Modelos No Paramétricos

Javier Adanaque

17 de Abril, 2017

Resumen

Se revisan, rápidamente, algunos modelos no paramétricos aprendidos en clase

1. Introducción

El objetivo del presente documento es aplicar los métodos No Paramétricos aprendidos en clase. Para ello se está usando una base de datos con diferentes características de las viviendas y suburbios/ciudades en Boston, data otorgada en clase junto a otros datasets.

El foco será en la aplicación de los métodos, sin dar mayor detalle sobre el marco teórico o utilidad.

Nos concentraremos en conocer cómo se relaciona el valor de las viviendas en Boston con respecto al número de habitaciones.

2. Data

Para el análisis se está usando la data `bostonvivienda.txt`, ubicada entre los datasets distribuidos para la clase.

```
valor_viviendas <- read.table("data/bostonvivienda.txt",
                              header = TRUE, stringsAsFactors = FALSE)
valor_viviendas[1:6, c(1:3, 6, 7, 10, 13, 14)]
```

crim	zn	indus	rm	edad	impuesto	lstat	medv
0.00632	18	2.31	6.575	65.2	296	4.98	24.0
0.02731	0	7.07	6.421	78.9	242	9.14	21.6
0.02729	0	7.07	7.185	61.1	242	4.03	34.7
0.03237	0	2.18	6.998	45.8	222	2.94	33.4
0.06905	0	2.18	7.147	54.2	222	5.33	36.2
0.02985	0	2.18	6.430	58.7	222	5.21	28.7

Como se puede observar en las primeras seis observaciones (y algunas columnas), nos muestra diferentes características sobre los suburbios, incluido el valor medio de las viviendas en el suburbio (ver última columna).

Descripción de variables:

- `crim`: tasa de delincuencia per cápita por ciudad.

- `zn`: proporción de suelo residencial dividido en zonas para lotes de más de 25,000 pies cuadrados.
- `indus`: proporción de acres de negocios no minoristas por la ciudad.
- `chas`: variable ficticia (dummy) Charles River (1 si sale de las vías fluviales; 0 en caso contrario).
- `nox`: concentración de óxidos de nitrógeno (partes por 10 millones).
- `rm`: número promedio de habitaciones por vivienda.
- `edad`: proporción de unidades ocupadas por sus propietarios construidas antes de 1940.
- `dis`: media ponderada de las distancias a cinco centros de empleo de Boston.
- `rad`: Índice de la accesibilidad a las autopistas radiales.
- `impuesto`: tasa de impuestos a la propiedad por el valor total por $ 10,000.
- `ptratio`: proporción de alumnos por profesor por ciudad.
- `negro`: 1000(Bk - 0,63)2, donde Bk es la proporción de negros por la ciudad.
- `lstat`: estatus más bajo de la población (por ciento).
- `medv`: valor mediano de las viviendas ocupadas por sus propietarios en $ 1000s.

3. Modelamiento con Métodos No Paramétricos

3.1. Regresorgrama

```
minimo=min(valor_viviendas[,6])  # Variable: "rm"
maximo=max(valor_viviendas[,6])
plot(valor_viviendas[, 6], valor_viviendas[, 14], col='tomato', pch=19,
     xlab = "rm", ylab = "medv")
particion=c(minimo, 4.2, 4.8, 5.2, 5.9, 6.2, 6.7, 7.2, 7.9, 8.2, maximo)
n1=dim(valor_viviendas)
n2=length(particion)
s=rep(0,n2-1)
x1=rep(0,2)
y1=rep(0,2)

for (j in 1:(n2-1)) {
    suma=0
    cont=0
  for (i in 1:n1[1]) {
    if (valor_viviendas[i,"rm"]>=particion[j] &
        valor_viviendas[i,"rm"]<particion[j+1]) {
      suma=suma+valor_viviendas[i,"medv"]
      cont=cont+1
    }
  }
  s[j]=suma/cont
  x1[1]=particion[j]
  x1[2]=particion[j+1]
  y1[1]=s[j]
```

```
  y1[2]=s[j]
  lines(x1,y1,type="l",col="darkred")
}

for (j in 2:(n2-1)) {

  x1[1]=particion[j]
  x1[2]=particion[j]
  y1[1]=s[j]
  y1[2]=s[j-1]
  lines(x1,y1,type="l",col="darkred")
}
```

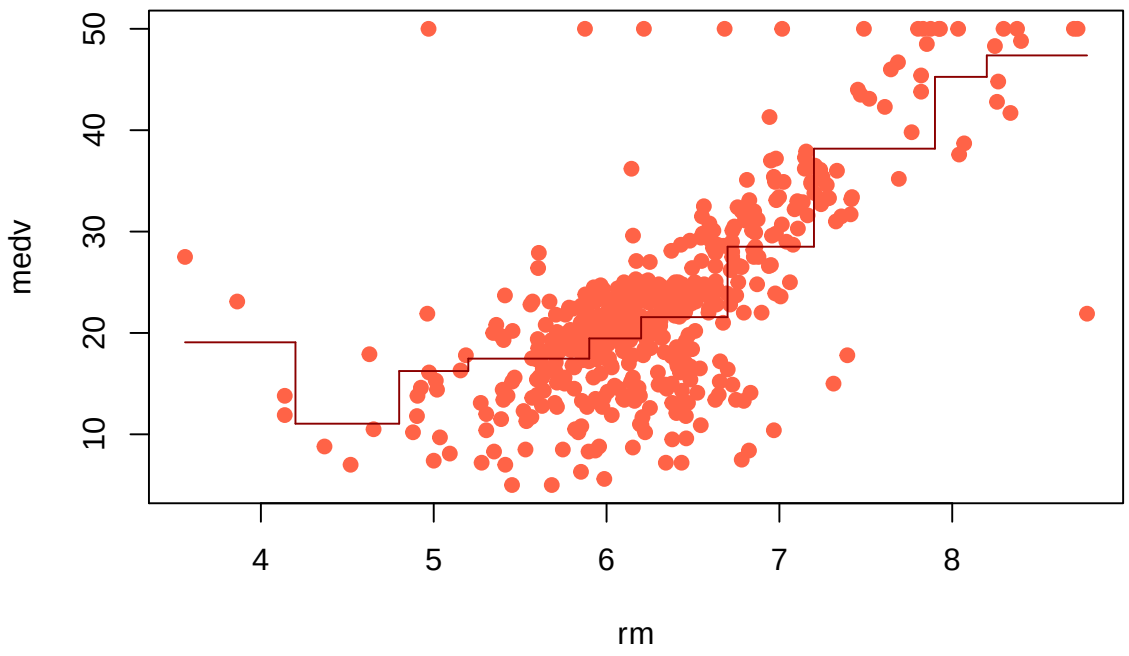

Figura 1: Regresorgrama

3.2. Running Means

```
valor_viviendas <- valor_viviendas[order(valor_viviendas$rm), ]
k=30
n=dim(valor_viviendas)
s=rep(0,n[1]-2*k)

for (i in (k+1):(n[1]-k)) {
  j=seq(i-k, i+k)
  s[i]=mean(valor_viviendas[j,"medv"])
}

i = seq(k+1,n[1]-k)
plot(valor_viviendas[, 6], valor_viviendas[, 14], col='tomato', pch=19,
     xlab = "rm", ylab = "medv")
lines(valor_viviendas[i,6],s[i],type = "s", col = "blue")
```

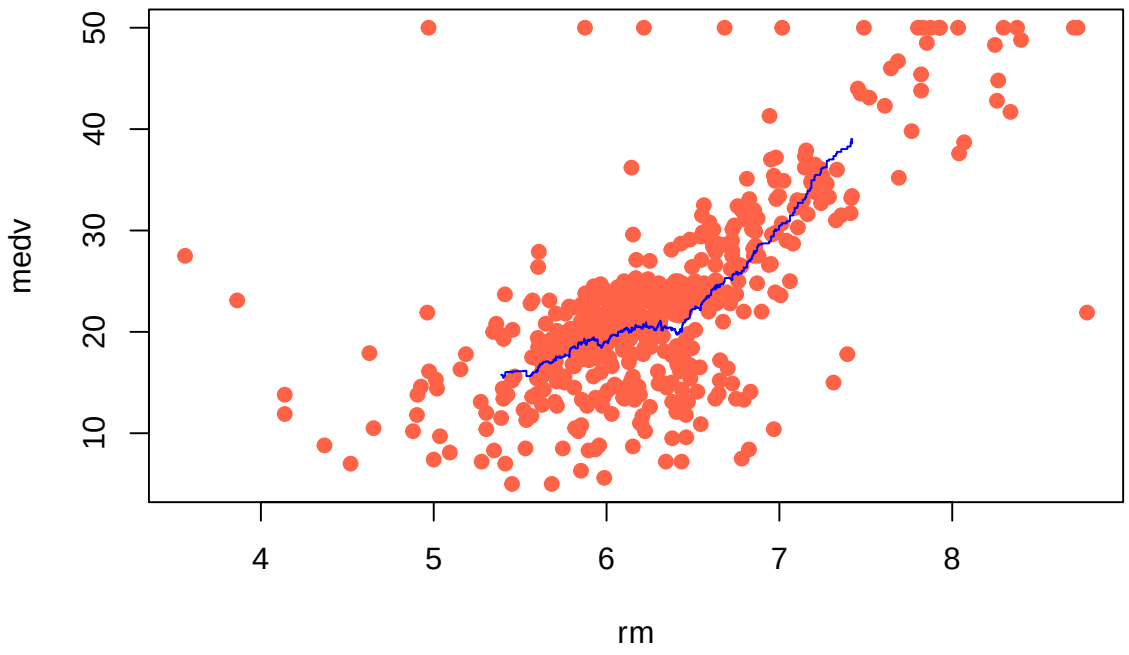

Figura 2: Running Means with k = 30. La línea es más "suave". Ojo: la línea no se encuentra definida en los extremos.

3.3. Running Medians

```
k=30
n=dim(valor_viviendas)
s=rep(0,n[1]-2*k)

for (i in (k+1):(n[1]-k)) {
  j=seq(i-k, i+k)
  s[i]=median(valor_viviendas[j,"medv"])
}

i = seq(k+1,n[1]-k)
plot(valor_viviendas[, 6], valor_viviendas[, 14], col='tomato', pch=19,
     xlab = "rm", ylab = "medv")
lines(valor_viviendas[i,6],s[i],type = "s", col = "blue")
```

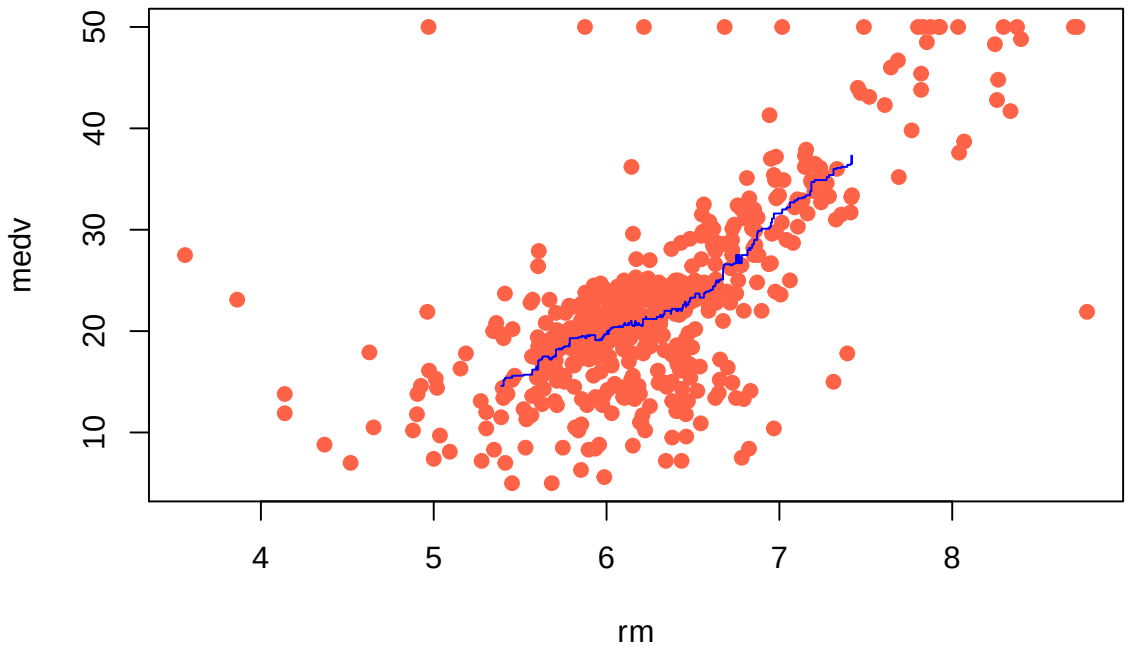

Figura 3: Running Medians with k = 30. La línea es más "suave" aún; esto se debe a la menor influencia de los datos atípicos sobre nuestro estimado.

3.4. Running Lines

```
k=30
n <- dim(valor_viviendas)
cont <- 0
x1 <- rep(0,n[1]-2*k)
y1 <- rep(0,n[1]-2*k)

for (i in (k+1):(n[1]-k)) {
  j=seq(i-k,i+k)
  modelo <- lm(valor_viviendas[j, 14]~valor_viviendas[j, 6])
  cont <- cont+1
  x1[cont] <- valor_viviendas[i,6]
  y1[cont] <- modelo$coefficients[1]+modelo$coefficients[2]*x1[cont]
}

plot(valor_viviendas[, 6], valor_viviendas[, 14], col='tomato', pch=19,
     xlab = "rm", ylab = "medv")
lines(x1, y1, type = "s", col = "blue")
```

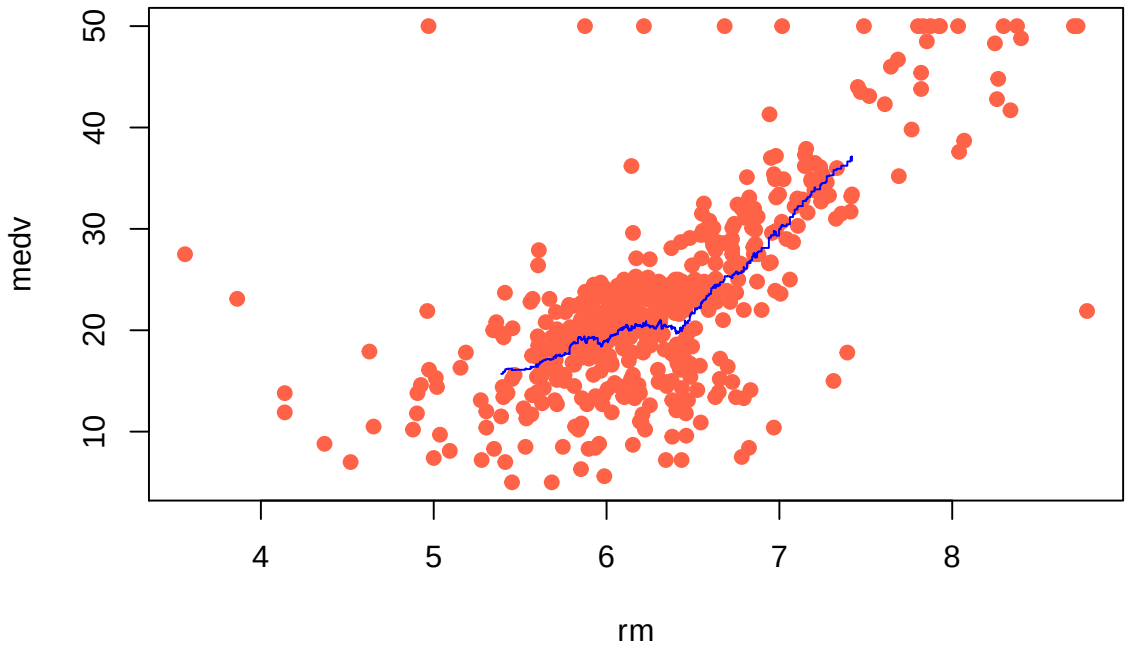

Figura 4: Running Lines with k = 30.