**UNIVERSIDADE FEDERAL DE MINAS GERAIS**
Graduação em Engenharia de Sistemas
**Introdução ao Reconhecimento de Padrões - Aula 6**
Nikolas Dias Magalhães Fantoni - 2018019400

# Classificador de Bayes aplicado a um problema multivariado (Base Spam)

No exercício proposto, a base de dados *spambase.data*, fornecida pelo professor, foi utilizada para demonstrar um classificador bayesiano. Primeiro, as amostras foram separadas em classes do tipo 1 (consideradas spam) ou do tipo 2 (não spams), de acordo com a última coluna de cada amostra observada. Depois, os dados foram separados em 10 *folds* e foi utilizada a técnica de validação cruzada, em que a cada iteração, um dos folds (correspondente a aproximadamente 10% dos dados) é utilizado como teste e o restante serve de base para o treinamento do classificador. Para uma implementação correta da separação dos *folds* de maneira aleatória, foi utilizada a biblioteca do *R caret*.

Após realizados os 10 testes, a acurácia média obtida foi de 96.08%, com um desvio padrão de 2.17%. O código do exercício encontra-se abaixo.

```
#Universidade Federal de Minas Gerais
#Introducao ao Reconhecimento de Padroes
#Nikolas Dias Magalhaes Fantoni
#AULA 6 - Base Spam
#2019/2

#Limpando o ambiente
rm(list=ls())

#Adicionando biblioteca
library(caret)

#Funcao estimativa densidade para n variaveis
pdfnvar <- function(x, m, K, n) {
  if (det(K) == 0) 999999999 else (1/(sqrt((2 * pi)^(n) * (det(K))))) *
  exp(-0.5*(t(x-m) %*% (solve(K)) %*% (x-m)))
}

#Lendo os dados
spam <- read.csv( "spambase.data", sep=",",header = FALSE)
ic1 <- which(spam$V58 == 1)
ic2 <- which(spam$V58 == 0)
spamc1 <- spam[ic1,]
spamc2 <- spam[ic2,]

#Criando os folds que irao separar os dados
fl1 <- createFolds(spamc1[,1], k = 10, list = TRUE, returnTrain = FALSE)
fl2 <- createFolds(spamc1[,1], k = 10, list = TRUE, returnTrain = FALSE)

#Looping para os 10 testes
acuraciaacumulada <- NULL
for (j in 1:10){
  test <- rbind(spamc1[fl1[[j]],],spamc1[fl1[[j]],])
  trainc1 <- spamc1[-fl1[[j]],]
  trainc2 <- spamc2[-fl1[[j]],]

  #Calculando as medias
```

```
  u1<- NULL
  u2<- NULL
  for (m in 1:57){
  u1 <- rbind(u1, mean(trainc1[,m]))
  u2 <- rbind(u2, mean(trainc2[,m]))
  }

  #Covariancias dos dados
  cov1 <- cov(trainc1[,1:57])
  cov2 <- cov(trainc2[,1:57])

  #Testando o algoritmo
  erro <- 0
  for (i in 1:length(test[,1])){
    l <- t(test[i,1:57])
    f1 <- pdfnvar(l, u1 , cov1, 57)
    f2 <- pdfnvar(l, u2 , cov2, 57)
    if (f2 == 0){
      c <- 1
    } else {
    c <- if ((f1/f2 >= 1) == TRUE) 1 else 0
    if ((c-test[i,58]) != 0) erro <- erro +1
    }
  }

  #Obtendo a acuracia
  acerto <- 1 - erro/length(test[,1])
  acuraciaacumulada <- c(acuraciaacumulada, acerto)
}

#Imprimindo a saida
cat("\n acuracia: [", acuraciaacumulada, "]\n", "media: ",
    mean(acuraciaacumulada)*100,"%", "\n", "desvio: ",
    sd(acuraciaacumulada)*100,"%","\n\n")
```