Clustering: Algoritmos

Clasificación no supervisada

Javier G. Sogo

10 de marzo de 2015

Introducción

Acotar el problema

Complejidad del algoritmo

- Implementación näive de clustering jerárquico: $O(N^3)$
- Una implementación mejor puede llegar a $O(N^2 logN)$

Tamaño del problema

- ¿Podemos cargar todos los datos en memoria?
- ¿Es crítico el tiempo de ejecución del algoritmo?

Introducción

Medida de distancia

- Depende del número de dimensiones d y de los valores que puedan tomar. Ejs.:
  - Documento como conjunto de palabras –> distancia **Jaccard**
  - Documento como punto $\mathbf{x} = (x_1, \ldots, x_d)$ donde $x_i = 1$ si el documento contiene la palabra i –> distancia **Euclidea**
  - Documento como vector en un *espacio de palabras* –> distancia **coseno**.
- Cuando hay muchas dimensiones d todos los puntos están cerca.
- Atención a la escala de las variables (standarizar o distancia **Mahalanobis**).

Cuándo detener el algoritmo

Clúster jerárquico

- Elegir un número k de clases *a priori* y detener el algoritmo cuando se alcance ese número.
- Medir la **cohesión** del cluster:
- **Diámetro**: máxima distancia entre dos puntos del cluster.
- **Radio**: máxima distancia de un punto al centroide.
- Basarse en la **densidad**: número de puntos por unidad de volumen (utilizar radio o diámetro).

Clúster particional

- Criterio de **convergencia**: detener el algoritmos cuando los puntos no se muevan entre clusters y los centroides no cambien.

Introducción

A tener en cuenta

- Cómo tratar attributos no numéricos.
- Cómo tratar valores no disponibles: imputación.

Algoritmo: K-medias

K-medias

Inicialización

- Seleccionar una medida de distancia.
- Seleccionar un método para medir la distancia entre clusters (simple, completo, media,...).
- Seleccionar el número de clusters k
- Inicializar los clusters escogiendo un punto para cada uno de ellos.

K-medias

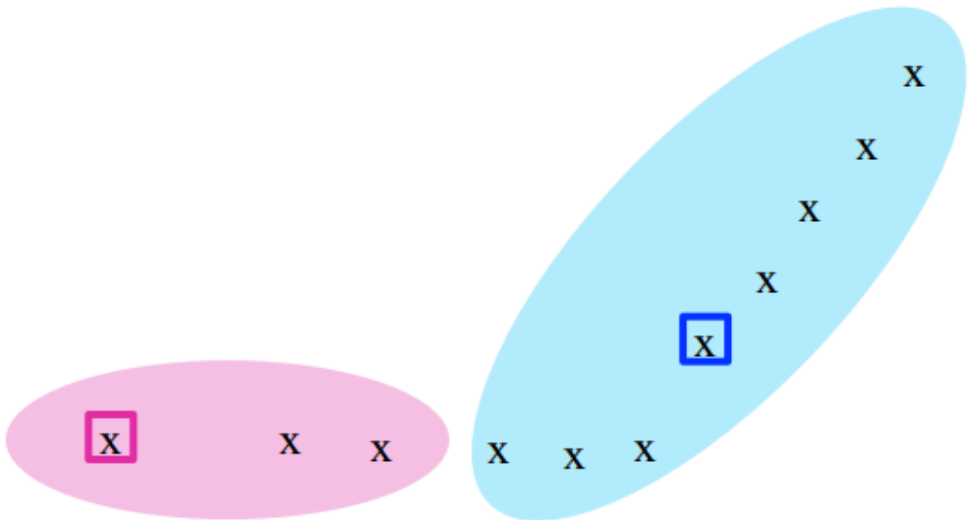
Algoritmo paso a paso

1. Para cada punto, calcular la distancia a los clusters y asignarlo a aquél más próximo.
2. Una vez que todos los puntos han sido asignados, actualizar los centroides de los k clusters.
3. Reasignar todos los puntos al cluster más cercano.

- Repetir los pasos **2** y **3** hasta lograr la convergencia (los puntos no cambian de cluster y los centroides no se mueven).

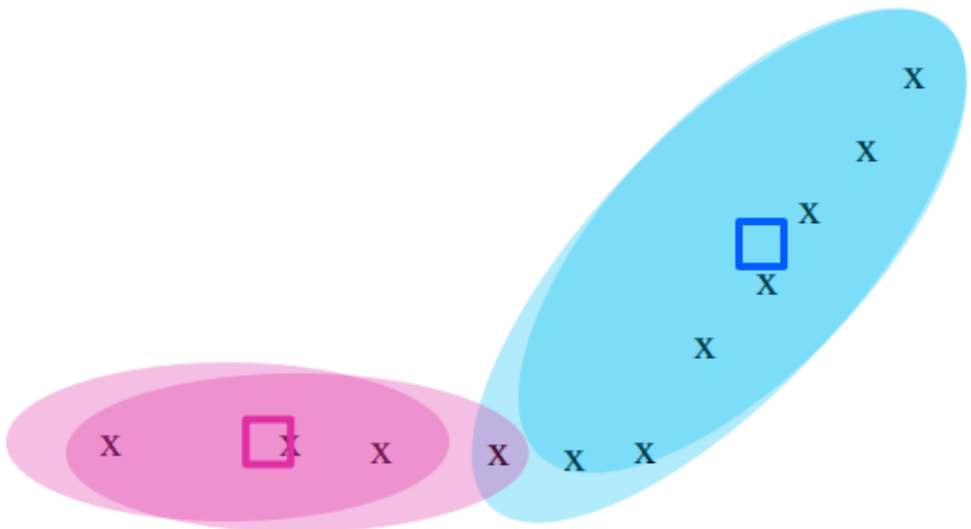
K-medias: paso a paso

Iteracion #1



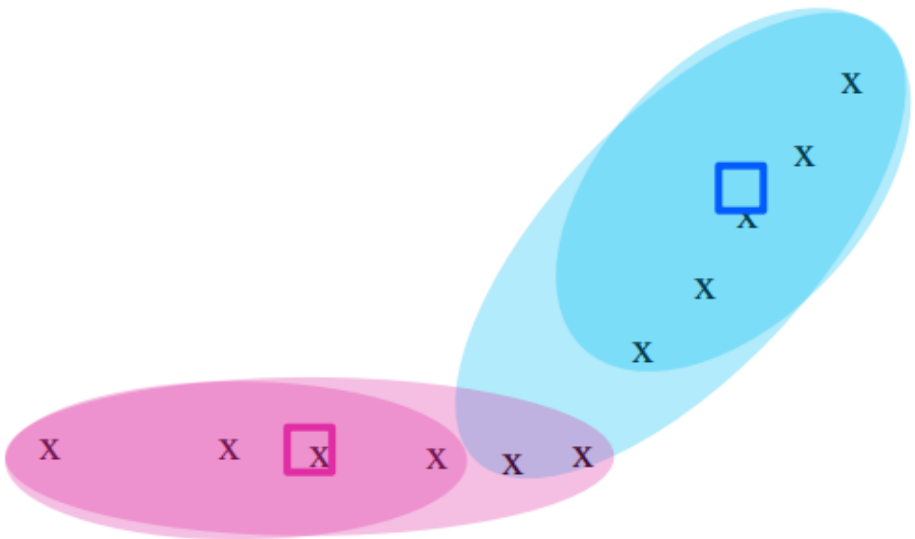
K-medias: paso a paso

Iteracion #2



K-medias: paso a paso

Iteracion #3



K-medias: selección de k

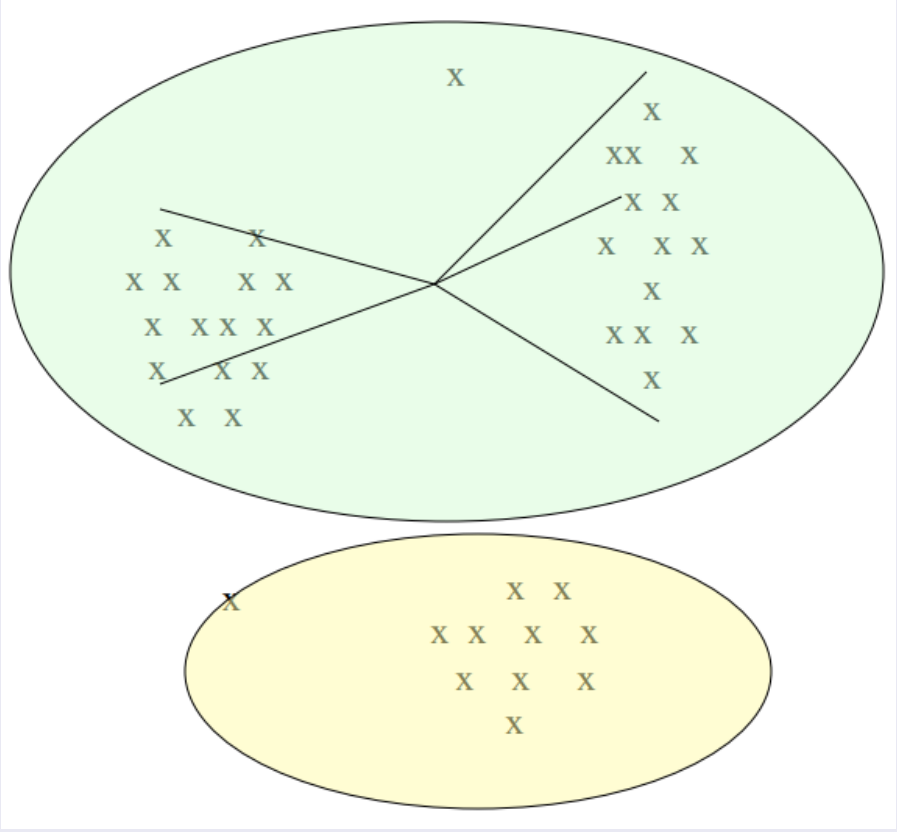
Cómo seleccionar k

- Sabemos *a priori* en cuántas clases se dividen los datos.
- Probar diferentes valores de k registrando el cambio de la distancia media a los centroides a medida que se modifica k.

K-medias: selección de k

$k = 2$

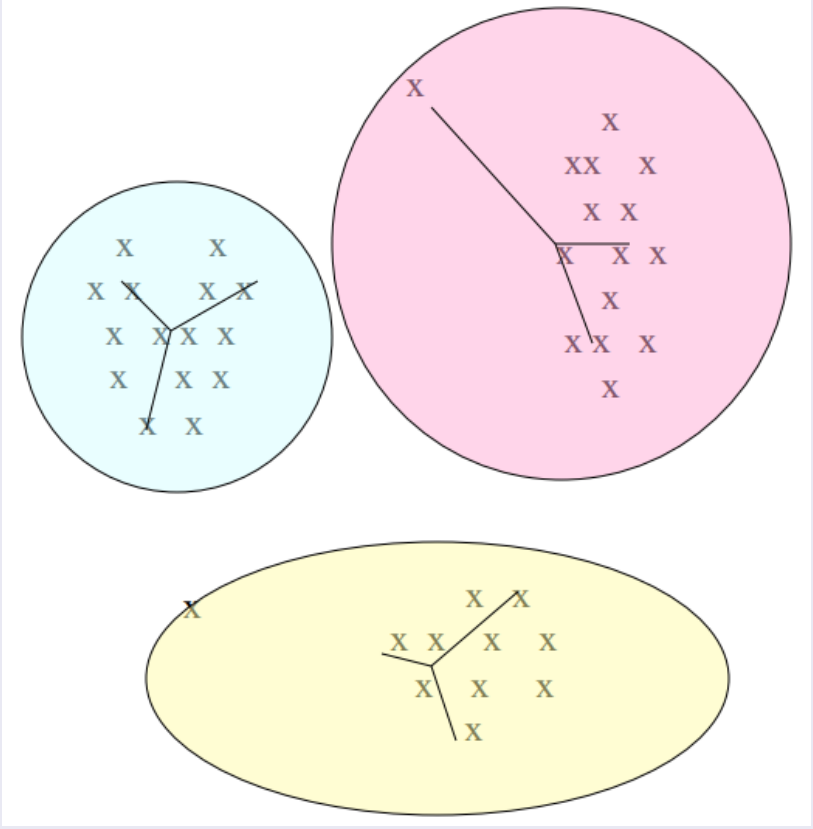
- Pocos centroides, las distancias son grandes.



K-medias: selección de k

$k = 3$

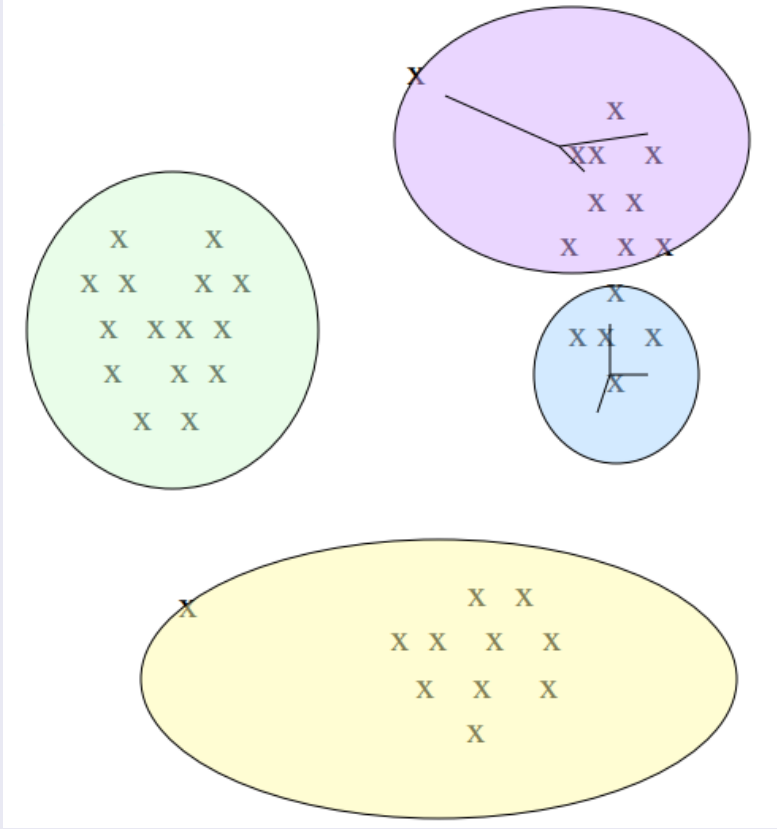
- Tiene buena pinta, los clusters parecen compactos.



K-medias: selección de k

$k = 4$

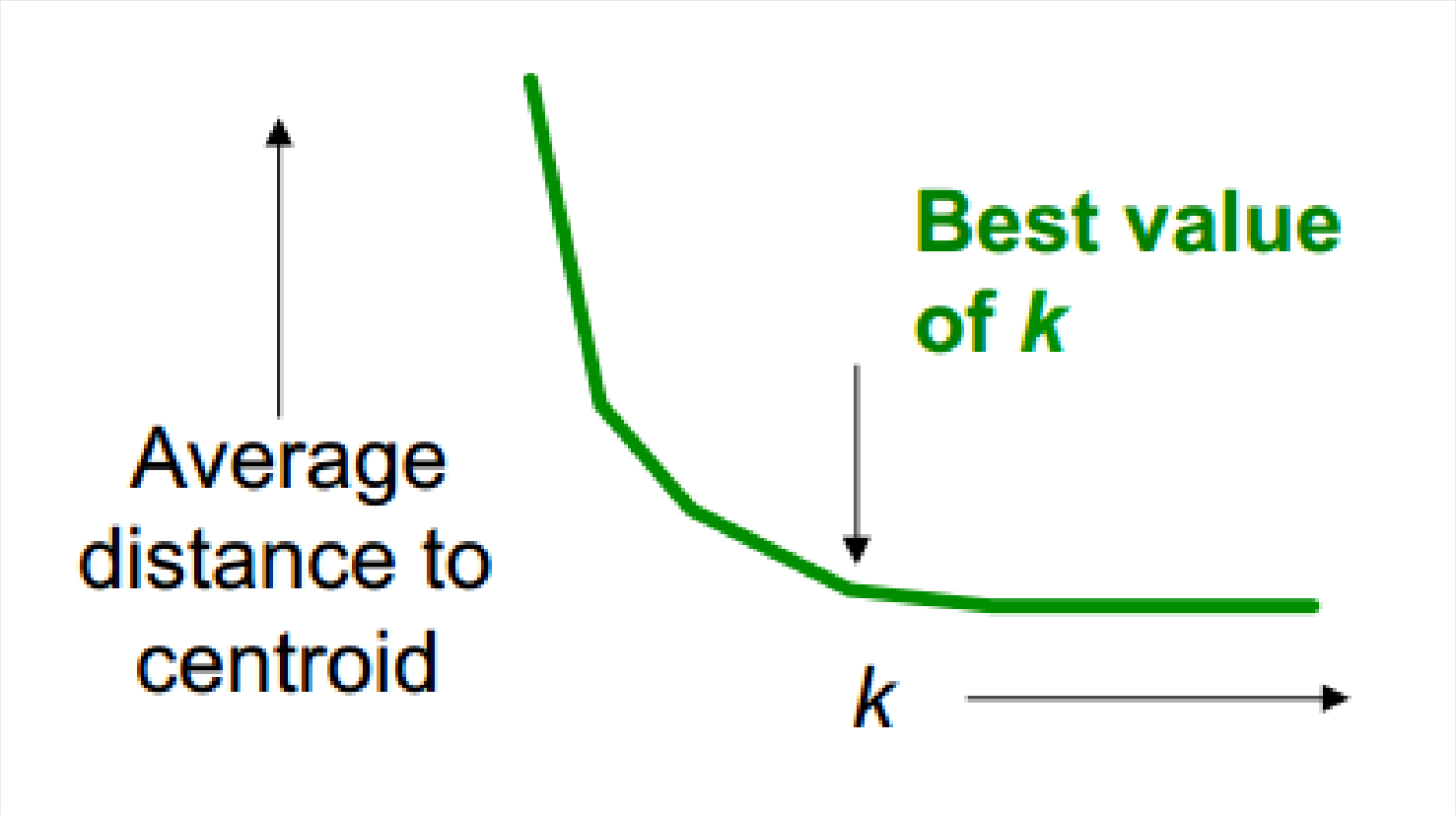
- Demasiados clusters, hemos mejorado poco con respecto a $k = 2$



K-medias: selección de k

Criterio para seleccionar k

- La distancia media a los centroides se estabiliza a medida que aumenta k.



K-medias: inicialización de los clusters

Criterio para seleccionar los k puntos iniciales

1. Opción 1: **Muestreo**
   1. Ejecutar un clustering jerárquico sobre una muestra de los datos para obtener k clusters.
   2. Seleccionar un punto de cada cluster (ej. el más próximo al centroide)
   3. (La muestra entra en memoria)
2. Opción 2: **Dispersión**
   1. Elegir un punto aleatoriamente.
   2. Elegir el siguiente punto de tal forma que la mínima distancia a los *puntos ya seleccionados* sea la máxima posible.
   3. Repetir el proceso hasta tener k puntos.

K-medias: complejidad

Complejidad

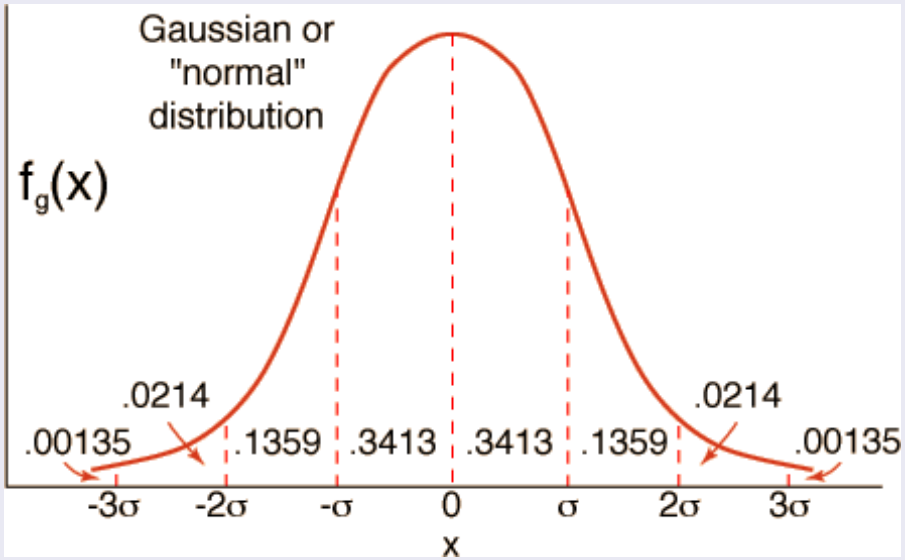
- En cada iteración examinamos cada punto una vez para encontrar el centroide más próximo.
- Cada iteración es $O(kN)$ con N puntos y k clusters.
- . . . pero el número de iteraciones hasta converger puede ser muy elevado.

Algoritmo: BFR

BFR

Algoritmo Bradley-Fayyad-Reina (BFR)

- Es una variante de k-medias para conjuntos de datos muy grandes.
- No es un algoritmo de cluster probabilista puesto que asigna los puntos a un único cluster, aunque puede utilizarse su salida de forma probabilista.
- Asume que cada cluster se distribuye según una normal (gaussian) en torno a un centroide en un espacio euclideo.

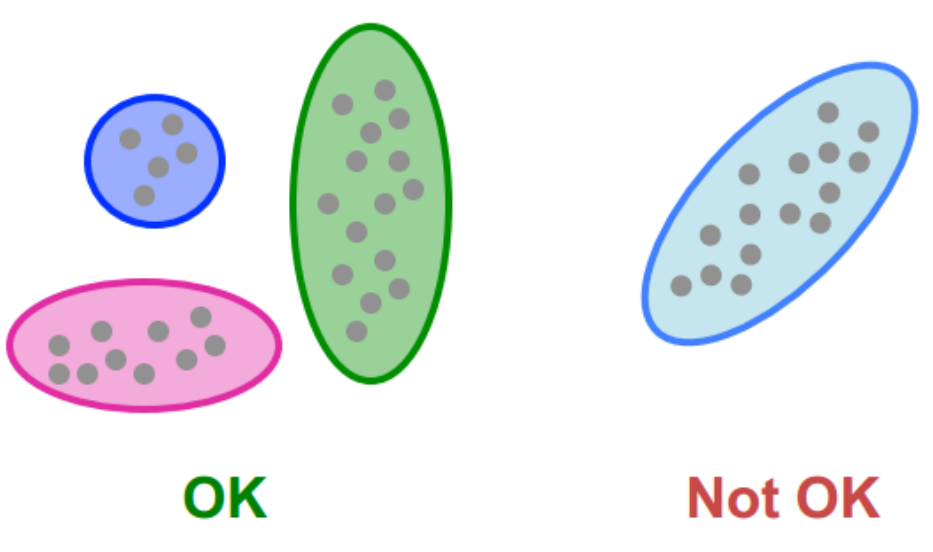


Probabilidad de encontrar un punto en un cluster a cierta distancia (según cada dimensión) de un centroide.

BFR: Pros and cons

Limitaciones

- Asume sólo una distribución normal.
- Las distribuciones están alineadas según los ejes definidos por las dimensiones.



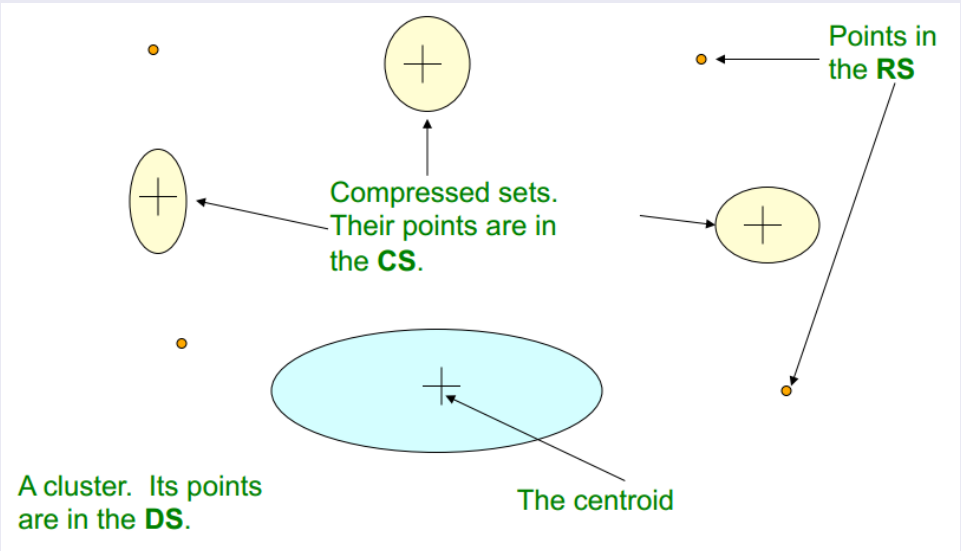
Ventajas

- La mayoría de los puntos se resumen estadísticamente (una única lectura de los datos).

BFR: Tipos de puntos

Tipos de puntos

- A medida que se leen los datos estos son incorporados a un conjunto:
  - **Discard set** (DS): puntos que están suficientemente cerca de un centroide y se incorporar a él.
  - **Compression set** (CS): grupos de puntos que están próximos entre sí, pero no están cerca de ningún centroide. También se resumen en términos estadísticos.
  - **Retained set** (RS): puntos aislados a la espera de ser asignados a un *compression set*.



BFR: Cluster

Cómo resumir un conjunto de puntos

Cada cluster (*discard set*) se resume utilizando las siguientes variables:

- Número de puntos, N
- Vector SUM cuya i-ésima componente se corresponde con la suma de la i-ésima componente de cada punto.
- Vector $SUMSQ$, su i-ésima componente es la suma de los cuadrados de la i-ésima componente de cada punto.

A medida que nuevos puntos son incorporados al cluster, se actualizan estos valores.

Estadísticos

- Centroide: puede calcular como

$$c_i = \frac{SUM_i}{N} \quad i = 1, \ldots, d$$

- Varianza:

$$var_i = \frac{SUMSQ_i}{N} - \frac{SUM_i}{N}^2$$
$$\sigma_i = \sqrt{var_i}$$

BFR: Paso a paso

Para cada *subset* de datos

- Los puntos que están ***"suficientemente cerca"*** de un centroide:
  1. Se añaden al cluster correspondiente.
  2. Se descartan.
- El resto de puntos son tratados por un algoritmo **en memoria**:
  - Los clusters irán al *compression set* (resumido también por sus estadísticos)
  - Los *outliers* se mantienen en el *retained set* (RS)

Última iteración

- Los puntos del *retained set* son asignados al cluster más cercano.
- Considerar la unión de varios *compressed sets*.

BFR: Cómo evaluar el "suficientemente cerca".

Mahalanobis distance

- Si el cluster C tiene como centroide $(c_1, \ldots, c_d)$ y desviación estándar $(\sigma_i, \ldots, \sigma_d)$
- Estamos considerando el punto $P = (x_1, \ldots, x_d)$
- La distancia normalizada según la dimensión i será:

$$d_i = \frac{x_i - c_i}{\sigma_i}$$

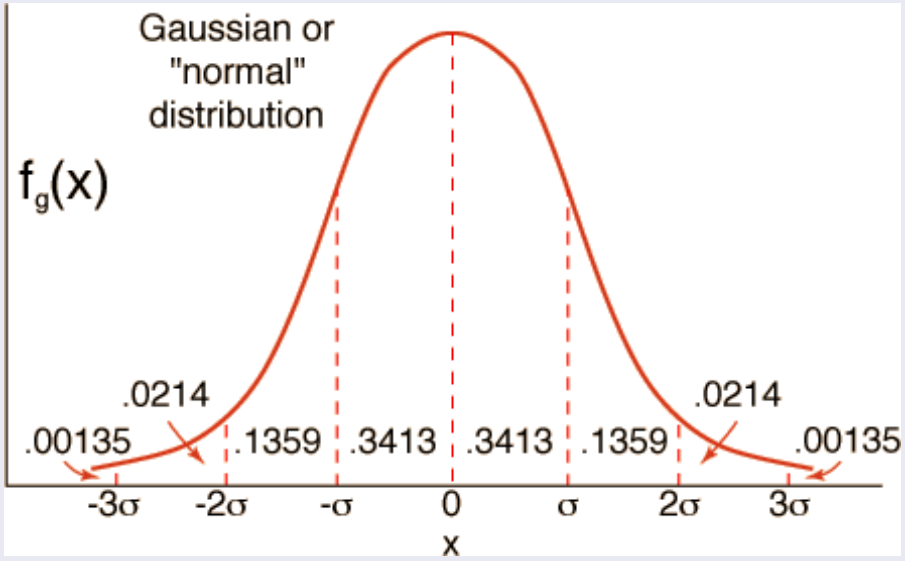
- Distancia Mahalanobis:

$$MD = \sqrt{\sum_{i=1}^{d} y_i^2}$$

BFR: Cómo evaluar el "suficientemente cerca".

Criterio de aceptación Mahalanobis

- Un punto que esté a una *desviación estándar* del centroide en cada dimensión tendrá $MD = \sqrt{d}$.



- Según la distribución normal:
  - 68% de los puntos tienen $MD \leq \sqrt{d}$
  - 95% de los puntos tienen $MD \leq 2\sqrt{d}$
  - 99% de los puntos tienen $MD \leq 3\sqrt{d}$

Aceptar en un cluster todos los puntos cuya MD sera menor que un umbral seleccionado, p.ej. $3\sqrt{d}$

BFR: Cómo decidir si dos CS deben ser combinados

Combinación de dos clusters del CS

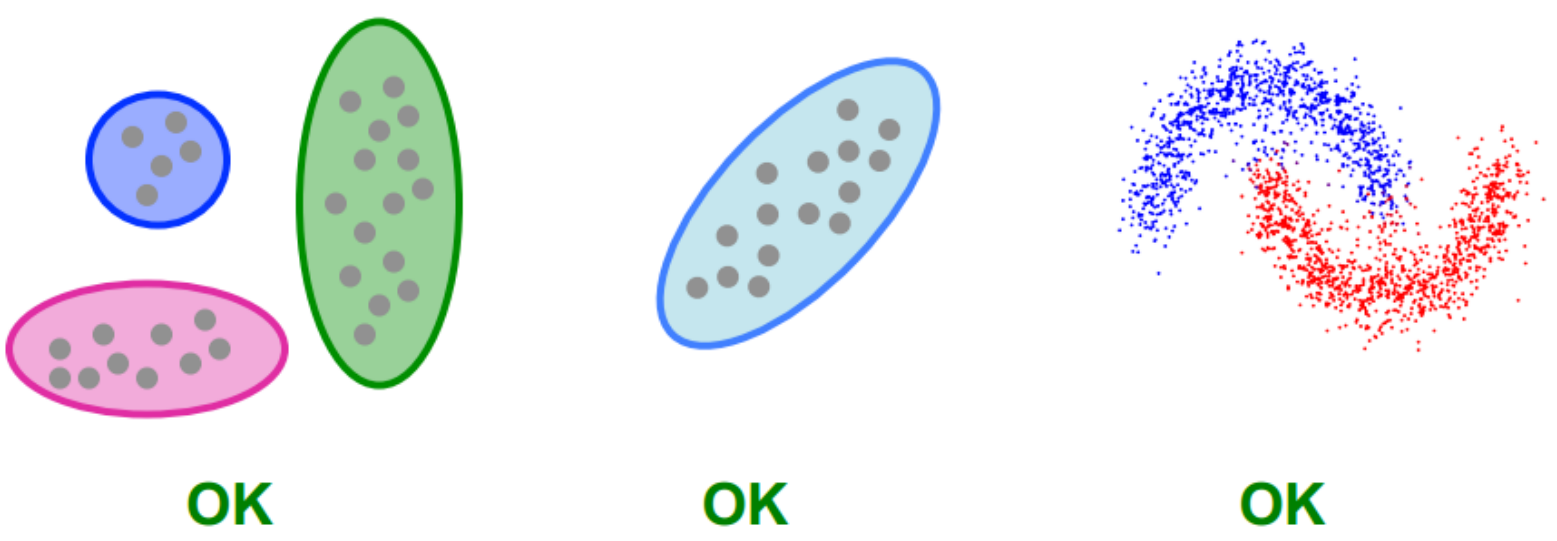
- Calcular la varianza del cluster combinado: ¡muy rápido puesto que tenemos las variables N, SUM y $SUMSQ$ de cada uno!
- Combinarlos si la varianza del cluster combinado está por debajo de un límite.

Algoritmo: CURE

CURE

Algoritmo CURE (Clustering Using REpresentatives)

- Asume distancia **euclidea**.
- Los clusters pueden adoptar cualquier forma.
- Utiliza un conjunto de **puntos representativos** de cada cluster.



CURE: Paso a paso

Primera pasada (de dos)

- Obtener una muestra aleatoria de los datos que entre en memoria.
- Utilizar un algoritmo de cluster jerárquico para generar los clusters iniciales
- Escoger los **puntos representativos**:
  - Dentro de cada cluster escoger r (ej. 4) puntos representativos tan dispersos como sea posible.
  - Mover cada punto un porcentaje (ej. 20%) hacia el centroide del cluster.

CURE: Paso a paso

Segunda pasada (de dos)

- Volver a evaluar cada punto situándolo en el cluster más cercano:
  - Distancia: cluster que tenga un punto representativo más próximo.
- *Et voilá*.

¡Muchas gracias!

@jgsogo

https://github.com/jgsogo/talks