Reconocimiento Estadístico de Patrones, parte 4

Johan Van Horebeek

horebeek@cimat.mx

Enero - Junio 2011

1. Recordando el Temario del curso

1. Métodos exploratorios para datos multivariados
2. Métodos de agrupamiento
   (a) agrupamiento jerarquica
   (b) k-medias
   (c) agrupamiento basado en mezclas
   (d) relación PCA y agrupamiento
   (e) agrupamiento espectral

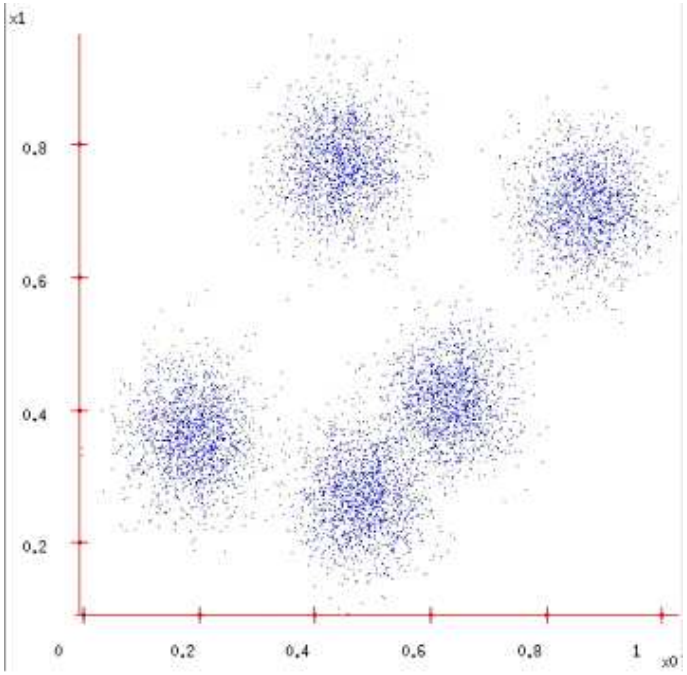
Métodos de agrupamiento

Problema:

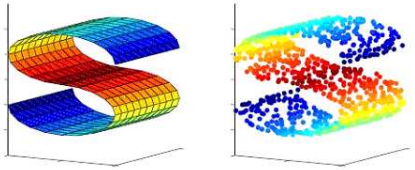
Segmentar datos en subgrupos homogeneos;

Encontrar grupos en base de semejanza;

Situación idonea:



dolor de cabeza:

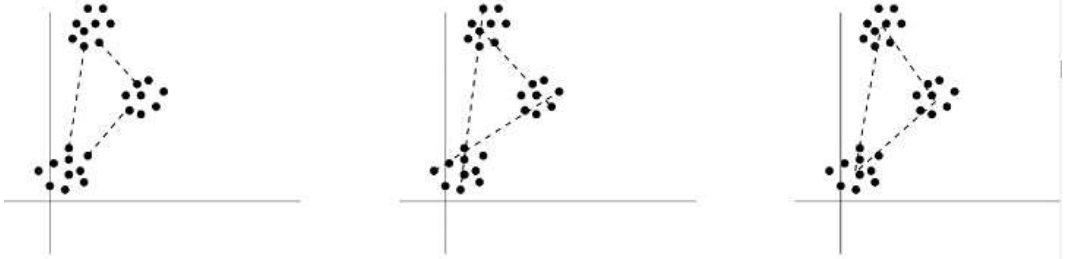


En cómputo, se considera lo anterior como un problema de **aprendizaje no supervisado**.

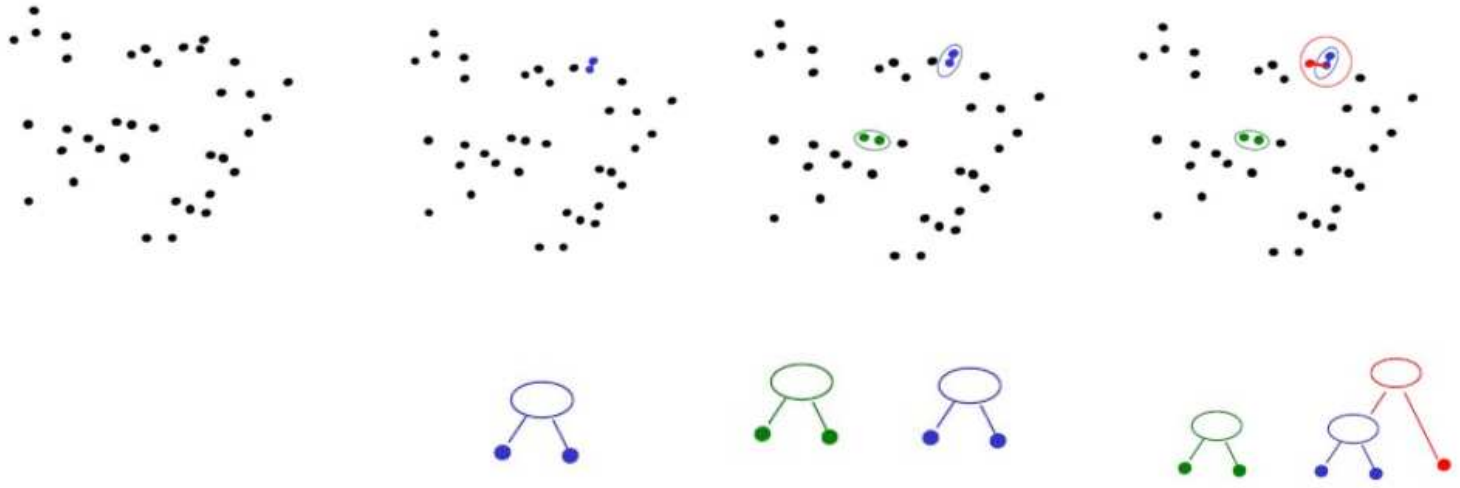
1. Agrupamiento jerárquico

Define distancia entre grupos de observaciones a partir de distancias entre observaciones

Single linkage: $d_{cc'} = \min_{n \in c, n' \in c'} d_{nn'}$
Complete linkage: $d_{cc'} = \max_{n \in c, n' \in c'} d_{nn'}$
Average linkage: $d_{cc'} = \text{mean}_{n \in c, n' \in c'} d_{nn'}$

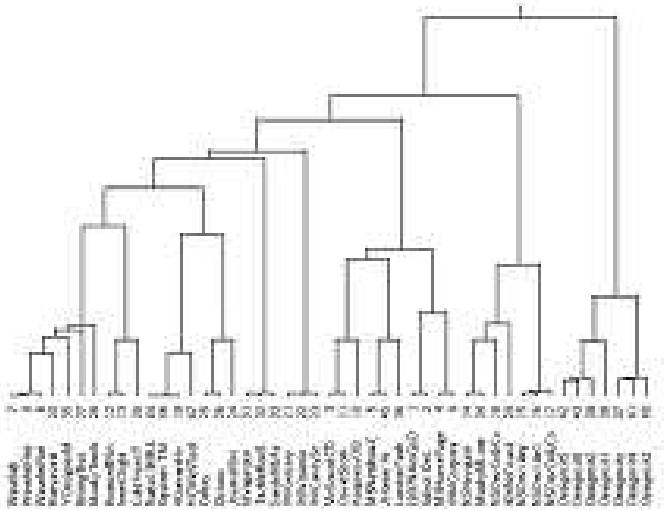


```
Define cada dato como un cluster
Repite  hasta tener un solo cluster:
     Une los dos clusters m'as cercanos seg'un d() en un solo cluster nuevo.
```



```
Define cada dato como un cluster
Repite  hasta tener un solo cluster:
     Une los dos clusters m'as cercanos seg'un d() en un solo cluster nuevo.
```

En general, se obtiene un dendograma:



En R: `hclust(d, method = "complete", ...  )`

Insertar `demo4.r`

Cuál d() elegir? Dónde cortar el dendograma?: misma situación con la elección del ancho de los Kernels de la clase anterior.

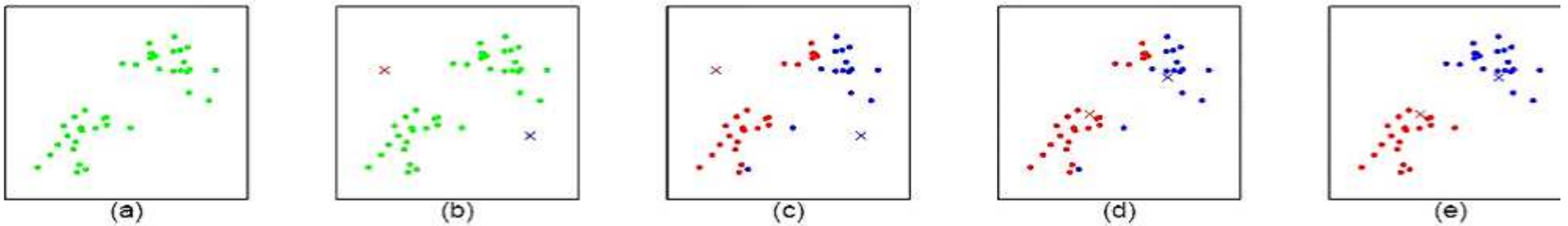
2. Algoritmo k-medias (k-means)

Define distancia Euclideana entre observaciones $d(\cdot, \cdot)$.

Representa cada grupo por un nucleo o centroide.

```
Elige al azar k nucleos;
  Repite  hasta convergencia:
     Asigna cada dato al nucleo m'as cercano seg'un d(,)
     Toma como nuevos nucleos, las medias de los datos de cada nucleo.
```

En R: `kmeans(x, centers, iter.max = 10, ...)`



Problema: Cómo elegir k?

Insertar `demo3.r`

k-medias minimiza una **función (de costo)** de una manera particular:

Sea c_g nucleo de grupo g, y indica con $g(i)$ el grupo de x_i, entonces k-medias minimiza:

$$\min_{g()} \min_{c_g} \sum_{g} \sum_{i:g(i)=g} \|x_i - c_g\|^2$$

en dos pasos descoplados: minimizando fijando g y minimizando fijando c_g, o sea, se minimiza:

$$\min_{g()} \sum_{g} \sum_{i:g(i)=g} \|x_i - \bar{x}_g\|^2, \text{ con } \bar{x}_g \text{ centroide de grupo } g$$

Interpretación:

$$\|x_i - c_{g(i)}\|^2 = \|x_i - \text{decodifica}(\text{codifica}(x_i))\|^2 \tag{1}$$

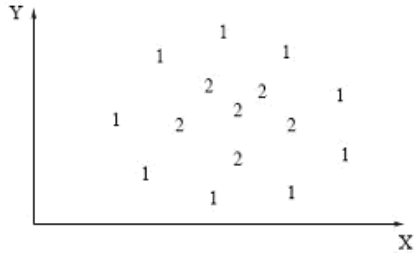
Se trata de **minimizar la varianza dentro de cada grupo**. Entre mayor k menor (1).

Observaciones:

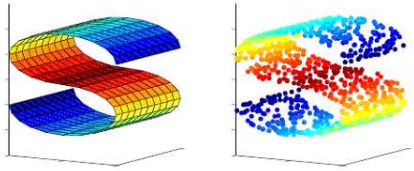
1. Como la distancia Euclideana da igual peso a cada dimensión, mejor normalizar primero los datos.

2. Se puede usar lo anterior para imputación de datos faltantes.

3. En muchas situaciones, k-means falla: por ejemplo:



   Solución: primero transformar los datos; otro ejemplo:



   Solución: usar méodos donde $d()$ captura la forma local del superficie;

4. Si los datos no son muy continuos, se prefiere tomar como nucleos puntos de la muestra.

5. Miles de variantes! (también porque para conjuntos de datos grandes, el algoritmo básico es demasiado costoso).

Es buena idea correr el algoritmo con diferentes puntos de arranque para evitar óptimos locales.

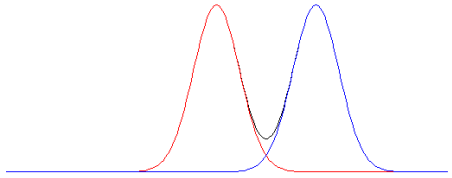
3. Métodos basados en mezclas

Vimos que k-medias minimiza una **función (de costo)** de una manera particular:

```
Elige al azar k nucleos (=centroide);
  Repite  hasta convergencia:
     Asigna cada dato al nucleo m'as cercano seg'un d(,)
     Toma como nuevos nucleos, las medias de los datos asoicados con cada nucleo.
```

Se pueden definir funciones de costo de muchas maneras.

Una manera es suponer un modelo generativo y usar menos la **verosimilitud** como función.

→ **Modelo de mezclas:** 

Por ejemplo:

$$P(X = x) = (1 - \alpha_1)P_0(X = x) + \alpha_1 P_1(X = x),$$

con P_0, P_1 distribucions gausianas.

Caso general: se supone X proviene de $P(X = x) = \sum_k \alpha_k P_k(X = x)$.

La Log Verosimilitud es de la forma:

$$\sum_i \log\{(1 - \alpha_1)P_0(X = x_i) + \alpha_1 P_1(X = x_i)\},$$

en general es muy dificil de optimizar.

Como $0 \leq \alpha_1 \leq 1$, se puede pensar que haya una v.a. latente $Y \sim Bern(\alpha_1)$:

$$P(X = x) = P(X = x|Y = 0)P(Y = 0) + P(X = x|Y = 1)P(Y = 1).$$

Conociendo $\{X_i, Y_i\}$, si n_j denota $\#\{y_i = j\}$, la log verosimilitud es:

$$\sum_{i:y_i=0} \log P_0(X = x_i) + n_0 \log(1 - \alpha_1) + \sum_{i:y_i=1} \log P_1(X = x_i) + n_1 \log \alpha_1,$$

obtenemos un problema descoplado; por ejemplo si $P_j \sim N(\mu_j, \sigma_j)$, la solución es:

$\hat{\mu}_j$ es promedio muestral; $\quad \hat{\sigma}_j$ es desviación estandar muestral.

$$\hat{\alpha_1} = \frac{n_1}{n_0 + n_1}.$$

IDEA

- Si conocemos $\{Y_i\}$, hay una solución cerrada; por ejemplo:

$$\hat{\mu_0} = \frac{\sum_i (1-y_i)x_i}{\sum_i (1-y_i)} \qquad \hat{\mu_1} = \frac{\sum_i y_i x_i}{\sum_i y_i} \tag{2}$$

(y algo similar para $\hat{\sigma_0}, \hat{\sigma_1}$) y

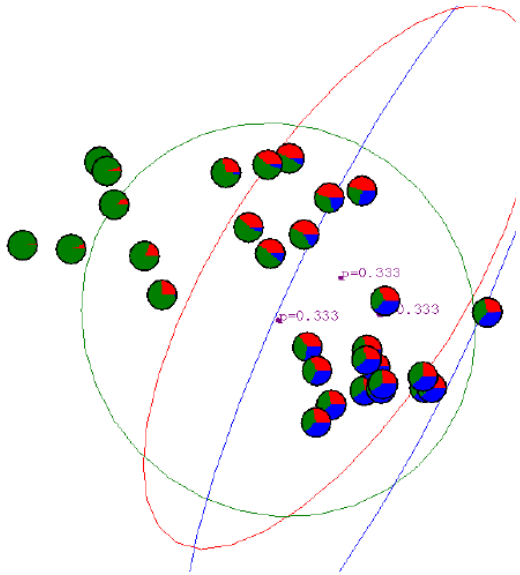
$$\hat{\alpha_1} = \frac{n_1}{n_0 + n_1} = \frac{\sum_i y_i}{n}$$

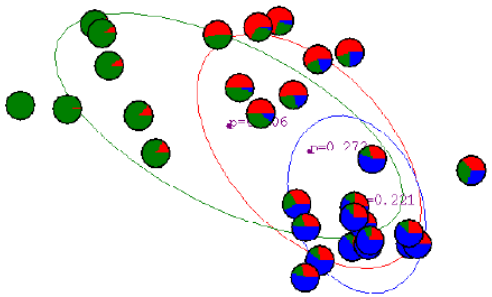
- Si conocemos los parámetros θ, podriamos calcular:

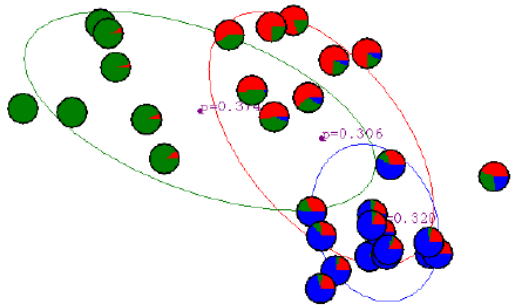
$$E(Y_i|\{X_i\}, \theta) = E_\theta Y_i|\{X_i\} = P_\theta(Y_i = 1|X_i) = \frac{P_\theta(X_i|Y_i = 1)P_\theta(Y_i = 1)}{P_\theta(X_i)}.$$

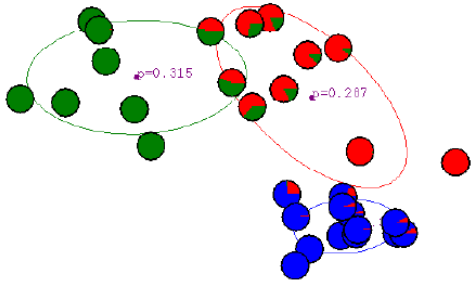
$\Rightarrow$ Iteramos lo anterior, usando en (2) en lugar de y_i, $E_\theta Y_i|\{X_i\}$.

En general para una mezcla de K distribuciones: para cada x_i, tenemos un vector γ_i de longitud K con $\gamma_i(k) = P(Y_i = k|X_i, \theta)$.


p=0.333
p=0.333
0.333


p=0.221


p=0.306


p=0.315
p=0.287

Forma general

Sea $T = (Z, Z^m)$; Z^m refiere a la parte faltante.

Define $l_0(\theta, T)$ como la log verosimilitud basada en los datos completos

Adivina $\hat{\theta^0}$

REPITE

Calcula: $Q(\theta|\hat{\theta^t}) = E_{\hat{\theta^t}}(l_0(\theta, T)|Z)$

Define $\hat{\theta^{t+1}}$ como el máximo de $Q(.|\hat{\theta^t})$

Se puede mostrar que converge a un óptimo **local**.

En el caso que vimos de la mezcla de gausianas: $T = (Z, Z^m)$ es (X, Y); $\theta = (\mu_0, \mu_1, \sigma_0, \sigma_1, \gamma)$

$$l_0(\theta, T) = \sum_{i:y_i=0} \log P_{0,\theta}(X = x_i) + \sum_{i:y_i=1} \log P_{1,\theta}(X = x_i) + (1 - \sum_i \frac{y_i}{n}) \log(1 - \alpha_1) + \sum_i \frac{y_i}{n} \log \alpha_1,$$

$$E_{\hat{\theta^t}}(l_0(\theta, T)|Z) = \sum_i (1 - E_{\hat{\theta^t}} Y_i|\{X_i\}) \log P_{0,\theta}(X = x_i) + \sum_i E_{\hat{\theta^t}}(Y_i|\{X_i\}) \log P_{1,\theta}(X = x_i) +$$

$$(1 - \sum_i E_{\hat{\theta^t}}(\frac{Y_i}{n}|\{X_i\})) \log(1 - \alpha_1) + \sum_i E_{\hat{\theta^t}}(\frac{Y_i}{n}|\{X_i\}) \log \alpha_1 =$$

$$\sum_i (1 - \gamma_i(1)) \log P_{0,\theta}(X = x_i) + \sum_i \gamma_i(1) \log P_{1,\theta}(X = x_i) + (1 - \sum_i \frac{\gamma_i(1)}{n} \log(1 - \alpha_1) + \sum_i \frac{\gamma_i(1)}{n} \log \alpha_1$$

En el caso de una mezcla de gausianas: si se compara EM con k-medias se observa que EM usa asignación suave: los γ_i no son típicamente 0 y 1 como en k-medias.

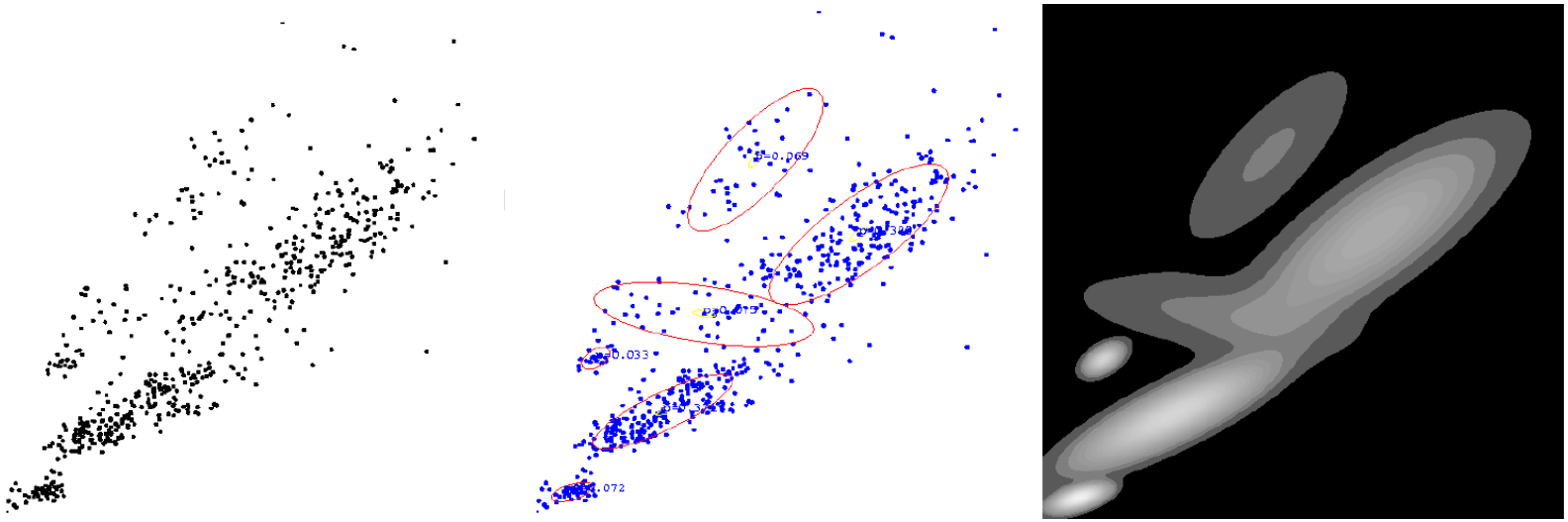
Uso en R: en `library(mclust)`: `me()` o `em()`.

```
m<-me(modelName = "EII", data = data,z=z)
m$z
```

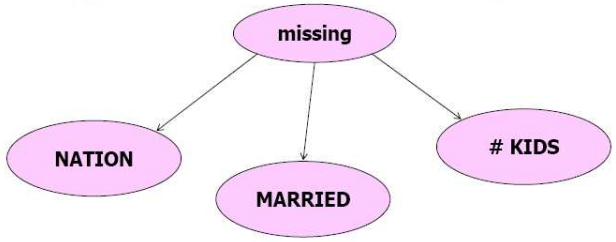
```
"E": equal variance  (one-dimensional)
 "V": variable variance(one-dimensional)
"EII": spherical, equal volume "VII": spherical, unequal volum
"EEI": diagonal, equal volume and shape "VVI": diagonal, varying
           etc etc
```

Se puede usar EM en muchas más situaciones !!!!

Ejemplo de estimación de densidades:



Ejemplo con variables latentes:



4 Agrupamiento Espectral