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[1] e sua validação por meio de *k-folds* com busta 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 γ (gamma). Faça um grid search de para o 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 γ 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 γ.

A função que, dados os dados e o número de folds, procura os melhores parâmetros C e γ 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 γ 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:

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

[1] do inglês: *Support Vector Machines*

3 Código-fonte

```
#packages
library(caret)
library(e1071)

#default values:
#file path of data
data_filepath <- "data1.csv"
#k for external k-fold
ext_k <- 5
#k for internal k-fold
inn_k <- 3
#cost parameters
cost_params <- c(2^-5, 2^-2, 2^0, 2^2, 2^5)
#gamma parameters
gamma_params <- c(2^-15, 2^-10, 2^-5, 2^0, 2^5)

#wrapper for sprintf
printf <- function(...) cat(sprintf(...))

#calculates accuracy of prediction
accuracy <- function(pred_y, y, thresh=0.5)
{
  pred_y <- as.numeric(pred_y >= thresh)
  return (sum(pred_y == y)/length(y))
}

#performs grid search to find best C and gamma for svm
grid_search <- function(x_train, y_train, x_test, y_test, costs, gammas)
{
  best_cost <- cost_params[1]
  best_gamma <- gamma_params[1]
  max_accuracy <- 0.0

  for(cost in costs)
  {
    for(gamma in gammas)
    {
      model <- svm(x_train, y_train, cost=cost, gamma=gamma,
        kernel="radial")

      y_pred <- predict(model, x_test)

      acc <- accuracy(y_pred, y_test)
      if(acc > max_accuracy)
      {
        max_accuracy <- acc
        best_cost <- cost
        best_gamma <- gamma
      }
    }
  }

  return (c(best_cost, best_gamma, max_accuracy))
}

#gets best svm parameters (C, gamma) within given folds
get_best_svm_params <- function(x, y, num_folds, costs, gammas)
{
  folds <- createFolds(y, k=num_folds)

  count <- 1
  max_accuracy <- 0.0
  best_cost <- costs[1]
  best_gamma <- gammas[1]

  for(fold in folds)
```

```
  {
    printf("fold n. %d: ", count)

    x_train <- x[-fold, ]
    x_test <- x[fold, ]
    y_train <- y[-fold]
    y_test <- y[fold]

    params <- grid_search(x_train, y_train, x_test, y_test, costs, gammas)
    cost <- params[1]
    gamma <- params[2]
    acc <- params[3]

    printf("cost=%f, gamma=%f, accuracy=%.2f%% ", cost, gamma, 100*acc)
    if(acc > max_accuracy)
    {
      printf("(best so far!)")
      max_accuracy <- acc
      best_cost <- cost
      best_gamma <- gamma
    }
    printf("\n")

    count <- count + 1
  }

  printf("\t--------\n")
  printf("\tbest cost: %f, best gamma: %f, max accuracy: %.2f%%\n",
    best_cost, best_gamma, 100*max_accuracy)

  return (max_accuracy)
}

#k-fold inside a k-fold. used to estimate accuracy of svm
mean_accuracy_estimate <- function(x, y, num_external_folds, num_inner_folds)
{
  #preparing data
  ext_folds <- createFolds(y, k=num_external_folds)

  ext_count <- 1
  max_accuracies <- c()

  for(ext_fold in ext_folds)
  {
    printf("external fold n. %d:\n", ext_count)

    x_test <- x[ext_fold, ]
    y_test <- y[ext_fold]
    x_train <- x[-ext_fold, ]
    y_train <- y[-ext_fold]

    max_acc <- get_best_svm_params(x_train, y_train, inn_k,
      cost_params, gamma_params)
    max_accuracies <- c(max_accuracies, max_acc)

    ext_count <- ext_count + 1
    printf("\n")
  }

  printf("mean of maximum accuracies: %.2f%%\n", 100*mean(max_accuracies))
}

#main method for whole challenge
main <- function()
{
  #reading data
  data <- read.csv(data_filepath)
  x <- data[, 1:ncol(data)-1]
  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

```
TIMATING ACCURACY:
        external k-folds: 5
        inner k-folds: 3
        costs: [1]  0.03125  0.25000  1.00000  4.00000 32.00000
        gammas: [1] 3.051758e-05 9.765625e-04 3.125000e-02 1.000000e+00 3.200000e+01
        total iterations: 375

external fold n. 1:
fold n. 1: cost=1.000000, gamma=0.031250, accuracy=92.13% (best so far!)
fold n. 2: cost=32.000000, gamma=0.000977, accuracy=86.61%
fold n. 3: cost=32.000000, gamma=0.000977, accuracy=91.34%
        --------
        best cost: 1.000000, best gamma: 0.031250, max accuracy: 92.13%

external fold n. 2:
fold n. 1: cost=4.000000, gamma=0.031250, accuracy=85.04% (best so far!)
fold n. 2: cost=4.000000, gamma=0.031250, accuracy=91.34% (best so far!)
fold n. 3: cost=1.000000, gamma=0.031250, accuracy=91.34%
        --------
        best cost: 4.000000, best gamma: 0.031250, max accuracy: 91.34%

external fold n. 3:
fold n. 1: cost=32.000000, gamma=0.000977, accuracy=85.83% (best so far!)
fold n. 2: cost=32.000000, gamma=0.000977, accuracy=93.70% (best so far!)
fold n. 3: cost=4.000000, gamma=0.031250, accuracy=87.40%
        --------
        best cost: 32.000000, best gamma: 0.000977, max accuracy: 93.70%

external fold n. 4:
fold n. 1: cost=4.000000, gamma=0.031250, accuracy=93.70% (best so far!)
fold n. 2: cost=4.000000, gamma=0.031250, accuracy=88.19%
fold n. 3: cost=32.000000, gamma=0.000977, accuracy=90.55%
        --------
        best cost: 4.000000, best gamma: 0.031250, max accuracy: 93.70%

external fold n. 5:
fold n. 1: cost=4.000000, gamma=0.031250, accuracy=85.71% (best so far!)
fold n. 2: cost=4.000000, gamma=0.031250, accuracy=91.34% (best so far!)
fold n. 3: cost=32.000000, gamma=0.000977, accuracy=90.55%
        --------
        best cost: 4.000000, best gamma: 0.031250, max accuracy: 91.34%

mean of maximum accuracies: 92.44%

------------------------------------------------------------
GETTING FINAL CLASSIFIER:
        k-folds: 3
        costs: [1]  0.03125  0.25000  1.00000  4.00000 32.00000
        gammas:[1] 3.051758e-05 9.765625e-04 3.125000e-02 1.000000e+00 3.200000e+01
        total iterations: 75

fold n. 1: cost=4.000000, gamma=0.031250, accuracy=93.04% (best so far!)
fold n. 2: cost=4.000000, gamma=0.031250, accuracy=88.68%
fold n. 3: cost=1.000000, gamma=0.031250, accuracy=89.94%
        --------
        best cost: 4.000000, best gamma: 0.031250, max accuracy: 93.04%
```