Evaluación bootstrap de clústeres

El objetivo del análisis de clúster es agrupar las observaciones de los datos en clústeres de forma que cada dato de un clúster sea más similar a otros datums del mismo clúster que a los de otros clústeres. Este es un método de análisis ideal cuando no se dispone de datos de entrenamiento anotados.

Una pregunta importante al evaluar clústeres es si un clúster dado es "real" - ¿representa la estructura real de los datos o es un artefacto del algoritmo de clúster? Esto es especialmente importante con algoritmos de clúster como k-medias, donde el usuario debe especificar el número de clústeres a priori. En nuestra experiencia, los algoritmos de clústeres suelen producir varios clústeres que representan la estructura o las relaciones reales de los datos, y luego uno o dos clústeres que son contenedores que representan "otros" o "misceláneos". Los clústeres de "otros" tienden a estar compuestos por puntos de datos que no tienen una relación real entre sí; simplemente no encajan en ningún otro lugar.

Una forma de evaluar si un clúster representa una estructura real es observar si se mantiene ante variaciones plausibles en el conjunto de datos. El paquete **fpc** incluye una función llamada clusterboot() que utiliza el remuestreo bootstrap para evaluar la estabilidad de un clúster determinado. clusterboot() es una función integrada que realiza la agrupación y evalúa los clústeres finales generados. Tiene interfaces con varios algoritmos de clustering de R, como hclust y kmeans.

El algoritmo de clusterboot utiliza el coeficiente de Jaccard, una medida de similitud entre conjuntos. La similitud de Jaccard entre dos conjuntos A y B es el cociente entre el número de elementos en la intersección de A y B y el número de elementos en la unión de A y B. La estrategia general básica es la siguiente:

1. Agrupar los datos como de costumbre.
2. Extraer un nuevo conjunto de datos (del mismo tamaño que el original) remuestreando el conjunto de datos original con reemplazo (lo que significa que algunos puntos de datos pueden aparecer más de una vez y otros no). Agrupa el nuevo conjunto de datos.
3. Para cada clúster del clustering original, encuentra el clúster más similar en el nuevo clustering (el que proporcione el coeficiente de Jaccard máximo) y registra ese valor. Si este coeficiente de Jaccard máximo es menor que 0.5, el clúster original se considera *disuelto* (dissolved-it); no apareció en el nuevo clustering. Un clúster que se disuelve con demasiada frecuencia probablemente no sea un clúster "real".
4. Repite los pasos 2 y 3 varias veces.

La estabilidad de cada c clúster en el clustering original es el valor medio de su coeficiente de Jaccard en todas las iteraciones de bootstrap. Como regla general, los clústeres con un valor de estabilidad inferior a 0.6 deben considerarse inestables. Valores entre 0.6 y 0.75 indican que el clúster mide un patrón en los datos, pero no hay mucha certeza sobre qué puntos deben agruparse.

Los clústeres con valores de estabilidad superiores a 0.85 pueden considerarse altamente estables (es probable que sean clústeres reales).

Distintos algoritmos de agrupamiento pueden generar distintos valores de estabilidad, incluso cuando producen agrupaciones muy similares, por lo que clusterboot() también mide la estabilidad del algoritmo de agrupamiento.

El conjunto de datos de proteínas

Para demostrar clusterboot(), utilizaremos un pequeño conjunto de datos de 1973 sobre el consumo de proteínas de nueve grupos de alimentos diferentes en 25 países europeos. El conjunto de datos original se puede encontrar adjunto al documento.
El archivo de datos se llama protein.txt.

1. Country: Country name
2. RdMeat: Red meat
3. WhMeat: White meat
4. Eggs: Eggs
5. Milk: Milk
6. Fish: Fish
7. Cereal: Cereals
8. Starch: Starchy foods
9. Nuts: Pulses, nuts, and oil-seeds
10. Fr&Veg: Fruits and vegetables

El objetivo es agrupar los países según sus patrones de consumo de proteínas. El conjunto de datos se carga en R como un frame de datos llamado protein, como se muestra en el siguiente listado.

```
> protein <- read.table("protein.txt", header=TRUE, sep="\t")
> summary(protein)
```

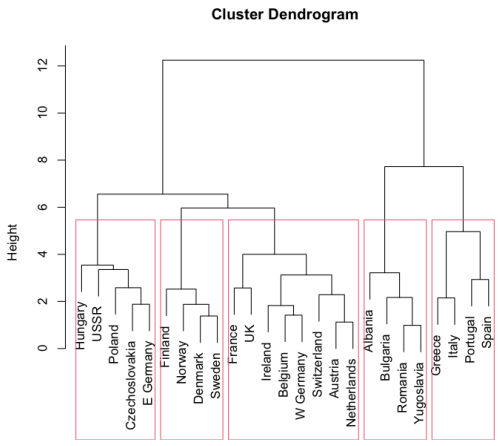
```
   Country            RedMeat        WhiteMeat          Eggs           Milk           Fish          Cereals
 Length:25          Min.   : 4.400   Min.   : 1.400   Min.   :0.500   Min.   : 4.90   Min.   : 0.200   Min.   :18.60
 Class :character   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700   1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30
 Mode  :character   Median : 9.500   Median : 7.800   Median :2.900   Median :17.60   Median : 3.400   Median :28.00
                    Mean   : 9.828   Mean   : 7.896   Mean   :2.936   Mean   :17.11   Mean   : 4.284   Mean   :32.25
                    3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700   3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10
                    Max.   :18.000   Max.   :14.000   Max.   :4.700   Max.   :33.70   Max.   :14.200   Max.   :56.70
     Starch            Nuts           Fr.Veg
 Min.   :0.600   Min.   :0.700   Min.   :1.400
 1st Qu.:3.100   1st Qu.:1.500   1st Qu.:2.900
 Median :4.700   Median :2.400   Median :3.800
 Mean   :4.276   Mean   :3.072   Mean   :4.136
 3rd Qu.:5.700   3rd Qu.:4.700   3rd Qu.:4.900
 Max.   :6.500   Max.   :7.800   Max.   :7.900
```

```
> # Use all the columns except the first (Country).
> vars.to.use <- colnames(protein)[-1]
> # Scale the data columns to be zero mean and unit variance.
> # The output of scale() is a matrix.
> pmatrix <- scale(protein[,vars.to.use])
> # optionally, store the centers and standard deviations of the original data,
> # so you can "unscale" it later.
> pcenter <- attr(pmatrix, "scaled:center")
> pscale <- attr(pmatrix, "scaled:scale")
```

Antes de ejecutar clusterboot(), agruparemos los datos mediante un algoritmo de agrupamiento jerárquico (método de Ward):

```
> # Create the distance matrix.
> d <- dist(pmatrix, method="euclidean")
> # Do the clustering.
> pfit <- hclust(d, method="ward.D2")
> # Plot the dendrogram.
> plot(pfit, labels=protein$Country)
```

El dendrograma sugiere cinco clústeres, como se muestra en la figura siguiente. Puedes dibujar los rectángulos en el dendrograma con el comando rect.hclust(pfit, k=5).



Extraigamos e imprimamos los clústeres:

Una función conveniente para imprimir los países en cada grupo, junto con los valores de consumo de carne roja, pescado y frutas/verduras.

```
> print_clusters <- function(labels, k) {
+  for(i in 1:k) {
+    print(paste("cluster", i))
+    print(protein[labels==i,c("Country","RedMeat","Fish","Fr.Veg")])
+  }
+ }
> groups <- cutree(pfit, k=5) # get the cluster labels
> print_clusters(groups, 5) # --- results –
```

```
[1] "cluster 1"
      Country RedMeat Fish Fr.Veg
1     Albania    10.1  0.2    1.7
4    Bulgaria     7.8  1.2    4.2
18    Romania     6.2  1.0    2.8
25 Yugoslavia     4.4  0.6    3.2
[1] "cluster 2"
       Country RedMeat Fish Fr.Veg
2      Austria     8.9  2.1    4.3
3      Belgium    13.5  4.5    4.0
9       France    18.0  5.7    6.5
12     Ireland    13.9  2.2    2.9
14 Netherlands     9.5  2.5    3.7
21 Switzerland    13.1  2.3    4.9
22          UK    17.4  4.3    3.3
24   W Germany    11.4  3.4    3.8
[1] "cluster 3"
          Country RedMeat Fish Fr.Veg
5  Czechoslovakia     9.7  2.0    4.0
7       E Germany     8.4  5.4    3.6
11        Hungary     5.3  0.3    4.2
16         Poland     6.9  3.0    6.6
23           USSR     9.3  3.0    2.9
[1] "cluster 4"
   Country RedMeat Fish Fr.Veg
6  Denmark    10.6  9.9    2.4
8  Finland     9.5  5.8    1.4
15  Norway     9.4  9.7    2.7
20  Sweden     9.9  7.5    2.0
[1] "cluster 5"
    Country RedMeat Fish Fr.Veg
10   Greece    10.2  5.9    6.5
13    Italy     9.0  3.4    6.7
17 Portugal     6.2 14.2    7.9
19    Spain     7.1  7.0    7.2
```

Estos clústeres tienen cierta lógica: los países de cada clúster tienden a estar en la misma región geográfica. Es lógico que los países de la misma región tengan hábitos alimentarios similares. También se puede observar que:

- El clúster 2 está compuesto por países con un consumo de carne roja superior al promedio.
- El clúster 4 contiene países con un consumo de pescado superior al promedio, pero bajo consumo de productos agrícolas.

- El clúster 5 contiene países con un alto consumo de pescado y productos agrícolas.

Ejecutemos clusterboot() con los datos de proteínas, utilizando la agrupación jerárquica con cinco clústeres.

```
> library(fpc)
> # set the desired number of clusters
> kbest.p<-5
> #   Run clusterboot() with hclust
> #   ('clustermethod=hclustCBI') using Ward's method
> #   ('method="ward"') and kbest.p clusters
> #   ('k=kbest.p'). Return the results in an object
> #   called cboot.hclust.
>
> cboot.hclust <- clusterboot(pmatrix,clustermethod=hclustCBI,
+   method="ward.D2", k=kbest.p)
```

Los resultados de la agrupación se encuentran en cboot.hclust$result. La salida de la función hclust() se encuentra en cboot.hclust$result$partition. cboot.hclust$result$partition devuelve un vector de etiquetas de clúster.

```
> groups<-cboot.hclust$result$partition
> # -- results --
> print_clusters(groups, kbest.p)
```

```
[1] "cluster 1"
      Country RedMeat Fish Fr.Veg
1     Albania    10.1  0.2    1.7
4    Bulgaria     7.8  1.2    4.2
18    Romania     6.2  1.0    2.8
25 Yugoslavia     4.4  0.6    3.2
[1] "cluster 2"
       Country RedMeat Fish Fr.Veg
2      Austria     8.9  2.1    4.3
3      Belgium    13.5  4.5    4.0
9       France    18.0  5.7    6.5
12     Ireland    13.9  2.2    2.9
14 Netherlands     9.5  2.5    3.7
21 Switzerland    13.1  2.3    4.9
22          UK    17.4  4.3    3.3
24   W Germany    11.4  3.4    3.8
[1] "cluster 3"
         Country RedMeat Fish Fr.Veg
5  Czechoslovakia     9.7  2.0    4.0
7       E Germany     8.4  5.4    3.6
11        Hungary     5.3  0.3    4.2
16         Poland     6.9  3.0    6.6
23           USSR     9.3  3.0    2.9
[1] "cluster 4"
   Country RedMeat Fish Fr.Veg
6  Denmark    10.6  9.9    2.4
8  Finland     9.5  5.8    1.4
15  Norway     9.4  9.7    2.7
20  Sweden     9.9  7.5    2.0
[1] "cluster 5"
    Country RedMeat Fish Fr.Veg
10   Greece    10.2  5.9    6.5
13    Italy     9.0  3.4    6.7
17 Portugal     6.2 14.2    7.9
19    Spain     7.1  7.0    7.2
```

```
> # The vector of cluster stabilities.
> # Values close to 1 indicate stable clusters
> cboot.hclust$bootmean
[1] 0.7863333 0.7804563 0.6463730 0.8793214 0.7573333
> # The count of how many times each cluster was dissolved. By default
> # clusterboot() runs 100 bootstrap iterations.
> # Clusters that are dissolved often are unstable.
> cboot.hclust$bootbrd
[1] 26 17 46 17 29
```

Los resultados anteriores muestran que el clúster de países con alto consumo de pescado (clúster 4) es altamente estable (estabilidad del clúster = 0.88). Los clústeres 1 y 2 también son bastante estables; el clúster 5 (estabilidad del clúster = 0.75) lo es menos. El clúster 3 (estabilidad del clúster = 0.64) presenta las características de lo que hemos denominado el "otro" clúster. Ten en cuenta que clusterboot() tiene un componente aleatorio, por lo que los valores exactos de estabilidad y el número de veces que se disuelve cada clúster variarán entre ejecuciones.

Con base en estos resultados, podemos afirmar que los países del clúster 4 tienen hábitos alimentarios muy similares, distintos de los de los demás países (alto consumo de pescado y carne roja, con una cantidad relativamente baja de frutas y verduras); también podemos afirmar que los países de los clústeres 1 y 2 representan patrones alimentarios distintos. Los países del clúster 3, por otro lado, muestran patrones alimentarios diferentes a los de los países de los otros clústeres, pero no son tan similares entre sí.

El algoritmo clusterboot() asume que se tiene el número de clústeres, k. Obviamente, determinar k será más difícil para conjuntos de datos más grandes que nuestro ejemplo de proteínas. Hay maneras de estimar k, pero quedan fuera del alcance de este documento. Sin embargo, una vez que se tenga una idea del número de clústeres, clusterboot() es una herramienta útil para evaluar la solidez de los patrones descubiertos.