AEM - Tema 6 - Regresión y Clasificación mediante KNN. Trabajo de evaluación.

Jerónimo Carranza Carranza

20 de febrero de 2017

Contents

1 Conjunto de datos para el estudio

Conjunto de datos: datawork.csv

- Dimensiones: n=4000 casos, m=42 variables
- Variables: “clasobj” “varobj” “x01” “x02” .... “x40”.
- clasobj : variable nominal con modalidades: AA, BB, CC, DD.
- varobj : variable continua
- atributos: x01, ..., x40.

```
datawork <- read.table("datawork.csv", header=TRUE, sep=";")
str(datawork)
```

```
## 'data.frame':    4000 obs. of  42 variables:
##  $ clasobj: Factor w/ 4 levels "AA","BB","CC",..: 3 1 4 1 2 3 2 2 2 4 ...
##  $ varobj : num  12.6 25.1 28.8 26.6 18.5 ...
##  $ x01    : num  0.64 0.7 0.45 0.15 0.7 0.37 0.36 0.5 0.5 0.64 ...
##  $ x02    : num  0.68 0.72 0.71 0.56 0.54 0.75 0.36 0.52 0.62 0.63 ...
##  $ x03    : num  25.1 16.1 57.8 17.8 17.5 ...
##  $ x04    : num  33.4 11 48.8 10 25.5 ...
##  $ x05    : num  103.2 38.3 148.2 30.3 82.4 ...
##  $ x06    : num  1.08 1.62 0.34 0.86 1.04 0.92 1.16 1.84 0.94 0.7 ...
##  $ x07    : num  80.3 57.3 174.5 57.4 57.5 ...
##  $ x08    : num  14.8 30.7 50.7 167.5 40.1 ...
##  $ x09    : num  18.09 17.13 2.72 23.09 1.7 ...
##  $ x10    : num  0.2 0.1 0.46 0.36 0.19 0.15 0.1 0.02 0.05 1.8 ...
##  $ x11    : num  2.78 1.73 4.81 2.94 2.22 ...
##  $ x12    : num  17.6 12 21.6 12.7 15.8 ...
##  $ x13    : num  18.9 15.5 21.3 12.9 17.4 ...
##  $ x14    : num  21 13 23.7 12 15.6 ...
##  $ x15    : num  18.7 12.6 20.5 12.8 14.6 ...
##  $ x16    : num  18.5 12.2 23.2 12.2 15.8 ...
##  $ x17    : num  18 12.7 21.1 12.9 17.4 ...
##  $ x18    : num  19.3 13.8 20.6 15.1 13.7 ...
##  $ x19    : num  18.6 12.1 21.9 12.6 16.7 ...
##  $ x20    : num  18.5 14.3 22.1 12.4 18.1 ...
##  $ x21    : num  18.5 10.4 24.9 12.6 15.6 ...
##  $ x22    : num  17.9 14.1 22.5 13.8 16.2 ...
##  $ x23    : num  18.3 11.5 22.1 14.2 15.2 ...
##  $ x24    : num  19.6 11.6 23.8 11.6 14.8 ...
##  $ x25    : num  18.7 12.3 22 13.3 15.7 ...
##  $ x26    : num  19.1 14.3 21.2 13.8 14.8 ...
##  $ x27    : num  20.7 16.9 20.7 15.5 14.5 ...
##  $ x28    : num  19.8 13.1 21.3 15 18.5 ...
##  $ x29    : num  0.61 0.54 0.32 0.46 0.3 0.25 0.15 0.78 0.11 0.16 ...
##  $ x30    : int  21 14 20 11 17 17 13 15 15 24 ...
##  $ x31    : num  145.5 83.3 189.1 91.3 119.3 ...
##  $ x32    : num  133.9 99.6 172.9 96.7 122.2 ...
##  $ x33    : num  147 94.1 171.7 108.3 112.8 ...
##  $ x34    : num  170 113 193 110 125 ...
##  $ x35    : num  142 92.1 168.1 103.9 124.9 ...
##  $ x36    : num  82.7 60.2 86.4 56.1 62.5 ...
##  $ x37    : num  50.07 37.2 9.47 24.95 12.78 ...
```

```
##  $ x38    : num  397 226 515 245 345 ...
##  $ x39    : num  393 276 451 262 296 ...
##  $ x40    : num  415 313 505 314 358 ...
```

```
summary(datawork)
```

```
##  clasobj       varobj            x01              x02
##  AA: 626   Min.   : 10.49   Min.   :0.0700   Min.   :0.1300
##  BB:1315   1st Qu.: 14.35   1st Qu.:0.4000   1st Qu.:0.5100
##  CC:1372   Median : 19.34   Median :0.5000   Median :0.6000
##  DD: 687   Mean   : 20.86   Mean   :0.4989   Mean   :0.5977
##            3rd Qu.: 25.48   3rd Qu.:0.6000   3rd Qu.:0.7000
##            Max.   :713.81   Max.   :0.9400   Max.   :0.9300
##       x03              x04              x05              x06
##  Min.   :  0.94   Min.   :-0.64   Min.   :  2.96   Min.   :0.080
##  1st Qu.: 15.55   1st Qu.:21.56   1st Qu.: 69.82   1st Qu.:0.660
##  Median : 25.98   Median :29.54   Median : 94.06   Median :0.940
##  Mean   : 30.42   Mean   :29.48   Mean   : 93.91   Mean   :1.015
##  3rd Qu.: 40.17   3rd Qu.:37.34   3rd Qu.:117.63   3rd Qu.:1.270
##  Max.   :166.17   Max.   :62.67   Max.   :194.56   Max.   :3.260
##       x07              x08                x09                x10
##  Min.   :  3.61   Min.   :    0.02   Min.   :   0.000   Min.   : 0.0000
##  1st Qu.: 51.86   1st Qu.:    5.58   1st Qu.:   1.147   1st Qu.: 0.1000
##  Median : 83.12   Median :   21.86   Median :   2.020   Median : 0.2500
##  Mean   : 96.34   Mean   :  162.95   Mean   :   4.502   Mean   : 0.4398
##  3rd Qu.:125.90   3rd Qu.:   80.82   3rd Qu.:   3.480   3rd Qu.: 0.5700
##  Max.   :502.70   Max.   :68676.15   Max.   :1097.640   Max.   :24.1800
##       x11               x12              x13              x14
##  Min.   :  0.490   Min.   :10.57   Min.   : 9.26   Min.   :10.21
##  1st Qu.:  2.158   1st Qu.:15.45   1st Qu.:15.45   1st Qu.:15.40
##  Median :  2.960   Median :17.66   Median :17.60   Median :17.65
##  Mean   :  4.197   Mean   :17.56   Mean   :17.58   Mean   :17.57
##  3rd Qu.:  4.152   3rd Qu.:19.70   3rd Qu.:19.70   3rd Qu.:19.69
##  Max.   :688.530   Max.   :25.16   Max.   :24.86   Max.   :24.64
##       x15              x16              x17              x18
##  Min.   :10.15   Min.   : 9.65   Min.   :10.13   Min.   :10.47
##  1st Qu.:15.42   1st Qu.:15.37   1st Qu.:15.46   1st Qu.:15.43
##  Median :17.58   Median :17.55   Median :17.60   Median :17.62
##  Mean   :17.55   Mean   :17.55   Mean   :17.57   Mean   :17.57
##  3rd Qu.:19.68   3rd Qu.:19.66   3rd Qu.:19.67   3rd Qu.:19.67
##  Max.   :24.73   Max.   :25.50   Max.   :25.15   Max.   :25.22
##       x19              x20              x21              x22
##  Min.   :10.06   Min.   :10.26   Min.   : 9.59   Min.   : 9.44
##  1st Qu.:15.44   1st Qu.:15.37   1st Qu.:15.45   1st Qu.:15.39
##  Median :17.61   Median :17.66   Median :17.58   Median :17.63
##  Mean   :17.58   Mean   :17.58   Mean   :17.54   Mean   :17.57
##  3rd Qu.:19.72   3rd Qu.:19.67   3rd Qu.:19.63   3rd Qu.:19.71
##  Max.   :25.08   Max.   :24.95   Max.   :25.09   Max.   :24.99
##       x23              x24              x25              x26
##  Min.   : 9.99   Min.   : 9.63   Min.   : 9.03   Min.   : 9.89
##  1st Qu.:15.47   1st Qu.:15.41   1st Qu.:15.44   1st Qu.:15.39
##  Median :17.59   Median :17.50   Median :17.64   Median :17.59
##  Mean   :17.57   Mean   :17.55   Mean   :17.55   Mean   :17.59
##  3rd Qu.:19.66   3rd Qu.:19.70   3rd Qu.:19.75   3rd Qu.:19.73
```

```
##  Max.   :25.29   Max.   :25.15   Max.   :24.95   Max.   :25.05
##       x27             x28             x29             x30
##  Min.   : 5.91   Min.   : 7.81   Min.   :0.020   Min.   :10.00
##  1st Qu.:15.23   1st Qu.:15.92   1st Qu.:0.250   1st Qu.:14.00
##  Median :17.74   Median :18.41   Median :0.360   Median :16.00
##  Mean   :17.78   Mean   :18.47   Mean   :0.375   Mean   :16.76
##  3rd Qu.:20.36   3rd Qu.:21.14   3rd Qu.:0.490   3rd Qu.:19.00
##  Max.   :47.94   Max.   :28.27   Max.   :0.920   Max.   :29.00
##       x31             x32             x33             x34
##  Min.   : 75.25   Min.   : 79.55   Min.   : 81.6   Min.   : 83.37
##  1st Qu.:119.00   1st Qu.:118.01   1st Qu.:118.2   1st Qu.:132.24
##  Median :135.91   Median :134.82   Median :135.5   Median :151.30
##  Mean   :135.30   Mean   :133.66   Mean   :134.2   Mean   :150.74
##  3rd Qu.:151.52   3rd Qu.:149.53   3rd Qu.:150.0   3rd Qu.:169.27
##  Max.   :191.41   Max.   :185.77   Max.   :189.7   Max.   :241.19
##       x35             x36             x37             x38
##  Min.   : 75.3   Min.   : 35.80   Min.   :-26.19   Min.   :203.3
##  1st Qu.:119.4   1st Qu.: 63.55   1st Qu.: 10.10   1st Qu.:326.0
##  Median :136.8   Median : 73.68   Median : 19.36   Median :374.3
##  Mean   :136.5   Mean   : 73.64   Mean   : 19.57   Mean   :372.6
##  3rd Qu.:154.4   3rd Qu.: 83.84   3rd Qu.: 28.77   3rd Qu.:418.2
##  Max.   :197.3   Max.   :110.56   Max.   :206.12   Max.   :534.5
##       x39             x40
##  Min.   :210.2   Min.   :226.3
##  1st Qu.:312.5   1st Qu.:349.1
##  Median :357.7   Median :398.4
##  Mean   :355.8   Mean   :397.5
##  3rd Qu.:397.9   3rd Qu.:446.0
##  Max.   :493.2   Max.   :553.8
```

2 Problema de Clasificación

Determinación de un clasificador basado en kNN ponderado para la variable objetivo "clasobj" con los atributos x01...x40.

- Seleccionar aleatoriamente un conjunto test de tamaño n/3 y un conjunto de aprendizaje de tamaño 2n/3.

```
set.seed(123456789)
n = dim(datawork)[1]
test = sample(1:n, size = round(n/3), replace = FALSE, prob = rep(1/n, n))
datawork.test = datawork[test,-2] # no incluye la variable varobj
datawork.apre = datawork[-test,-2] # no incluye la variable varobj
# datawork.apre.summary = summary(datawork.apre)
# datawork.test.summary = summary(datawork.test)
# rbind(datawork.apre.summary,'-----test------',datawork.test.summary)
```

- Con el conjunto de aprendizaje, selecciona el mejor núcleo y el mejor k (entre 1 y 20) a través de validación cruzada.

```
library(kknn)
datawork.clasif.1 = train.kknn(clasobj ~ ., datawork.apre, kmax = 20,
   kernel = c("triangular", "rectangular", "epanechnikov", "optimal",
              "biweight", "triweight", "cos", "inv", "gaussian"))
datawork.clasif.1
```

```
##
## Call:
## train.kknn(formula = clasobj ~ ., data = datawork.apre, kmax = 20,     kernel = c("triangular", "rec
##
## Type of response variable: nominal
## Minimal misclassification: 0.01424822
## Best kernel: rectangular
## Best k: 12
```

- Aplicar el clasificador óptimo obtenido para clasificar los casos del conjunto test y obtener una medida del error de clasificación y la tabla de confusión asociada.

```
datawork.clasif.1.confusion = addmargins(table(predict(
  datawork.clasif.1,datawork.test), datawork.test$clasobj))
library(knitr)
kable(datawork.clasif.1.confusion, caption = "Matriz de Confusión")
```

Table 1: Matriz de Confusión

	AA	BB	CC	DD	Sum
AA	205	0	1	7	213
BB	2	423	7	2	434
CC	1	3	434	1	439
DD	1	1	1	244	247
Sum	209	427	443	254	1333

A partir de la matriz de confusión pueden calcularse varias medidas del error de clasificación; la principal, la proporción de casos (test) incorrectamente clasificados.

```
sumNoError = 0
for (i in (1:4)){
  sumNoError = sumNoError + datawork.clasif.1.confusion[i,i]
}
pError = (1-sumNoError/dim(datawork.test)[1])
'Error de clasificación: '
```

```
## [1] "Error de clasificación: "
```

```
pError
```

```
## [1] 0.02025506
```

3 Problema de Regresión

3.1 kNN ponderado

Determinación de un predictor basado en kNN ponderado para la variable objetivo "varobj" con los atributos x01...x40.

```
datawork.vtest = datawork[test,-1] # no incluye la variable clasobj
datawork.vapre = datawork[-test,-1] # no incluye la variable clasobj
```

- Con el conjunto de aprendizaje, selecciona el mejor núcleo y el mejor k (entre 1 y 20) a través de validación cruzada.

```
datawork.kknn = train.kknn(varobj ~ ., datawork.vapre, kmax = 20,
   kernel = c("triangular", "rectangular", "epanechnikov", "optimal",
              "biweight", "triweight", "cos", "inv", "gaussian"))
datawork.kknn
```

```
##
## Call:
## train.kknn(formula = varobj ~ ., data = datawork.vapre, kmax = 20,     kernel = c("triangular", "rec
##
## Type of response variable: continuous
## minimal mean absolute error: 1.752477
## Minimal mean squared error: 112.964
## Best kernel: optimal
## Best k: 2
```

- Aplicar el predictor óptimo obtenido para predecir los casos del conjunto test y obtener una medida del error de predicción.

```
datawork.kknn.predict = predict(datawork.kknn, datawork.vtest)
summary(datawork.kknn.predict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.16   14.24   19.23   20.29   25.42  147.80
```

```
datawork.kknn.error =
  sqrt ((datawork.vtest$varobj - datawork.kknn.predict)**2)
summary(datawork.kknn.error)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.000238  0.325400  0.685600  1.637000  1.480000 27.720000
```

```
"Error Cuadrático Medio"
```

```
## [1] "Error Cuadrático Medio"
```

```
datawork.kknn.ECM = mean(datawork.kknn.error**2)
datawork.kknn.ECM
```

```
## [1] 12.16573
```

```
# datawork.kknn.ECM =
#   sum(datawork.kknn.error**2)/dim(datawork.vtest)[1]
# datawork.kknn.ECM

# test_reg = lm(datawork.vtest$varobj ~ datawork.kknn.predict)
# summary(test_reg)
```

3.2 kNN aleatorio

Determinación de un predictor basado en kNN aleatorio para la variable objetivo "varobj" con los atributos x01...x40.

```
library(rknn)
```

```
## Loading required package: gmp
```

```
##
## Attaching package: 'gmp'
```

```
## The following objects are masked from 'package:base':
##
##     %*%, apply, crossprod, matrix, tcrossprod
```

```
datawork.vapre.norm = data.frame(normalize.softmax(datawork.vapre))
datawork.vtest.norm = data.frame(normalize.softmax(datawork.vtest))

# Elección de r (número de clasificadores / regresores)

p = (dim(datawork.vapre)[2]-1)

datawork.vapre.r.binomial =
  r(p,m = floor(sqrt(p)),eta = 0.99,method = "binomial")
datawork.vapre.r.poisson =
  r(p,m = floor(sqrt(p)),eta = 0.99,method = "poisson")

datawork.vapre.r.binomial
```

```
## [1] 52
```

```
datawork.vapre.r.poisson
```

```
## [1] 52
```

```r
# Aplicación rkNNReg

datawork.rknn =
  rknnReg(datawork.vapre.norm, datawork.vtest.norm,
          y=datawork.vapre$varobj, k = 1, r=60, seed=123456789)

datawork.rknn$k
```

```
## [1] 1
```

```r
datawork.rknn$n
```

```
## NULL
```

```r
datawork.rknn$mtry
```

```
## [1] 6
```

```r
datawork.rknn.error =
  sqrt ((datawork.vtest$varobj - datawork.rknn$pred)**2)
summary(datawork.rknn.error)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3632  0.8272  1.3760  1.4820 38.3700
```

```r
"Error Cuadrático Medio"
```

```
## [1] "Error Cuadrático Medio"
```

```r
datawork.rknn.ECM = mean(datawork.rknn.error**2)
datawork.rknn.ECM
```

```
## [1] 8.453614
```

3.3 Estudio comparativo

Realizar un estudio comparativo entre ambos resultados

Se comparan mediante el error (test) cuadrático medio de cada modelo; kknn y rknn.

```r
datawork.compara =
  cbind(kknn = datawork.kknn.ECM,
        rknn = datawork.rknn.ECM)
rownames(datawork.compara) = 'Error Cuadrático Medio'

library(knitr)
kable(datawork.compara, caption = "Comparación de Modelos")
```

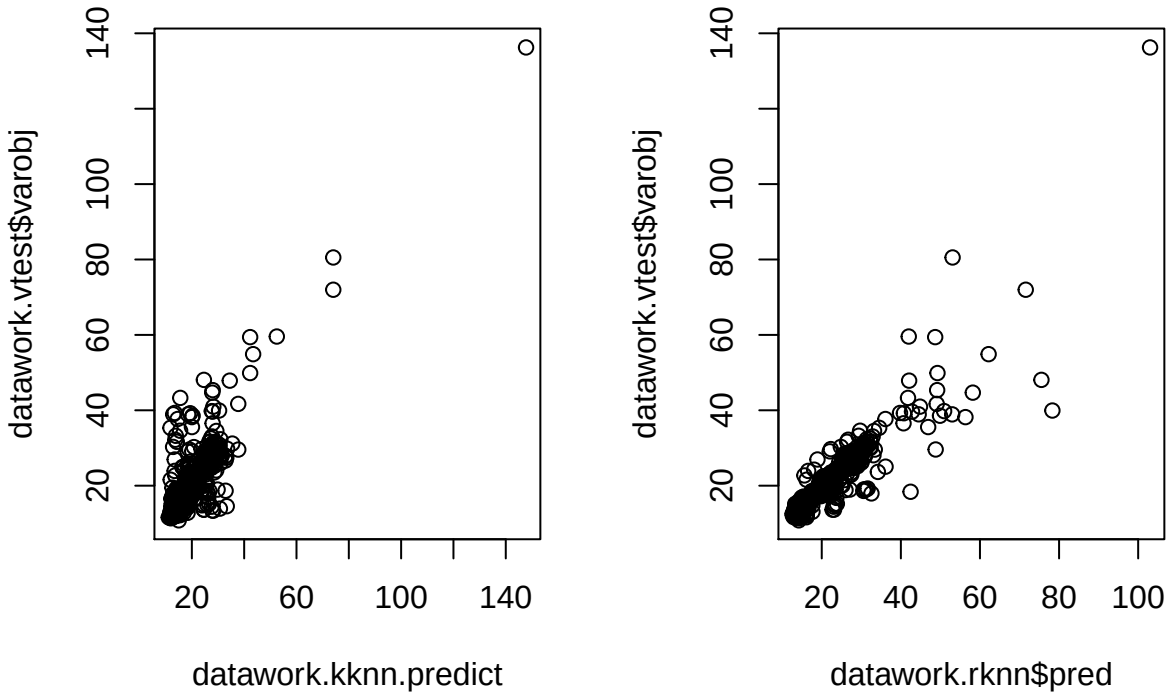
Table 2: Comparación de Modelos

	kknn	rknn
Error Cuadrático Medio	12.16573	8.453614

Los resultados de ECM muestran que el modelo rknn proporciona predicciones más ajustadas a los valores de la variable respuesta que el modelo kknn.

Se representa finalmente las predicciones de cada modelo frente a los valores observados de la variable respuesta en el conjunto test.

```
par(mfrow = c(1,2))
plot(datawork.kknn.predict,datawork.vtest$varobj)
plot(datawork.rknn$pred,datawork.vtest$varobj)
```




```
par(mfrow = c(1,1))
```