

Métodos multivariantes basados en distancias con R

Francesc Carmona
Departament d'Estadística

5 de marzo de 2007

1. Introducción

En este documento se resuelven con el programa estadístico R algunos de los problemas del curso de doctorado *Métodos multivariantes basados en distancias* de C.M. Cuadras[2]. Los enunciados de los problemas se encuentran en los apuntes de dicho curso.

Si se quiere aprender R desde el principio o practicar su utilización en la Estadística elemental un buen libro es el de J. Verzani[5]. Para profundizar en la teoría subyacente a la regresión se puede consultar, entre otros, el libro *Modelos lineales*[1]. Para estudiar modelos lineales avanzados con R se puede leer el libro de J.J. Faraway[3].

2. Regresión basada en distancias

2.1. Introducción de los datos

En el apartado 2.2 *Regresión DB en dimensión reducida* del curso de Cuadras[2] se explica un modelo de regresión con variables regresoras mixtas (cuantitativas, binarias y cualitativas) basado en la disimilaridad de Gower. Vamos a resolver el ejercicio 2.4 según este método y utilizando las funciones de R más adecuadas.

Los datos consisten en un conjunto de variables continuas (5), binarias (2) y cualitativas (3) que hacen de regresoras para una variable cuantitativa dependiente (Y). Se trata de los consumos de gasolina `mpg` (número de millas recorridas por galón) en función de las 10 características. El número de observaciones (automóviles) es 32. Estos datos se encuentran en la base de datos `mtcars` de R.

```
> data(mtcars)
> str(mtcars)
> help(mtcars)
```

Sin embargo y como ilustración, vamos a construir la base de datos desde el principio. En primer lugar procedemos a gravar los datos en un archivo de texto ASCII o ANSI. En mi caso un copia y pega directamente del archivo PDF al archivo de texto `gasolina.txt`. Todas las variables en columnas y como separador entre datos un espacio en blanco. En este caso, el separador decimal es un punto. En la primera fila del archivo ponemos unos nombres de variable de modo que las primeras filas del archivo `gasolina.txt` son:

```
Y c1 c2 c3 c4 c5 b1 b2 q1 q2 q3
21.0 160.0 110 3.90 2.620 16.46 0 1 6 4 4
21.0 160.0 110 3.90 2.875 17.02 0 1 6 4 4
22.8 108.0 93 3.85 2.320 18.61 1 1 4 4 1
...
```

Ahora ya podemos capturar los datos en R mediante la instrucción `read.table`.

```
> gasolina <- read.table("gasolina.txt", header = T, sep = " ")
> dim(gasolina)
> n <- dim(gasolina)[1]
> p <- dim(gasolina)[2] - 1
> colnames(gasolina)
```

Como todos los datos leídos son numéricos, de entrada para R todas las variables son numéricas. Vamos a corregir este malentendido convirtiendo las variables cualitativas en factores.

```
> gasolina[, 7:11] <- lapply(gasolina[, 7:11], as.factor)
> summary(gasolina)
```

Este resumen es distinto en función del tipo de variable, de modo que R ha comprendido que algunas variables son factores (cualitativas). De momento no se distingue entre binarias y multinivel.

En el data.frame mtcars las variables cualitativas ya son factores y este último paso no es necesario.

2.2. Cálculo de la disimilaridad

Para calcular la disimilaridad entre las observaciones vamos a utilizar la función `daisy` del paquete `cluster` que utiliza la disimilaridad de Gower cuando las variables son mixtas como en este caso.

```
> library(cluster)
> Y <- gasolina[, 1]
> Delta <- daisy(gasolina[, -1], type = list(asymm = c("b1", "b2")))
> class(Delta)
```

```
[1] "dissimilarity" "dist"
```

Delta es la distancia de Gower al cuadrado.

Observemos que el argumento `type` de `daisy` señala que las variables binarias son asimétricas, es decir, los casos de coincidencia (0,0) no cuentan. Esta es una condición de Gower que no es la opción por defecto en la función `daisy`.

La función `daisy` tiene en cuenta incluso variables con escala ordinal, pero si todas son numéricas utiliza la distancia euclídea o la `manhattan`.

Para calcular la distancia de Gower cuando todas las variables son numéricas podemos utilizar `vegdist` del paquete `vegan` o también `gdist` del paquete `mvpart`.

2.3. Análisis de coordenadas principales

A partir de la matriz de distancias $\Delta^{(1/2)}$ podemos aplicar la técnica de reducción de la dimensión de Gower o análisis de coordenadas principales. Se trata de un Multidimensional Scaling (MDS) clásico que R calcula con la función `cmdscale`.

```
> mds <- cmdscale(Delta^(1/2), k = n - 1, eig = TRUE)
> names(mds)
```

```
[1] "points" "eig"      "x"        "ac"        "GOF"
```

```
> round(mds$points[, 1], 4)
```

1	2	3	4	5	6	7	8	9	10	11
-0.1570	-0.1508	-0.3957	-0.0289	0.3241	-0.0379	0.3808	-0.2919	-0.2919	-0.1577	-0.1687
12	13	14	15	16	17	18	19	20	21	22
0.3457	0.3374	0.3315	0.3983	0.4035	0.4024	-0.4285	-0.4082	-0.4390	-0.1990	0.3091
23	24	25	26	27	28	29	30	31	32	
0.2956	0.3518	0.3437	-0.4259	-0.2306	-0.2306	0.1235	-0.0757	0.1322	-0.3619	

En el gráfico 1 se representan las 32 observaciones de las variables regresoras en dimensión 2 con las dos primeras coordenadas principales. Las instrucciones para generar el gráfico son:

```
> plot(mds$points[, 1], mds$points[, 2],
+      main = "Representación en dimensión reducida dim=2",
+      xlab = "Coordenada principal 1",
+      ylab = "Coordenada principal 2", pch = 19, col = "red")
> abline(h = 0)
> abline(v = 0)
```

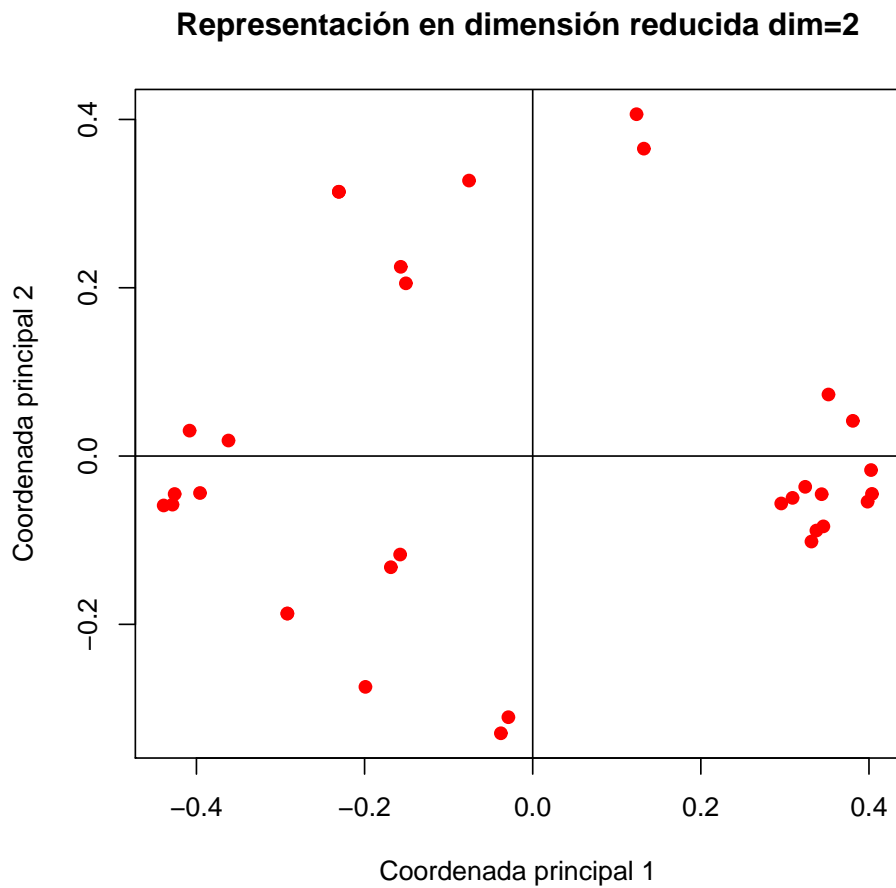


Figura 1: Representación de las observaciones con las dos primeras coordenadas principales.

La función `cmdscale` se queja por algún valor propio negativo porque utilizamos un número excesivo de coordenadas principales. Vamos a fijar el rango de la matriz **B** y calcular la matriz **X** que contiene todas las coordenadas principales para ese rango.

```
> m <- sum(mds$eig > 1e-15)
> mds <- cmdscale(Delta^(1/2), k = m, eig = TRUE)
> X <- mds$points
```

2.4. Selección de coordenadas principales como regresoras

Para seleccionar las coordenadas principales como regresoras se pueden utilizar varios criterios. En todo caso, parece razonable seleccionar las de mayor valor propio. Sin embargo, como vamos a utilizarlas en regresión conviene no olvidar alguna que tenga una correlación alta con la variable respuesta. Así pues, la decisión se basará en los valores propios y en las correlaciones al cuadrado que tenemos en los siguientes vectores:

```
> ValoresPropios <- mds$eig
> CorrCuadrado <- cor(Y, X)^2
```

Cuadras[2] sugiere un método de selección basado en una combinación de ambos criterios que llama predictibilidad. El vector de predictibilidades empieza en cero y va sumando la contribución de cada coordenada principal (ordenadas por sus respectivos valores propios) como producto de su valor propio y de la correlación al cuadrado. La serie de valores de predictibilidad se obtiene así:

```
> aux <- CorrCuadrado * ValoresPropios/sum(CorrCuadrado * ValoresPropios)
> c.pred <- c(0, cumsum(aux))
```

Ahora podemos dibujar el gráfico que nos permitirá decidir el número y las coordenadas principales seleccionadas.

```
> plot(0:m, 1 - c.pred, xlab = "Número de coordenadas principales",
+      ylab = "1 - Predictibilidad",
+      ylim = c(0, 0.2), type = "l")
> abline(v = 3, lty = 2, col = "blue")
```

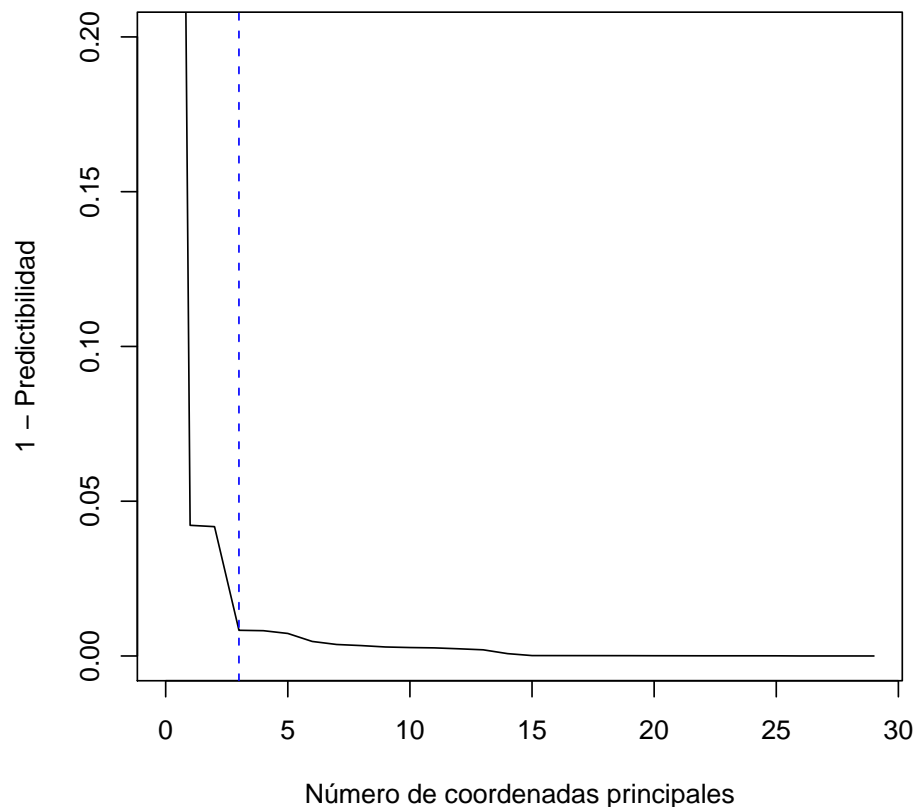


Figura 2: Gráfico de no predictibilidad para todas las coordenadas principales.

En la figura 2 se representa el gráfico de la serie `1-c.pred` que sugiere la selección de las 3 primeras coordenadas principales. Sin embargo, los datos concretos de sus valores propios y correlaciones al cuadrado ponen en duda la selección de la segunda coordenada.

```
> ValoresPropios[1:3]
[1] 2.9713331 1.1354895 0.8245557

> CorrCuadrado[1:3]
[1] 0.7209264007 0.0008456909 0.0908266557
```

En el siguiente apartado decidiremos.

2.5. Regresión con las coordenadas seleccionadas

La regresión con las tres primeras coordenadas principales se realiza como es habitual en R con la función `lm`.

```

> Xr <- X[, 1:3]
> rdb <- lm(Y ~ Xr)
> summary(rdb)

Call:
lm(formula = Y ~ Xr)

Residuals:
    Min       1Q   Median       3Q      Max
-5.9140 -1.6664 -0.1956  1.7813  6.1556

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.0906     0.4853  41.398 < 2e-16 ***
Xr1          -16.5291     1.5926 -10.379 4.21e-11 ***
Xr2           0.9158     2.5763   0.355 0.724904
Xr3          -11.1372     3.0233  -3.684 0.000974 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.745 on 28 degrees of freedom
Multiple R-Squared:  0.8126,    Adjusted R-squared:  0.7925
F-statistic: 40.47 on 3 and 28 DF,  p-value: 2.594e-10

```

El p-valor correspondiente a la segunda coordenada principal no es significativo de modo que, en este caso, podemos eliminarla del modelo.

El modelo de regresión definitivo con la primera y la tercera coordenada principal es:

```

> Xr <- X[, c(1, 3)]
> rdb <- lm(Y ~ Xr)
> summary(rdb)

Call:
lm(formula = Y ~ Xr)

Residuals:
    Min       1Q   Median       3Q      Max
-5.8971 -1.7287 -0.2628  1.7830  6.1018

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.0906     0.4779  42.036 < 2e-16 ***
Xr1          -16.5291     1.5684 -10.539 1.98e-11 ***
Xr2          -11.1372     2.9774  -3.741 0.000805 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.704 on 29 degrees of freedom
Multiple R-Squared:  0.8118,    Adjusted R-squared:  0.7988
F-statistic: 62.53 on 2 and 29 DF,  p-value: 3.045e-11

```

Aunque observamos un buen ajuste, seria necesario un análisis de los residuos como se puede ver en el apartado 9.4 del libro de Carmona[1] o el capítulo 4 del libro de Faraway[3].

También podemos hacer una regresión con los datos de la variable respuesta centrados. El modelo es más simple:

```

> Yc <- Y - mean(Y)
> rdbc <- lm(Yc ~ 0 + Xr)
> summary(rdbc)

```

```

Call:
lm(formula = Yc ~ 0 + Xr)

Residuals:
    Min       1Q   Median       3Q      Max
-5.8971 -1.7287 -0.2628  1.7830  6.1018

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
Xr1   -16.529      1.542  -10.719 8.87e-12 ***
Xr2   -11.137      2.927   -3.805 0.000651 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.658 on 30 degrees of freedom
Multiple R-Squared:  0.8118,    Adjusted R-squared:  0.7992
F-statistic: 64.68 on 2 and 30 DF,  p-value: 1.321e-11

```

2.6. Predicción de nuevos valores

Supongamos que se han observado los valores de todas las variables explicativas sobre un nuevo individuo. Como se sabe, es posible estimar las coordenadas principales para ese individuo en función de las distancias al cuadrado a los n individuos iniciales.

```

> gasolina[33, ] <- c(NA, 183.5, 151, 3.8, 3.908, 18.22, 1, 0, 4, 4, 2)
> d <- as.matrix(daisy(gasolina[, -1], type = list(asymm = c("b1", "b2"))))[-33, 33]
> b <- diag(X %*% t(X))
> x.new <- (1/2) * diag(ValoresPropios^(-1)) %*% t(X) %*% (b - d)
> y.pred <- t(c(1, x.new[c(1, 3)])) %*% rdb$coef
> y.pred

      [,1]
[1,] 24.29385

> y.pred <- mean(Y) + sum(x.new[c(1, 3)] * rdb$coef)
> y.pred

[1] 24.29385

```

Observemos que para calcular `x.new` utilizamos únicamente las coordenadas principales seleccionadas (primera y tercera). La predicción se puede hacer con el modelo sin centrar y con el modelo centrado. El resultado es el mismo.

2.7. Distancias euclídeas y no euclídeas

En el paquete `ade4` (Analysis of Environmental Data) de R hay un conjunto de funciones que nos pueden ayudar cuando utilizamos distancias. Para utilizar estas funciones deberemos instalar el paquete desde el CRAN.

```

> library(ade4)
> is.euclid(Delta^(1/2))

[1] TRUE

```

Puede ocurrir, por diversas razones, que la distancia que utilizemos no sea euclídea. Entonces la matriz B tendría algunos valores propios negativos y por lo tanto nos encontraríamos con algunas variables (columnas de X) con “varianzas negativas”. Consecuentemente el modelo de regresión DB no funcionaría correctamente. Para solucionar este inconveniente, podemos considerar la transformación q -aditiva de la distancia.

$$\tilde{\delta}_{ij}^2 = \begin{cases} 0 & i = j \\ \delta_{ij}^2 + 2a & i \neq j \end{cases}$$

La función `lingoes` usa la menor constante positiva a que transforma la distancia en euclídea. Otra posibilidad para hacer euclídea una distancia es utilizar la función `cailliez` del mismo paquete. Por otra parte, cuando la distancia es euclídea pero algunos valores propios son muy pequeños, el ajuste de la regresión DB puede ser muy pobre. La transformación q-aditiva puede mejorar la regresión. Existe una cierta analogía con la regresión cresta¹ (Hoel y Kennard, 1970).

Referencias

- [1] F. Carmona, *Modelos lineales*, Publicacions UB, 2005.
- [2] C.M. Cuadras, *Métodos multivariantes basados en distancias*. Curso de doctorado, enero 2007.
- [3] J.J. Faraway, *Linear Models with R*, Chapman & Hall/CRC, 2004.
- [4] P. Murrell, *R Graphics*, Chapman & Hall/CRC, 2005.
- [5] J. Verzani, *Using R for Introductory Statistics*. Chapman & Hall/CRC, 2004.

¹Ridge regression.