PECL5 - Fundamentos de la ciencia de datos

Mario Adán Herrero Alberto González Martínez
Branimir Stefanov Yanev Diego Gutiérrez Marco

29 de diciembre de 2020

Resumen

En el siguiente documento se presentan los resultadostrucción de árboles de deci

Índice

1. Introducción

La práctica consta de dos partes:
En la primera parte se reaejercicio que contenga modifi

2. Ejercicio 1 - Análisis de detección de datos anómalos

Utilizaremos las librerías tree y rpart para realizar la clasificación de los datos de las calificaciones de los alumnos en distintas partes de la asignatura (teoria, lab, práctica). Realizaremos el análisis con ambas librerías y contrastaremos los resultados obtenidos.

2.1. Apartado 1 - Detección de datos anómalos. Medidas de ordenación

En este ejercicio, empleamos el algoritmo K-vecinos para analizar la misma muestra de calificaciones empleada en clase. El algoritmo es tan sencillo que no necesitaremos usar ninguna librería; simplemente lo programaremos. Emplearemos un valor de $K = 4$ y un grado de outlier $d = 2{,}5$.

Primero, cargamos la muestra data en la variable `muestra`.

```
> muestra <- matrix(c(4, 4, 4, 3, 5, 5, 1, 1, 5, 4), 2,5)
> muestra <- t(muestra)
```

A continuación, creamos una matriz que almacene las distancias entre puntos de la muestra y ordenamos las distancias de menor a mayor.

```
> distancias <- as.matrix(dist(muestra))
> distancias <- matrix(distancias, 5, 5)
> for(i in 1:5){
+     distancias[,i] = sort(distancias[,i])
+ }
> distanciasOrdenadas <- distancias
```

Con esto, ya podemos distinguir datos anómalos: Consideraremos un dato anómalo cualquier dato cuyo vecino K esté a una distancia mayor que el grado de outlier d. En nuestro caso, será anómalo cualquier valor superior cuyo vecino $K = 4$ esté a una distancia mayor que $d = 2{,}5$.

```
> for(i in 1:5){
+     if(distanciasOrdenadas[4,i] > 2.5) {
+         cat("[", i, "] es un suceso anomalo o outlier\n")
+     }
+ }

[ 4 ] es un suceso anomalo o outlier
```

2.2. Apartado 2 - Detección de datos anómalos. Medidas de dispersión

En este ejercicio, empleamos medidas de ordenación, método de caja y bigotes, para la detección de datos anómalos sobre la resistencia de la muestra de tipos de hormigón vista en clase.

Primero, cargamos en la variable `muestra` los datos de la muestra.

```
> muestra <- t(matrix(c(3, 2, 3.5, 12, 4.7, 4.1, 5.2, 4.9, 7.1,
+                       6.1, 6.2, 5.2, 14, 5.3), 2, 7,
+                     dimnames = list(c("r","s"))))
> muestra = data.frame(muestra)
```

La función `boxplot` nos permite mostrar en pantalla, entre otros datos, los outliers de una muestra según el método de caja y bigotes. Normalmente, esta función también dibuja un plot, pero podemos desactivar esa funcionalidad.

A continuación, empleamos dicha función para identificar datos anómalos por este método.

```
> boxplot(muestra$r, range=1.5, plot = FALSE)

$stats
     [,1]
[1,] 3.00
[2,] 4.10
[3,] 5.20
[4,] 6.65
[5,] 7.10

$n
[1] 7

$conf
         [,1]
[1,] 3.677181
[2,] 6.722819

$out
[1] 14

$group
[1] 1

$names
[1] "1"
```

Esta función, en cambio, no nos permite obtener dichos outliers como variables. Por tanto, implementaremos también el algoritmo de caja y bigotes.

Primero, calculamos los cuartiles de la muestra.

```
> q1 <- quantile(muestra$r, 0.25)
> q3 <- quantile(muestra$r, 0.75)
```

Ahora calculamos el intervalo de valores que consideramos normales.

```
> s = 1.5
> intervalo <- c(q1 - s * (q3 - q1), q1 + s*(q3-q1))
```

Finalmente, identificamos los datos que se encuentran fuera del intervalo.

```
> for (i in 1:length(muestra$r))
+ {
+     # si se sale del intervalo, es un outlayer
+     if(muestra$r[i] < intervalo[1] || muestra$r[i] > intervalo[2])
+     {
+         cat("El dato", muestra$r[i], "es un outlayer.")
+     }
+
+ }
```

```
El dato 14 es un outlayer.
```

2.3. Apartado 3 - Detección de datos anómalos. Regresion

En este ejercicio, empleamos medidas de dispersión, desviación típica, para determinar datos anómalos sobre la misma muestra del ejercicio anterior.

Empezamos cargando en una variable la muestra.

```
> muestra <- t(matrix(c(3, 2, 3.5, 12, 4.7, 4.1, 5.2, 4.9, 7.1,
+ 6.1, 6.2, 5.2, 14, 5.3), 2, 7, dimnames=list(c("r", "d"))))
> muestra <- data.frame(muestra)
```

Ahora calculamos el intervalo de datos normales usando la desviación típica.

```
> intDesv <- c(mean(muestra$d) - 2*sd(muestra$d), mean(muestra$d) + 2*sd(muestra$d))
> sdd <- sqrt(var(muestra$d) * (length(muestra$d)-1 / length(muestra$d)))
```

Finalmente, comprobamos en bucle que los sucesos se encuentren dentro del intervalo definido.

```
> for(i in 1:length(muestra$d)){
+     # Si estan fuera del rango, imprimimos el suceso por pantalla
+     if(muestra$d[i] < intDesv [1] || muestra$d[i] > intDesv [2]) {
+         cat("El suceso [" , i, "] = ", muestra$d[i] ,
+          "es un suceso anomalo o outlier\n")
+     }
+ }
```

```
El suceso [ 2 ] =  12 es un suceso anomalo o outlier
```

2.4. Apartado 4 - Detección de datos anómalos. Algoritmo K-vecinos

En este ejercicio, empleamos un modelo de regresión para identificar datos anómalos sobre las densidades de la muestra usada en el anterior ejercicio, utilizando error estándar de los residuos.

Primero cargamos en una variable la muestra.

```
> muestra <- t(matrix(c(3,2,3.5,12,4.7,4.1,5.2,4.9,7.1,
+ 6.1,6.2,5.2,4,5.3), 2, 7, dimnames = list(c("r","d"))))
> muestra = data.frame(muestra)
```

Calculamos ahora la regresión de la muestra y extraemos los residuos.

```
> regresion = lm(muestra$d~muestra$r)
> residuos = summary(regresion)$residuals
> print(residuos)
```

```
         1          2          3          4          5          6          7
-3.8171799  6.2269248 -1.5672239 -0.7231192  0.6444787 -0.3349098 -0.4289705
```

Calculamos ahora el error estándar, en función de los residuos.

```
> error_estandar = sqrt(sum(residuos**2)/length(muestra$d))
```

Finalmente, identificamos como anómalos los datos cuyo valor absoluto supere el rango correspondiente al grado de outlier $d = 1{,}5$.

```
> grado_outlier = 1.5
> dsr = grado_outlier * error_estandar
> for (i in 1:length(muestra$r))
+ {
+     if(abs(residuos[i]) > dsr)
+     {
+         cat("El dato", muestra$d[i], "es un outlayer. \n")
+     }
+ }
```

```
El dato 12 es un outlayer.
```

3. Ejercicio 2 - Análisis de detección de datos anómalos. Metodos de teoria

Presentamos en este apartado dos enunciados desarrollados por nosotros y sus soluciones.

3.1. Apartado 1 - Árboles de decisión. Algoritmo Hunt

En este ejercicio usamos un modelo multivariante para identificar outliers en una muestra. En concreto, creamos un modelo de regresión y usamos un criterio basado en distancia de Cook para determinar qué datos son anómalos.

Primero cargamos la muestra.

```
> url <- "https://raw.githubusercontent.com/selva86/datasets/master/ozone.csv"
> ozono <- read.csv(url)  # Leemos el csv desde una URL
```

Construímos ahora el modelo de regresión, y calculamos las distancias de Cook entre los puntos. Además, identificamos la distancia máxima de un dato normal.

```
> modelo <- lm(ozone_reading~., data=ozono)
> distancia_cooks <- cooks.distance(modelo)
> valor_maximo <- 4*mean(distancia_cooks, na.rm=T)
```

Con esto, podemos dibujar un plot que separe los datos anómalos de los normales.

```
> plot(distancia_cooks, pch="*", cex=2,
+ # Pintamos la distancia de cooks
+ main="Estudio de valores anómalos con la distancia de Cooks")
> # Pintamos la linea que separa los datos normales de los outliers
> abline(h = valor_maximo, col="red")
> text(x=1:length(distancia_cooks)+1, y=distancia_cooks,
+ labels=ifelse(distancia_cooks>valor_maximo,
+ names(distancia_cooks),""), col="red")
> # Añadimos etiquetas para identificar los outliers
```

Mediante el criterio de la distancia de Cook, identificamos los valores anómalos individualmente y mostramos una lista de ellos.

```
> outliers <- list() # Creamos la lista para almacenar los outliers
> for(i in 1:length(distancia_cooks)){
+     if(distancia_cooks[i] > valor_maximo) {
+         # Si los datos estan fuera del rango de valores normales
+         outliers <- append(outliers, i) # Se añaden a la lista de anomalos
+     }
+ }
> matriz <- matrix(unlist(outliers))
> cat("Los datos outliers son:\n")

Los datos outliers son:

> cat(matriz, sep =', ')

11, 14, 34, 82, 84, 86, 130, 146, 154

> cat("\n")
```

Y genera el siguiente gráfico:

3.2. Apartado 2 - Análisis de regresión lineal

En este ejercicio se emplea aprendizaje mediante K-vecinos-próximos (K-nearest-neighbours) para clasificar una muestra; en este caso, datos de flores del dataset `iris`.

Para esto, empleamos la librería `kknn`, que realiza clasificación de usando un training set.

```
> # install.packages("kknn")
> library(kknn)
```

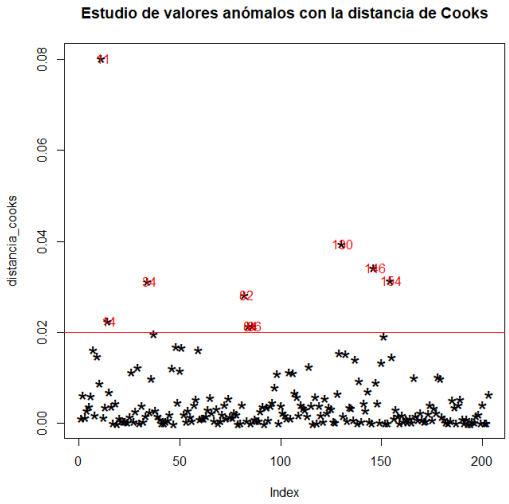

Figura 1: Gráfico de valores anómalos

Cargamos el data set y la muestra.

```
> Data<-iris
> Muestra <- sample(1:150, 50)
```

Separamos la muestra en set de test y set de aprendizaje.

```
> conjunto_test <- Data[Muestra, ]
> conjunto_aprendizaje <- Data[-Muestra, ]
```

Ejecutamos el algoritmo y entrenamos el modelo.

```
> modelo <- train.kknn(Species ~ ., data = conjunto_aprendizaje, kmax = 9)
```

Finalmente, evaluamos el modelo en el conjunto de test y comprobamos cómo rinde.

```
> resultado_prediccion <- predict(modelo, conjunto_test[, -5])
> # Matriz de confusión
> matriz_confusion <- table(conjunto_test[, 5], resultado_prediccion)
> print(matriz_confusion)

           resultado_prediccion
            setosa versicolor virginica
  setosa        24          1         0
  versicolor     0         13         1
  virginica      0          2         9

> # Precisión
> acierto <- (sum(diag(matriz_confusion)))/sum(matriz_confusion)
> cat("Acierto: ", acierto, "\n")
```

```
Acierto:  0.92

> # Error
> error <- 1 - acierto
> cat("Error: ", error, "\n")

Error:  0.08

> # Calidad de la clasificación en función del nº de vecinos
> plot(modelo)
```

Y obtenemos el siguiente gráfico del modelo:

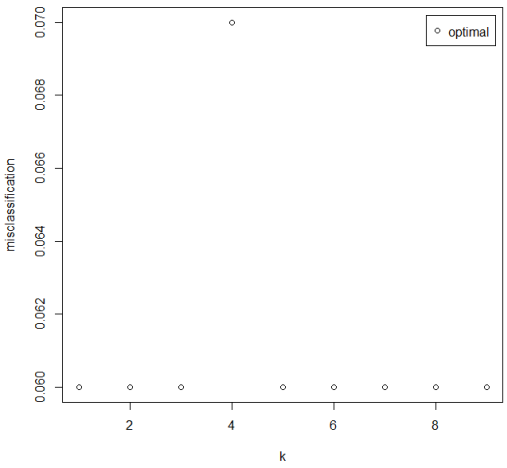

Figura 2: Gráfico de modelo

4. Ejercicio 3 - Análisis de detección de datos anómalos. Algoritmo LOF

Para este segundo ejercicio se deben realizar los mismo cálculos que para el ejercicio 1, pero con otras muestras y cambiando la forma en la que se calculan los resultados. Las muestras utilizadas para este ejercicio serán "muestra1.txt" y "muestra2.txt".

4.1. Apartado 1 - Árboles de decisión. Algoritmo Hunt

Para el primer apartado, a partir de una muestra con datos sobre las características de los vehículos, se pide realizar un análisis de clasificación de los datos sobre el atributo TipoVehiculo.

La clasificación se realizará a partir de los siguientes atributos, sie

5. Conclusiones

Mediante esta práctica hemos empleado herramientas que permiten la realización de análisis de clasificación para un conjunto de datos mediante R. Se ha aprendido a utilizar el algoritmo de construcción de árboles de decisión de Hunt y a mostrar e interpretar este. También, se ha aprendido a realizar un análisis de regresión lineal e interpretar los resultados obtenidos.