

Trabajo Práctico 3: Clustering

1. Ejemplos prácticos de clustering

a) Analice el dataset crabs en el paquete MASS de R usando k-means y hclust. Tiene dos columnas (1 y 2) con especie y género de cangrejos, y después 5 columnas de mediciones (4 a 8). Para cargarlo y verlo se usa:

```
library(MASS)
data(crabs)
summary(crabs)
plot(crabs[,4:8],col=as.numeric(crabs[,1]),pch=as.numeric(crabs[,2]))
```

El objetivo es ver si se pueden encontrar algunas de las clases con clustering. En este dataset se sugiere usar una transformación logarítmica de los datos en primer lugar, y a partir de allí usar los datos con distintos escalados (por ejemplo, usar scale() o usar PCA -prcomp()- escalando los datos previamente o usar primero PCA y después escalar los datos, una vez girados).

b) Analice también el dataset lampone que está en la página:

```
load("lampone.Rdata")
```

Tiene dos clasificaciones distintas, una es el año de la medición (columna 1) y otra la especie de blueberry (columna 143). Nuevamente, hay que ver si se pueden recuperar con clustering divisivo o jerárquico, usando distintas escalas. Para visualizar los datos, es conveniente usar PCA ya que son muchas dimensiones.

Comentario: para comparar dos soluciones de clustering o una de ellas contra las clases originales se suele usar una tabla, como por ejemplo:

```
>#hago una tabla de confusion para comparar
>cont.table <- table(clusters.kmeans$cluster,clusters.otro.metodo)
>print(cont.table)
```

pero se puede optimizar el match entre los dos clusterings, para hacer mejor la tabla, usando:

```
library(e1071)
# Find optimal match between the two classifications
class.match <- matchClasses(as.matrix(cont.table),method="exact")
# Print the confusion table, with rows permuted to maximize the diagonal
print(cont.table[class.match])
```

2. Prepare código en R para los métodos:

a) GAP statistic

b) Estabilidad

Código R con ejemplo de como calcular el score de estabilidad de dos soluciones de clustering:

```
x<-iris[, -5]
n<-dim(x)[1]
#fijo el numero de clusters
k=3
#creo dos indices al azar y hago los clusters
ind1<-sample(n,0.9*n)
cc1<-kmeans(x[ind1,],k,nsta=10)$cluster
ind2<-sample(n,0.9*n)
cc2<-kmeans(x[ind2,],k,nsta=10)$cluster
#pongo los clusters de nuevo en longitud n - quedan 0 los puntos fuera del sample
v1<-v2<-rep(0,n)
v1[ind1]<-cc1
```

```

v2[ind2]<-cc2
#creo una matriz m con 1 donde los dos puntos estan en el mismo cluster, -1 en distinto cluster y 0 si alguno no esta, para cada
clustering
a<-sqrt(v1%*%t(v1))
m1<-a / -a + 2*(a==round(a))
m1[is.nan(m1)]<-0
a<-sqrt(v2%*%t(v2))
m2<-a / -a + 2*(a==round(a))
m2[is.nan(m2)]<-0
#calculo el score, los pares de puntos que estan en la misma situacion en los dos clustering dividido el total de pares validos.
validos<-sum(v1*v2>0)
score<-sum((m1*m2)[upper.tri(m1)]>0)/(validos*(validos-1)/2)
print(score)

```

3. Aplíquelos a los problemas de las 4 gaussianas de las slides, iris y lampone.

4. Opcional (1 punto). Busque un dataset que considere interesante. Aplique alguno de los métodos de clustering discutidos y alguno de los métodos para determinar la cantidad de clusters presentes.

Entrega por mail. Como siempre, todo el código R que se usa, en un script que funcione y que se entienda mínimamente. Aparte, un informe sobre los resultados.