Técnicas de agrupamiento (clustering)

Reconocimiento de Patrones

Departamento de Procesamiento de Señales
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR

2018

- Duda, Hart, Stork. "*Pattern Classification*", capítulo 10.
- Jain, Duin, Mao . "*Statistical Pattern Recognition: A Review*," IEEE-PAMI 2000.
- Jain, "*Data clustering: 50 years beyond K-means*,", Pattern Recognition Letters 2009.

Introducción

Objetivo:
Descubrir estructura dentro de un conjunto de datos $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, dividiéndolo en subconjuntos que muestren una cierta "coherencia".

- "Coherencia": muestras dentro de un mismo grupo o *cluster* son más "parecidas" entre sí que a las muestras de otros clusters.
- nada
- Muestras "parecidas": noción de similitud o de distancia entre muestras.

La mayor parte de los métodos de clustering son de dos tipos:

- **Particionales**: producen una única partición, $\mathcal{D}_1, \ldots, \mathcal{D}_c$, que optimiza una función criterio
- **Jerárquicos**: jerarquía de particiones "anidadas"; cada nivel de la jerarquía es en si mismo una partición, obtenida por unión de clusters de la jerarquía inferior.

OBS: Un método de clustering siempre produce clusters, aunque éstos no existan realmente $\Rightarrow$ todo método de clustering debe ser seguido de una **etapa de validación** de los clusters obtenidos.

1. Medidas de similitud entre patrones

- Para encontrar los agrupamientos naturales, la noción de similitud debe ser adaptada al problema particular. La elección de una medida adecuada no es trivial.
- No tienen porqué ser distancias.

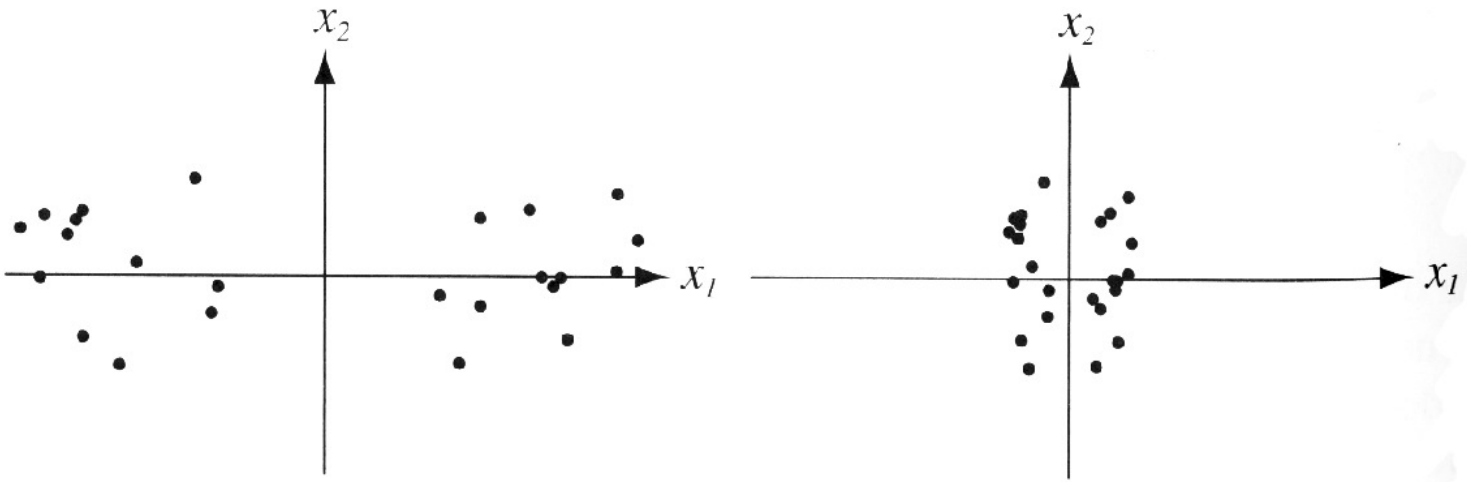
Ejemplos:

- Distancias de Minkowski (normas ℓ^p),

$$d(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^{d} |x_i - x_i'|^p \right)^{1/p}, \ p \geq 1.$$

  - $p = 2$:Euclídea; $p = 1$: "Manhattan"; $p = \infty$: $d(\mathbf{x}, \mathbf{x}') = \max_i |x_i - x_i'|$
  - Solo válidas para espacios de características suficientemente isotópicos (distribución similar en todas las direcciones).
  - Correlaciones entre características pueden distorsionar estas distancias; es común normalizar los datos previamente (transformación de blanqueado). Si $p = 2$, esto es usar distancia de Mahalanobis. Ojo:sólo vale para datos con distribución normal; catastrófico si se aplica a distribuciones multimodales.

Medidas de similitud (2)



Efecto de estandarizar datos no unimodales

- Ejemplo de función de similitud que no es una distancia: $d(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'/\|\mathbf{x}\|\|\mathbf{x}'\|$.
- Error frecuente: definir métricas como normas en espacios de características que no son espacios vectoriales (ejemplo: los parámetros de una transformación afin).

2. Métodos Particionales

Objetivo: dado un conjunto de patrones $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ y una medida de similitud entre patrones, identificar una partición $\mathcal{D}_1, \ldots, \mathcal{D}_c$ que optimice una cierta función criterio.

Aproximadamente $c^n/c!$ particiones posibles (si $n = 100$ y $c = 5$, esto es 10^{67})

$\Rightarrow$ Optimización del criterio por búsqueda exhaustiva fuera de consideración

$\Rightarrow$ Utilización de métodos iterativos, aunque éstos no siempre garanticen convergencia a óptimos globales.

2.1 Funciones criterio (2)

Evalúan la calidad de una partición. De la elección del criterio dependerá la forma de los clusters que se obtienen al optimizarlo.

Criterio SSE (sum of squared errors), criterios de mínima varianza

$$SSE = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathcal{D}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2, \text{ con } \boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}, \; n_i = \#\mathcal{D}_i.$$

- Medida de las varianzas intra-cluster de la partición. Estrictamente, sólo tiene sentido cuando los clusters presentes en los datos son isotrópicos, de distribución normal multivaluada.
- Forma parte de una familia de criterios llamados de mínima varianza.

2.1 Funciones criterio (3)

Observando que el SSE se escribe $SSE = \sum_{i=1}^{c} n_i \bar{d}_i$ con

$$\bar{d}_i = \frac{1}{2n_i^2} \sum_{\mathbf{x} \in \mathcal{D}_i} \sum_{\mathbf{x}' \in \mathcal{D}_i} \|\mathbf{x} - \mathbf{x}'\|_2^2,$$

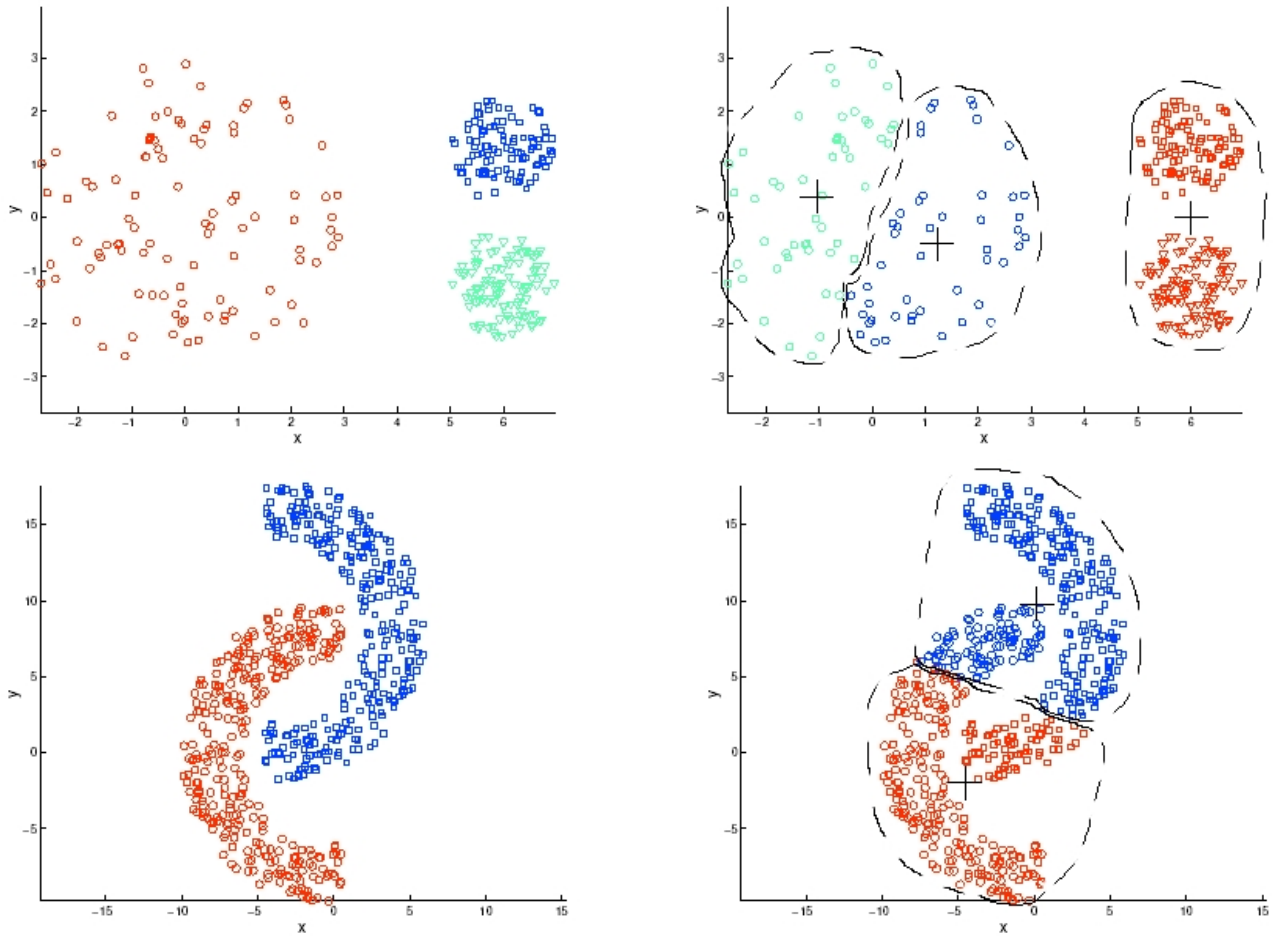
podemos utilizar en $\bar{d}_i$ cualquier otra medida de similitud:

$$\bar{d}_i = \frac{1}{2n_i^2} \sum_{\mathbf{x} \in \mathcal{D}_i} \sum_{\mathbf{x}' \in \mathcal{D}_i} d(\mathbf{x}, \mathbf{x}').$$

Inconvenientes de los criterios de mínima varianza

- Favorecen soluciones que dividen clusters grandes.
- Si la cantidad de puntos en los distintos clusters es muy dispar, no permiten alcanzar soluciones que revelan la estructura intrínseca de los datos.

2.1 Funciones criterio (4)



Resultado de minimizar un criterio de mínima varianza mediante k-means

2.2 Optimización iterativa de la función criterio

El algoritmo k-means

- Es uno de los métodos usados.
- Consiste en una minimización iterativa del SSE
- Algoritmo:
  1. Elegir una partición inicial en c grupos al azar
  2. Calcular las medias $\boldsymbol{\mu}_i$ de cada cluster
  3. Seleccionar secuencialmente un punto $\mathbf{x} \in \mathcal{D}$, y si corresponde, reasignarlo al cluster que minimiza $\|\mathbf{x} - \boldsymbol{\mu}_i\|_2$
  4. Si no hay más reasignaciones en todo $\mathcal{D}$, terminar; si no, volver al paso 2.

2.2 Optimización iterativa de la función criterio (2)

Este algoritmo busca minimizar en cada iteración la SSE:
$J_e = \sum_{i=1}^{c} J_i, \quad J_i = \sum_{\mathbf{x}\in\mathcal{D}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2.$

Veamos que pasa cuando una muestra $\hat{\mathbf{x}}$ pasa del cluster $\mathcal{D}_i$ al $\mathcal{D}_j$:

$$\boldsymbol{\mu}_j^{\text{new}} = \frac{n_j\boldsymbol{\mu}_j + \hat{\mathbf{x}}}{n_j+1} = \frac{(n_j+1)\boldsymbol{\mu}_j + \hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1} = \boldsymbol{\mu}_j + \frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1}.$$

$$\begin{aligned}
J_j^{\text{new}} &= \sum_{\mathbf{x}\in\mathcal{D}_j} \|\mathbf{x} - \boldsymbol{\mu}_j^{\text{new}}\|_2^2 + \|\hat{\mathbf{x}} - \boldsymbol{\mu}_j^{\text{new}}\|_2^2 \\
&= \sum_{\mathbf{x}\in\mathcal{D}_j} \left\|\mathbf{x} - \boldsymbol{\mu}_j - \frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1}\right\|_2^2 + \left\|\hat{\mathbf{x}} - \boldsymbol{\mu}_j - \frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1}\right\|_2^2 \\
&= \sum_{\mathbf{x}\in\mathcal{D}_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|_2^2 - 2\left(\frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1}\right)^{\mathrm{T}} \underbrace{\sum_{\mathbf{x}\in\mathcal{D}_j} (\mathbf{x} - \boldsymbol{\mu}_j)}_{0} + n_j \left\|\frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_j}{n_j+1}\right\|_2^2 + \left\|\frac{n_j(\hat{\mathbf{x}} - \boldsymbol{\mu}_j)}{n_j+1}\right\|_2^2 \\
&= J_j + \frac{n_j}{n_j+1}\|\hat{\mathbf{x}} - \boldsymbol{\mu}_j\|_2^2
\end{aligned}$$

2.2 Optimización iterativa de la función criterio (3)

La misma manipulación sobre el cluster $\mathcal{D}_i$ conduce a:

$$\boldsymbol{\mu}_i^{\text{new}} = \boldsymbol{\mu}_i - \frac{\hat{\mathbf{x}} - \boldsymbol{\mu}_i}{n_i + 1}.$$
$$J_i^{\text{new}} = J_i - \frac{n_i}{n_i - 1}\|\hat{\mathbf{x}} - \boldsymbol{\mu}_i\|_2^2$$

Tenemos

$$J_i^{\text{new}} + J_j^{\text{new}} < J_i + J_j \Leftrightarrow \frac{n_j}{n_j + 1}\|\hat{\mathbf{x}} - \boldsymbol{\mu}_j\|_2^2 < \frac{n_i}{n_i - 1}\|\hat{\mathbf{x}} - \boldsymbol{\mu}_i\|_2^2.$$

La reasignación que más hace decrecer J_e es aquíella que minimiza

$$\frac{n_j}{n_j + 1}\|\hat{\mathbf{x}} - \boldsymbol{\mu}_j\|_2^2.$$

2.2 Optimización iterativa de la función criterio (4)

Variantes del algoritm k-means

Fuzzy k-means

- El algoritmo $k-$means asume que cada punto pertenece a un úúnico cluster (la probabilidad a posteriori de clase vale $\hat{P}(\omega_i|\mathbf{x}_j) = 1$ para el cluster al cual se asigna $\mathbf{x}_j$).
- Fuzzy k-means relaja esta condición asignando a cada muestra $\mathbf{x}_j$, un grado de membresía *fuzzy* a cada cluster, $\hat{P}(\omega_i|\mathbf{x}_j)$. Estos parámetros verifican $\sum_{i=1}^{c} \hat{P}(\omega_i|\mathbf{x}_j) = 1, \forall\ j = 1, .., n.$
- La función objetivo es ahora

$$J_{\text{fuzzy}} = \sum_{i=1}^{c} \sum_{j=1}^{n} \left(\hat{P}(\omega_i|\mathbf{x}_j)\right)^b \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2,$$

donde b ajusta el grado de mezcla de lo distintos clusters ($b = 0$ corresponde al k-means clásico).

2.2 Optimización iterativa de la función criterio (5)

(Ejercicio) Se puede ver que J_{fuzzy} es mínima (anulando el gradiente del Lagrangeano) para

$$\boldsymbol{\mu}_j = \frac{\sum_{j=1}^{n} \left(\hat{P}(\omega_i|\mathbf{x}_j)\right)^b \mathbf{x}_j}{\left(\hat{P}(\omega_i|\mathbf{x}_j)\right)^b}, \quad \hat{P}(\omega_i|\mathbf{x}_j) = \frac{1}{\sum_{r=1}^{c} \left(\frac{\|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2}{\|\mathbf{x}_j - \boldsymbol{\mu}_r\|_2^2}\right)^{1/(b-1)}}$$

El algoritmo es similar al k-means:

1. Inicialización de los $\hat{P}(\omega_i|\mathbf{x}_j)$ (con normalización)
2. Cálculo de los $\boldsymbol{\mu}_j$ con la ecuación de arriba
3. Actualización de los $\hat{P}(\omega_i|\mathbf{x}_j)$ con la ecuación de arriba
4. Si los $\hat{P}(\omega_i|\mathbf{x}_j)$ y los $\boldsymbol{\mu_j}$ varían menos que un cierto umbral con respecto a la iteración anterior, salir. Si no, volver al paso 2.

Obs: en general mejor que el k-means si la cantidad de clases c es la real; si no, la performance se degrada mucho (las membresías dependen implícitamente de c).

2.2 Optimización iterativa de la función criterio (6)

k-medians

- Variante del k-means, más robusto a outliers (basado en medianas en lugar de medias).
- Minimización iterativa de $\sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathcal{D}_i} \|\mathbf{x} - \mathbf{m}_i\|_1$, donde la coord. k-ésima de $\mathbf{m}_i$ es la mediana en la dirección k de los $\mathbf{x} \in \mathcal{D}_i$.

El algoritmo es igual al k-means, con los cambios siguientes:

- Paso 2: se calculan los $\mathbf{m}_i$ en lugar de los $\boldsymbol{\mu}_i$
- Paso 3: $\mathbf{x}$ se asigna al cluster que minimiza la $\|\mathbf{x} - \mathbf{m}_i\|_1$.

2.3 Elección del número de clusters

Inconvenientes de los métodos particionales:

1. Pueden ser sensibles a la inicialización. Como son algoritmos rápidos, se resuelve corriendo el algoritmo para varias inicializaciones y tomando la mejor solución (valor de función criterio mínimo).
2. La cantidad de clusters es una entrada de los algoritmos, y es difícil de establecer.

Guías para seleccionar el número de clusters

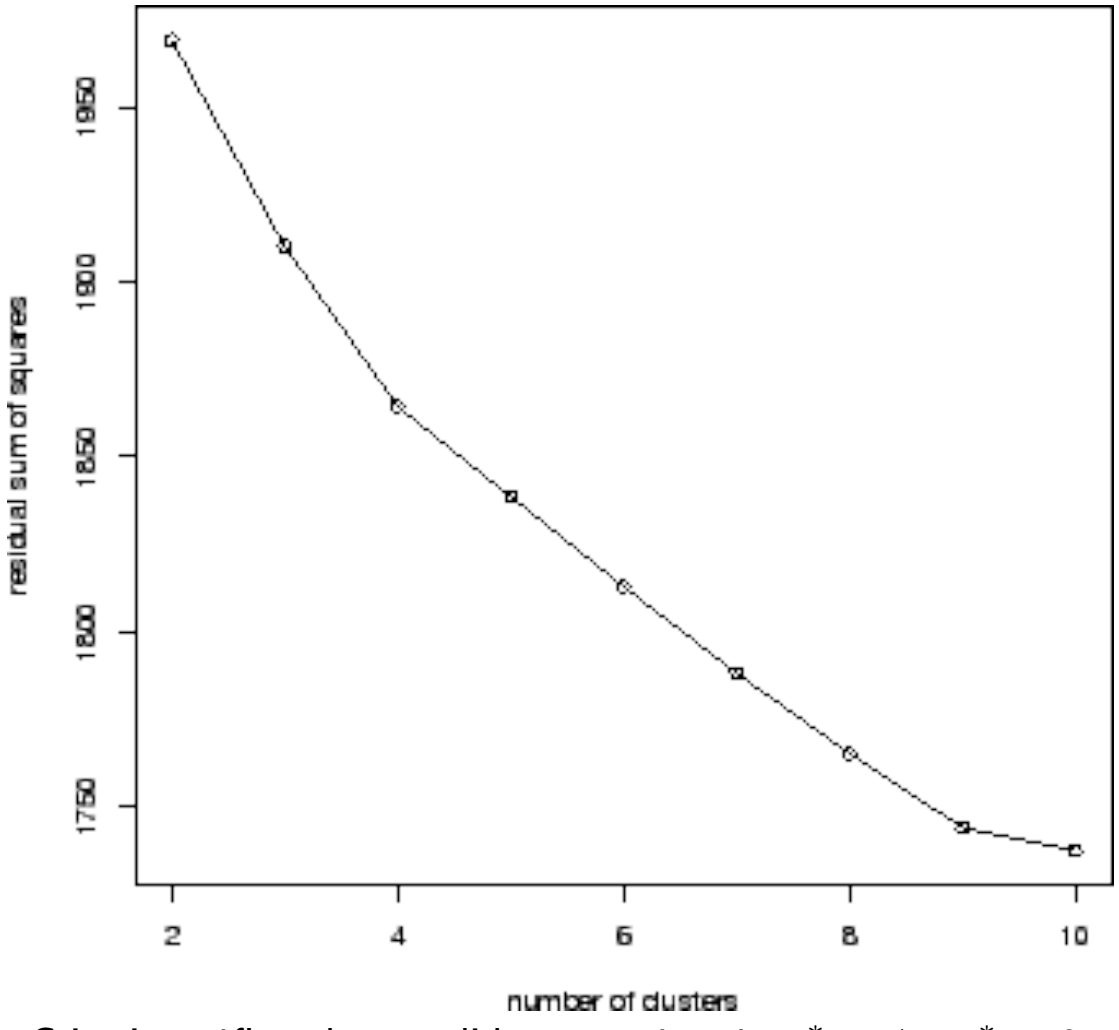
- Criterio gráfico

  Intuitivamente: elegir una cantidad de clusters tal que agregar más clusters no aporta información, i.e. no explica "significativamente mejor" la varianza.

  Para cada número de clusters $c = 1, \ldots, C$, se minimiza (repitiendo para varias inicializaciones) la función criterio.

  Esto corresponde a cantidades de clusters c^* en la gráfica *num. de clusters vs. valor de función criterio* donde el criterio baja poco con respecto a $c^* - 1$.

2.3 Elección del número de clusters (2)



Criterio gráfico: los candidatos aquí serán $c^* = 4$ y $c^* = 9$.

2.3 Elección del número de clusters (3)

- Criterios AIC, BIC, MDL

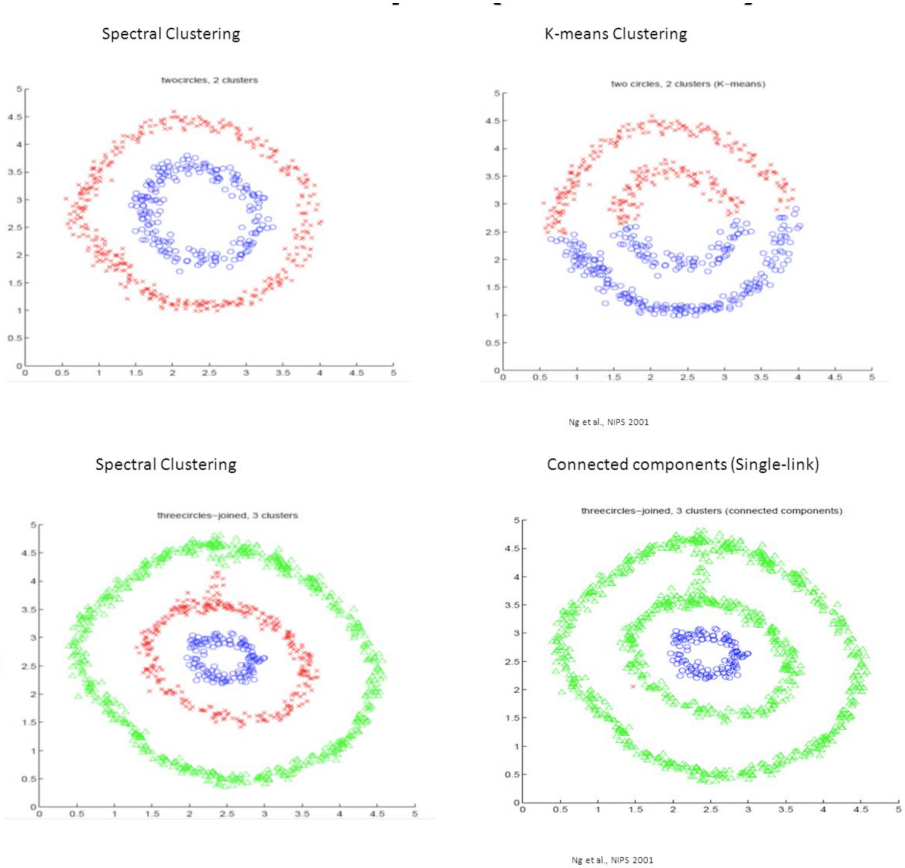
Ejemplo: adaptación de AIC al clustering.

Criterio AIC: minimizar $-\text{logverosimilitud} + m_c$.
Si los clusters son gaussianos, con mismas matrices de covarianza y con priors uniformes, la log-verosimilitud es proporcional a $-SSE(c)/2$, y $m_c = c \times d$.
$\Rightarrow$ AIC: minimizar $SSE(c) + 2 \times c \times d$.

Más general: minimizar Función-Criterio-mínimo$(c) + \lambda \times c \times d$.
El parámetro λ es un parámetro de escala.

2.4. Otros métodos particionales populares

- Self Organizing Maps
- Métodos basados en kernels (e.g. Kernel K-Means)
- Métodos basados grafos. Spectral clustering (algoritmos de Shi & Malik, y de Ng, Jordan & Weiss)
- Non-negative Matrix Factorization



3. Métodos jerárquicos

- No compiten necesariamente con los métodos particionales, ya que representan a los datos de manera diferente.
- Construyen jerarquías de particiones; son una representación natural cuando la descripción de los datos debe ser en términos de clases, subclases, subsubclases (e.g. una taxonomía biológica).
- En aplicaciones en las que los datos no son inherentemente jerárquicos, uno debe elegir entre métodos de ambos tipos.
- Son más versátiles que los métodos particionales y pueden lidiar con clusters de formas variadas, pero son computacionalmente más complejos (típicamente $O(n^2 \log n)$ vs. $O(n)$ para el k-means).

3. Métodos Jerárquicos (2)

Pueden ser **aglomerativos** (*bottom-up*) o **divisivos** (*top-down*). Los primeros son computacionalmente más simples. Comienzan con cada punto como un cluster, y van uniendo iterativamente los pares de clusters más cercanos según alguna medida de similitud.

Algoritmo: clustering jerárquico aglomerativo

1. Inicialización: calcular la matriz de proximidad (la matriz con las medidas de similitud entre todos los pares de patrones)
2. Encontrar el par de clusters más similares usando la matriz de proximidad. Unir ese par de clusters.
3. Actualizar la matriz de proximidad de acuerdo a esta unión de clusters
4. Repetir los pasos 2 y 3 hasta que todos los patrones estén en un único cluster.

3. Métodos Jerárquicos (3)

- En cada iteración se unen dos clusters. El resultado es un árbol o *dendrograma*; las hojas son las muestras originales.
- El nivel l del árbol contiene $n - l$ nodos, cada uno correspondiente a un cluster
- El nivel $l + 1$ se obtiene del nivel l, uniendo los dos cluster más cercanos
- Clusters "más cercanos": involucra medida de proximidad entre clusters, $\delta(\mathcal{C}_i, \mathcal{C}_j)$
- Diferentes estrategias para actualizar la matriz de proximidad conducen a diferentes métodos de clustering.

3.1. Proximidad entre clusters

Fórmula de Lance y Williams

Tenemos tres clusters $\mathcal{C}_i$, $\mathcal{C}_j$ y $\mathcal{C}_k$, y los dos primeros se unen por ser el par de menor $\delta(\cdot,\cdot)$. Entonces, suponiendo que $\delta(\mathcal{C}_j,\mathcal{C}_k) < \delta(\mathcal{C}_i,\mathcal{C}_k)$:

$$\delta(\mathcal{C}_i \cup \mathcal{C}_j, \mathcal{C}_k) = \alpha_i \delta(\mathcal{C}_i,\mathcal{C}_k) + \alpha_j \delta(\mathcal{C}_j,\mathcal{C}_k) + \beta \delta(\mathcal{C}_i,\mathcal{C}_j) + \gamma \left[\delta(\mathcal{C}_i,\mathcal{C}_k) - \delta(\mathcal{C}_j,\mathcal{C}_k)\right].$$

Todo método de clustering que se puede expresar mediante esta fórmula no requiere guardar la matriz de proximidad original entre patrones.

Métodos más usados:

– *Single-link*: $\alpha_i = \alpha_j = \frac{1}{2}$, $\beta = 0$, $\gamma = -\frac{1}{2}$, $\quad \delta(\mathcal{C}_p,\mathcal{C}_q) = \min_{\mathbf{x}\in\mathcal{C}_p,\mathbf{x}'\in\mathcal{C}_q} d(\mathbf{x},\mathbf{x}')$.

– *Complete-link*: $\alpha_i = \alpha_j = \frac{1}{2}$, $\beta = 0$, $\gamma = \frac{1}{2}$, $\quad \delta(\mathcal{C}_p,\mathcal{C}_q) = \max_{\mathbf{x}\in\mathcal{C}_p,\mathbf{x}'\in\mathcal{C}_q} d(\mathbf{x},\mathbf{x}')$.

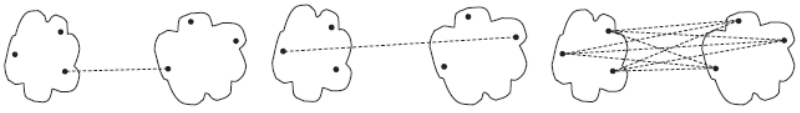
– *Promedio*: $\alpha_i = \frac{n_i}{(n_i+n_j)}$, $\alpha_j = \frac{n_j}{(n_i+n_j)}$, $\beta = \gamma = 0$,

$$\delta(\mathcal{C}_p,\mathcal{C}_q) = \frac{1}{n_p n_q} \sum_{\mathbf{x}\in\mathcal{C}_p} \sum_{\mathbf{x}'\in\mathcal{C}_q} d(\mathbf{x},\mathbf{x}').$$

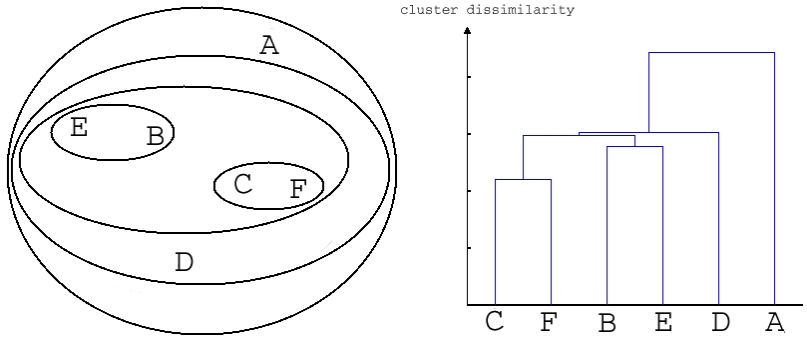
– *Mínima varianza (Ward)*: $\alpha_i = \frac{(n_i+n_k)}{(n_i+n_j+n_k)}$, $\alpha_j = \frac{(n_j+n_k)}{(n_i+n_j+n_k)}$, $\beta = -\frac{n_k}{(n_i+n_j+n_k)}$, $\gamma = 0$,

$$\delta(\mathcal{C}_p,\mathcal{C}_q) = \frac{n_p n_q}{n_p+n_q} \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\|_2^2.$$

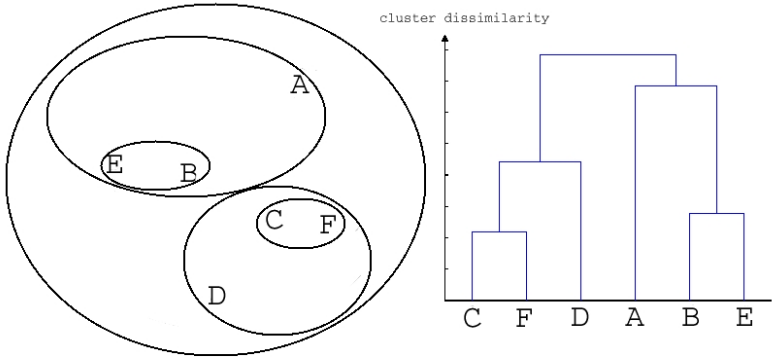
3.2. Métodos de clustering jerárquico más usados



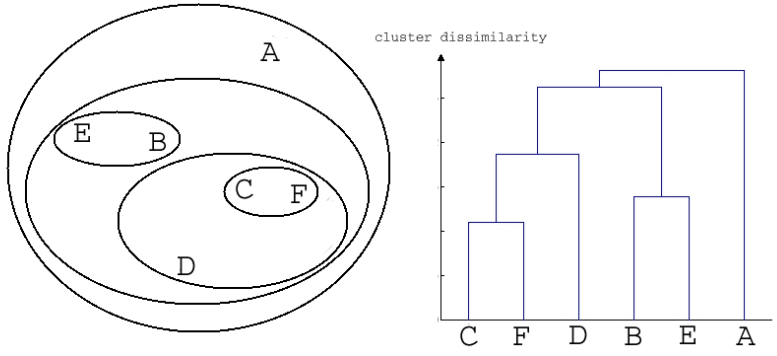
(a)MIN(singlelink.)(b)MAX(completelink.)(c)Groupaverage.



Single linkeage



Complete linkage



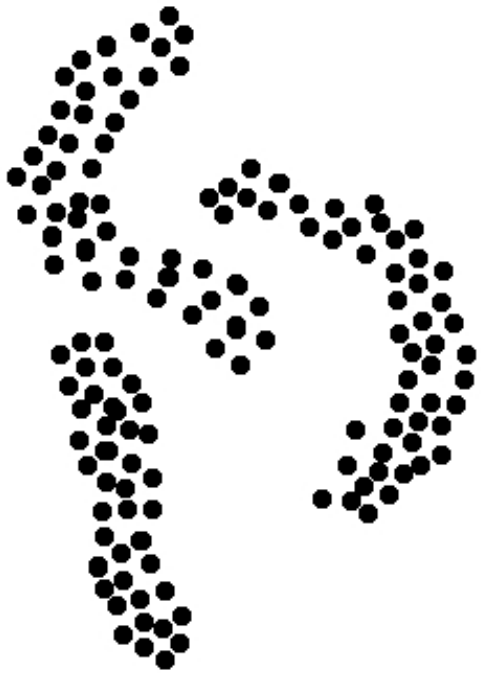
Promedio

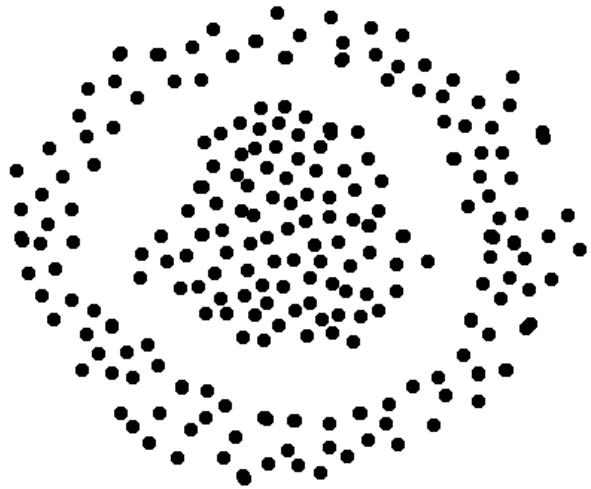
3.2. Métodos de clustering jerárquico más usados (2)

Observaciones

- Si los clusters son compactos y etán bien separados, todos los métodos dan el mismo resultado.
- Single-linkage: bueno para detectar clusters alargados (como contraparte, sufre del “chainning effect")
- Complete-linkage: Produce clusters compactos, de diámetro chico. Los patrones asignados a un cluster pueden estar más cerca de patrones en otros clusters que de patrones del mismo cluster.
- Promedio, Ward: un compromiso. Menos sensibles a outliers que los otros dos (que se basan en valores extremos).

3.2. Métodos de clustering jerárquico más usados (3)





3.3. Métodos Jerárquicos: comentarios generales

- Los métodos jerárquicos son buenos tomando decisiones locales sobre la unión de clusters, ya que usan directamente la matriz de proximidades.
- Pueden ser demasiado locales; una vez que la decisión de unir dos clusters fue tomada, no hay marcha atrás en la construcción de la jerarquía.
- Esta diferencia en la estrategia de minimización de la energía con respecto a métodos particionales conduce a resultados distintos, aún cuando la energía a minimizar sea la mismo.
- Ejemplo: el método de Ward es de mínima varianza, pero una c partición obtenida por este método puede no conducir a la misma solución que un k-means con c clusters (a menos que los clusters sean compactos y bien separados).

3.4. Criterios de parada, validación de particiones

Particiones anidadas $\Rightarrow$ La cantidad óptima de clusters puede ser determinada mediante reglas de parada del proceso de aglomeración.

Validación de partición

- Dos aspectos distintos de validación: validación de clusters individuales y validación de un partición.
- Una parte importante de la validación es la determinación del número de clusters c. Esto no asegura que los c clusters sean válidos.

Reglas de parada para determinar el número de clusters

- Reglas globales: se optimiza una energía $E(c)$, basada en varianzas intra-cluster e inter-cluster, para $c = 1, \ldots, c_{max}$.
- Reglas locales: en el árbol, se decide caso a caso sobre la unión de dos clusters en uno.

3.4. Criterios de parada, validación de particiones (2)

Ejemplos:

- Regla de Calinski y Harabasz.

  Regla global. Se elige la partición de tamaño c del dendrograma que minimiza el cociente de las varianzas intra e inter-cluster:

$$E(c) = \frac{\frac{1}{c-1}\sum_{i=1}^{c}\sum_{\mathbf{x}\in\mathcal{D}_i}\|\mathbf{x}-\boldsymbol{\mu}_i\|_2^2}{\frac{1}{n-c}\sum_{i=1}^{c} n_i\,\|\boldsymbol{\mu}_i-\boldsymbol{\mu}\|_2^2}$$

  Tiende a partir los datos en clusters hiperesféricos, con cantidades de puntos parecidas (es un criterio tipo SSE).

3.4. Criterios de parada, validación de particiones (2)

- Regla "$J_e(2)/J_e(1)$" de Duda y Hart.

  Regla local, basada en test de hipótesis para clusters isotrópicos con distribución normal multivariada (la idea se podría aplicar a otro tipo de distribuciones).

  - La hipótesis nula: en un cluster, todas las muestras son i.i.d $\sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$.
  - El cluster $\mathcal{D}_i$ surge de la unión $\mathcal{D}_j \cup \mathcal{D}_k$.
  - Se define $J_e(1) = \sum_{\mathbf{x}\in\mathcal{D}_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2$, $J_e(2) = \sum_{l\in\{j,k\}} \sum_{\mathbf{x}\in\mathcal{D}_l} \|\mathbf{x} - \boldsymbol{\mu}_l\|_2^2$.
  - Se rechaza la hipótesis nula al nivel de confianza p% si

$$\frac{J_e(2)}{J_e(1)} < 1 - \frac{2}{\pi d} - \alpha\sqrt{\frac{2(1 - 8/\pi^2 d)}{nd}},$$

  con $p = \frac{100}{\sqrt{2\pi}} \int_\alpha^\infty \exp(-u^2/2)du$.

Ver libro Duda, Hart & Stork para la deducción de este criterio.

3.4. Criterios de parada, validación de particiones (3)

- Bayesian Hierarchical Clustering

  [Heller, Ghahramani. Bayesian Hierarchical Clustering. ICML, 2005]

$\mathcal{D}_i$ (resp. $\mathcal{D}_j$) $\subset \mathcal{D}$ conjunto de todas las hojas del subárbol T_i (resp. T_j).
T_k subárbol que surge de la fusión de T_i y T_j, $\mathcal{D}_k = \mathcal{D}_i \cup \mathcal{D}_j$.

Hipótesis $\mathcal{H}_1^k$: "Todos los puntos de $\mathcal{D}_k$ son realizaciones independientes del mismo modelo de probabilidad $p(\mathbf{x}|\theta)$".
Hipótesis $\mathcal{H}_2^k$: "$\mathcal{D}_k$ tiene dos clusters, consistentes con T_i y T_j".

$$p(\mathcal{D}_k|\mathcal{H}_1^k) = \int p(\mathcal{D}_k|\theta)p(\theta|\beta)d\theta = \int \left[\prod_{\mathbf{x}_k \in \mathcal{D}_k} p(\mathbf{x}_k|\theta)\right] p(\theta|\beta)d\theta.$$

$$p(\mathcal{D}_k|\mathcal{H}_2^k) = p(\mathcal{D}_i|T_i)p(\mathcal{D}_j|T_j)$$

$$\pi_k := p(\mathcal{H}_1^k)$$
$$p(\mathcal{D}_k|T_k) = \pi_k p(\mathcal{D}_k|\mathcal{H}_1^k) + (1-\pi_k)p(\mathcal{D}_k|\mathcal{H}_2^k)$$

$$r_k := p(\mathcal{H}_1^k|\mathcal{D}_k) = \frac{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k)}{p(\mathcal{D}_k|T_k)}$$

3.4. Criterios de parada, validación de particiones (3)

- Bayesian Hierarchical Clustering (cont.)

input: data $\mathcal{D} = \{\mathbf{x}^{(1)} \dots \mathbf{x}^{(n)}\}$, model $p(\mathbf{x}|\theta)$, prior $p(\theta|\beta)$
initialize: number of clusters $c = n$, and $\mathcal{D}_i = \{\mathbf{x}^{(i)}\}$ for $i = 1 \dots n$
while $c > 1$ **do**
Find the pair $\mathcal{D}_i$ and $\mathcal{D}_j$ with the highest probability of the merged hypothesis:

$$r_k = \frac{\pi_k p(\mathcal{D}_k|\mathcal{H}_1^k)}{p(\mathcal{D}_k|T_k)}$$

Merge $\mathcal{D}_k \leftarrow \mathcal{D}_i \cup \mathcal{D}_j$, $T_k \leftarrow (T_i, T_j)$
Delete D_i and D_j, $c \leftarrow c - 1$
end while
output: Bayesian mixture model where each tree node is a mixture component
The tree can be cut at points where $r_k < 0.5$

3.5. Detección de clusters en ruido (validación individual de clusters)

Detección *a contrario* de clusters

[Cao, Delon, Desolneux, Musé, Sur. A Unified Framework for Detecting Groups and Application to Shape Recognition, JMIV, 2007]

- No produce particiones (puntos pueden no pertenecer a ningún cluster).
- Método *a contrario*: test de múltiples hipótesis que busca grandes desvíos de un modelo de fondo estimado, que corresponde a la ausencia de clusters.
- $NFA(G)$: estimador del número esperado de clusters de misma características que G bajo el modelo de fondo.
- Un cluster G es ε-significativo si $NFA(G) < \varepsilon$. Si $\varepsilon = 1$ se dice que es significativo (no puede ser una realización del azar).
- $NFA(G_1, G_2)$: estimador del número esperado de pares de clusters de mismas características que G_1 y G_2 bajo el modelo de fondo.
- Los NFA son cantidades comparables (independientemente del tipo de estructura que cuantifiquen).

3.5. Detección de clusters en ruido (validación individual de clusters)

Detección *a contrario* de clusters (cont.)

- Busca clusters validos entre los propuestos por un clustering jerárquico.

  Permite:

  - Decir si un cluster es válido;
  - Detectar el más significativo por inclusión: un cluster válido puede incluir o estar incluído en otros clusters válidos; se elige el de menor NFA;
  - Optar por dos clusters separados o por la unión de ellos (comparar $NFA(G_1, G_2)$ con $NFA(G_1 \cup G_2)$).