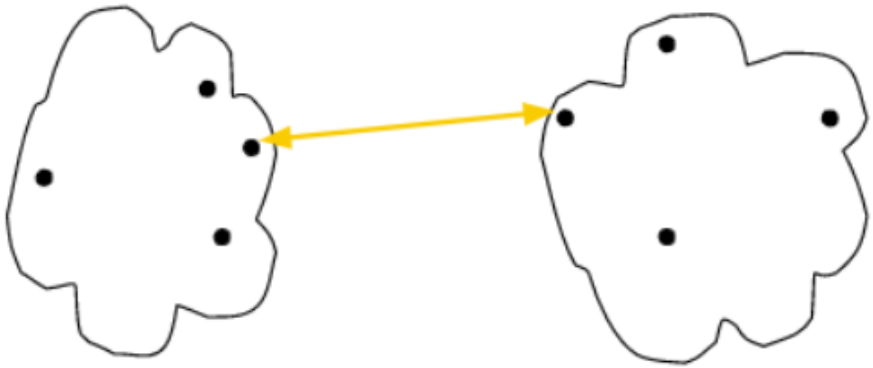
.

.

*Matriz de distancias*

Basado en el curso de F. Bravo y B. Poblete

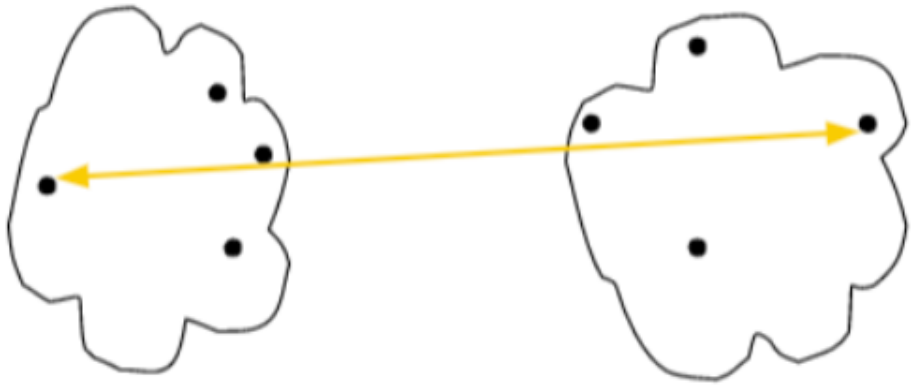
# *Single link*



- MIN (single link)
  - Considero los dos puntos más cercanos entre sí (cada uno de un cluster distinto)

Basado en el curso de F. Bravo y B. Poblete

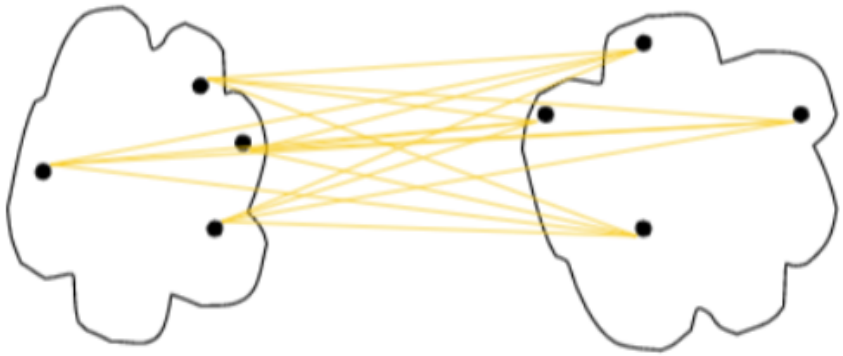
# *Complete link*



- MAX (complete link)
  - Considero los dos puntos más lejanos entre sí (cada uno de un cluster distinto)

Basado en el curso de F. Bravo y B. Poblete

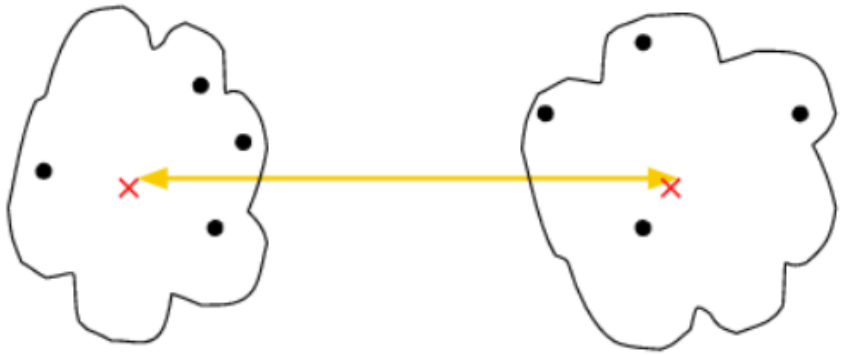
# Promedio del grupo



- Promedio del grupo
  - Distancia promedio de todos los pares de puntos (cada par tiene un punto por cluster)

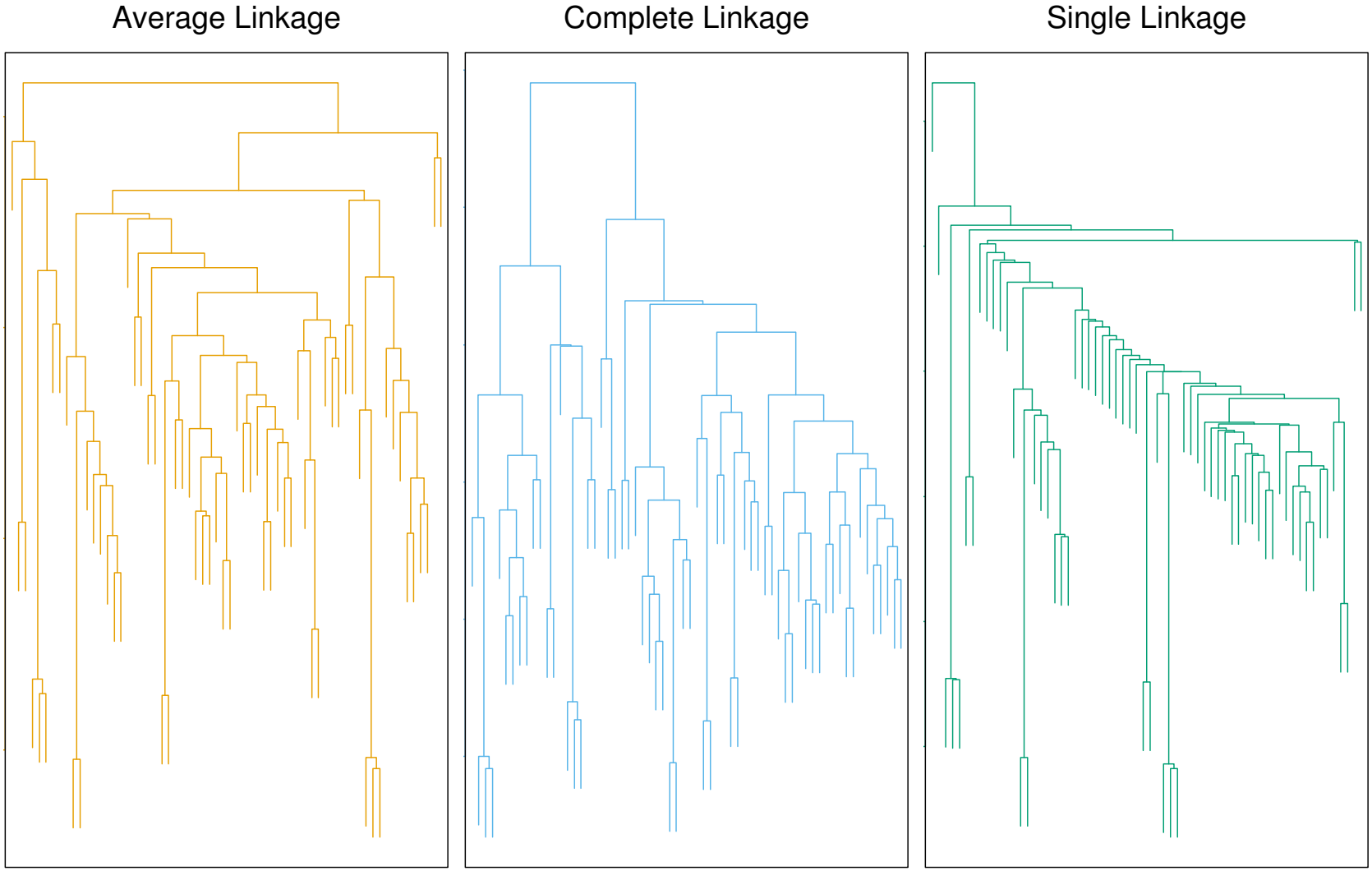
Basado en el curso de F. Bravo y B. Poblete

# Distancia entre centroides



- Distancia entre centroides
  - distancia entre los centroides de cada grupo

Basado en el curso de F. Bravo y B. Poblete



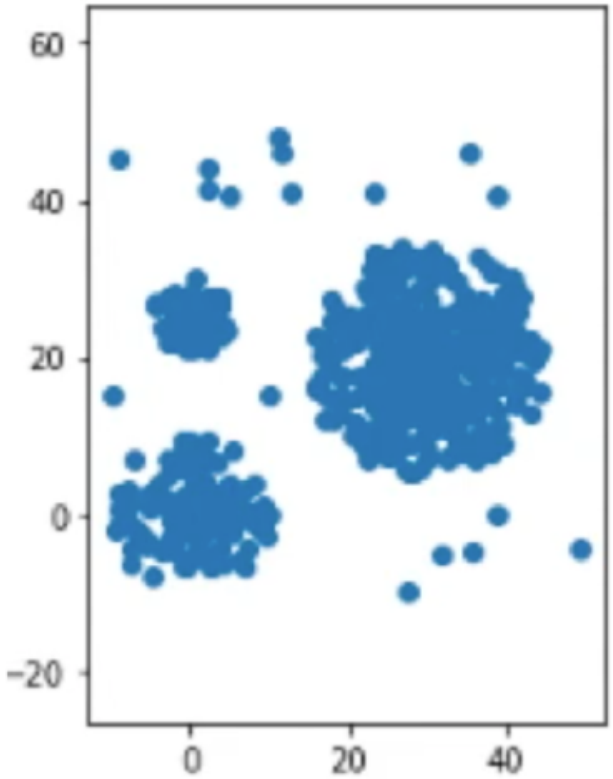

https://www.statlearning.com/

# Density-based spatial clustering of applications of noise (DBSCAN)

# DBSCAN

- El 2014 este algoritmo ganó "the test of time award" en la conferencia lider de data mining ACM SIGKDD.
- La idea es encontrar regiones de alta densidad y reconocer en ellas los clusters.
- En los métodos anteriores nos veíamos forzados a asignar a cada punto un cluster, lo que hace el método sensible a outliers y ruido. En DBSCAN puedo tener puntos dentro del cluster (**core**) pero también existen puntos en el borde (**border**) y puntos de ruido (**noise**)

# DBSCAN




Los humanos pueden claramente identificar los clusters mirando la densidad

`https://www.youtube.com/watch?v=c10ujiY1ZQ8`

# Parámetros

**Parámetros:**

1. **Eps**: radio especificado
2. **MinPts**: número mínimo de puntos en una región

**Tipos de puntos:**

1. **Core**: con más puntos que MinPts a una distancia Eps. Dentro del cluster
2. **Border**: menos puntos que MinPts pero en la vecindad de un punto core
3. **Noise**: un punto que no es ni core ni border.

Basado en el curso de F. Bravo y B. Poblete

# Comentarios finales de clustering

- *Small decisions with big consequences*: hay muchas decisiones que tomar y no hay una respuesta directa.

# Comentarios finales de clustering

- *Small decisions with big consequences*: hay muchas decisiones que tomar y no hay una respuesta directa.
- Una opción para validar los clusters obtenidos es hacer varios con muestras de datos. Además comparar los clusters que se obtienen con las diferentes elecciones.

# Comentarios finales de clustering

- *Small decisions with big consequences*: hay muchas decisiones que tomar y no hay una respuesta directa.
- Una opción para validar los clusters obtenidos es hacer varios con muestras de datos. Además comparar los clusters que se obtienen con las diferentes elecciones.
- Clustering puede no ser un método robusto por lo que sus resultados deben ser analizados con cautela.

https://www.statlearning.com/