MC884/MO444 - Aprendizado de Máquina

Regressão Logística e LDA

Erik de Godoy Perillo - RA135582

Universidade Estadual de Campinas

23 de setembro de 2016

1 Introdução

O objetivo do trabalho era experimentar com as técnicas de PCA[1], *Logistic Regression* e *LDA*[2].

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 59 da seção 3) executa todos os itens em ordem, mostrando os resultados.

2 Itens

1. *faça o PCA dos dados (sem a última coluna). Se você quiser que os dados transformados tenham 80% da variância original, quantas dimensões do PCA vc precisa manter? Gere os dados transformados mantendo 80% da variância.*

   O PCA é calculado na linha 69 do código da seção 3. Os dados são normalizados antes da função `prcomp` ser chamada, pois já precisaremos deles normalizados no momento em que gerarmos os dados nas novas dimensões. A função `pca_min_pcs` da linha 19 da seção 3 contém o código necessário para determinar o número mínimo de componentes dada uma variância. Para o valor de 80%, é necessário manter 13 das bases com maior variância dos componentes principais. A matriz de transformação dos dados antigos para os nas novas 13 dimensões é obtida na linha 78 da seção 3. Os dados novos são gerados na linha 80.

2. *Treine uma regressão logística no conjunto de treino dos dados originais e nos dados transformados. Qual a taxa de acerto no conjunto de teste nas 2 condições (sem e com PCA)?*

   A função que calcula a regressão logística e sua acurácia é dada na função `logit_reg` da linha 32 da seção 3. Os dados de treino e teste são selecionados entre as linhas 84 e 91. Como resultado, obteve-se que a acurácia do modelo obtido pela regressão com todos os dados foi menor que a obtida com as dimensões que mantinham 80% da variância. Os valores foram, respectivamente, 65.58% e 75.72%.

3. *Treine o LDA nos conjuntos de treino com e sem PCA e teste nos respectivos conjuntos de testes. Qual a acurácia nas 2 condições?*

   A função que calcula a LDA e sua acurácia é dada na função `lda_reg` da linha 45 da seção 3. Como resultado, obteve-se que a acurácia do modelo obtido pela regressão com todos os dados foi menor que a obtida com as dimensões que mantinham 80% da variância, assim como no item 2. Os valores foram, respectivamente, 67.75% e 78.62%.

4. *Qual a melhor combinação de classificador e PCA ou não?*

   A combinação que mostrou o melhor resultado foi o uso de PCA com LDA. O resultado é interessante, ainda mais que na regressão logística, sob o uso de todas as dimensões, o algoritmo demonstrou mensagens de aviso do tipo `algorithm did not converge`. Uma pesquisa para os motivos de tal resultado sugere [1] que isso pode ser pelo fato de o modelo estar

[1] do inglês: *Principal Component Analisys*
[2] do inglês: *Linear Discriminant Analisys*

''perfeito demais" que, devido a algum detalhe de implementação da função em R, faz com que alguns parâmetros fiquem com valores muito pequenos/grandes, sendo assim difíceis de serem representados com precisão pelo computador e então gerando resultados piores que os com PCA.

3 Apêndice: código-fonte

```
#package for lda
library(MASS)

#default values:
#file path of data
data_filepath <- "data1.csv"
#minimum variance required
min_var <- 0.80
#number of lines to use in training
train_n_lines <- 200
#number of lines to use in test
test_n_lines <- 276

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

#gets minimum number of principal components to keep in order to conserve
#min_var of variance
pca_min_pcs <- function(pcs, min_var)
{
  #getting k minimum number of components required for minimum variance
  pcs_var_cumsum <- cumsum(pcs$sdev^2/sum(pcs$sdev^2))
  min_pcs <- which(pcs_var_cumsum >= min_var)[1]
  #printing result
  printf("\t-Number of components to keep %.2f%% variance: %d\n",
    min_var*100, min_pcs)

  return(min_pcs)
}

#calculates logistic regression and displays accuracy
logit_reg <- function(x_train, y_train, x_test, y_test)
{
  #computing logistic regression
  lr <- glm(y_train ~ ., data=x_train, family=binomial(link="logit"))
  #getting predictions
  pred <- as.matrix(predict(lr, x_test)) >= 1
  #getting score
  score = sum(pred == y_test)/length(y_test)
  #printing accuracy
  printf("accuracy: %.2f%%\n", score*100)
}

#calculates LDA and displays accuracy
lda_reg <- function(x_train, y_train, x_test, y_test)
{
  #computing lda
  ldar <- lda(y_train ~ ., data=x_train)
  #getting predictions
  pred <- predict(ldar, x_test, prior=ldar$prior)
  pred <- pred$posterior[, 2] > pred$posterior[, 1]
  #getting score
  score = sum(pred == y_test)/length(y_test)
  #printing accuracy
  printf("accuracy: %.2f%%\n", score*100)
}

```

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

  #scaling data prior to pca. we would have to do it anyway...
  x <- scale(x)

  #getting principal components
  pcs <- prcomp(x, scale=FALSE)

  #1. Faca o PCA dos dados (sem a ultima coluna).
  #Se voce quiser que os dados transformados tenham 80% da variancia original,
  #quantas dimensoes do PCA vc precisa manter?
  #Gere os dados transformados mantendo 80\% da variancia.
  printf("Item 1:\n")
  min_pcs <- pca_min_pcs(pcs, min_var)
  #getting transformation matrix
  transf_mat <- t(pcs$rotation[, 1:min_pcs])
  #transforming data into k dimensions while keeping percentage of variance
  transf_x <- as.matrix(x) %*% t(transf_mat)

  #preparing data for regression
  #train data
  y <- as.matrix(data[, ncol(data)])
  y_train <- y[1:train_n_lines, ]
  x_full_var_train <- x[1:train_n_lines, ]
  x_part_var_train <- transf_x[1:train_n_lines, ]
  #test data
  x_full_var_test <- x[(train_n_lines+1):nrow(x), ]
  x_part_var_test <- as.matrix(x_full_var_test) %*% t(transf_mat)
  y_test <- y[(train_n_lines+1):nrow(y), ]

  #2. Treine uma regressao logistica no conjunto de treino dos dados originais
  #e nos dados transformados.
  #Qual a taxa de acerto no conjunto de teste nas 2 condicoes (sem e com PCA)?
  printf("\nItem 2:\n")
  #logistic regression on all dimensions
  printf("\t-All dimensions: ")
  logit_reg(as.data.frame(x_full_var_train), y_train,
    as.data.frame(x_full_var_test), y_test)
  #logistic regression on k principal components
  printf("\t-First %d dimensions from PCA: ", min_pcs)
  logit_reg(as.data.frame(x_part_var_train), y_train,
    as.data.frame(x_part_var_test), y_test)

  #3. Treine o LDA nos conjuntos de treino com e sem PCA e teste nos
  #respectivos conjuntos de testes. Qual a acuracia nas 2 condicoes?
  printf("\nItem 3:\n")
  #lda on all dimensions
  printf("\t-All dimensions: ")
  lda_reg(as.data.frame(x_full_var_train), y_train,
    as.data.frame(x_full_var_test), y_test)
  #lda on k principal components
  printf("\t-First %d dimensions from PCA: ", min_pcs)
  lda_reg(as.data.frame(x_part_var_train), y_train,
    as.data.frame(x_part_var_test), y_test)

  printf("\n")
}
```

Referências

[1] *Logistic regression model does not converge*. 2010. URL: http://stats.stackexchange.com/questions/5354/logistic-regression-model-does-not-converge (acesso em 19/04/2016).