DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Clustering basado en particiones

© Fernando Berzal, berzal@acm.org

Clustering basado en particiones

- k-Means
- k-Medoids
  - PAM [Partitioning Around Medoids]
  - CLARA [Clustering for LARge Applications]
  - CLARANS [CLAra based on RANdomized Search]
- BFR [Bradley, Fayyad & Reina]

Métodos de agrupamiento

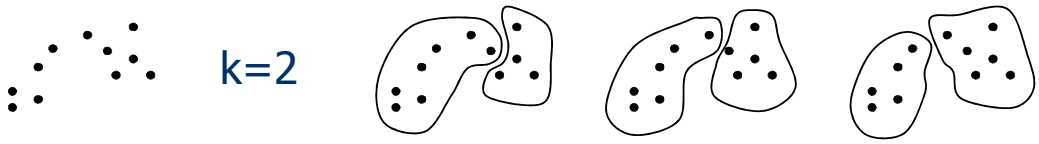
Familias de algoritmos de clustering:

- **Agrupamiento por particiones**
  k-Means, PAM/CLARA/CLARANS, BFR
- **Métodos basados en densidad**
  DBSCAN, Optics, DenClue
- **Clustering jerárquico**
  Diana/Agnes, BIRCH, CURE, Chameleon, ROCK

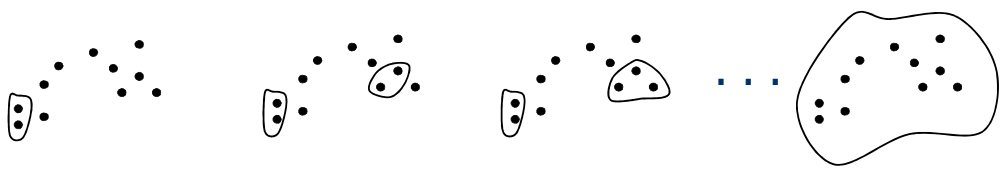
...

Métodos de agrupamiento

Clustering por particiones (suele fijarse k)



Clustering jerárquico (no se fija k)

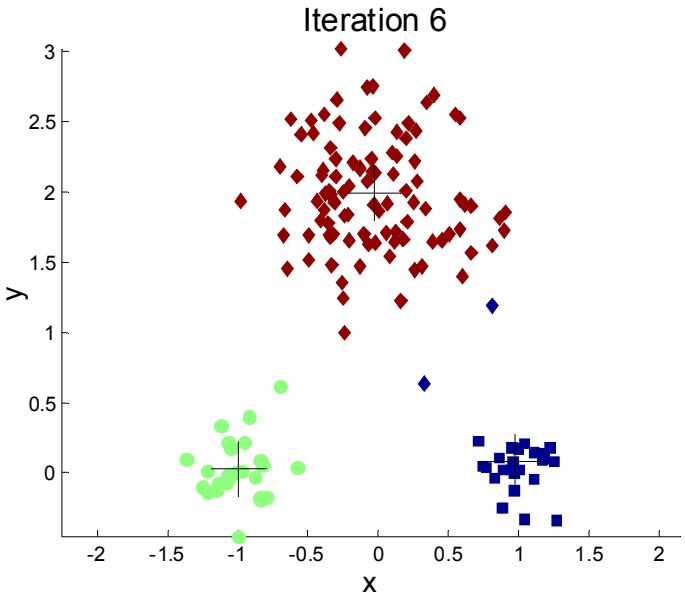


Se obtiene como resultado final
un conjunto de agrupamientos.

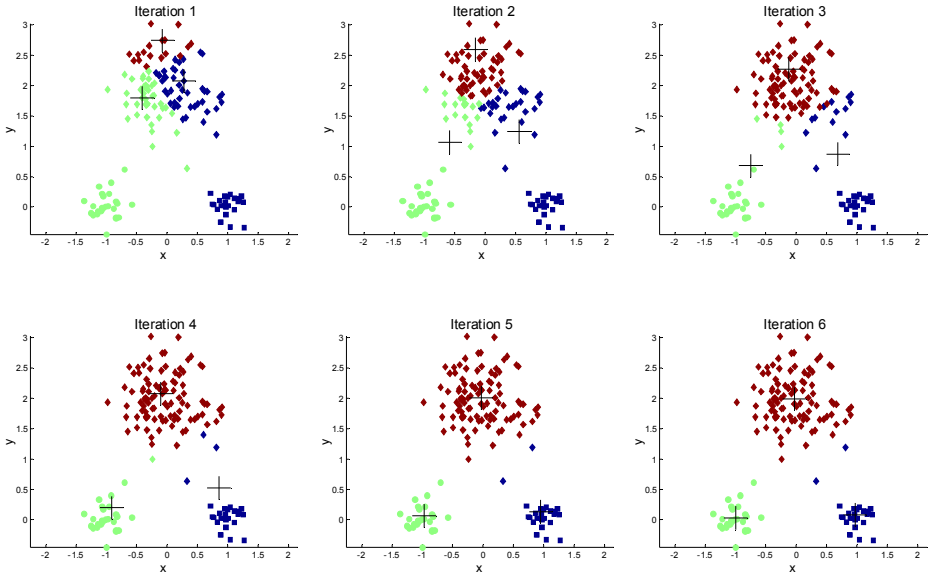
k-Means

- Algoritmo de agrupamiento por particiones.
- Número de clusters conocido (k).
- Cada cluster tiene asociado un centroide (centro geométrico del cluster).
- Los puntos se asignan al cluster cuyo centroide esté más cerca (utilizando cualquier métrica de distancia).
- Iterativamente, se van actualizando los centroides en función de las asignaciones de puntos a clusters, hasta que los centroides dejen de cambiar.

k-Means



k-Means



k-Means

Inicialización

- Escoger k centroides aleatoriamente (hay métodos más sofisticados).
- Formar k grupos, asignando cada punto al centroide más cercano

Proceso iterativo

Mientras que los centroides cambien:

- Calcular las distancias de todos los puntos a los k centroides.
- Formar k grupos, asignando cada punto al centroide más cercano.
- Recalcular los nuevos centroides.

k-Means

Complejidad

n = número de puntos,
k = número de clusters,
I = número iteraciones,
d = número de atributos

Problema NP si k no se fija.

Ahora bien, si fijamos los parámetros n, k, d, I:

O(n * k * I * d)

k-Means

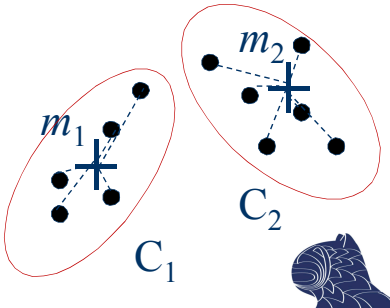
¿Cómo se recalculan los centroides?

Partiendo de k grupos de puntos,
cada grupo determinará un nuevo centroide m_i.

- Se elige una medida global (función objetivo)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d^2(m_i, x)$$

- Se escogen los valores de m_i que minimizan dicha función.



k-Means

¿Cómo se recalculan los centroides?

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d^2(m_i, x)$$

- Cuando se utiliza la distancia euclídea, SSE se minimiza usando la media aritmética (por cada atributo o variable)

x_a = (0.4, 0.6, 0.6, 0.7)
x_b = (0.3, 0.2, 0.1, 0.4)
x_c = (0.3, 0.2, 0.2, 0.4)
m_1 = (0.33, 0.33, 0.3, 0.5)

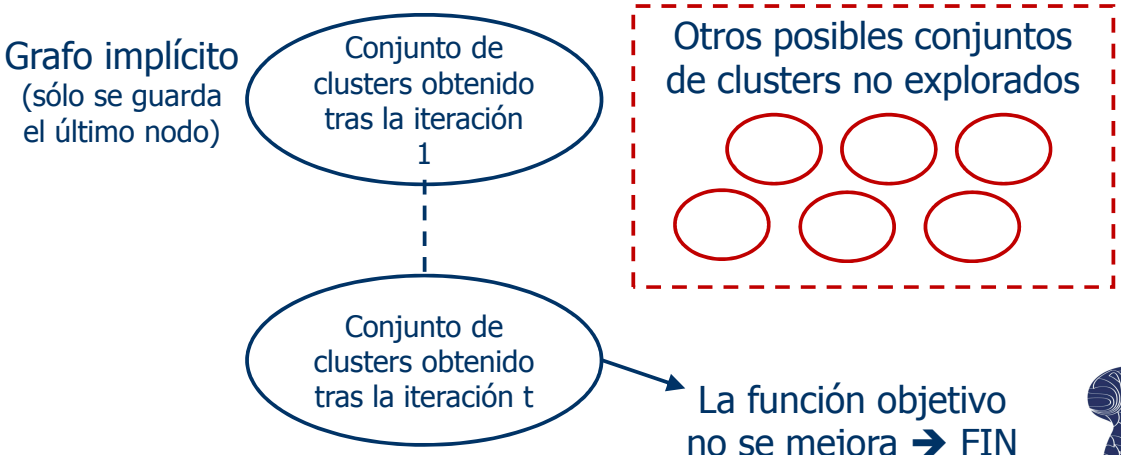
- Cuando se emplea la distancia de Manhattan, SSE se minimiza usando la mediana.
- La media funciona bien con muchas distancias (por ejemplo, cualquier divergencia de Bregman).

k-Means

Estrategia de control en la búsqueda:

Ascensión de colinas por la máxima pendiente

Después de cada iteración, no se recuerda el estado para volver atrás y probar otros posibles centroides.



k-Means

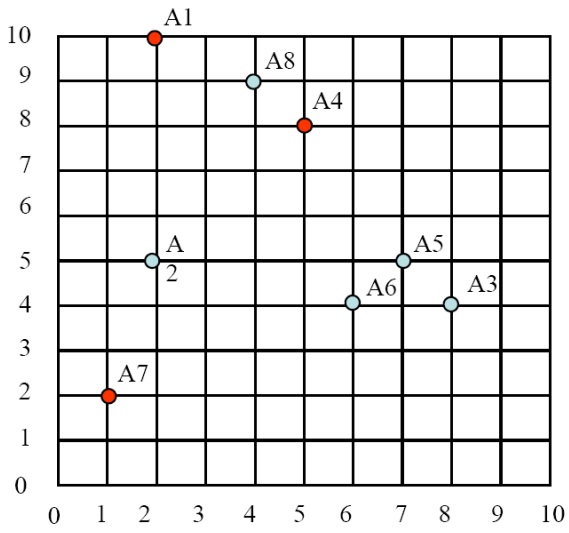
Ejercicio

Agrupar los 8 puntos de la figura en 3 clusters usando el algoritmo de las K medias.

Centroides iniciales:
A1, A7 y A8

Métricas de distancia:

- Distancia euclídea.
- Distancia de Manhattan.
- Distancia de Chebyshev.



k-Means

Ejercicio resuelto

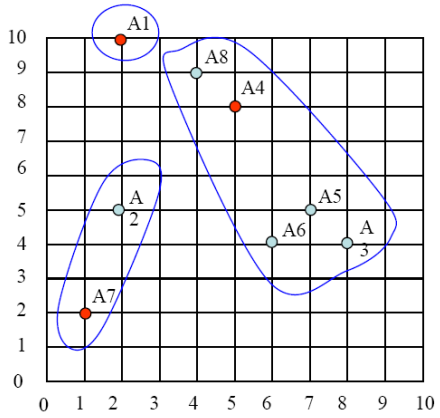
Distancia euclídea

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

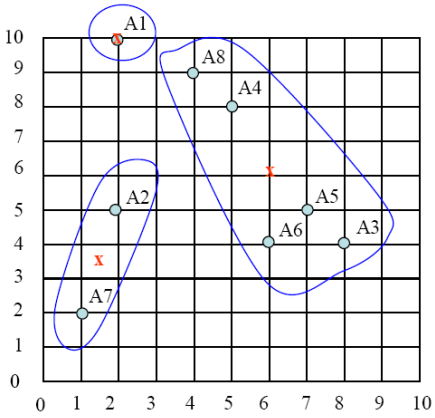
k-Means

Ejercicio resuelto

Distancia euclídea



Primera iteración

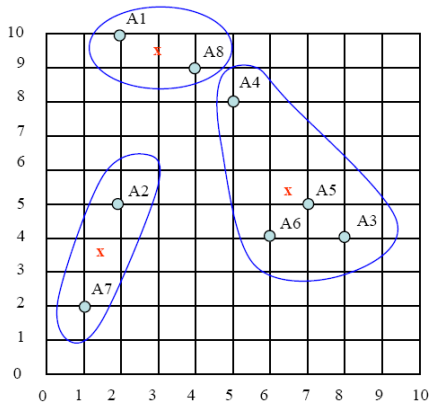


Segunda iteración

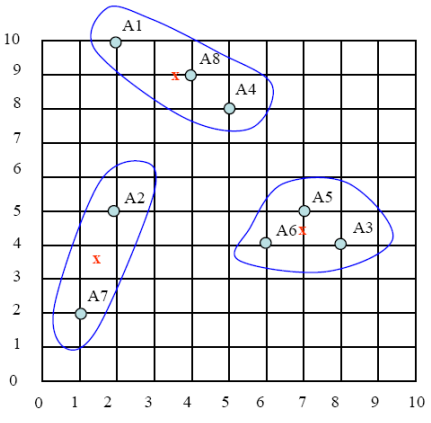
k-Means

Ejercicio resuelto

Distancia euclídea



Tercera iteración



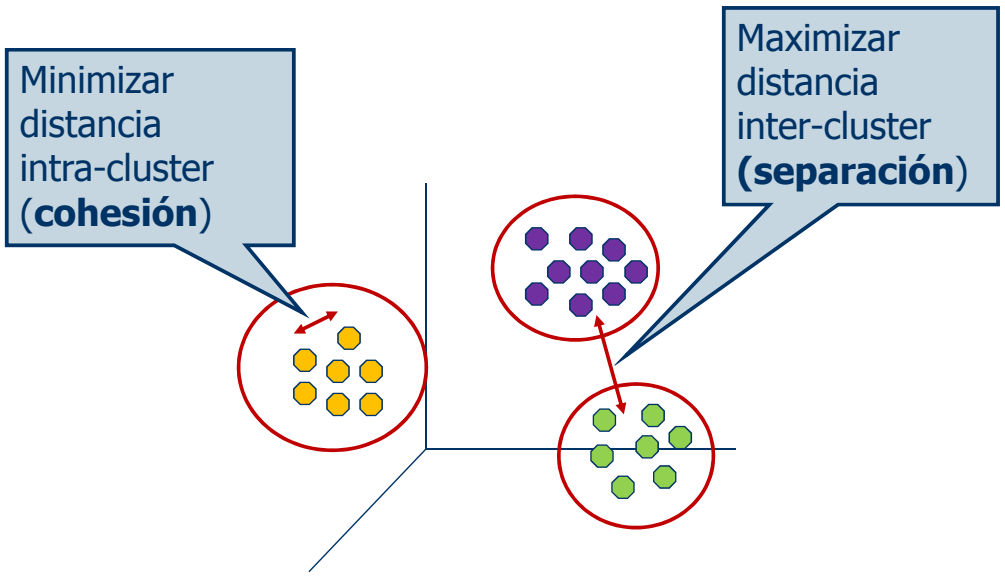
Configuración final

k-Means

DEMO

http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html

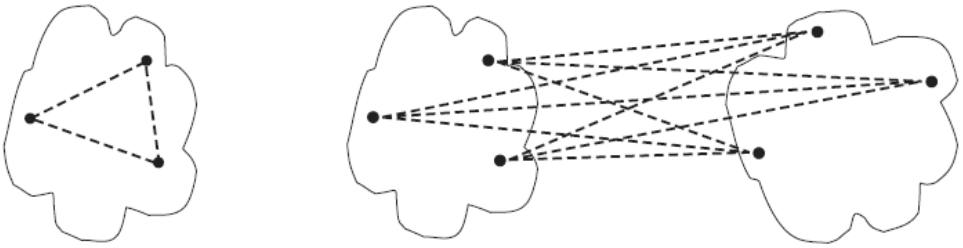
Evaluación de resultados



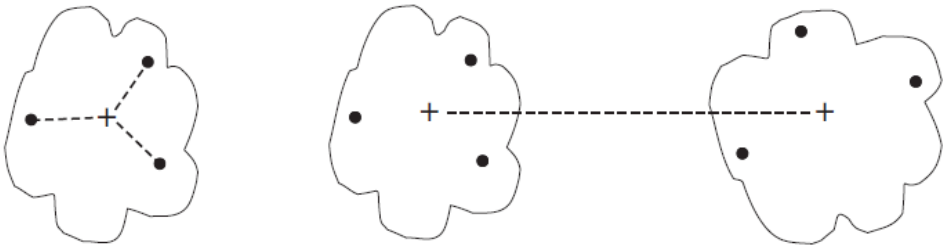
Evaluación de resultados

Formas de medir cohesión y separación:

- Sin usar centroides:



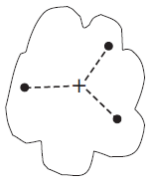
- Usando centroides:



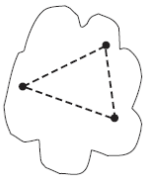
Evaluación de resultados

Cuando la distancia utilizada es la distancia euclídea:

1. El uso de centroides no influye al medir la cohesión:



$$\sum_{x \in C_i} dist^2(m_i, x) = \frac{1}{2(\#C_i)} \sum_{x \in C_i} \sum_{y \in C_i} dist^2(x, y)$$



$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) = \sum_{i=1}^{K} \frac{1}{2(\#C_i)} \sum_{x \in C_i} \sum_{y \in C_i} dist^2(x, y)$$

2. Minimizar cohesión y maximizar separación son equivalentes.

Evaluación de resultados

Así pues, cuando se usa la distancia euclídea, SSE es una buena medida del grado de ajuste (cohesión y separación) de los centroides hallados.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d^2(m_i, x)$$

Por otro lado, ya sabíamos que, en cada iteración del algoritmo de las k-medias, se minimizaba SSE al calcular los centroides usando la media aritmética.

¿Garantiza lo anterior que los centroides finales sean los que minimicen SSE globalmente? **NO**

Evaluación de resultados

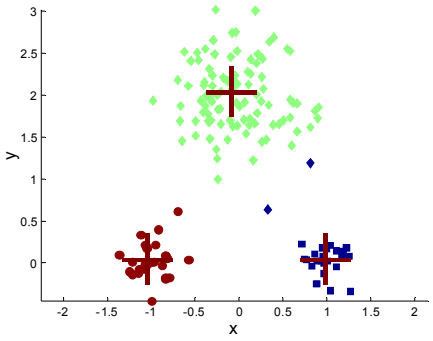
Cuando usamos la distancia euclídea, el centroide determinado en cada iteración por el vector de medias garantiza la mejor solución con respecto a SSE, pero considerando:

- un valor de k fijo, y
- los centroides dados por la iteración anterior.

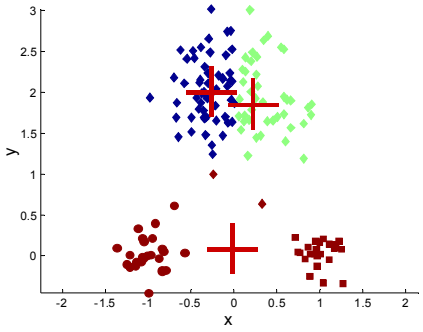
La solución final no será la óptima:

El algoritmo de las k medias **no garantiza** que los centroides finales obtenidos sean los que minimizan globalmente la función objetivo SSE.

Evaluación de resultados

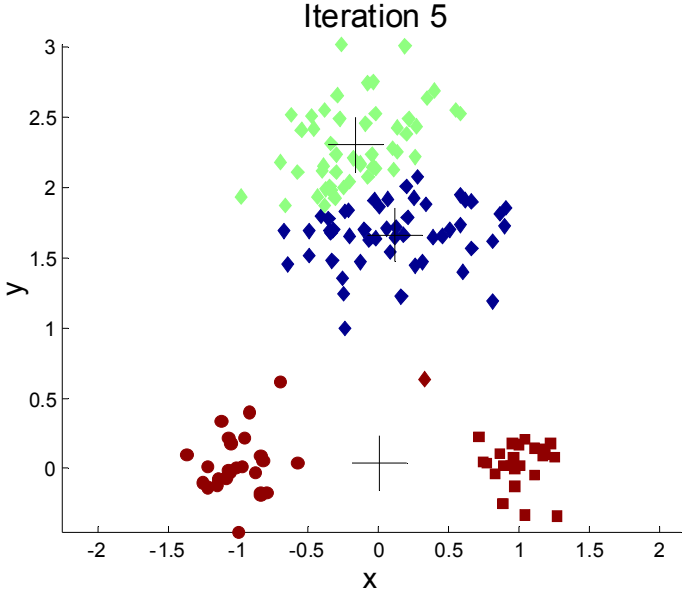


Solución óptima

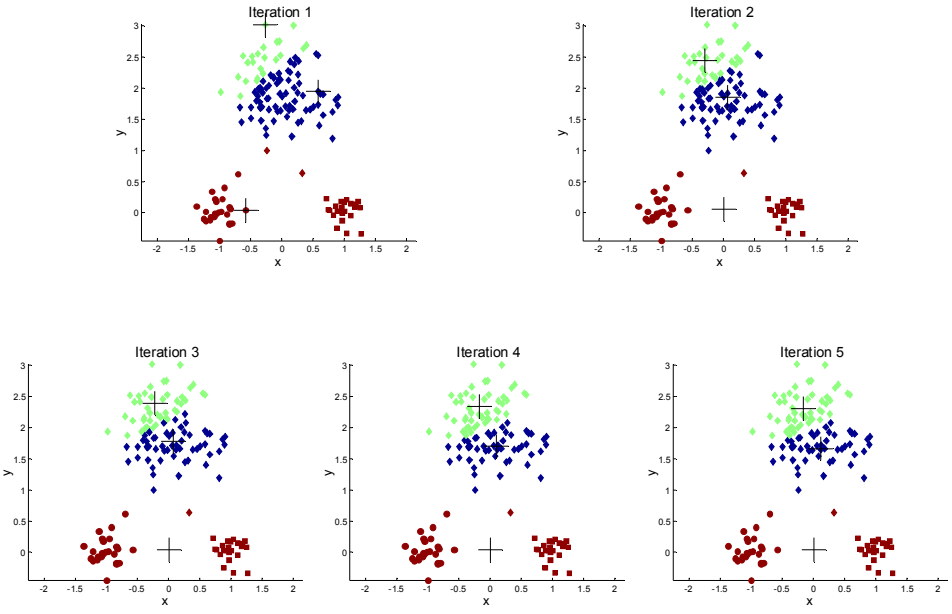


Posible resultado proporcionado por k-means
Óptimo local

Evaluación de resultados



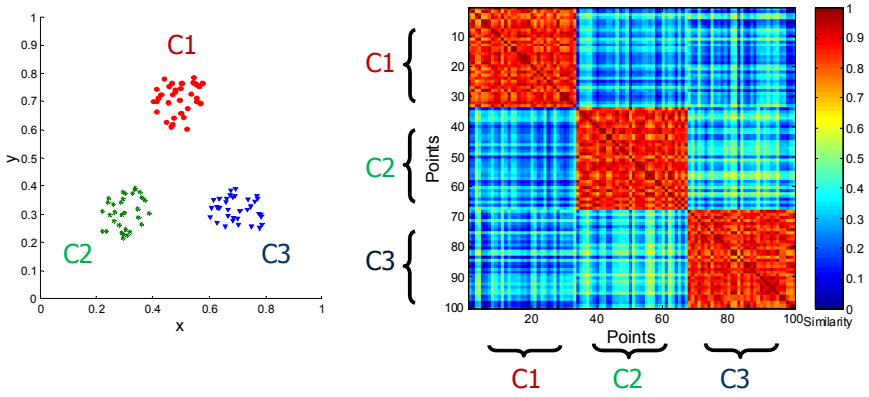
Evaluación de resultados



Evaluación de resultados

Matriz de similitud

Ordenamos los datos en la matriz de similitud con respecto a los clusters en los que quedan los datos e inspeccionamos visualmente...



k-Means

Problema

Sensible a la elección inicial de los centroides.

Posibles soluciones

- Realizar varias ejecuciones con varios conjuntos de centroides iniciales y comparar resultados (GRASP).
- Estimar a priori unos buenos centroides:
  - Métodos específicos: k-means++
  - Escoger una muestra y aplicar un método jerárquico.

k-Means

Problema

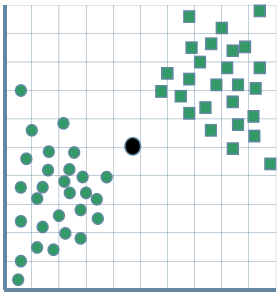
Hay que elegir a priori el valor de k

(a priori, no sabemos cuántos grupos puede haber).

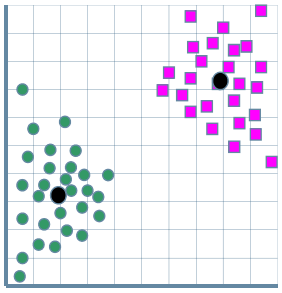
Posibles soluciones

- Usar un método jerárquico sobre una muestra de los datos (por eficiencia) para estimar el valor de k.
- Usar un valor de k alto, ver los resultados y ajustar.
  Siempre que se aumente el valor de k, disminuirá el valor SSE.
  Lo normal será ir probando con varios valores de k
  y comprobar cuándo no hay una mejora significativa en SSE.

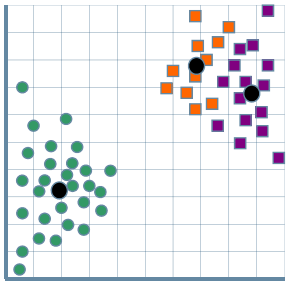
k-Means



k = 1
SSE = 873.0

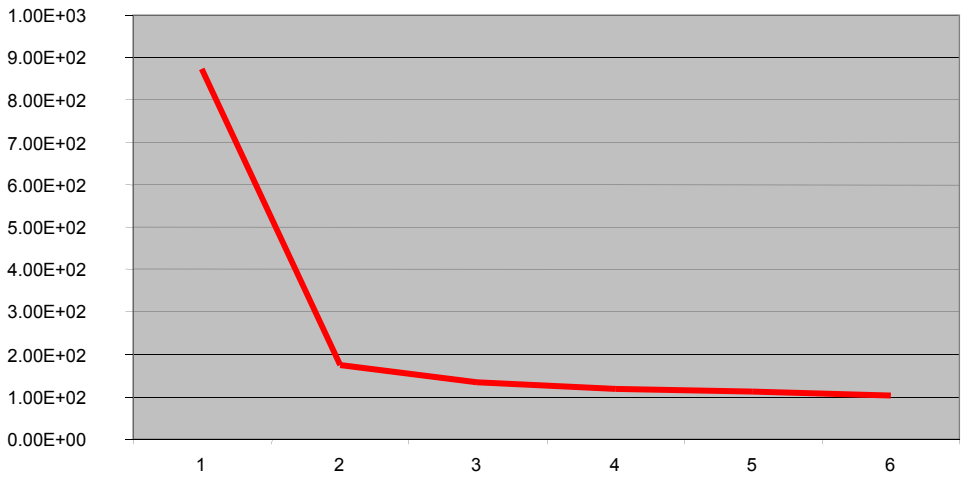


k = 2
SSE = 173.1



k = 3
SSE = 133.6

k-Means



El codo en k=2 sugiere que éste es el valor más adecuado para el número de agrupamientos.

k-Means

Problema

Cuando se usan la media para calcular los centroides, el método es sensible a outliers (valores anómalos).

Posibles soluciones

- Usar medianas en vez de medias (aun con la distancia euclídea).
- Eliminar previamente los outliers. ¡Ojo! Los outliers pueden ser valores interesantes
- Usar k-medoids: En vez de usar el vector de medias como centroide, se usa el vector correspondiente a un dato real (un representante).

k-Means

Problema

Manejo de atributos no numéricos.

Posibles soluciones

- Extender la medida de similitud para que incluya atributos nominales, p.ej.

  d(a,b) = 1 si a=b, 0 en otro caso

  Elegir como representante en el centroide la moda de los datos asignados a dicho cluster (método k-mode).

k-Means

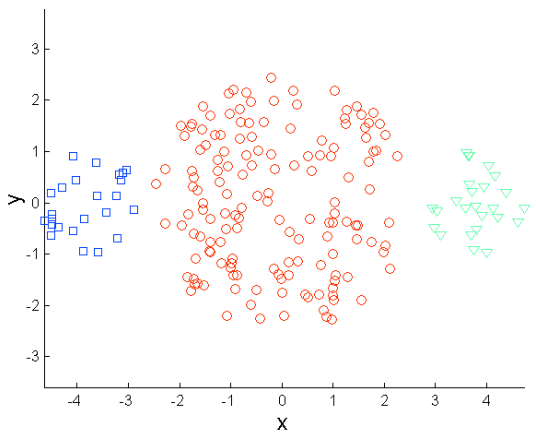
Problema

K-Means no funciona bien cuando los clusters son:

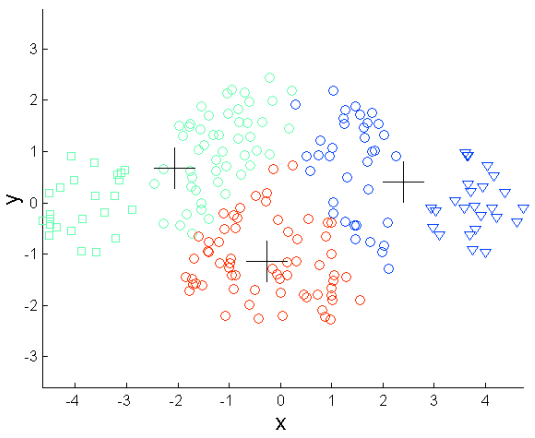
- de distinto tamaño
- de diferente densidad
- no convexos

k-Means

Clusters de distinto tamaño



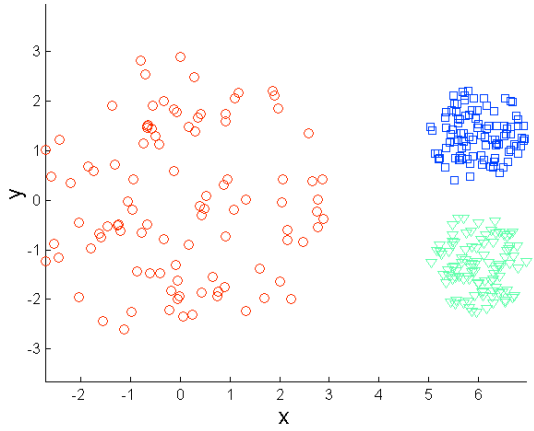
Puntos originales



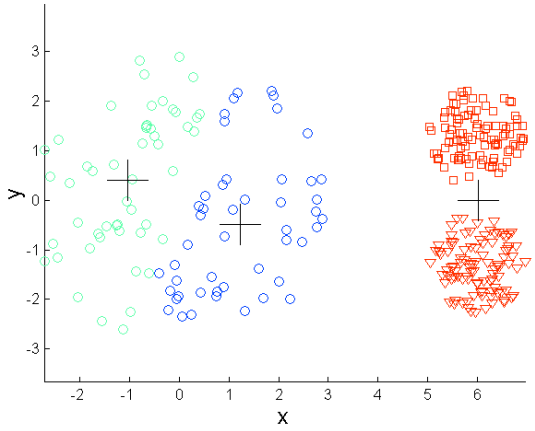
k-Means (3 clusters)

k-Means

Clusters de distinta densidad



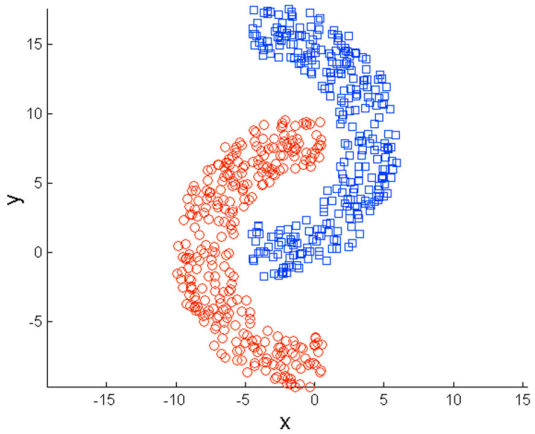
Puntos originales



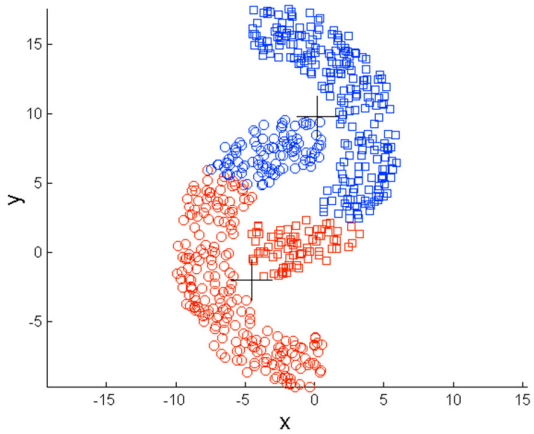
k-Means (3 clusters)

k-Means

Clusters no convexos



Puntos originales



k-Means (2 clusters)

k-Means

Problema

K-Means no funciona bien cuando los clusters son:

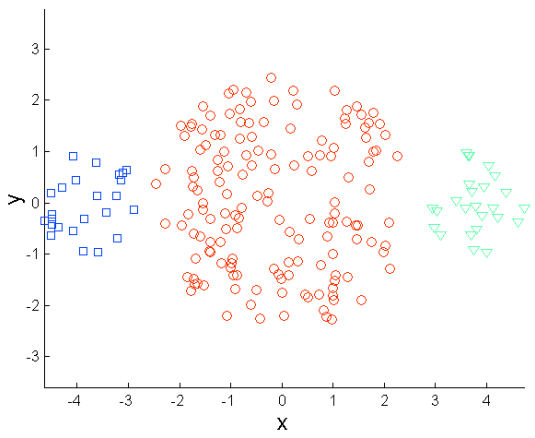
- de distinto tamaño
- de diferente densidad
- no convexos

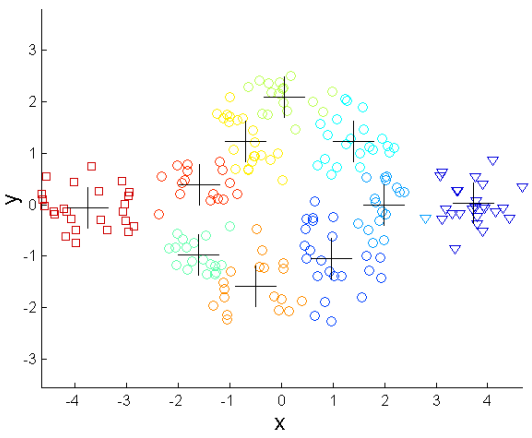
Posibles soluciones

- Métodos ad-hoc.
- Usar un valor de k alto y revisar los resultados.
- "Spectral clustering"

k-Means

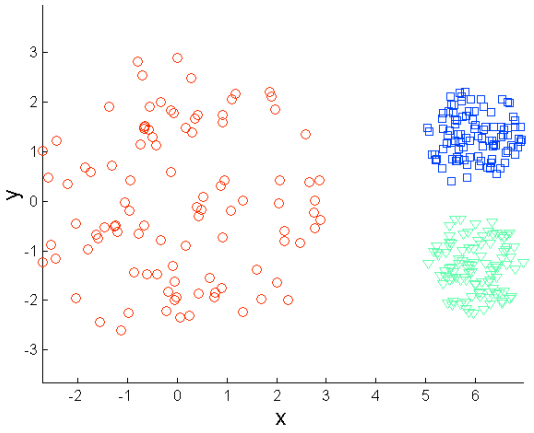
Clusters de distinto tamaño

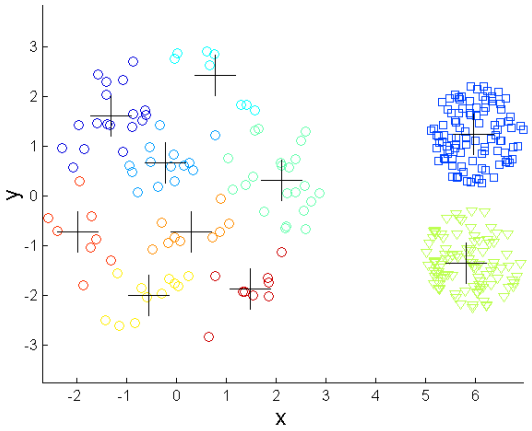




k-Means

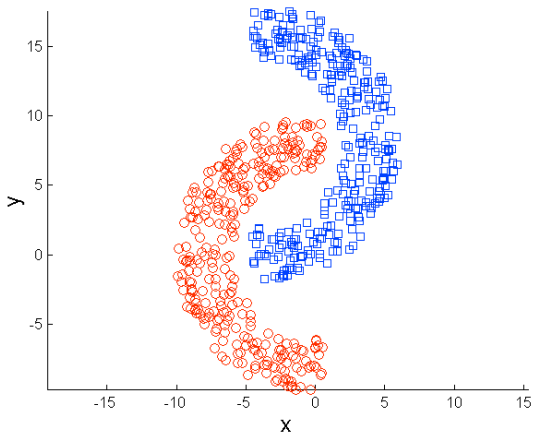
Clusters de distinta densidad

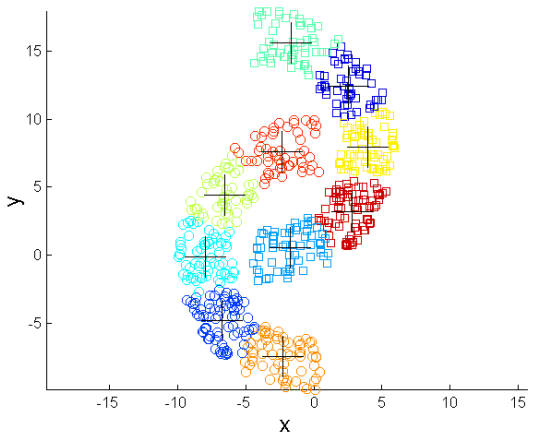




k-Means

Clusters no convexos





k-Means

Pre-procesamiento

- Normalizar los datos.
- Detectar outliers (eliminarlos, en su caso).

Post-procesamiento

- Eliminar pequeños clusters que puedan representar outliers.
- Dividir clusters dispersos ('loose' clusters); esto es, clusters con un SSE relativamente alto.
- Combinar clusters cercanos que tengan un SSE relativamente bajo.

NOTA: Estos criterios se pueden incluir en el propio algoritmo de clustering (p.ej. algoritmo ISODATA).

k-Means

Variantes

- **GRASP** [Greedy Randomized Adaptive Search Procedure] para evitar óptimos locales.
- **k-Modes** (Huang'1998) utiliza modas en vez de medias (para poder trabajar con atributos de tipo categórico).
- **k-Medoids** utiliza medianas en vez de medias para limitar la influencia de los outliers.

  vg. PAM (Partitioning Around Medoids, 1987)

  CLARA (Clustering LARge Applications, 1990)

  CLARANS (CLARA + Randomized Search, 1994)

Validación de resultados

¿Cómo se puede evaluar
la calidad de los clusters obtenidos?

Depende de lo que estemos buscando…

Hay situaciones en las que nos interesa:

- Evitar descubrir clusters donde sólo hay ruido.
- Comparar dos conjuntos de clusters alternativos.
- Comparar dos técnicas de agrupamiento.

Validación de resultados

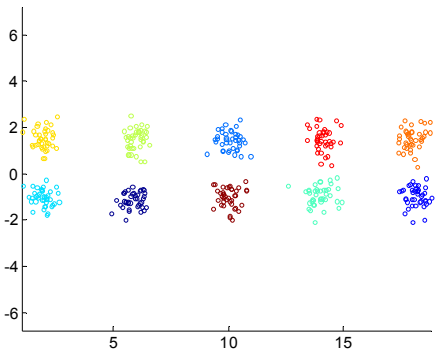
- **Criterios externos**
  (aportando información adicional)
  p.ej. entropía/pureza (como en clasificación)

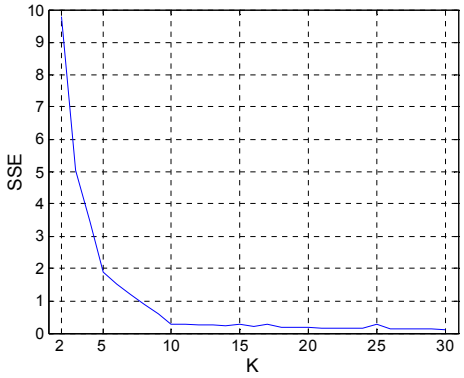
- **Criterios internos**
  (a partir de los propios datos),
  p.ej. SSE ("Sum of Squared Error")
  - para comparar clusters
  - para estimar el número de clusters

Otras medidas:
cohesión, separación, coeficientes de silueta…

Validación de resultados

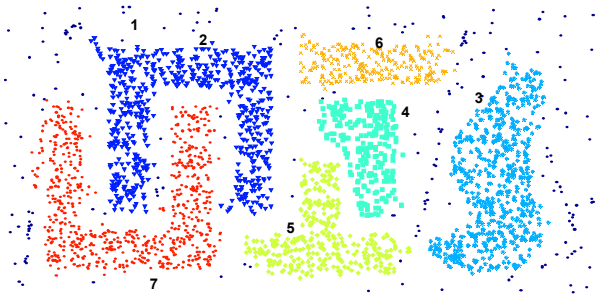
SSE ("Sum of Squared Error")

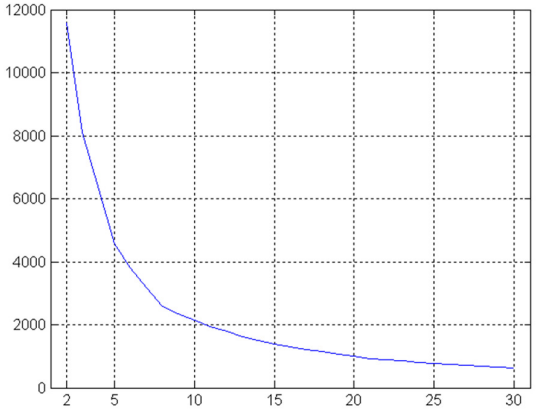




Validación de resultados

SSE ("Sum of Squared Error")

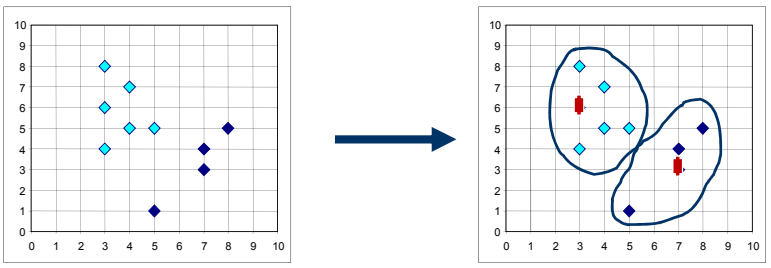




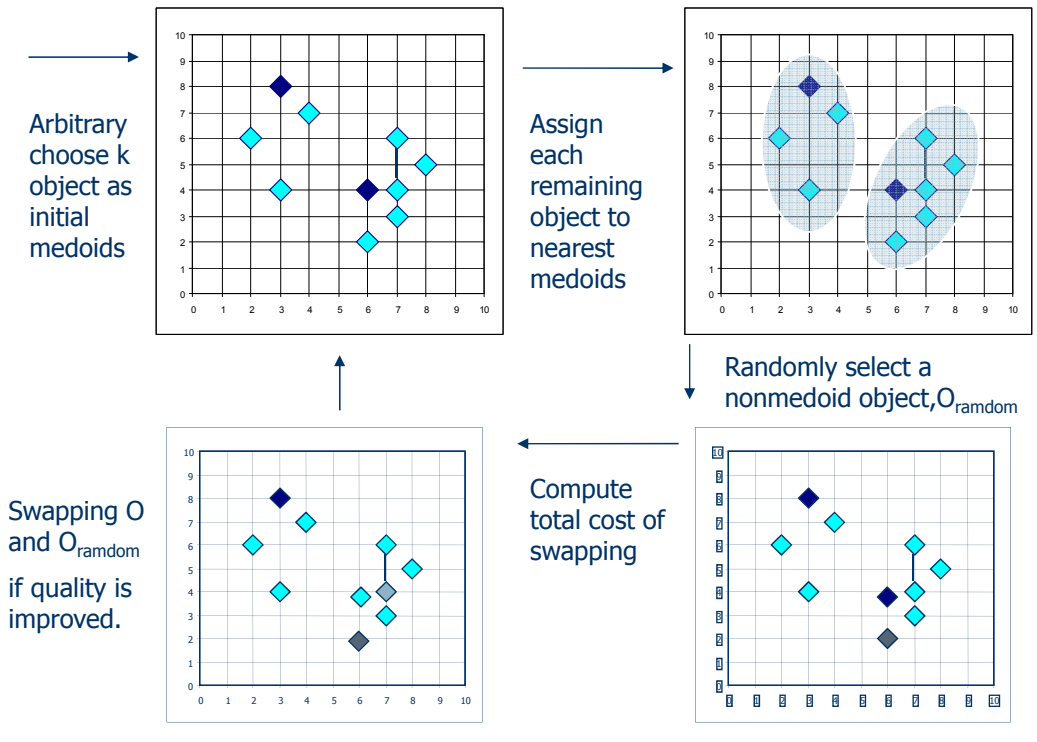
k-Medoids

El algoritmo k-means es sensible a la presencia de outliers…

En vez de utilizar centroides, podemos emplear medoides; esto es, tomar como referencia un objeto ya existente en el cluster (idealmente, el objeto más central del cluster)



PAM [Partitioning Around Medoids]



PAM [Partitioning Around Medoids]

Algorithm PAM

1. Select k representative objects arbitrarily.
2. Compute the total cost of replacing for all pairs of objects (Om, Op) where Om is currently selected, and Op is not.
3. Select the pair (Om, Op) which corresponds to the minimum total cost of replacinf. If the minimum is negative, replace Om with Op, and go back to Step 2.
4. Otherwise, for each non-selected object, find the most similar representative object. Halt.

PAM [Partitioning Around Medoids]

Eficiencia

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

vs. K-Means $O (n * k)$

- Funciona bien con conjuntos de datos pequeños, pero no escala bien debido a su complejidad computacional.

CLARA [Clustering for LARge Applications]

Algorithm CLARA

1. For i = 1 to 5, repeat the following steps:
2. Draw a sample of 40 + 2k objects randomly from the entire data set, and call Algorithm PAM to find k medoids of the sample.
3. For each object Oj in the entire data set, determine which of the k medoids is the most similar to Oj.
4. Calculate the average dissimilarity of the clustering obtained in the previous step. If this value is less than the current minimum, use this value as the current minimum, and retain the k medoids found in Step 2 as the best set of medoids obtained so far.
5. Return to Step 1 to start the next iteration

CLARA [Clustering for LARge Applications]

Eficiencia

$$\mathbf{O\ (ks^2 + k(n-k))}$$

vs. PAM $O\ (\ k * (n - k)^2\ )$

vs. K-Means $O(\ n * k * I * d\ )$

CLARA muestrea el conjunto de datos
para mejorar la eficiencia de PAM.

CLARANS [CLARA based on Randomized Search]

Algorithm CLARANS

Parameters:

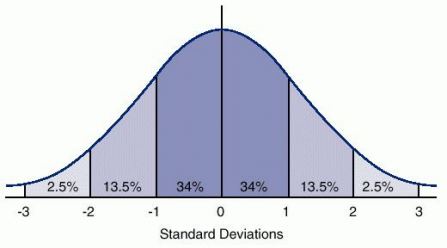
- numlocal (number of local minima obtained)
- maxneighbor (maximum number of neighbors examined).

1. Initialize i to 1, and mincost to a large number.
2. Set current to an arbitrary node in $G_{n;k}$.
3. Set j to 1.
4. Consider a random neighbor S of current and calculate the cost differential of the two nodes.
5. If S has a lower cost, set current to S, and go to Step 3.
6. Otherwise, increment j by 1. If j<=maxneighbor, go to Step 4.
7. Otherwise, when j > maxneighbor, compare the cost of current with mincost. If the former is less than mincost, set mincost to the cost of current and set bestnode to current.
8. Increment i by 1. If i > numlocal, output bestnode and halt. Otherwise, go to Step 2.

BFR Algorithm

- El algoritmo BFR [Bradley-Fayyad-Reina] es una variante del algoritmo de las k-medias para conjuntos de datos muy grandes (almacenados en disco).

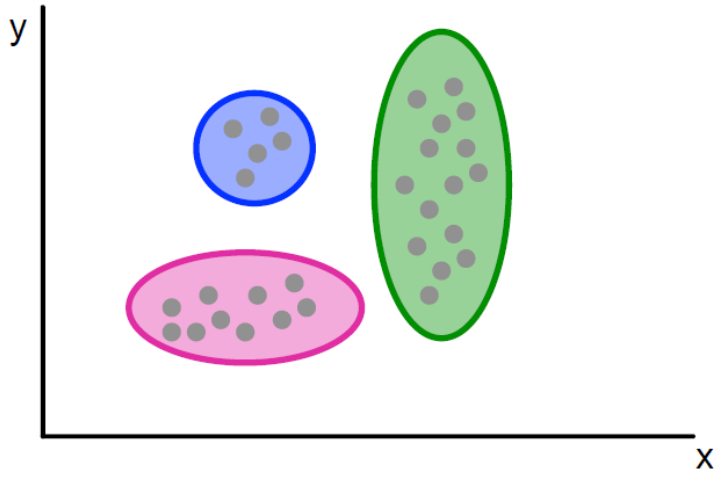
- El algoritmo BFR asume que cada cluster está normalmente distribuido en torno a un centroide en un espacio euclídeo.



- Se cuantifica la probabilidad de que un punto se encuentre en un cluster dada la distancia del punto al centroide, dimensión por dimensión.

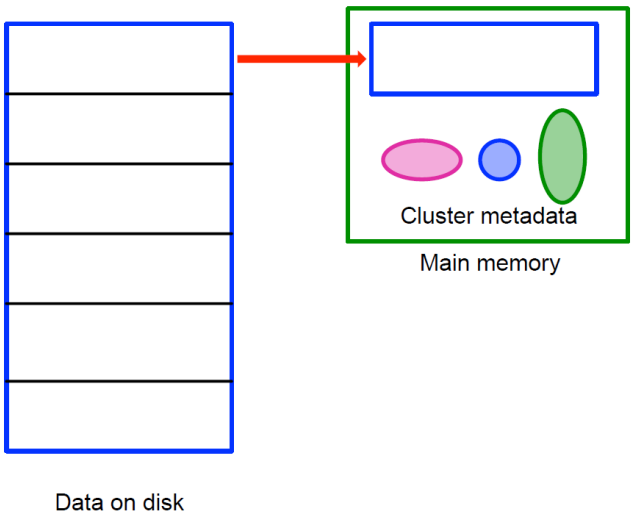
BFR Algorithm

- La suposición de normalidad implica que los clusters son elipsoides alineados con los ejes (las desviaciones en distintas dimensiones pueden ser diferentes).



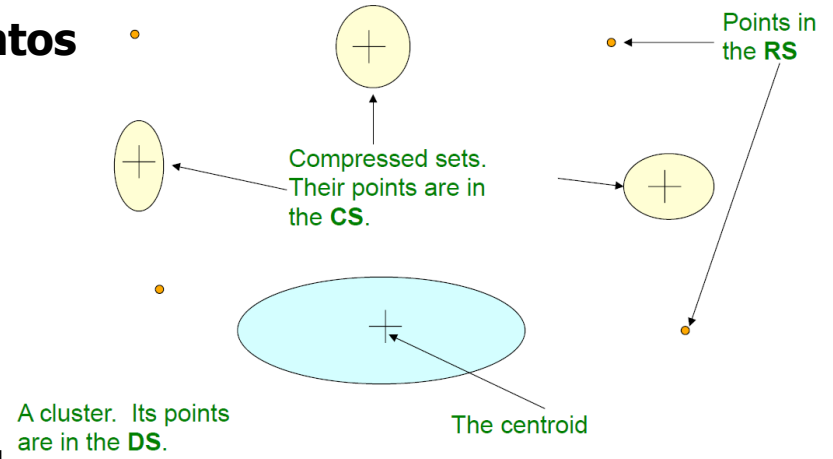
BFR Algorithm

- Los datos se leen de disco en bloques, hasta llenar la memoria disponible.
- La información de los bloques anteriores se resume con estadísticos sencillos.
- Al comenzar, se utiliza cualquier técnica razonable para elegir los centroides (k-means) dado el primer bloque de datos.



BFR Algorithm

Tres tipos de puntos



- **DS** [Discard Set]:
  Puntos cercanos al centroide (resumidos)
- **CS** [Compression Set]:
  Puntos cercanos entre sí pero lejos de centroides.
- **RS** [Retained Set]:
  Puntos aislados, a la espera de ser asignados a un CS.

BFR Algorithm

Para cada cluster,

el conjunto DS [discarded set] se resume con

- El número de puntos del conjunto, n
- El vector SUM de sumas de las coordenadas de los puntos (el i-ésimo componente del vector corresponde a la suma de la i-ésima dimensión).
- El vector SUMSQ de la suma de los cuadrados de las coordenadas de los puntos.

Total: **2d+1 valores por cluster** que permiten calcular media y varianza para cada dimensión.

BFR Algorithm

Algoritmo

- Encontrar los puntos "suficientemente" cercanos a un centroide.
  - Añadir los puntos al cluster y a su conjunto DS (ajustando sus estadísticas: n, SUM y SUMSQ).
  - Descartas esos puntos.
- Para los demás puntos, que no quedan cerca de ningún centroide:
  - Usar cualquier algoritmo de clustering en memoria para agrupar esos puntos y los antiguos outliers de RS (los clusters van a CS, los outliers a RS)
- Considerar la mezcla de conjuntos comprimidos en CS.

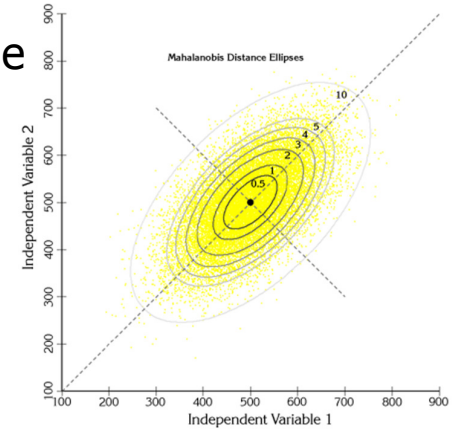
NOTA:
Al final, todos los CS y RS se asignan a su cluster más cercano

BFR Algorithm

Detalles de implementación

¿Cómo decidimos cuándo está un punto lo suficientemente cerca de un cluster para añadirlo a él?

Si la distancia del punto al centroide está por debajo de un umbral (parámetro del algoritmo), asumimos que es muy probable que el punto pertenezca al cluster:



- Distancia de Mahalanobis

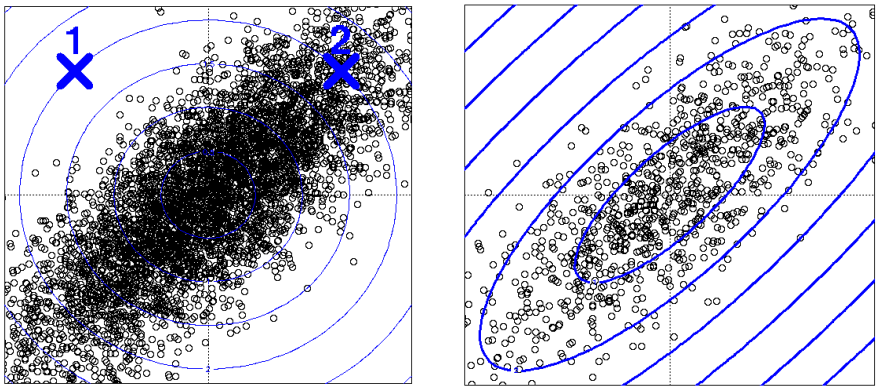
$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T \Sigma^{-1} (\vec{x}-\vec{y})}.$$

- Distancia euclídea sobre z-scores: $z = (x-\mu)/\sigma$

BFR Algorithm

Detalles de implementación

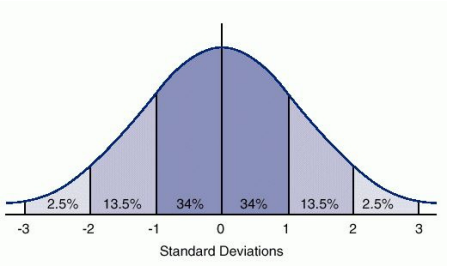
Distancia euclídea vs. Distancia de Mahalanobis



Puntos normalmente distribuidos
(contornos de puntos equidistantes del origen)

BFR Algorithm

Detalles de implementación



- Si un punto está a una desviación estándar del centroide en todas sus dimensiones (y no existe correlación entre las variables), su distancia de Mahalanobis es $\sqrt{d}$

- Dada una distribución gaussiana/normal, el 68% de los puntos están a distancia menor o igual a $\sqrt{d}$
  el 95% por debajo de $2\sqrt{d}$
  el 99% por debajo de $3\sqrt{d}$
  (p.ej. podemos usar este último valor como umbral).

BFR Algorithm

Detalles de implementación

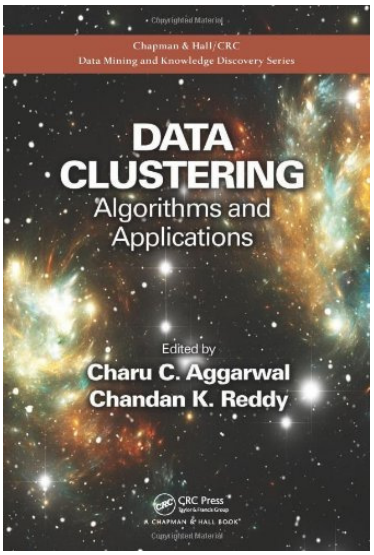
¿Cómo decidimos cuándo combinar dos conjuntos CS?

- Calculamos la varianza del cluster combinado (a partir de los estadísticos que resumen cada conjunto) y fusionamos los conjuntos si la varianza combinada queda por debajo de un umbral.

- Otras alternativas posibles: tratar cada dimensión de forma diferente, considerar la densidad…

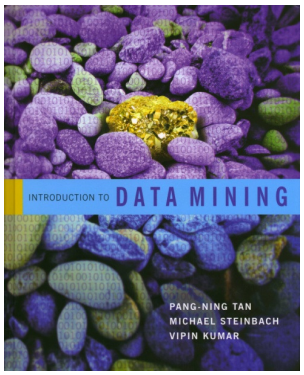
Bibliografía

- Charu C. Aggarwal & Chandan K. Reddy (editors): **Data Clustering: Algorithms and Applications**. Chapman & Hall / CRC Press, 2014. ISBN 1466558210.

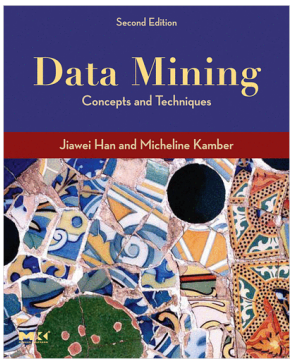


Bibliografía - Libros de texto

- Pang-Ning Tan,
  Michael Steinbach
  & Vipin Kumar:
  Introduction to Data Mining
  Addison-Wesley, 2006.
  ISBN 0321321367 [capítulos 8&9]



- Jiawei Han
  & Micheline Kamber:
  Data Mining:
  Concepts and Techniques
  Morgan Kaufmann, 2006.
  ISBN 1558609016 [capítulo 7]



Bibliografía - Algoritmos

K-Means

- J.B. MacQueen: **Some Methods for classification and Analysis of Multivariate Observations**. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1, pp. 281–297. University of California Press, 1967. MR 0214227. Zbl 0214.46201
- S.P. Lloyd: **Least squares quantization in PCM**. Bell Telephone Laboratories, 1957. Published much later in IEEE Transactions on Information Theory 28 (2): 129–137, 1982. DOI 10.1109/TIT.1982.1056489

PAM [Partitioning Around Medoids]

- L. Kaufman & Peter J. Rousseeuw: **Clustering by means of Medoids**, First International Conference on Statistical Data Analysis based on the L_1-Norm and Related Methods, Neuchatel (Switzerland), August 31 - September 4, 1987, pp. 405–416. ISBN 0444702733.

CLARA [Clustering for LARge Applications]

- L. Kaufman & P.J. Rousseeuw: **Finding Groups in Data: An Introduction to Cluster Analysis**. John Wiley & Sons, 1990. ISBN 0471878766

CLARANS [Clustering for Large Applications based on RANdomized Search]

- R. Ng & J. Han: **Efficient and Effective Clustering Methods for Spatial Data Mining**, VLDB'1994, pp. 144–155, 1994. http://www.vldb.org/conf/1994/P144.PDF
- R. Ng & J. Han: **CLARANS: a method for clustering objects for spatial data mining**, IEEE Transactions on Knowledge and Data Engineering, 14(5):1003-1016, 2002. DOI 10.1109/TKDE.2002.1033770.

Bibliografía - Algoritmos

BFR

- Paul S. Bradley, Usama M. Fayyad & Cory Reina: **Scaling Clustering Algorithms to Large Databases**. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998, pages 9-15. AAAI Press, 1998 ISBN 1-57735-070-7. http://www.aaai.org/Papers/KDD/1998/KDD98-002.pdf
- Fredrik Farnstrom, James Lewis & Charles Elkan: **Scalability for clustering algorithms revisited**. SIGKDD Explorations Newsletter 2(1):51-57, June 2000. DOI 10.1145/360402.360419

K-Means++

- David Arthur & Sergei Vassilvitskii: **k-means++: the advantages of careful seeding**. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07), pages 1027-1035. Society for Industrial and Applied Mathematics, 2007. ISBN 978-0-898716-24-5
- Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar & Sergei Vassilvitskii: **Scalable k-means++**. Proc. VLDB Endowment 5(7):622-633, March 2012. DOI 10.14778/2180912.2180915
- Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman & Chaitanya Swamy: **The effectiveness of lloyd-type methods for the k-means problem**. Journal of the ACM 59(6), article 28, December 2012. DOI 10.1145/2395116.2395117

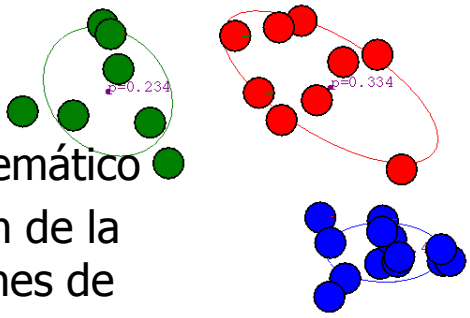
Apéndice: Notación O

El impacto de la eficiencia de un algoritmo...

n	10	100	1000	10000	100000
O(n)	10ms	0.1s	1s	10s	100s
$O(n \cdot \log_2 n)$	33ms	0.7s	10s	2 min	28 min
$O(n^2)$	100ms	10s	17 min	28 horas	115 días
$O(n^3)$	1s	17min	12 días	31 años	32 milenios

Apéndice
Otros métodos de clustering

Clustering basado en modelos



Ajustar los datos a un modelo matemático

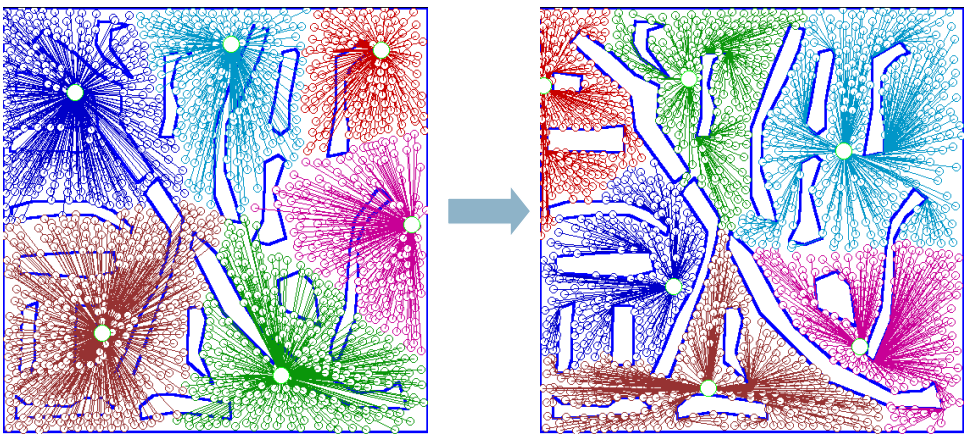
(se supone que los datos provienen de la superposición de varias distribuciones de probabilidades)

- Estadística:
  EM [Expectation Maximization], **AutoClass**
- Clustering conceptual (Machine Learning):
  COBWEB, CLASSIT
- Redes neuronales:
  SOM [Self-Organizing Maps]

Apéndice
Otros métodos de clustering

Clustering con restricciones

p.ej. Clustering con obstáculos



Posibles aplicaciones:

Distribución de cajeros automáticos/supermercados…