

**Questão 1:**

Qual a acurácia média (na validação de fora)

**Solução:**

A acurácia media calculada e exibida na linha **99** do script em R (ver Anexo I) foi de **0.8696912**.

**Questão 2:**

Quais são os valores de C e gamma a serem utilizados no classificador final(fazer o *3-fold* no conjunto todo)

**Solução:**

O valores de C e gamma obtidos pelo script em R (ver Anexo I), após o *cross-validation*, apresentados na linha **100**, foram respectivamente  $2^2$  e  $2^{-15}$ . A acurácia utilizando esses hiperparâmetros, após do *3-fold* sobre todo o conjunto de dados, foi de **0.864665**.

**Anexo I: Script fonte em R**

```
1 # -----
2 # Description:
3 #   Solutions for activity 2 (MO444)
4 #
5 # Version: 1.0
6 #
7 # Author:
8 #   Luiz Alberto , gomes.luiz@gmail.com
9 #
10 # History:
11 #   Sep 26th, 2016 started
12 #   Sep 27th, 2016 updated
13 #   Sep 28th, 2016 updated
14 #   Sep 29th, 2016 updated.
15 #   Sep 30th, 2016 updated.
16 #
17 # To do:
18 #   -
19 # -----
20 if (!require(caret))
21   install.packages('caret')
22 if (!require(e1071))
23   install.packages('e1071')
24
25 library('caret')
26 library('e1071')
27
28 # initializes execution environment.
29 rm(list=ls())
30 setwd('~\\Workspace\\doutorado\\disciplinas\\mo444b\\atividades\\2')
31
32 # reads raw data from file.
33 raw.data <- read.csv(file = './data/data1.csv', header = TRUE, sep = ',')
34
35 # selects 80% of raw data for train.
36 raw.data.train <- raw.data[trunc(nrow(raw.data) * 0.8), ]
37
38 # initializes cost and gamma grid.
39 grid.costs <- c(2 ** -5 , 2 ** -2 , 2 ** 0 , 2 ** 2, 2 ** 5)
```

```
40 grid.gammas <- c(2 ** -15, 2 ** -10, 2 ** -5, 2 ** 0, 2 ** 5)
41
42 # performs the external cross-validation using 5-folds methods.
43 external.folds <- createFolds(raw.data.train, k = 5, returnTrain = TRUE)
44 external.final.accuracy <- 0
45 for (e in external.folds) {
46   external.train.data <- raw.data[e, ]
47   external.test.data <- raw.data[-e, ]
48
49 # performs the internal cross-validation using 3-folds methods.
50 internal.folds <- createFolds(external.train.data, k = 3, returnTrain = TRUE)
51 internal.max.accuracy <- 0
52 for (i in internal.folds) {
53   internal.train.data <- external.train.data[i,]
54   internal.test.data <- external.train.data[-i,]
55
56 # searches in hyperparameters grid for the maximum accuracy.
57 for (cost.pos in 1:5) {
58   for (gamma.pos in 1:5) {
59
60     # uses hyperparameters in grid to train model.
61     internal.svm.fit <-
62       svm(clase ~ .,
63          data = internal.train.data,
64          cost = grid.costs[cost.pos],
65          gama = grid.gammas[gamma.pos],
66          kernel="radial")
67
68     # predicts with test data with internal svm model.
69     internal.svm.probs <- predict(internal.svm.fit, internal.test.data)
70     internal.svm.pred <- ifelse(internal.svm.probs > 0.5, 1, 0)
71     internal.svm.error <- mean(internal.svm.pred != internal.test.data$clase)
72     internal.accuracy <- 1 - internal.svm.error
73
74     if ( internal.accuracy > internal.max.accuracy ){
75       internal.max.accuracy <- internal.accuracy
76       h.max.cost <- grid.costs[cost.pos]
77       h.max.gamma <- grid.gammas[gamma.pos]
78     }
79   }
80 }
81 }
82 # uses hyperparameters to train model
83 external.svm.fit <-
84   svm(clase ~ .,
85      data = external.train.data,
86      cost = h.max.cost,
87      gama = h.max.gamma,
88      kernel="radial")
89
90 # predicts with test data with external svm model.
91 external.svm.probs <- predict(external.svm.fit, external.test.data)
92 external.svm.pred <- ifelse(external.svm.probs > 0.5, 1, 0)
93 external.svm.error <- mean(external.svm.pred != external.test.data$clase)
94 external.accuracy <- 1 - external.svm.error
95
96 external.final.accuracy <- external.accuracy + external.final.accuracy
97 }
98
99 cat(' accuracy after cross validation = ', external.final.accuracy/5, '\n')
100 cat(' cost and gama to use on classifier = ', h.max.cost, ' and ', h.max.gamma, '\n')
101
102 # performs the 3-folds method on all data.
103 final.folds <- createFolds(raw.data, k = 3, returnTrain = TRUE)
104 final.accuracy <- 0
105 for (e in final.folds) {
```

```
106 final.train.data <- raw.data[e, ]
107 final.test.data <- raw.data[-e, ]
108
109 # applies hyperparameters classifier on all data
110 final.svm.fit <-
111   svm(clase ~ .,
112       data = final.train.data,
113       cost = h.max.cost,
114       gama = h.max.gamma,
115       kernel="radial")
116
117 final.svm.probs <- predict(final.svm.fit, final.test.data)
118
119 final.svm.pred <- ifelse(final.svm.probs > 0.5, 1, 0)
120 final.svm.error <- mean(final.svm.pred != final.test.data$clase)
121 final.accuracy.local <- 1 - final.svm.error
122
123 final.accuracy <- final.accuracy.local + final.accuracy
124 }
125 cat('classifier accuracy in all data set = ', final.accuracy/3, '\n')
```