Problemes APA
Problema 12: Clustering de les dades artificials de Cassini

Lluc Bové

Q1 2016-17

Volem analitzar un problema d'agrupament amb dades en 2D usant la rutina mlbench.cassini. Generem dades en 3 grups amb el codi:

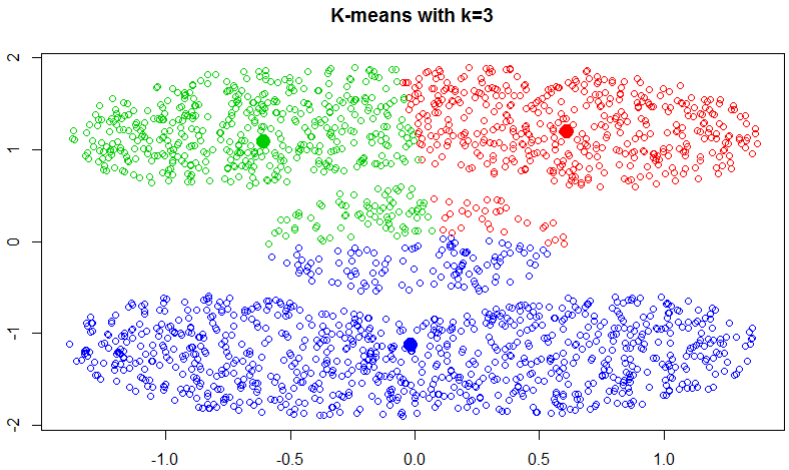
```
library(mlbench)
N <- 2000
data.1 <- mlbench.cassini(N, relsize = c(1,1,0.25))
plot(data.1)
```

Veiem que les estructures externes tenen forma de plàtan i entre elles hi ha un cercle amb menys densitat de dades. El plot anterior mostr la veritat de les dades (els 3 grups generats). Si ara fem:

```
plot(x=data.1$x[,1], y=data.1$x[,2])
```

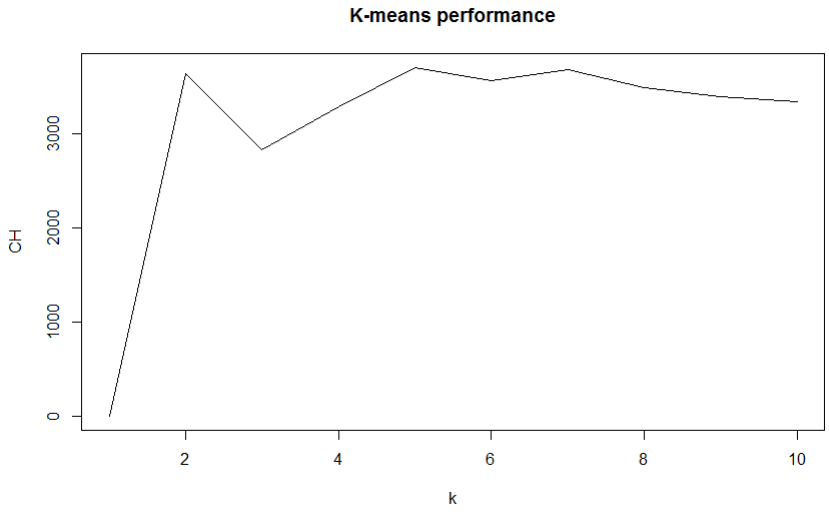
Veurem les dades en brut (el que rebrà el mètode de *clustering*). Es demana:

1. **Decidiu per endavant quin mètode de *clustering* hauria de treballar millor i amb quins paràmetres**
   El millor mètode hauria de ser barreja de gaussianes, donat que k-means aquí no tindria resultat, ja que a simple vista es pot veure que no seria el cas de gaussianes esfèriques. Pel que fa la família de la gaussiana en qüestió, sabent com s'han generat les gaussianes podríem elegir usar la família diagonal ja que les variables són independents. Però com que assumim que no sabem la veritat sobre les dades usarem general com a família de gaussianes.
2. **Apliqueu k-means un cert nombre de vegades amb $k = 3$ i observeu els reultats**
   Al aplicar k-means un cert nombre de vegades i quedant-nos amb la millor execució basant-nos en l'índex de *Calinski-Harabasz* amb $k = 3$ ens trobem amb la següent partició:

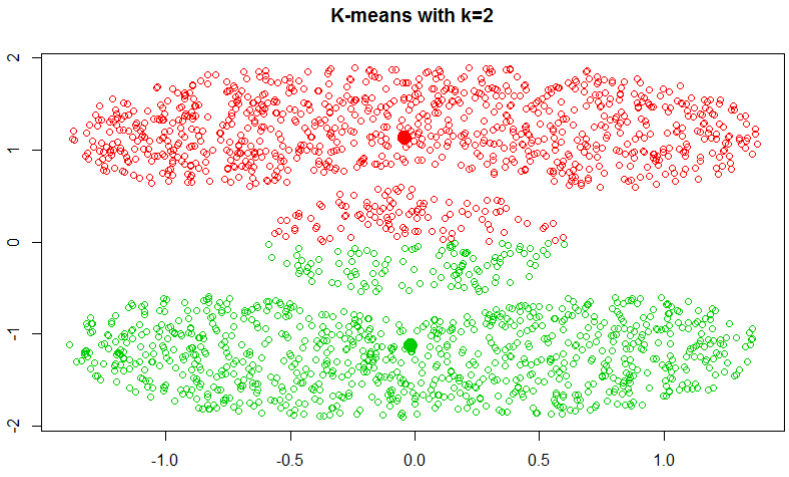


Podem veure que no s'acosta gens a la partició original, el *cluster* circular original fa que formi part de tots els *clusters* nous. El resultat és l'esperat ja que les dades no estan generades per gaussianes esfèriques i per tant k-means no ajusta massa bé.

3. **Apliqueu k-means amb una selecció de valors de k al vostre criteri (20 cops cadascun) i monitoritzeu l'índex de Calinski-Harabasz mitjà; quin k es veu millor?**
Usem valors de k en l'interval $[2, 10]$ i executem 20 cops per cada valor de k, trobem que la millor k varia depenent de l'execució general, entre els valors 2 i 5. En aquest cas mostrem el gràfic d'evolució de l'index CH segons el valor de k, en que el cas millor és $k = 2$:

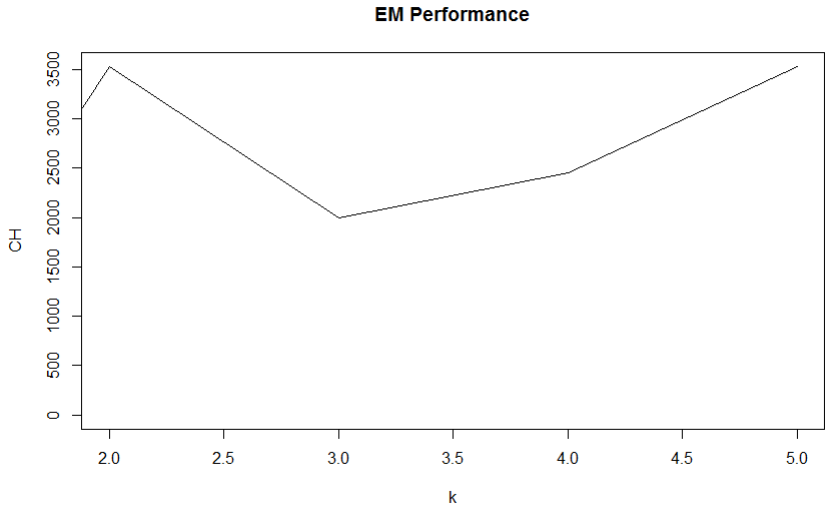


Veiem doncs que els valors més alts de k són 2 i 5, i el més alt entre ells dos és 2. Mostrem doncs el *clustering* amb $k = 2$.

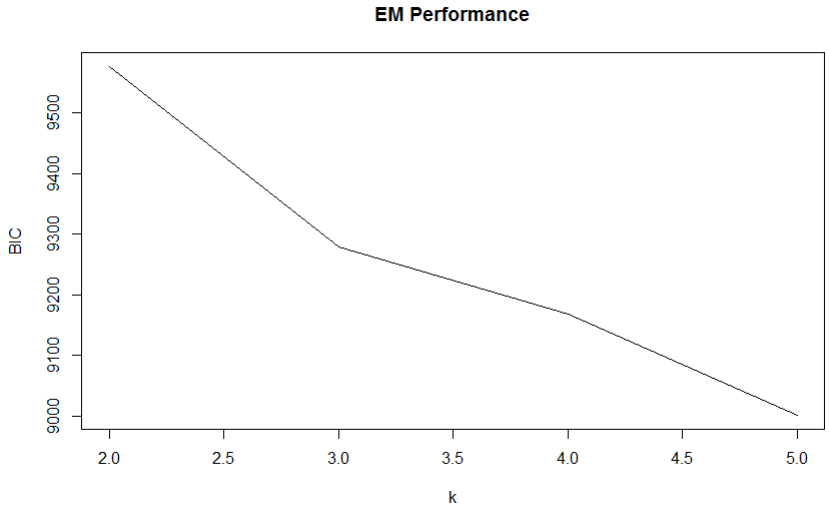


Segurament aquesta partició tingui un millor índex ja que es redueix la variància entre els *clusters* i també la variància dins d'ells. Tot i això no és el que busquem, com es pot veure a simple vista, ja que ni tant sols la millor k coincideix amb el nombre real de classes.

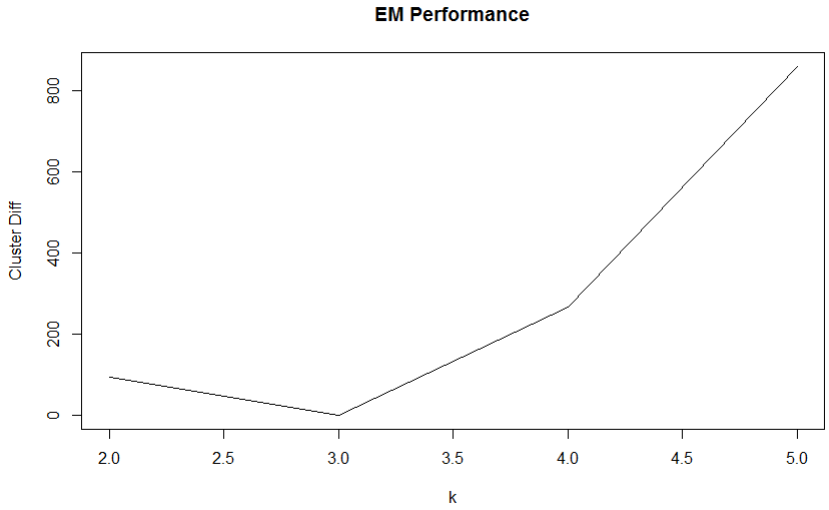
4. **Apliqueu l'algorisme E-M amb una selecció de valors de k al vostre criteri (10 cops cadascun) i observeu els resultats. Comproveu els resultats contra les vostres expectatives (apartat 1).**
Apliquem E-M per valors de k dins de l'interval $[2 - 5]$. Primer provem de mesurar l'índex CH per a tots els valors de k i obtenim el següent gràfic:



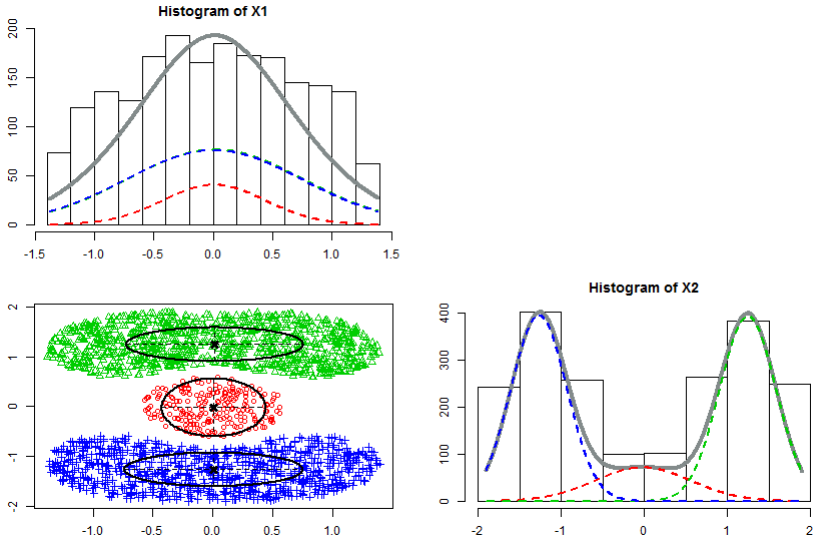
Podem veure com el millor valor de k és 5. Això contradiu el que esperàvem ja que sabem que el nombre de *clusters* és 3 i justament amb $k = 3$ tenim el valor més baix de l'index mesurat. A més a més hauria d'ajustar millor amb els paràmetres de EM que hem elegit. Així doncs pensem que l'índex CH potser no és el més adequat per mesurar la qualitat de la partició de les nostres dades i per això provem el mateix experiment amb l'índex BIC(*Bayesian information criterion*). Aquest índex es basa en la versemblança de la partició i com més baix és, més bona és aquesta. Obtenim el següent:



Podem veure com tampoc és gaire bo l'índex pel que fa la decisió de k ja que al augmentar-la aquest disminueix. Per tant la millor k tampoc seria 3 com ja sabem que ha de ser. Així doncs creem un nou criteri per a trobar la qualitat de la partició, ens basem en la veritat de les dades. Aquest criteri no és realista ja que normalment no saps la partició a priori de les dades, però com que estem analitzant el comportament dels algorismes de *clustering* farem veure que sabem a priori com és la partició. Així doncs anomenem aquest criteri *diff* i expressa quants punts no "encaixen"en la partició vertadera. És a dir, actuaria com una diferència entre la partició original i la que trobem amb l'algorisme. Mesurem també l'index per a la selecció de k i ho representem:



Aquí sí que es veu clarament com el millor valor de k és 3. Sabent això executem unes quantes vegades l'algorisme EM amb $k = 3$ i ens quedem amb la millor execució i podem observar el *clustering* que veiem a continuació:



Amb un bon nombre d'execucions hem pogut obtenir un *clustering* que és igual al vertader, és a dir, s'ajusta a la perfecció. Amb tot això podem dir que si no tenim dades a priori sobre el nombre de *clusters* que hem d'obtenir és molt complicat trobar un criteri que avaluï una partició concreta per poder optimitzar aquest nombre, i per tant, trobar quants *clusters* hem de buscar i que a més a més sigui independent de com són les dades.