Algoritmos de Clustering

Basado en las slides de Bárbara Poblete

Introducción

- ¿Qué es análisis de clusters?

- Técnica para encontrar grupos de objetos tal que los objetos en un grupo sean similares (o relacionados) entre sí y que sean diferentes (o no relacionados) a los objetos en otros grupos.
- Es una técnica de aprendizaje no-supervisado (no requiere etiquetas para los datos).

Introducción

Un clustering es una colección de clusters.

Tipos de clusterings:

- Clustering Particional
  - Divide los datos en subconjuntos sin traslape (clusters), tal que cada dato está en un solo subconjunto
- Clustering Probabilístico o Difuso
  - Cada objeto pertenece a cada cluster con un peso de pertenencia entre 0 y 1.
- Clustering Jerárquico
  - Un conjunto de clusters anidados, organizados como un árbol.

Métodos de clustering

- K-means
- Método jerárquico aglomerativo
- DBSCAN
- Mixture of Gaussians y algoritmo EM

K-means

Método de clustering particional

- Input: Un dataset de atributos numéricos.
- Párametro: número de clusters K
- Se asignan K centroides iniciales (aleatorios)
- Se realizan dos operaciones iterativamente: **asignar** y **recalcular centroides**
  a. Asignar: cada punto es asignado a su centroide más cercano.
  b. Recalcular centroides: se recalculan los centroides promediando sus puntos.

- Iterar hasta converger.

Cálculo de Centroide

Sean 3 vectores de tres de dimensiones:

$$\vec{x}_1 = [6, 4, 3], \vec{x}_2 = [4, 5, 1], \vec{x}_3 = [2, -3, 5]$$

El centroide de estos vectores es:

$$c(\vec{x}_1, \vec{x}_2, \vec{x}_3) = [(6 + 4 + 2)/3, (4 + 5 - 3)/3, (3 + 1 + 5)/3] = [4, 2, 3]$$

Algoritmo

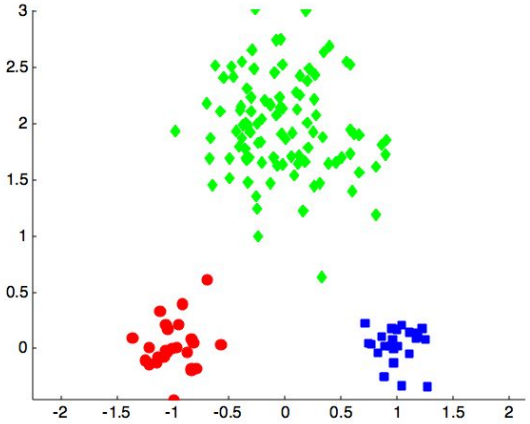
Algorithm 1 Basic K-means Algorithm.

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

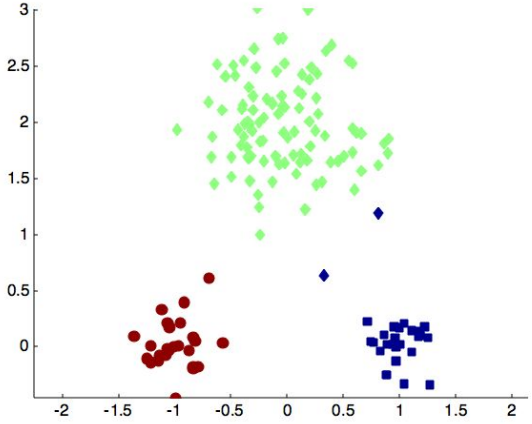
K-means

- Detalles del algoritmo
  - Centroides iniciales: aleatorios
    - Clusters varían dependiendo de la elección
  - “Cercanía” se mide con alguna distancia (generalmente usamos distancia euclidiana para variables númericas)
  - K-means converge para distancias “usuales”
  - En general la convergencia sucede con pocas iteraciones
    - Iterar hasta que cambien “pocos” puntos de cluster
  - Complejidad es O(n * K * I * d)
    - n puntos, K centros, I iteraciones, d dimensiones

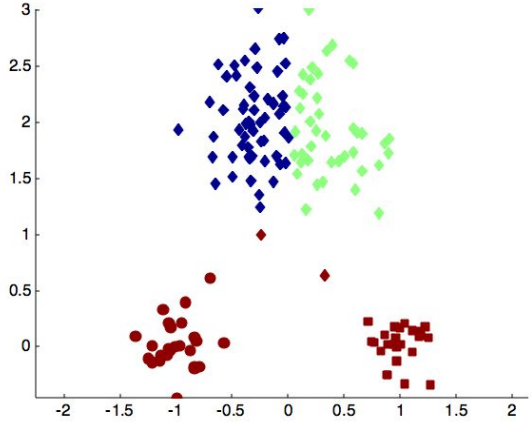
K-means no asegura encontrar los clusters óptimos



Puntos originales



Clustering óptimo



Clustering sub-optimal

K-means: escogiendo los centroides iniciales

- Escoger los centroides iniciales es una pieza clave en K-means.
- Enfoque tradicional: inicializar los centroides aleatoriamente.
- Cuando los centroides iniciales son escogidos aleatoriamente, diferentes ejecuciones de k-means producen distintos valores de SSE.
- **Solución simple**: correr K-means varias veces variando la semilla aleatoria y quedarse con el modelo de menor SSE.
  ¡Esto último no garantiza que encontremos los clusters óptimos!

Repaso SSE

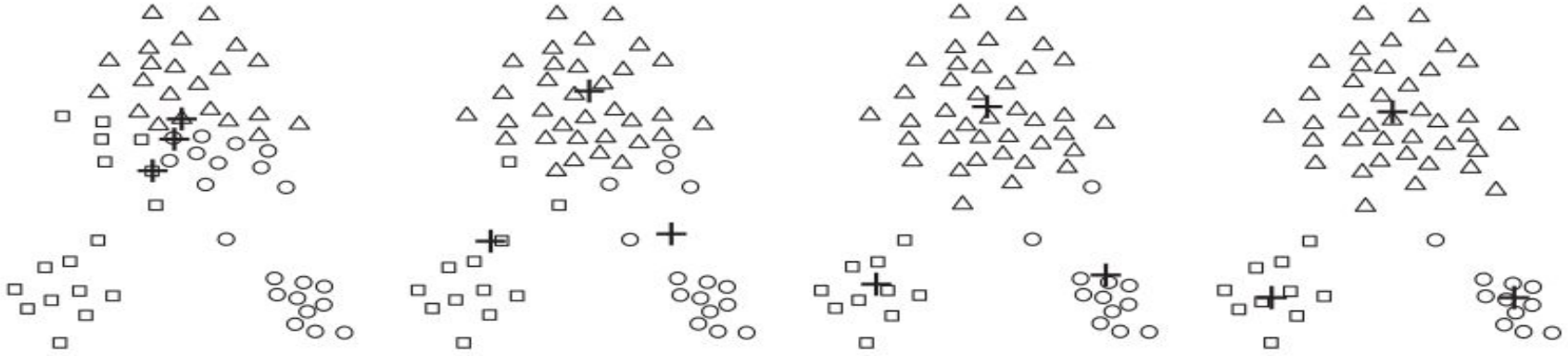
- SSE: Suma de las distancias cuadradas de cada punto al centroide de su cluster asignado.

$$\text{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

- Propiedad Interesante:
  - SSE permite calcular cual es el aporte individual de cada cluster al SSE total.
  - Eso nos permite juzgar si un cluster es bueno o no.

Comparando distintas inicializaciones

Caso1:

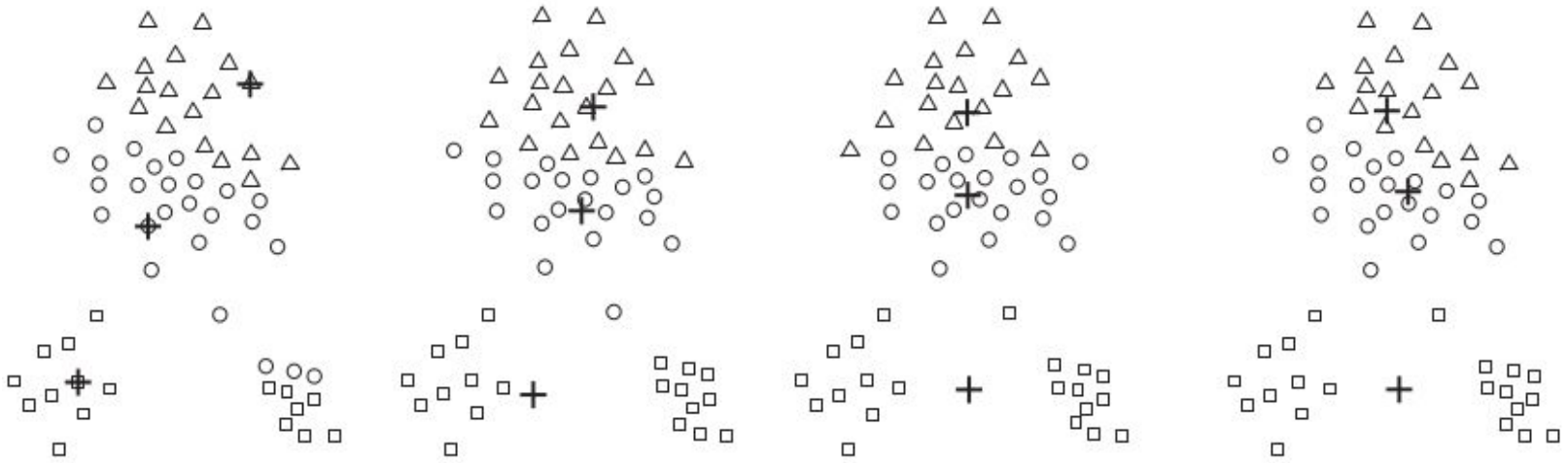


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Aunque los centroides iniciales están todos en único cluster, igual se converge a los clusters deseados.

Comparando distintas inicializaciones

Caso 2:

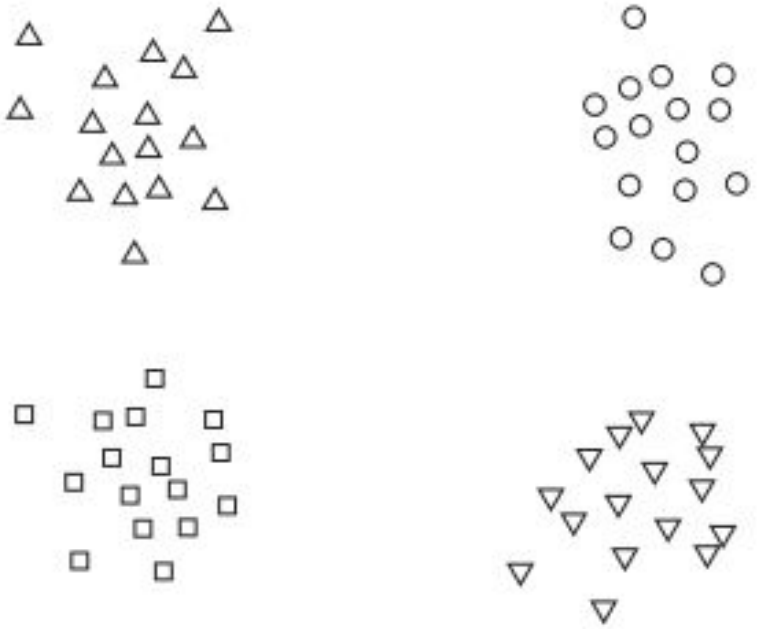


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

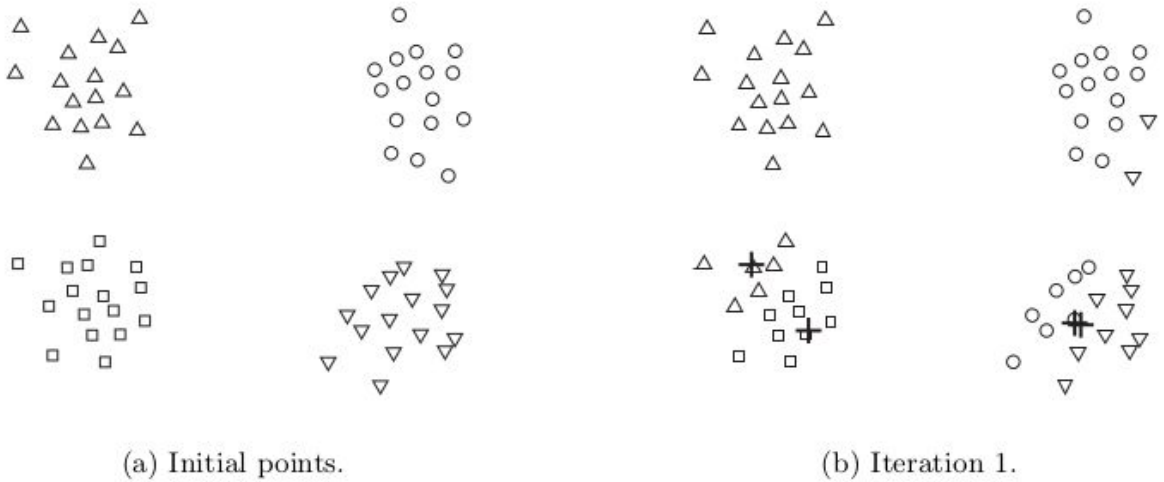
Aunque los centroides iniciales parecen estar mejor repartidos, llegamos a una solución peor que en el caso anterior.

Otro ejemplo

- Tenemos datos donde hay dos pares de clusters (el par izquierdo y el par derecho).
- Los clusters de cada par están más cerca entre sí que de los clusters del otro par.

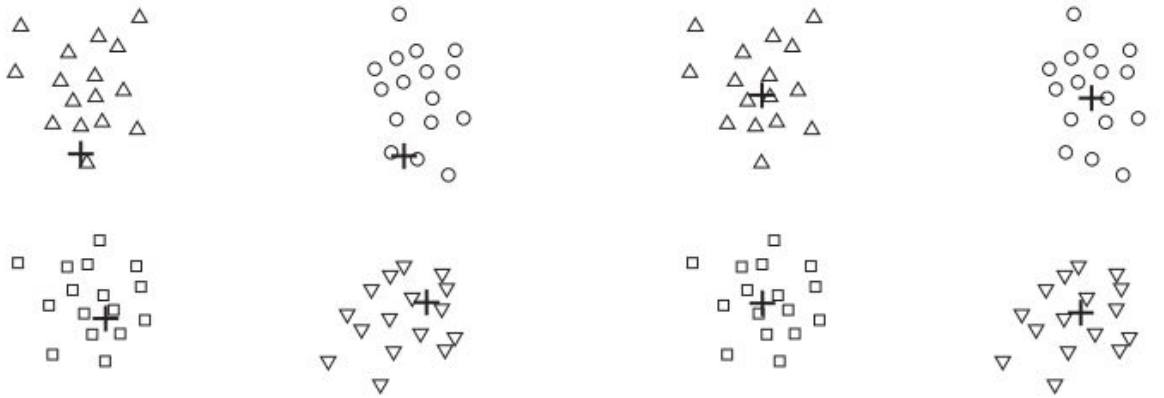


Caso 1



(a) Initial points.

(b) Iteration 1.

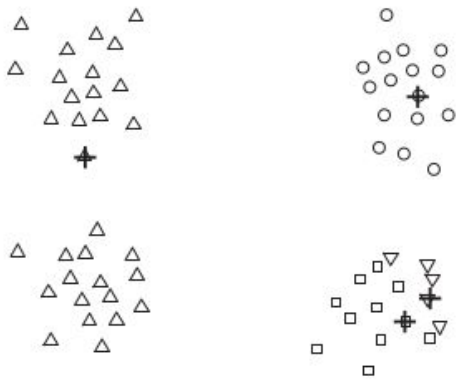


(c) Iteration 2.

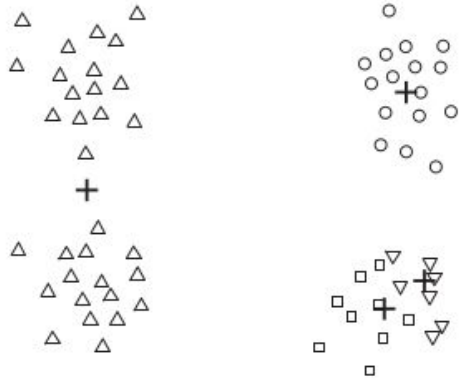
(d) Iteration 3.

Si empezamos con dos centroides iniciales en cada par, incluso si los pares de centroides están en el mismo cluster, los centroides se redistribuyen para encontrar los clusters reales.

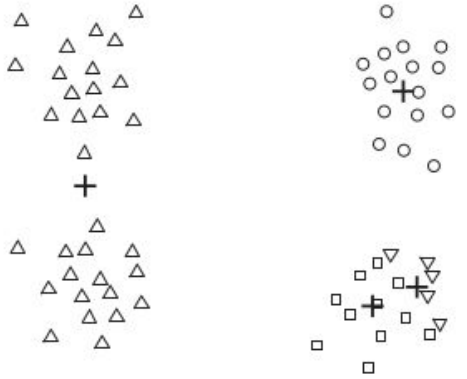
Caso 2



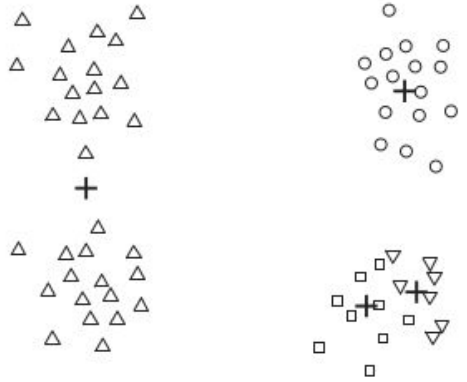
(a) Iteration 1.



(b) Iteration 2.



(c) Iteration 3.



(d) Iteration 4.

Por otro lado, si un par de clusters recibe sólo un centroide inicial y el otro par recibe 3, entonces 2 de los clusters reales se combinarán y un cluster será dividido.

K-means

- Idealmente nos gustaría partir con un centroide por cluster “real”.
- Si hay K clusters “reales”, probabilidad de escoger un centroide por cluster es baja

$$P = \frac{\text{\# formas de escoger un centroide de cada cluster}}{\text{\# formas de escoger K centroides}} = \frac{K! n^K}{(Kn)^K} = \frac{K!}{K^K}$$

Ej.: si K = 10, entonces $P = 10!/10^{10} = 0.00036$

K-means: Manejando clusters vacíos

- Algoritmo K-means puede retornar clusters vacíos si no se le asignan puntos al cluster en el paso de asignación.

- Estrategias para encontrar centroide de reemplazo:
  - Escoger el punto más lejano a todos los centroides como nuevo centroide (punto que contribuye más al SSE)

  - Escoger un punto aleatorio del cluster con mayor SSE.
    - Esto generalmente dividirá ese cluster y reducirá el SSE total.

K-means: Preprocesamiento

- Normalizar los datos: que todos los atributos aporten lo mismo a las distancias.
- Eliminar outliers: outliers producen centroides no representativos con alto SSE.

K-means: Postprocesamiento

- Aumentar el número de clusters es la solución trivial para bajar el valor del SSE.
- Eso no es lo queremos => tener K igual al número de datos nos daría un SSE de cero.

Bajar el SSE de forma más inteligente:

- Eliminar clusters pequeños que puedan representar outliers
- Dividir clusters “sueltos” (con alto SSE)
- Mezclar clusters cercanos y con bajo SSE.

Bisecting K-means

- Extensión simple de K-means
- Idea: Dividir el conjunto de todos los puntos en dos clusters, escoger uno de los dos para ser dividido, e iterar hasta producir K clusters.
- Cada división se obtiene ejecutando K-means (con k=2)

Algorithm 7.3 Bisecting K-means algorithm.

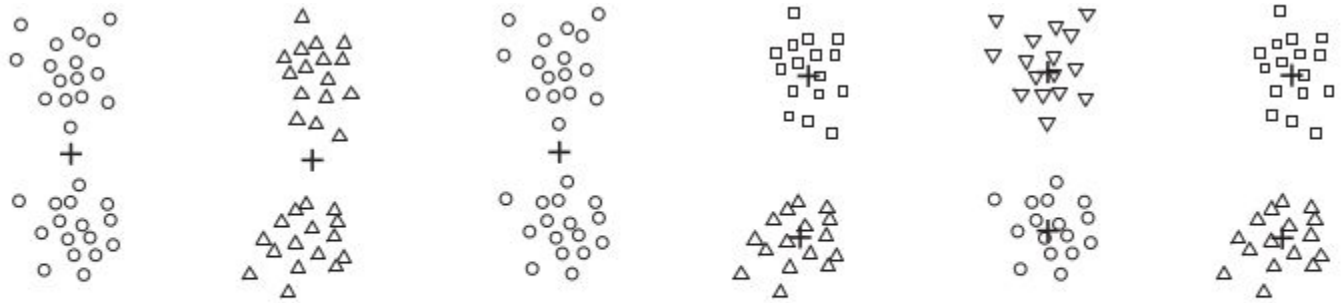
```
1: Initialize the list of clusters to contain the cluster consisting of all points.
2: repeat
3:    Remove a cluster from the list of clusters.
4:    {Perform several "trial" bisections of the chosen cluster.}
5:    for i = 1 to number of trials do
6:       Bisect the selected cluster using basic K-means.
7:    end for
8:    Select the two clusters from the bisection with the lowest total SSE.
9:    Add these two clusters to the list of clusters.
10: until The list of clusters contains K clusters.
```

Bisecting K-means

¿Cómo escojo los clusters a dividir (Paso 3 del algoritmo)?

- Opción 1: Escoger el cluster más grande en cada paso.
- Opción 2: Escoger el cluster con mayor SSE.
- Opción 3: Estrategia híbrida entre las dos anteriores

Bisecting K-means con datos del ejemplo anterior



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Iteración 1: se encuentran dos pares de clusters.

Iteración 2: El par de clusters de la derecha es dividido.

Iteración 3: El par de clusters de la izquierda es dividido.

Bisecting K-means

Bisecting K-means tiene menos problemas de inicialización que K-means.

- Esto es porque realiza varios intentos de bisección y toma la bisección de menor SSE.
- Además sólo se consideran dos centroides en cada paso.

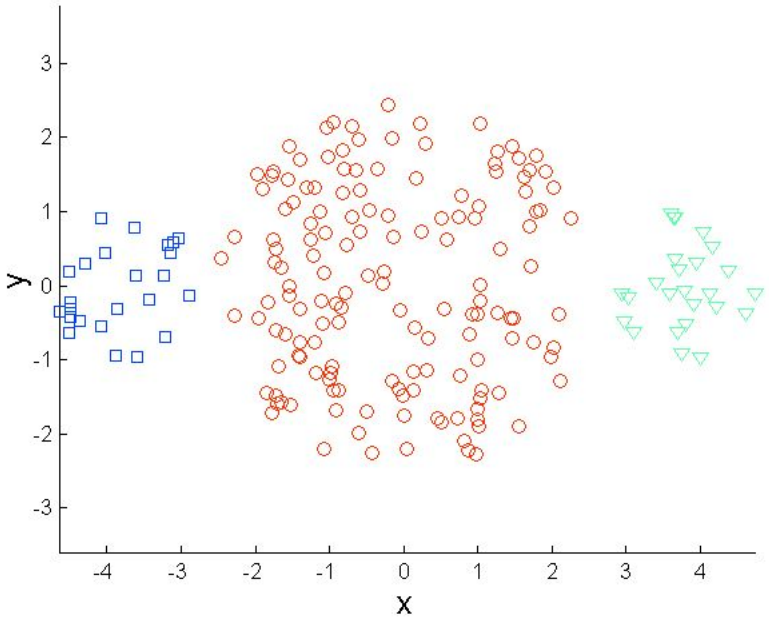
Si registramos la secuencia de clusters bisectados podemos producir un clustering jerárquico.

K-means

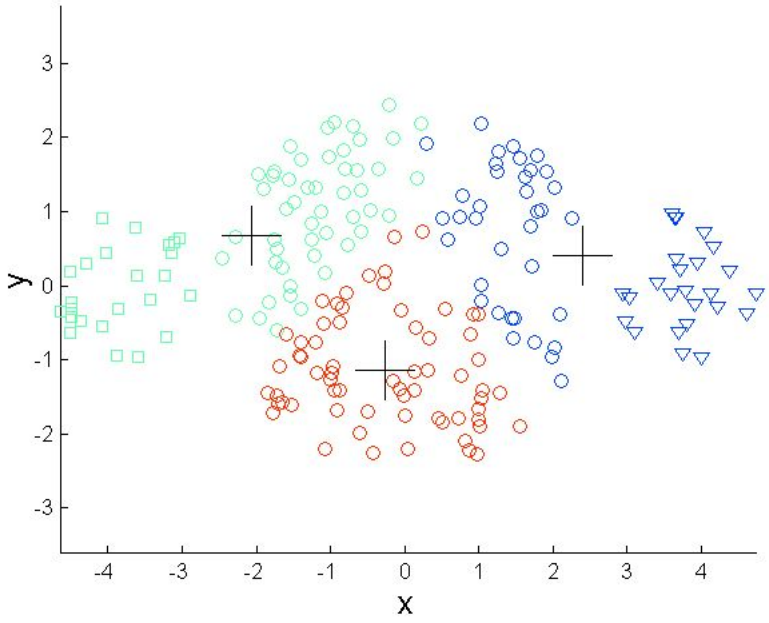
- Limitaciones de K-means
  - Clusters de diferente tamaño
  - Clusters de diferentes densidades
  - Clusters con formas no esféricas
- K-means no es robusto a outliers

K-means

- Ejemplo: tamaños diferentes



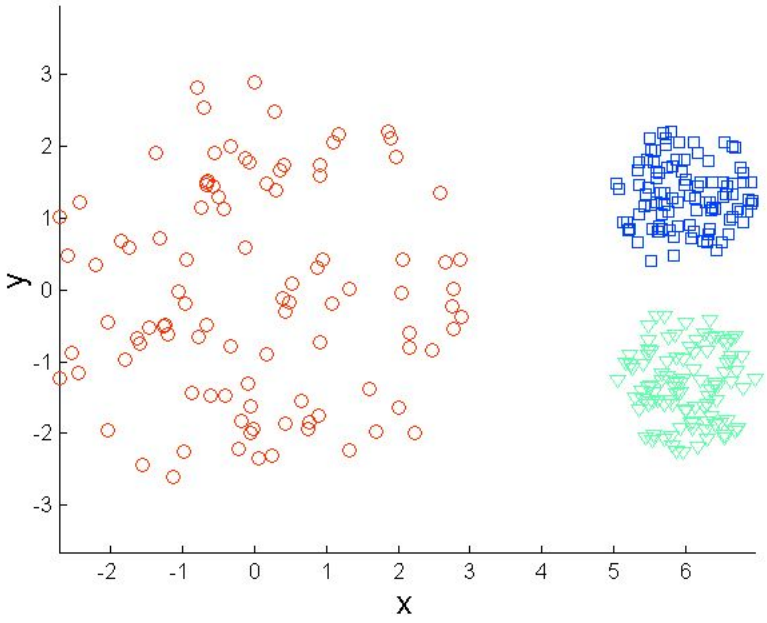
Puntos originales



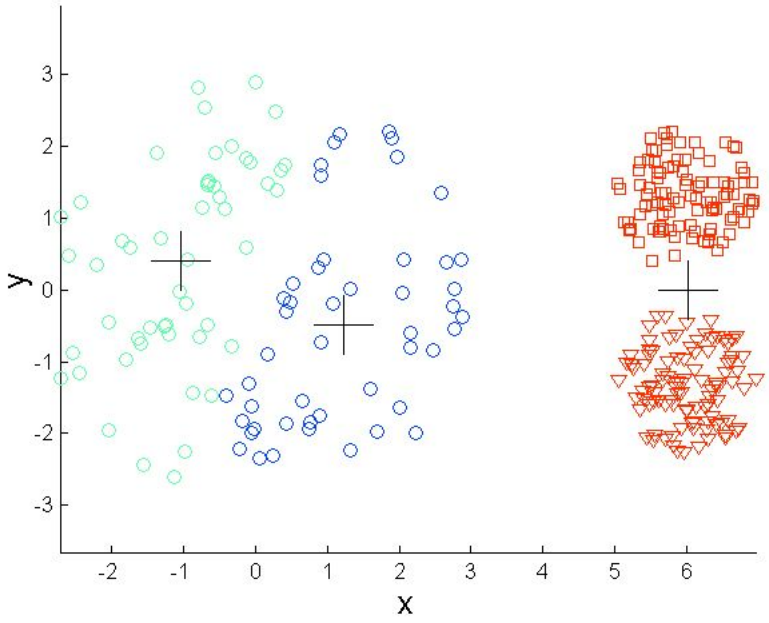
K-means (tres clusters)

K-means

- Ejemplo: densidades diferentes



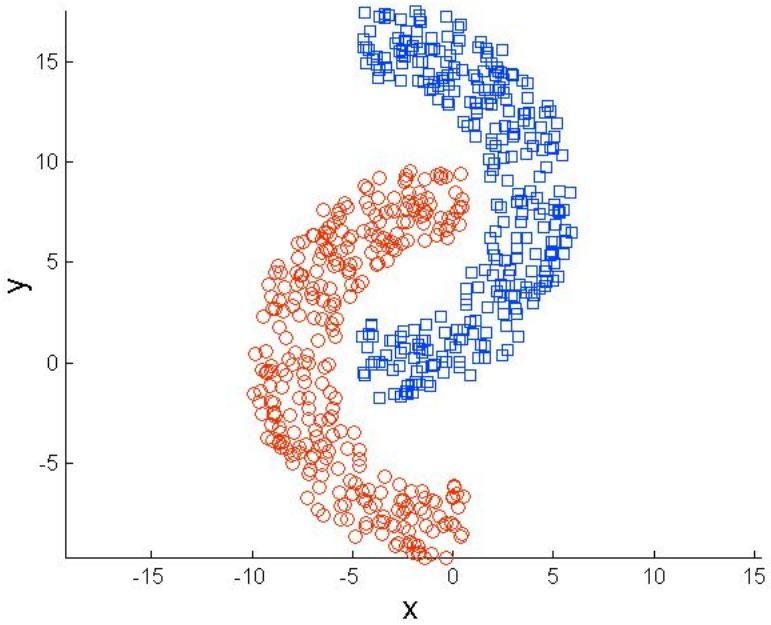
Puntos originales



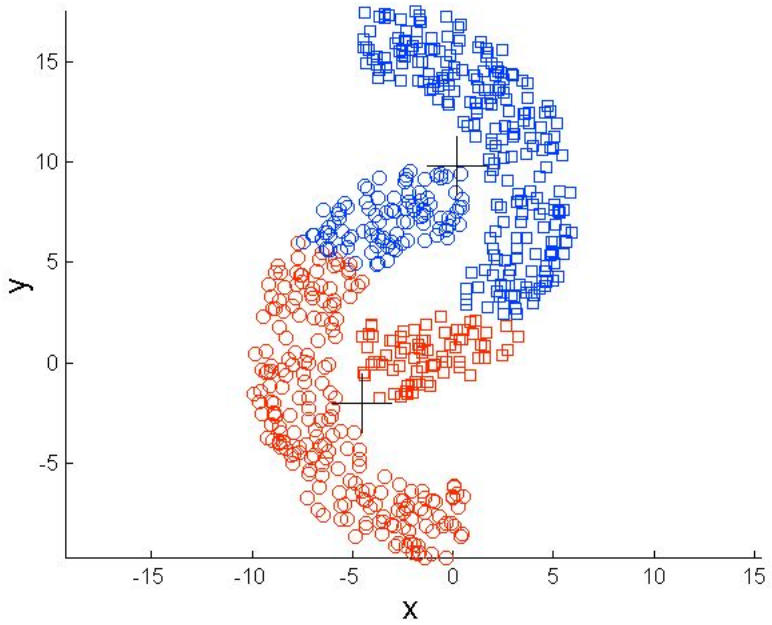
K-means (tres clusters)

K-means

- Ejemplo: formas no esféricas



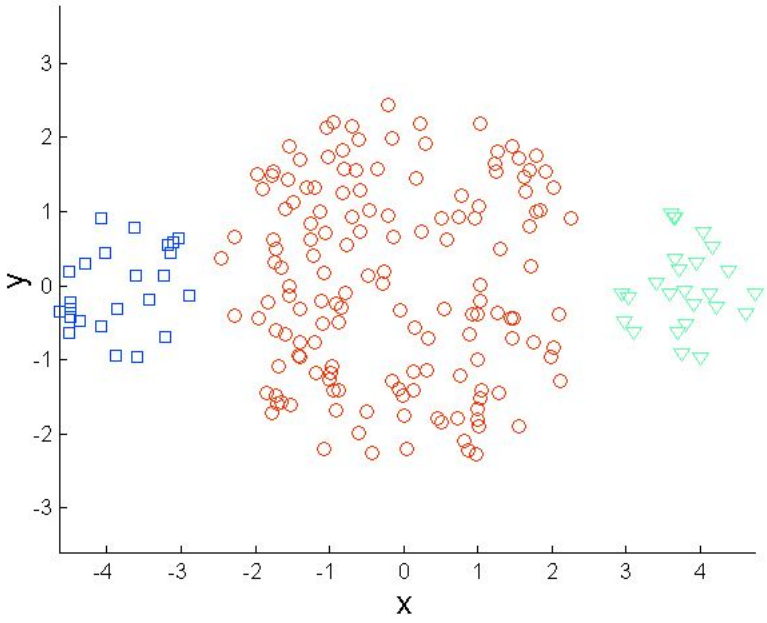
Puntos originales



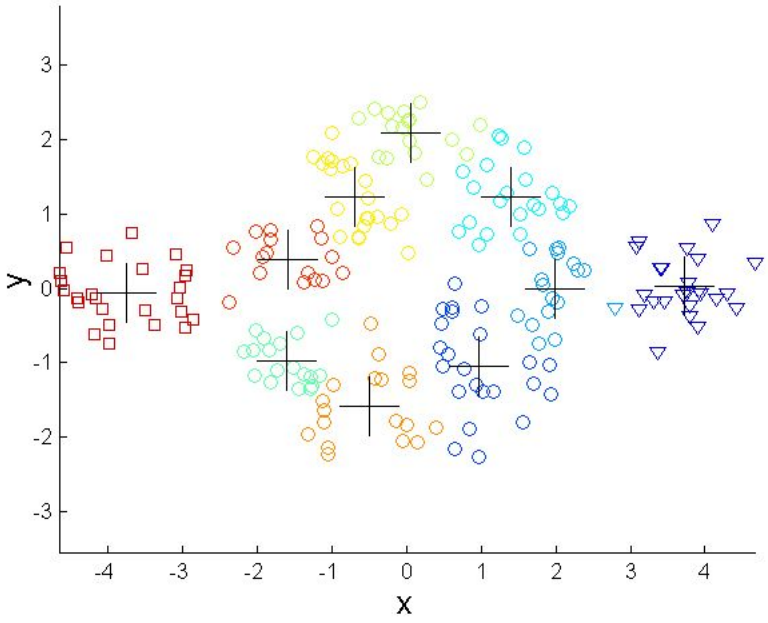
K-means (dos clusters)

K-means

- Solución: usar K alto, luego mezclar clusters



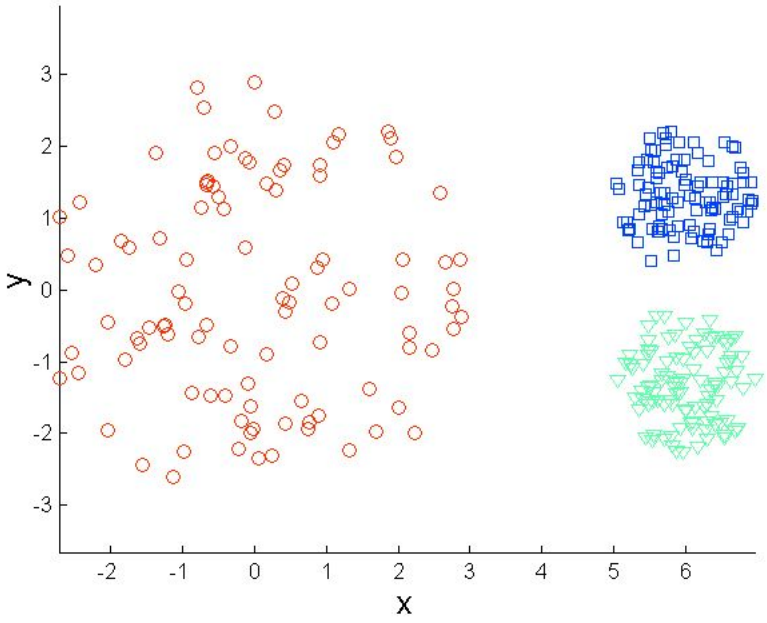
Puntos originales



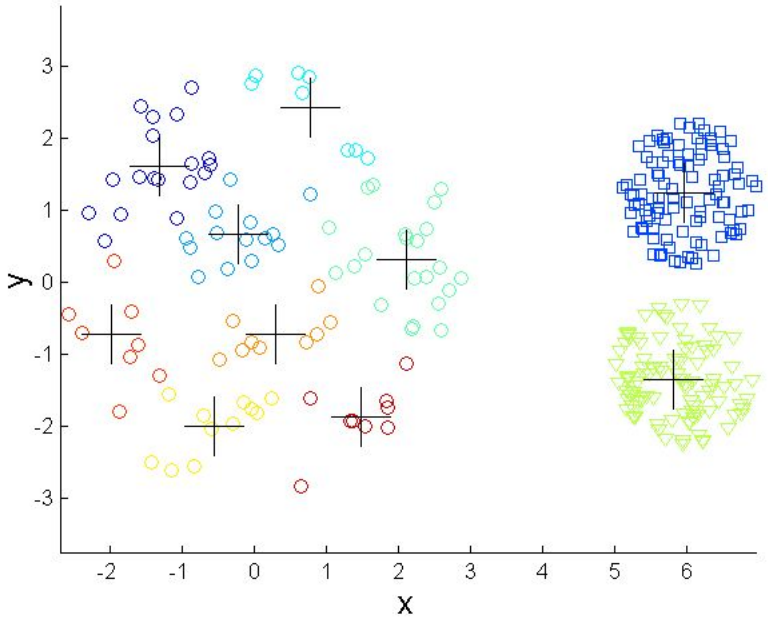
K-means clusters

K-means

- Solución: usar K alto, luego mezclar clusters



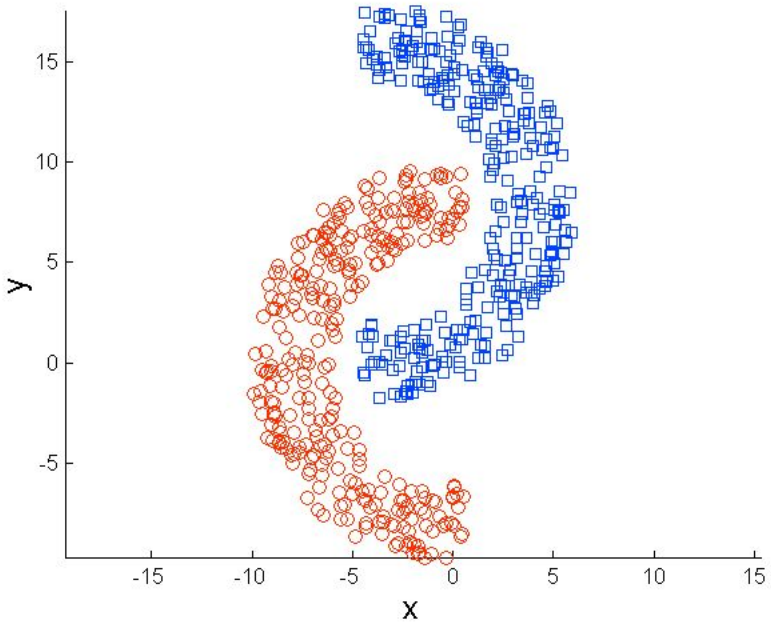
Puntos originales



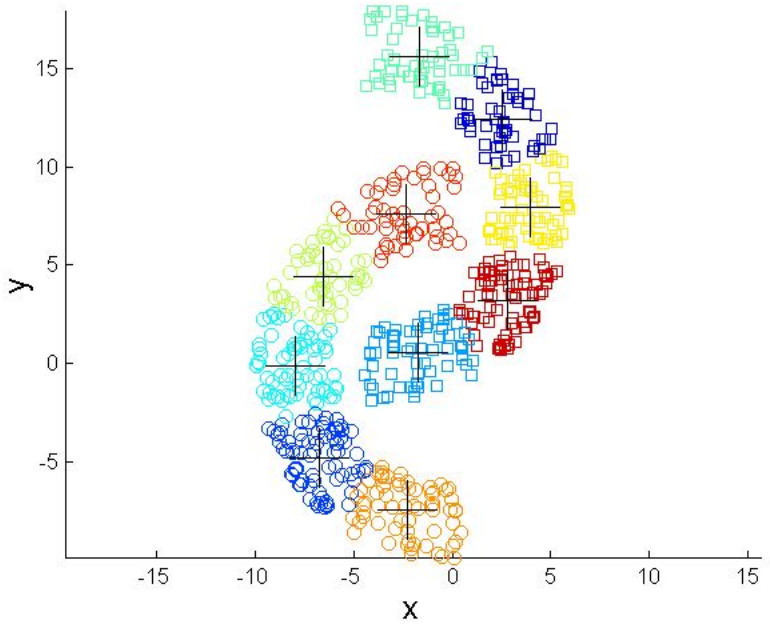
K-means clusters

K-means

- Solución: usar K alto, luego mezclar clusters



Puntos originales



K-means clusters

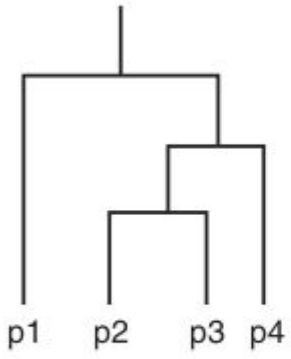
Clustering Jerárquico Aglomerativo

Produce un conjunto de clusters anidados organizados en un árbol jerárquico. Es una técnica antigua.

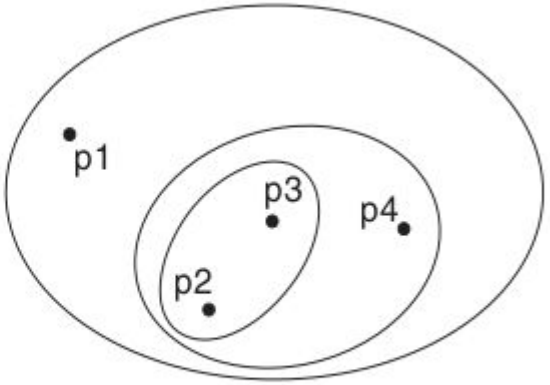
Visualizaciones:

a) **Dendograma**: árbol que muestra las relaciones cluster-subcluster y el orden en que los clusters fueron mezclados o divididos.

b) **Diagrama de clusters anidados**: sólo para puntos 2-dimensionales.



(a) Dendrogram.



(b) Nested cluster diagram.

Clustering Jerárquico Aglomerativo

- Fortalezas
  - No tiene que suponer un número a priori de clusters
    - Se puede obtener cualquier número de clusters deseado "cortando" el dendograma en el nivel apropiado
  - Clusters pueden corresponder a taxonomía
    - Ejemplos en biología.

Clustering Jerárquico Aglomerativo

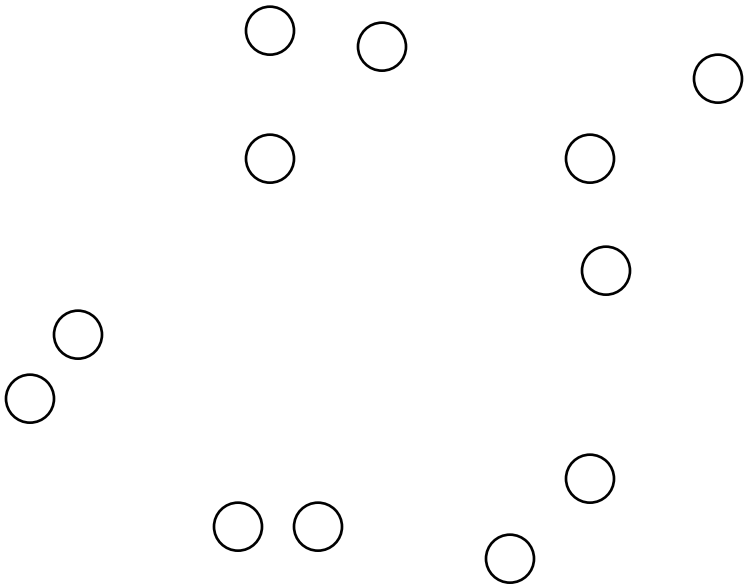
- Tipos principales de clustering jerárquico
  - Aglomerativo
    - Empezar con cada punto como cluster individual
    - En cada paso, mezclar el par de clusters más cercano hasta que quede sólo un cluster (o *k* clusters)
  - Divisivo
    - Empezar con un cluster que contenga todos los puntos
    - En cada paso, dividir un cluster en dos hasta que todo cluster contenga un solo punto (o haya *k* clusters)
- Requieren una definición de proximidad entre clusters.

Clustering Jerárquico Aglomerativo

- Algoritmo básico (aglomerativo)
  1. Calcular matriz de distancias
  2. Sea cada punto un cluster
  3. **Repetir**
  4. Mezclar par de clusters más cercano
  5. Actualizar matriz de distancias
  6. **Hasta** que quede sólo un cluster

Clustering Jerárquico Aglomerativo

- Situación inicial: empezar con clusters de puntos individuales y la matriz de distancias

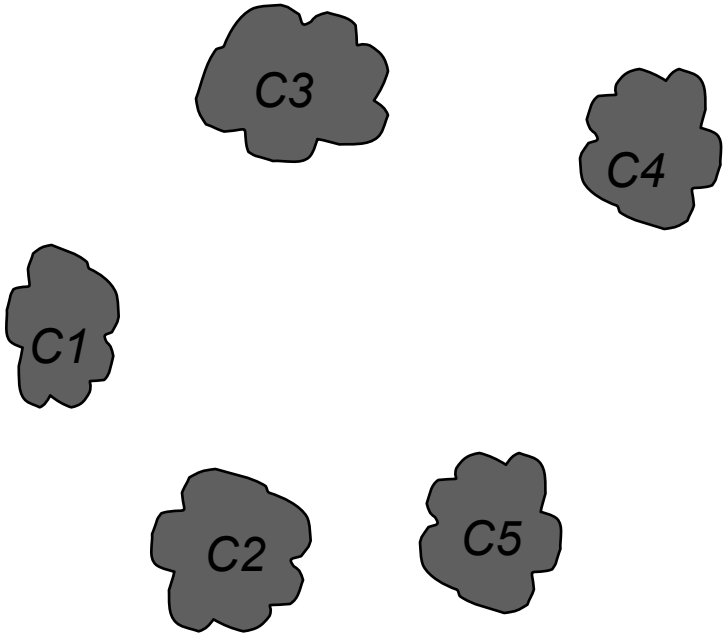


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Matriz de distancias

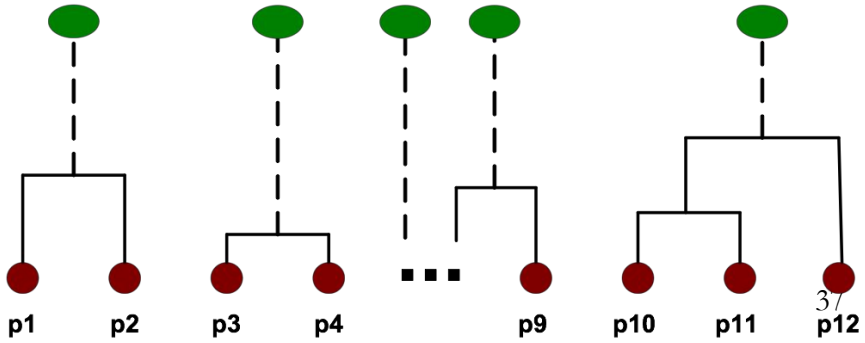
Clustering Jerárquico Aglomerativo

- Después de un par de iteraciones…



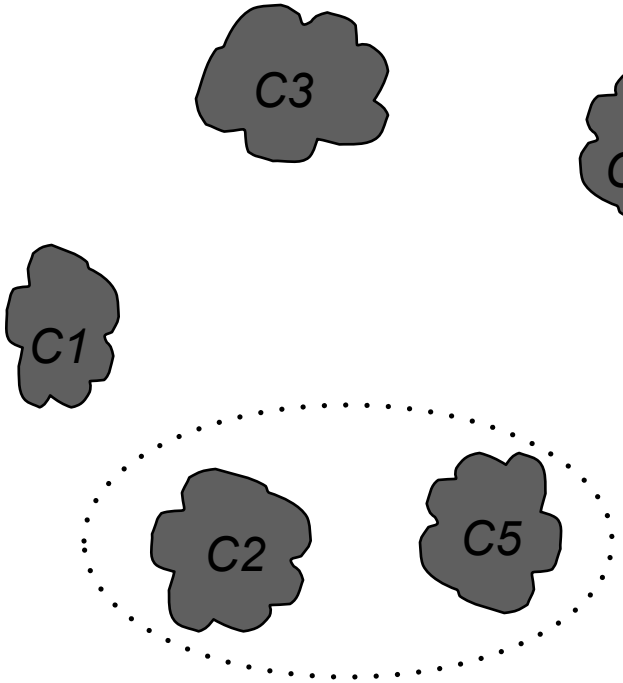
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Matriz de distancias



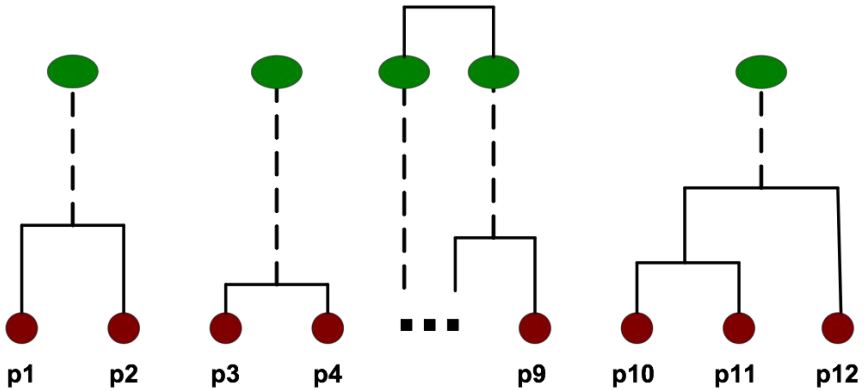
Clustering Jerárquico Aglomerativo

- … mezclar clusters más cercano (C2 y C5) y actualizar matriz de distancias



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Matriz de distancias

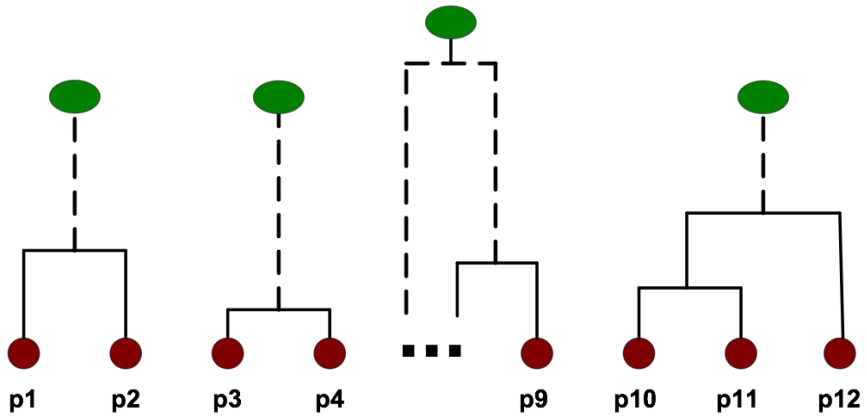


Clustering Jerárquico Aglomerativo

- ¿Cómo actualizar matriz de distancias?

	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Matriz de distancias



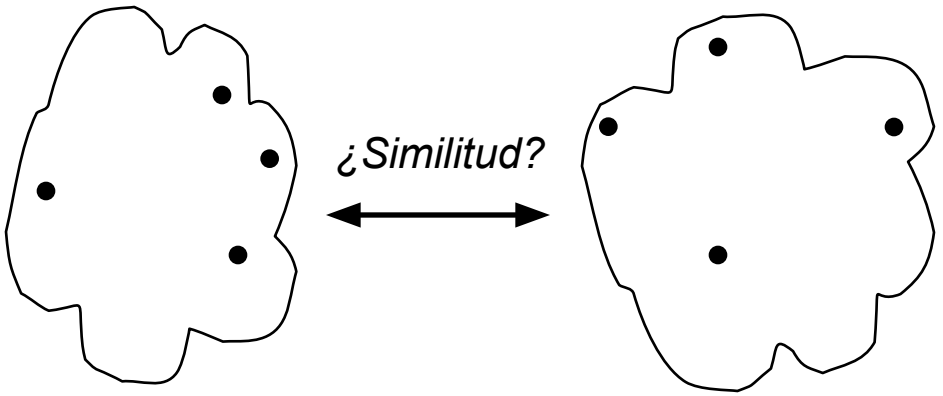
Clustering Jerárquico Aglomerativo

- Operación clave: cálculo de la distancia entre clusters
  - Diferentes formas de hacerlo distinguen a los diferentes algoritmos

- Intuición: Sabemos como calcular la distancia entre dos puntos, pero ¿cómo calculamos la distancia entre dos clusters? o ¿entre un punto y un cluster?

Clustering Jerárquico Aglomerativo

- ¿Cómo definir distancias entre clusters?



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

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

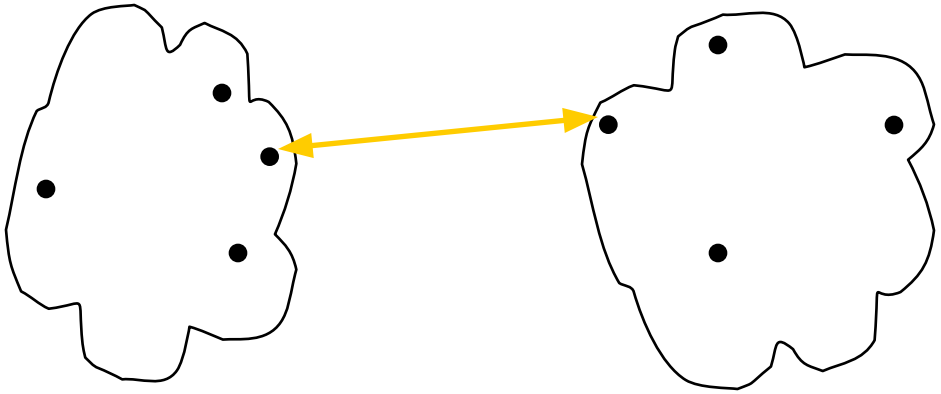
.

.

Matriz de distancias

Clustering Jerárquico Aglomerativo

- ¿Cómo definir distancias entre clusters?



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

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

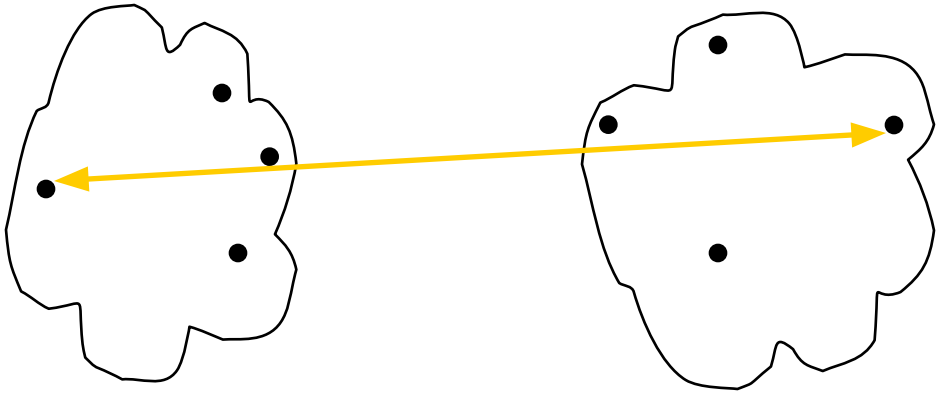
.

.

Matriz de distancias

Clustering Jerárquico Aglomerativo

- ¿Cómo definir distancias entre clusters?



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

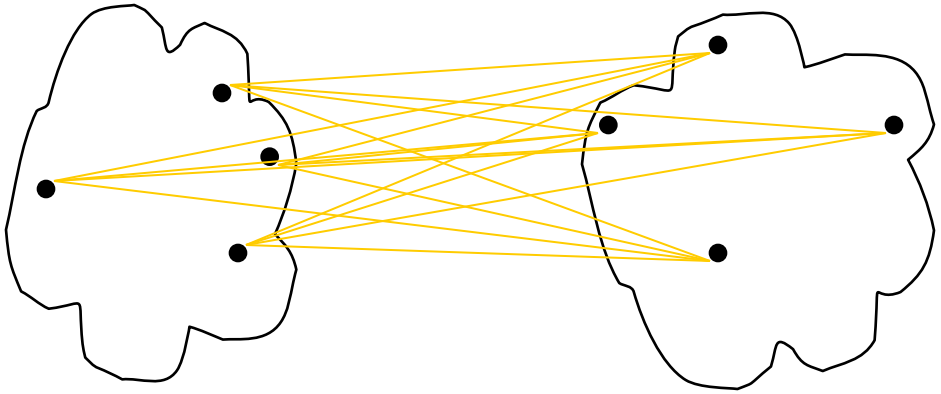
. .

. *Matriz de distancias*

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

Clustering Jerárquico Aglomerativo

- ¿Cómo definir distancias entre clusters?



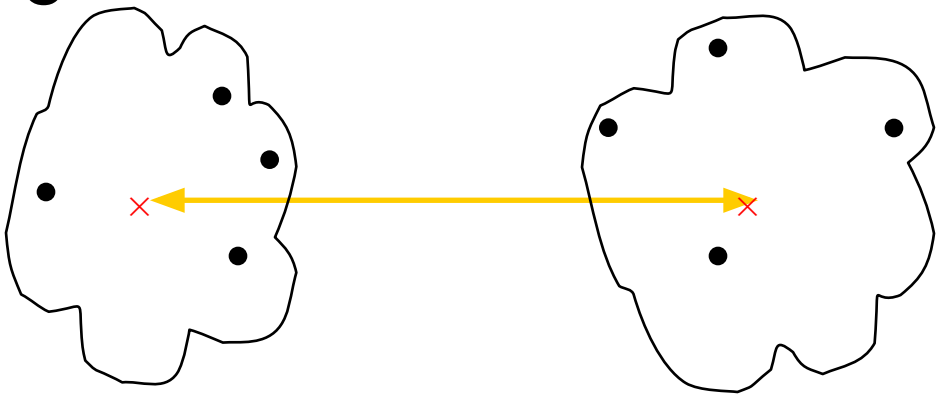
- Promedio del grupo
  - Distancia promedio de todos los pares de puntos (cada par tiene un punto por cluster)

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

Matriz de distancias

Clustering Jerárquico Aglomerativo

- ¿Cómo definir distancias entre clusters?



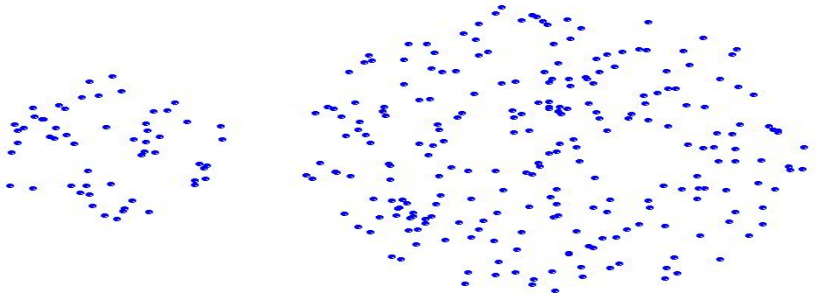
- Distancia entre centroides
  - distancia entre los centroides de cada grupo

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

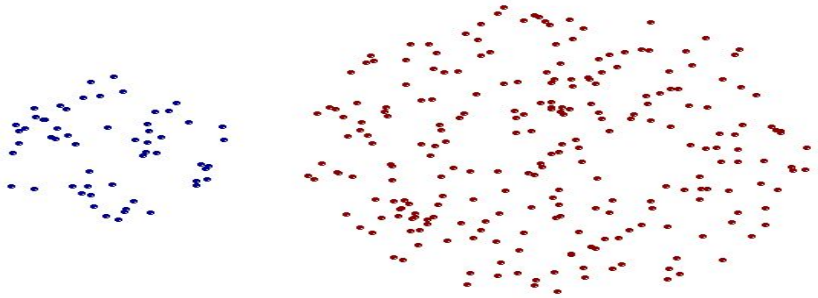
Matriz de distancias

Clustering Jerárquico Aglomerativo

- Fortaleza de distancia MIN



Puntos originales

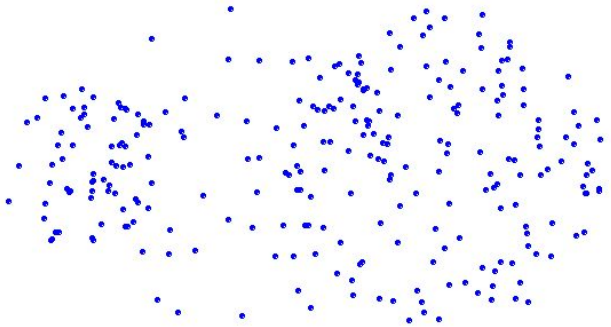


Dos Clusters

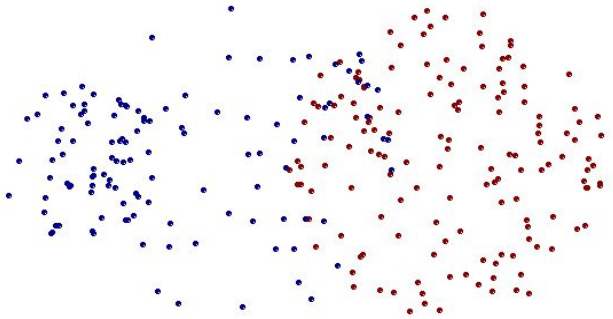
- *Puede manejar formas no-elípticas*

Clustering Jerárquico Aglomerativo

- Limitaciones de distancia MIN



Puntos originales

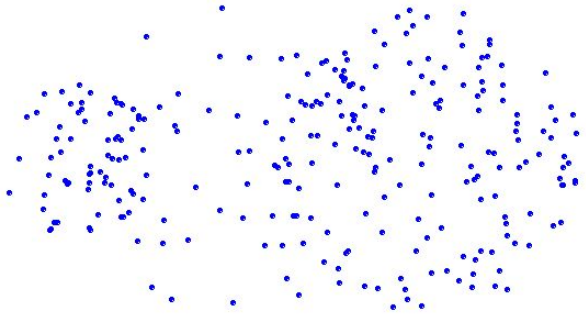


Dos Clusters

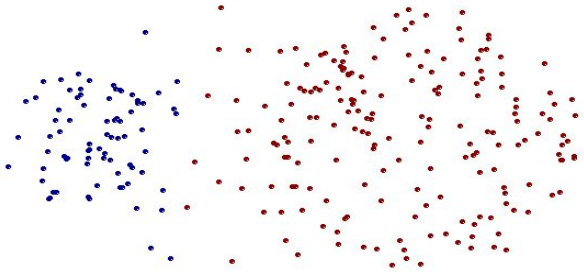
- *Sensible a ruido y outliers*

Clustering Jerárquico Aglomerativo

- Fortaleza de MAX



Puntos originales

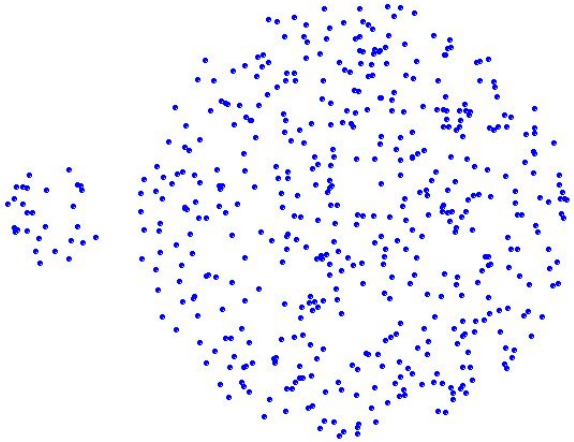


Dos Clusters

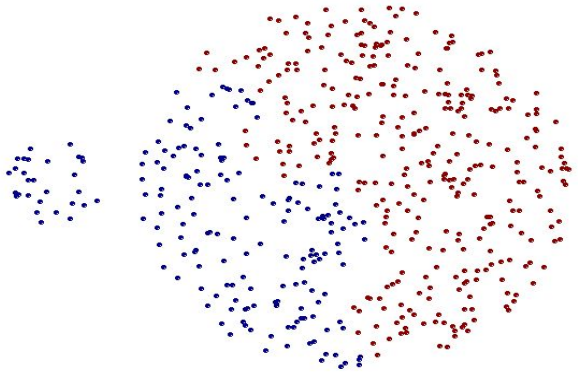
- *Menos susceptible a ruido y outliers*

Clustering Jerárquico Aglomerativo

- Limitaciones de MAX



Puntos originales



Dos Clusters

- *Tiende a quebrar clusters grandes*
- *Sesgado a clusters esféricos*

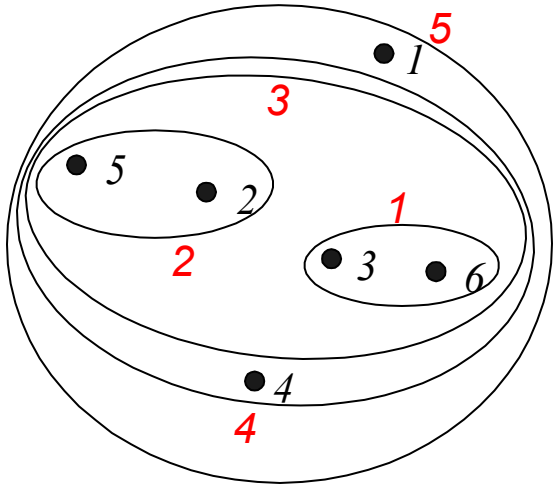
Clustering Jerárquico Aglomerativo

- Distancia promedio de grupo
  - Compromiso entre MIN y MAX
  - Fortalezas
    - Menos susceptible a ruido y outliers
  - Limitaciones
    - Sesgado a clusters esféricos

Clustering Jerárquico Aglomerativo

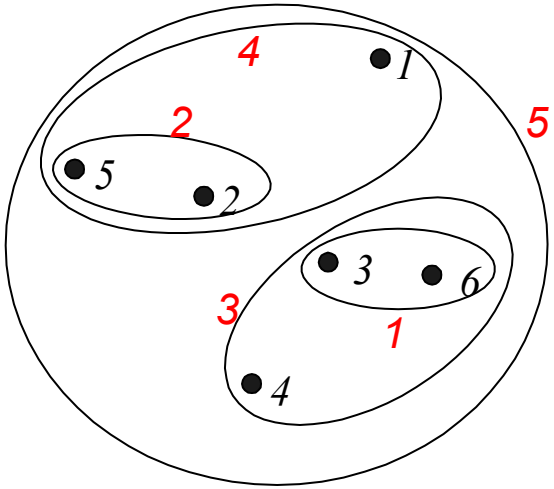
- Método de Ward
  - Similitud entre clusters se basa en el incremento del SSE cuando se mezclan dos clusters
    - Similar a distancia promedio de grupo si la distancia entre puntos es distancia cuadrada
  - Menos susceptible a ruido y outliers
  - Sesgado a clusters esféricos

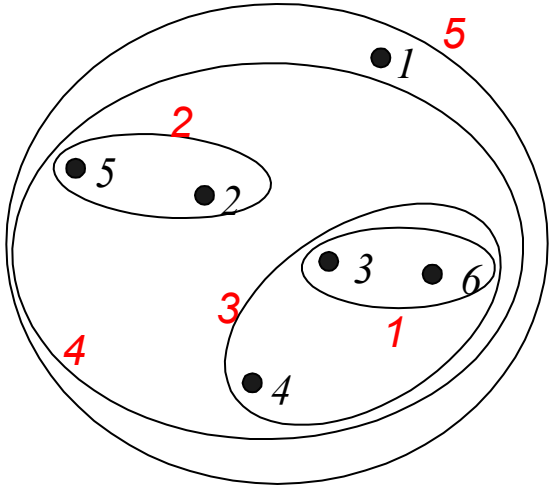
Clustering Jerárquico Aglomerativo



MIN

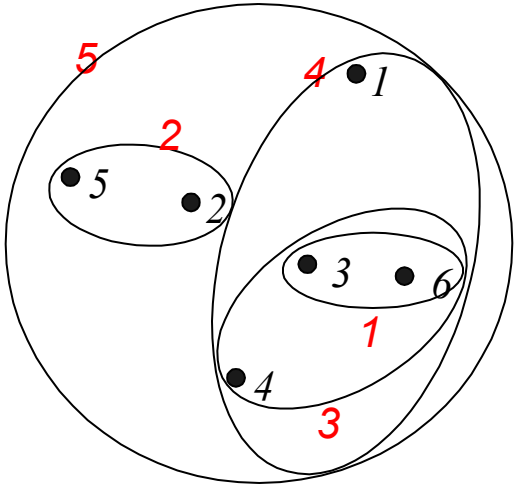
MAX





Promedio de grupo

Método de Ward



Clustering Jerárquico Aglomerativo

- Requerimientos de tiempo y espacio
  - Espacio: O(N^2) para guardar matriz de distancias
    - N: número de puntos
  - Tiempo: O(N^3) en muchos casos
    - Para N pasos, se debe actualizar matriz de similitud en cada paso
    - Complejidad puede reducirse a O(N^2 log N) usando listas ordenadas o heaps

Clustering Jerárquico Aglomerativo

- Problemas y limitaciones
  - Una vez decidido unir dos clusters, no se puede deshacer
  - No hay una función objetivo que sea directamente minimizada
  - Problemas de los diferentes esquemas:
    - Sensibles a ruido y outliers
    - Dificultad para manejar clusters de distinto tamaño
    - Pueden romper clusters grandes

DBSCAN

Algoritmo de clustering basado en densidad

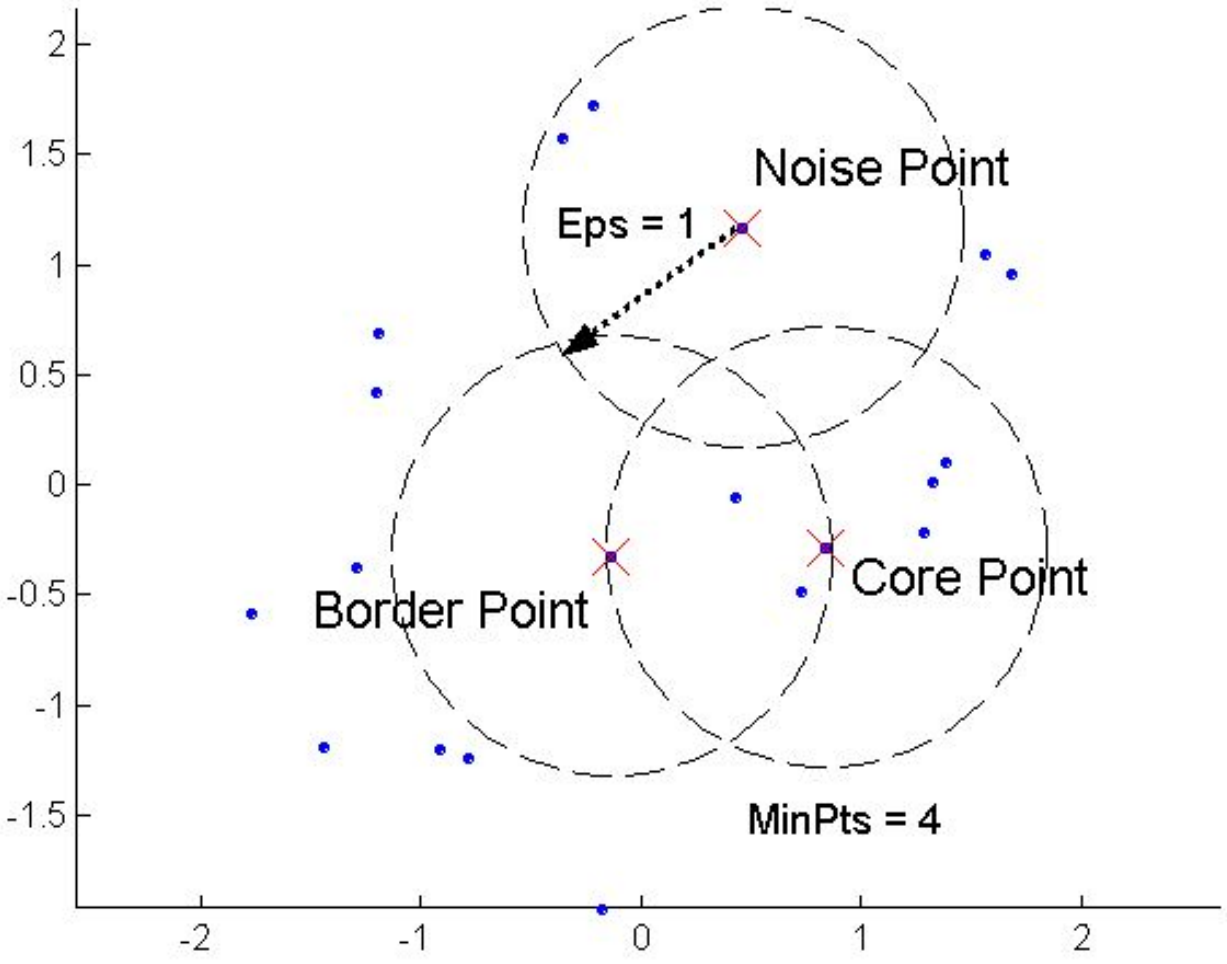
Idea: encontrar regiones de alta densidad de puntos separado por regiones de baja densidad.

- Las regiones densas corresponden a los clusters.
- La densidad de un punto es el número de puntos que tiene dentro de un radio dado.

DBSCAN

- Parámetros:
  - 1) **Eps**: radio especificado
  - 2) **MinPts**: número mínimo de puntos en una región.
- Tipos de punto:
  - Punto “**core**”: punto con más puntos que MinPts a distancia Eps
    - Éstos son los puntos dentro del cluster
  - Punto “border”: tiene menos que MinPts puntos en el radio Eps, pero está en la vecindad de un punto core.
  - Punto “noise”: cualquier punto que no sea core ni border.

DBSCAN


Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2

DBSCAN Algoritmo

- Cualquier par de puntos core que tengan una distancia entre sí menor que Eps son asignados al mismo cluster.

- Cualquier punto border que esté a una distancia menor que Eps de un punto core **pc** se le asigna el cluster de **pc.**
  - Hay que definir una estrategia cuando el punto border está cerca de dos puntos core de distinto cluster.

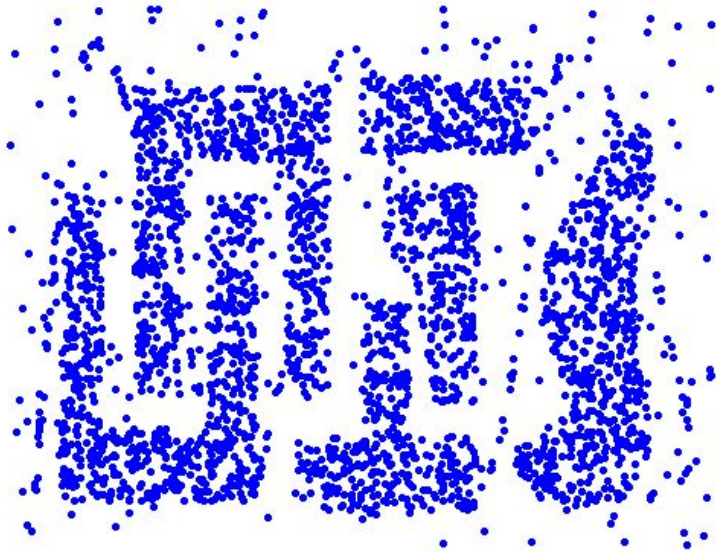
- Eliminar los puntos de ruido.

DBSCAN

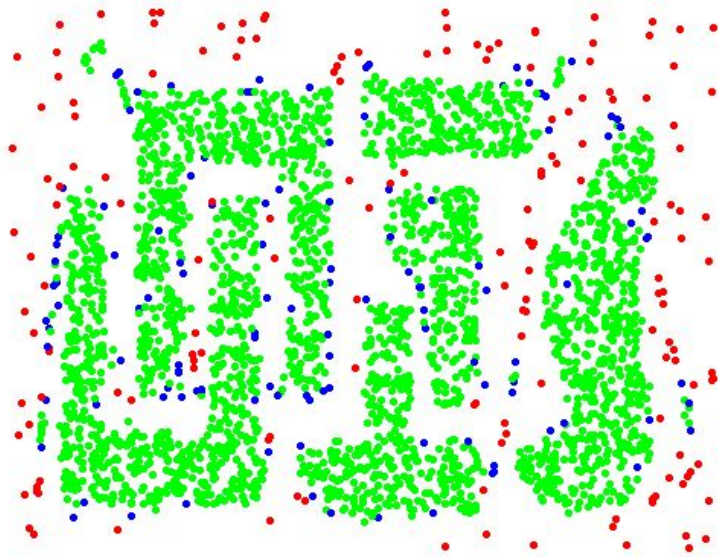
Algorithm 7.5 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points within a distance *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

DBSCAN



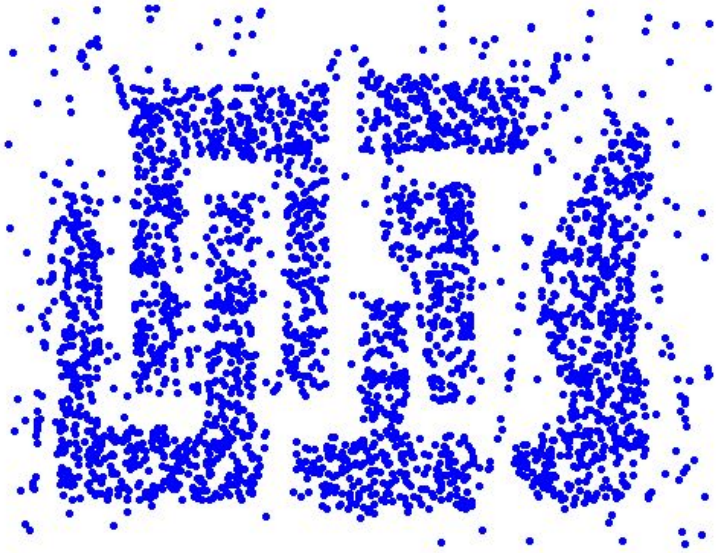
Puntos originales



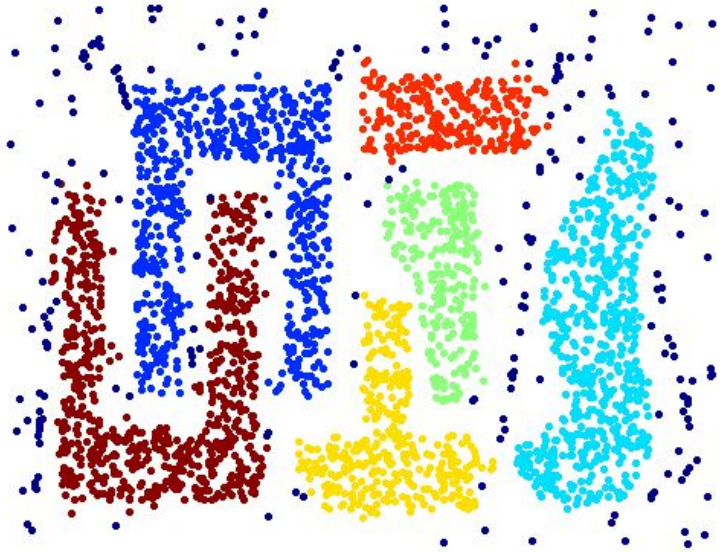
Tipos de punto: core, border y noise

Eps = 10, MinPts = 4

DBSCAN



Puntos originales

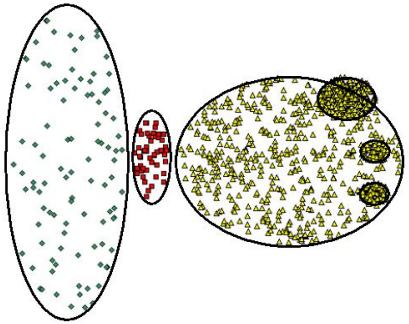


Clusters

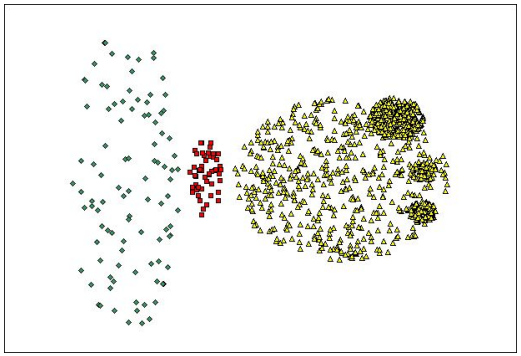
- *Resistente a ruido*
- *Puede encontrar clusters de diferentes formas y tamaños*

DBSCAN

- No funciona bien en:

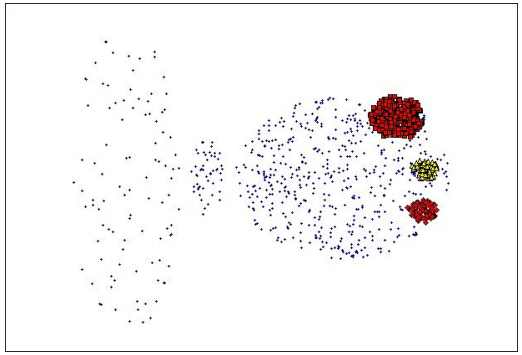


Puntos originales



(MinPts=4, Eps=9.92)

- *Densidades variables: clusters de baja densidad son confundidos con ruido.*
- *Datos de alta dimensionalidad: definición de densidad se vuelve compleja.*



(MinPts=4, Eps=9.75).

DBSCAN: Determinando Eps y MinPts

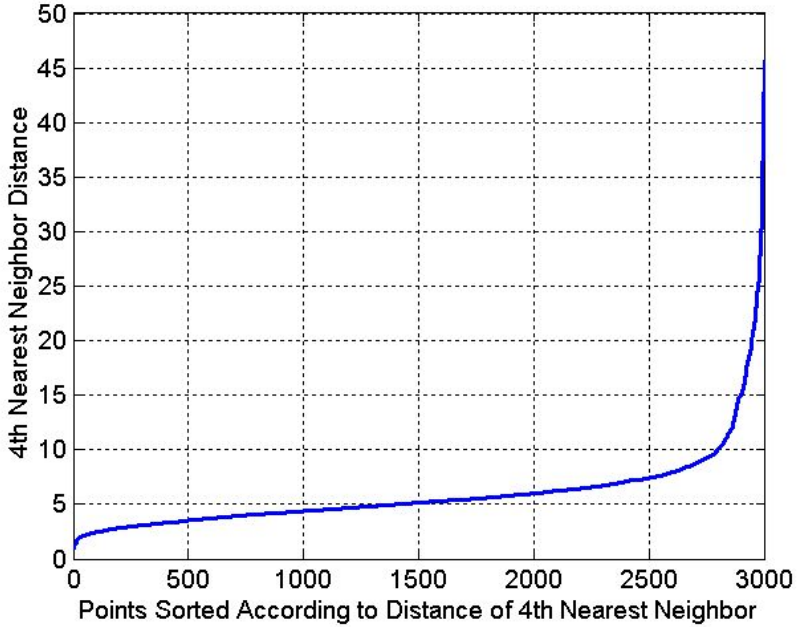
- Analizamos el comportamiento de la distancia de un punto a su k-ésimo vecino más cercano (**k-dist**).

- Para puntos que pertenecen a un cluster, el valor de k-dist será pequeño
  - siempre y cuando el valor de **k** no sea mayor que el tamaño del cluster al que pertenecen.

- Para puntos que no pertenecen a un cluster (como los noise points), el valor de k-dist será alto.

DBSCAN

- Calculamos el valor de k-dist para todos los puntos con un valor de k fijo y los ordenamos de manera creciente.
- Graficamos k-dist vs la cantidad de puntos con ese valor.



Eps seleccionado: valor de k-dist cuando ocurre el salto
MinPts: el valor de k.

Otras técnicas

- Mapas auto-organizados (SOM) o redes de Kohonen
- Fuzzy C-means
- Mezcla de Gaussianas y algoritmo EM

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE