

MC884/MO444 - APRENDIZADO DE MÁQUINA

# *Support Vector Machines* e validação cruzada

*Erik de Godoy Perillo - RA135582*

Universidade Estadual de Campinas

4 de outubro de 2016

# 1 Introdução

O objetivo do trabalho era experimentar com a técnica de *SVM*<sup>1</sup> e sua validação por meio de *k-folds* com busca de hiperparâmetros por *grid search*.

## 1.1 Implementação

A linguagem de implementação escolhida foi o R. Todo o código utilizado no relatório encontra-se na seção 3. Ao longo do documento, linhas do código serão citadas para referência no mesmo. A função `main` (linha 130 da seção 3) executa todos os itens em ordem, mostrando os resultados.

## 2 Enunciado

*Treine um SVM com kernel RBF nos dados do arquivos. A validação externa deve ser 5-fold estratificado. Para cada conjunto de treino da validação externa faça um 3-fold para escolher os melhores hiperparametros para  $C$  (cost) e  $\gamma$  (gamma). Faça um grid search para  $C$  nos valores  $2^{-5}, 2^{-2}, 2^0, 2^2, 2^5$  e gamma nos valores  $2^{-15}, 2^{-10}, 2^{-5}, 2^0, 2^5$ .*

Os valores especificados para os *folds*,  $C$  e  $\gamma$  são declarados das linhas 9 a 15 do código na seção 3. A função que faz o *grid search* está declarada na linha 28 do código. Ela faz a procura da melhor combinação de parâmetros  $C$  e  $\gamma$ .

A função que, dados os dados e o número de folds, procura os melhores parâmetros  $C$  e  $\gamma$  entre os  $k$  possíveis *folds* (fazendo *grid search* em cada um deles) é a `get_best_svm_params`, declarada na linha 57.

Além dessa procura de melhor combinação entre 3 folds e os hiperparâmetros, pede-se que esses 3 folds venham de 5 folds externos. A função que finalmente faz todas essas partes, além de estimar a acurácia média do sistema, é declarada como `mean_accuracy_estimate` na linha 101 do código.

Na função `main` da linha 130 são feitos os dois passos principais pedidos: primeiro, a acurácia é estimada na linha 147. Depois, os parâmetros finais para o sistema são escolhidos na linha 159.

A **saída** do código todo sendo executado pela `main` encontra-se na seção 4.

## 2.1 Perguntas

1. Qual a acurácia média na validação de fora?

A acurácia média, como indica a linha 43 da saída na seção 4, é de 92.44%.

2. Quais os valores de  $C$  e  $\gamma$  a serem usados no classificador final? (fazer 3-fold no conjunto todo)

Usando-se o 3-fold, obtivemos, como a linha 56 da saída indica, os parâmetros finais:

$$C = 4, \gamma = 0.031250$$

---

<sup>1</sup>do inglês: *Support Vector Machines*

### 3 Código-fonte

```
1 #packages
2 library(caret)
3 library(e1071)
4
5 #default values:
6 #file path of data
7 data_filepath <- "data1.csv"
8 #k for external k-fold
9 ext_k <- 5
10 #k for internal k-fold
11 inn_k <- 3
12 #cost parameters
13 cost_params <- c(2^-5, 2^-2, 2^0, 2^2, 2^5)
14 #gamma parameters
15 gamma_params <- c(2^-15, 2^-10, 2^-5, 2^0, 2^5)
16
17 #wrapper for sprintf
18 printf <- function(...) cat(sprintf(...))
19
20 #calculates accuracy of prediction
21 accuracy <- function(pred_y, y, thresh=0.5)
22 {
23   pred_y <- as.numeric(pred_y >= thresh)
24   return (sum(pred_y == y)/length(y))
25 }
26
27 #performs grid search to find best C and gamma for svm
28 grid_search <- function(x_train, y_train, x_test, y_test, costs, gammas)
29 {
30   best_cost <- cost_params[1]
31   best_gamma <- gamma_params[1]
32   max_accuracy <- 0.0
33
34   for(cost in costs)
35   {
36     for(gamma in gammas)
37     {
38       model <- svm(x_train, y_train, cost=cost, gamma=gamma,
39                   kernel="radial")
40
41       y_pred <- predict(model, x_test)
42
43       acc <- accuracy(y_pred, y_test)
44       if(acc > max_accuracy)
45       {
46         max_accuracy <- acc
47         best_cost <- cost
48         best_gamma <- gamma
49       }
50     }
51   }
52
53   return (c(best_cost, best_gamma, max_accuracy))
54 }
55
56 #gets best svm parameters (C, gamma) within given folds
57 get_best_svm_params <- function(x, y, num_folds, costs, gammas)
58 {
59   folds <- createFolds(y, k=num_folds)
60
61   count <- 1
62   max_accuracy <- 0.0
63   best_cost <- costs[1]
64   best_gamma <- gammas[1]
65
66   for(fold in folds)
```

```

67 {
68   printf("fold n. %d: ", count)
69
70   x_train <- x[-fold, ]
71   x_test <- x[fold, ]
72   y_train <- y[-fold]
73   y_test <- y[fold]
74
75   params <- grid_search(x_train, y_train, x_test, y_test, costs, gammas)
76   cost <- params[1]
77   gamma <- params[2]
78   acc <- params[3]
79
80   printf("cost=%f, gamma=%f, accuracy=%.2f%% ", cost, gamma, 100*acc)
81   if(acc > max_accuracy)
82   {
83     printf("(best so far!)")
84     max_accuracy <- acc
85     best_cost <- cost
86     best_gamma <- gamma
87   }
88   printf("\n")
89
90   count <- count + 1
91 }
92
93 printf("\t-----\n")
94 printf("\tbest cost: %f, best gamma: %f, max accuracy: %.2f%%\n",
95       best_cost, best_gamma, 100*max_accuracy)
96
97 return (max_accuracy)
98 }
99
100 #k-fold inside a k-fold. used to estimate accuracy of svm
101 mean_accuracy_estimate <- function(x, y, num_external_folds, num_inner_folds)
102 {
103   #preparing data
104   ext_folds <- createFolds(y, k=num_external_folds)
105
106   ext_count <- 1
107   max_accuracies <- c()
108
109   for(ext_fold in ext_folds)
110   {
111     printf("external fold n. %d:\n", ext_count)
112
113     x_test <- x[ext_fold, ]
114     y_test <- y[ext_fold]
115     x_train <- x[-ext_fold, ]
116     y_train <- y[-ext_fold]
117
118     max_acc <- get_best_svm_params(x_train, y_train, inn_k,
119                                   cost_params, gamma_params)
120     max_accuracies <- c(max_accuracies, max_acc)
121
122     ext_count <- ext_count + 1
123     printf("\n")
124   }
125
126   printf("mean of maximum accuracies: %.2f%%\n", 100*mean(max_accuracies))
127 }
128
129 #main method for whole challenge
130 main <- function()
131 {
132   #reading data
133   data <- read.csv(data_filepath)
134   x <- data[, 1:ncol(data)-1]
135   y <- data[, ncol(data)]

```

```

136
137 printf("ESTIMATING ACCURACY:\n")
138 printf("\texternal k-folds: %d\n\tinner k-folds: %d\n\tcosts: ",
139       ext_k, inn_k)
140 print(cost_params)
141 printf("\tgammas: ")
142 print(gamma_params)
143 printf("\ttotal iterations: %d\n\n",
144       length(cost_params)*length(gamma_params)*ext_k*inn_k)
145
146 #estimating accuracy for classifier
147 mean_accuracy_estimate(x, y, ext_k, inn_k)
148
149 printf("\n-----\n")
150 printf("GETTING FINAL CLASSIFIER:\n")
151 printf("\tk-folds: %d\n\tcosts: ", inn_k)
152 print(cost_params)
153 printf("\tgammas:")
154 print(gamma_params)
155 printf("\ttotal iterations: %d\n\n",
156       length(cost_params)*length(gamma_params)*inn_k)
157
158 #final classifier
159 params <- get_best_svm_params(x, y, inn_k, cost_params, gamma_params)
160 }

```

## 4 Saída do código

```
1 TIMATING ACCURACY:
2   external k-folds: 5
3   inner k-folds: 3
4   costs: [1] 0.03125 0.25000 1.00000 4.00000 32.00000
5   gammas: [1] 3.051758e-05 9.765625e-04 3.125000e-02 1.000000e+00 3.200000e+01
6   total iterations: 375
7
8 external fold n. 1:
9 fold n. 1: cost=1.000000, gamma=0.031250, accuracy=92.13% (best so far!)
10 fold n. 2: cost=32.000000, gamma=0.000977, accuracy=86.61%
11 fold n. 3: cost=32.000000, gamma=0.000977, accuracy=91.34%
12 -----
13 best cost: 1.000000, best gamma: 0.031250, max accuracy: 92.13%
14
15 external fold n. 2:
16 fold n. 1: cost=4.000000, gamma=0.031250, accuracy=85.04% (best so far!)
17 fold n. 2: cost=4.000000, gamma=0.031250, accuracy=91.34% (best so far!)
18 fold n. 3: cost=1.000000, gamma=0.031250, accuracy=91.34%
19 -----
20 best cost: 4.000000, best gamma: 0.031250, max accuracy: 91.34%
21
22 external fold n. 3:
23 fold n. 1: cost=32.000000, gamma=0.000977, accuracy=85.83% (best so far!)
24 fold n. 2: cost=32.000000, gamma=0.000977, accuracy=93.70% (best so far!)
25 fold n. 3: cost=4.000000, gamma=0.031250, accuracy=87.40%
26 -----
27 best cost: 32.000000, best gamma: 0.000977, max accuracy: 93.70%
28
29 external fold n. 4:
30 fold n. 1: cost=4.000000, gamma=0.031250, accuracy=93.70% (best so far!)
31 fold n. 2: cost=4.000000, gamma=0.031250, accuracy=88.19%
32 fold n. 3: cost=32.000000, gamma=0.000977, accuracy=90.55%
33 -----
34 best cost: 4.000000, best gamma: 0.031250, max accuracy: 93.70%
35
36 external fold n. 5:
37 fold n. 1: cost=4.000000, gamma=0.031250, accuracy=85.71% (best so far!)
38 fold n. 2: cost=4.000000, gamma=0.031250, accuracy=91.34% (best so far!)
39 fold n. 3: cost=32.000000, gamma=0.000977, accuracy=90.55%
40 -----
41 best cost: 4.000000, best gamma: 0.031250, max accuracy: 91.34%
42
43 mean of maximum accuracies: 92.44%
44 -----
45
46 GETTING FINAL CLASSIFIER:
47 k-folds: 3
48 costs: [1] 0.03125 0.25000 1.00000 4.00000 32.00000
49 gammas: [1] 3.051758e-05 9.765625e-04 3.125000e-02 1.000000e+00 3.200000e+01
50 total iterations: 75
51
52 fold n. 1: cost=4.000000, gamma=0.031250, accuracy=93.04% (best so far!)
53 fold n. 2: cost=4.000000, gamma=0.031250, accuracy=88.68%
54 fold n. 3: cost=1.000000, gamma=0.031250, accuracy=89.94%
55 -----
56 best cost: 4.000000, best gamma: 0.031250, max accuracy: 93.04%
```