

Saulo Henrique Pires de Oliveira

Mestrando em Bioinformática pelo Instituto de Bioinformática da USP. Atualmente, está sob a orientação do Dr. Paulo Sergio Lopes de Oliveira, dentro do Laboratório de Genética e Cardiologia Molecular - INCOR/HCFM-USP. Atua na área de Biologia Computacional, com ênfase em proteômica. O título do projeto com o qual trabalha é: “Desenvolvimento de um algoritmo para identificação e caracterização de cavidades em regiões específicas de estruturas tridimensionais de proteínas”.

Meus Exercícios

Exercício 1: [saulo_1_f.r](#)

Exercício 2: [saulo_2_f.r](#)

Exercício 3: [saulo_3_f.r](#)

Exercício 4: [saulo_4_f.r](#)

Exercício 5: [saulo_5_f.r](#)

Exercício 6: [saulo_6_f.r](#)

Exercício 7: [saulo_7_f.r](#)

Exercício 8: [saulo_8_f.r](#)

Exercício 9: [saulo_9_f.r](#)

Proposta de Trabalho Final

Plano A: Um Classificador Bayesiano para Dados em duas Dimensões

A idéia é implementar uma função que recebe:

-Duas matrizes de dimensões $n_1 \times 2$ e $n_2 \times 2$ com tuplas de dados que seguem uma distribuição aproximadamente normal.

-A proporção de cada uma das classes de interesse na população.

ou

-Um parâmetro indicando que um estimador das proporções na população deve ser calculado pela função.

A função gera um gráfico com os dados plottados em duas dimensões, assim como a superfície de decisão encontrada para o classificador bayesiano nesse conjunto de dados. Ela retornará a equação da superfície de decisão, que poderá então ser usada para a classificação de novos dados entre

classe 1 ou classe 2.

Plano B: Perceptron

A idéia é implementar uma função que recebe:

- Uma matriz D de n por m dimensões com tuplas de dados que seguem uma distribuição qualquer.
- Um vetor B com a classe de cada uma das amostras.

Obs: Os dados precisam ser linearmente separáveis.

A função encontra a superfície para um classificador linear utilizando o algoritmo Perceptron. Além disso, ela gera uma estimativa para o erro do classificador, utilizando o erro de Bolstering. Caso os dados possuam dimensão $m \leq 3$, a função permite que o usuário especifique como parâmetro que a função gere um gráfico com os dados e a superfície de decisão encontrada.

Página de Ajuda

class.bayes

package:nenhum

R Documentation

Cálculo de um Classificador de Bayes para duas classes de amostras com duas características

cada, que seguem uma distribuição aproximadamente normal.

Description:

Calcula a superfície de decisão de um classificador de Bayes para os dado de entrada e

imprime a equação dessa superfície. Produz um gráfico com os dados das duas classes, e

adiciona uma curva representando a superfície de decisão do classificador.

Usage:

```
class.bayes(x, y,prop=T, prop1=0.5, prop2=0.5, precision=10000)
```

Arguments:

x: Matriz numérica com 2 colunas representando os valores das duas características de uma

amostra da classe 1.

y: Matriz numérica com 2 colunas representando os valores das duas características de uma amostra da classe 2.

prop: Lógico. Indica se as proporções serão passadas como parâmetros ou se devem ser estimadas pela função.

prop1: A proporção da amostra de classe 1 na população.

prop2: A proporção da amostra de classe 2 na população.

precision: um inteiro indicando a precisão (ver detalhes)

Details:

A função inicialmente estima as proporções para cada uma das classes das duas amostras passadas como parâmetro. Em seguida, o vetor médio e a matriz de covariância de cada uma das amostras são calculados. Estes dados são utilizados para obter os parâmetros da equação da superfície de decisão do classificador.

Uma vez obtida a equação da superfície de decisão, a função gera um gráfico

com os dados da entrada. Para adicionar a curva ao gráfico, um número "n" de pontos da curva ("n" é fornecido com o parâmetro "precision") são computados, e então a curva é aproximada através desses pontos.

Value:

Um gráfico é gerado, contendo os dados e a curva de decisão do classificador. Uma string contendo a equação da superfície de decisão do classificador é impressa e retornada pela função.

Warning:

Cada uma das características de ambas as amostras necessariamente precisa seguir uma distribuição normal, sendo que as matrizes de covariância devem apresentar determinante não nulo para que o classificador de Bayes possa ser encontrado.

A função assume que a soma das proporções fornecidas como parâmetro somem 1.

Author(s):

Saulo Henrique Pires de Oliveira

sauloho@gmail.com

References:

Richard O. Duda, Peter E. Hart, David G. (2000). Pattern Classification (2nd

Edition) Stork. ISBN: 978-0-471-05669-0.

See Also:

0 pacote ROCR para manipulação e avaliação de desempenho de classificadores.

Examples:

```
# Primeiro Exemplo (Classificador linear):
```

```
d1=rnorm(10000,10,2)
d2=rnorm(10000,5,2)
d3=rnorm(10000,30,2)
d4=rnorm(10000,15,2)

D1=matrix(nrow=10000,ncol=2)
D2=matrix(nrow=10000,ncol=2)
D1[,1]=d1
D1[,2]=d2
D2[,1]=d3
D2[,2]=d4
class.bayes(D1,D2)
```

```
# Segundo Exemplo (classificador não linear):
```

```
d1=rnorm(10000,10,2)
d2=rnorm(10000,5,2)
d3=rnorm(10000,30,4)
d4=rnorm(10000,15,2)

D1=matrix(nrow=10000,ncol=2)
D2=matrix(nrow=10000,ncol=2)
D1[,1]=d1
D1[,2]=d2
D2[,1]=d3
D2[,2]=d4
class.bayes(D1,D2)
```

Código da Função

```
class.bayes <- function (class1,class2,prop=F,prop1,prop2,precision=10000)
{
  if(!prop)
  {
    prop1 <- length(class1)/(length(class1)+length(class2))
    prop2 <- length(class2)/(length(class1)+length(class2))
  }
  mu1 <- apply(class1,2,mean)
  mu2 <- apply(class2,2,mean)

  cov1 <- cov(class1)
  cov2 <- cov(class2)

  W1=-0.5*solve(cov1,ncol=2)
  w1=solve(cov1)%*%mu1
  w10= -0.5*(mu1%*%solve(cov1))%*%mu1 -0.5*log(det(cov1)) + log(prop1)
  W2=-0.5*solve(cov2,ncol=2)
  w2=solve(cov2)%*%mu2
  w20= -0.5*(mu2%*%solve(cov2))%*%mu2 -0.5*log(det(cov2)) + log(prop2)
  coefx2 <- W1[1,1]-W2[1,1]
  coefx <- w1[1,1]-w2[1,1]
  coefy2 <-W1[2,2]-W2[2,2]
  coefy <- w1[2,1]-w2[2,1]
  coefxy <- W1[1,2]*W1[2,1]- W2[1,2]*W2[2,1]
  coef0 <- w10 - w20

  xmin=min(c(min(class1[,1],class2[,1]))) -1
  xmax=max(c(max(class1[,1],class2[,1]))) +1
  ymin=min(c(min(class1[,2],class2[,2]))) -1
  ymax=max(c(max(class1[,2],class2[,2]))) +1

  plot(class1,col="green",xlim=c(xmin,xmax),ylim=c(ymin,ymax),pch=3,main="Dados das Classes 1 e 2 e Superfície de Decisão",
        xlab="Característica 1",ylab="Característica 2",cex.lab=1.4)
  points(class2,col="red",pch=4)

  if(coefy2>0.001)
  {
    surfacel<-matrix(ncol=2,nrow=2*precision)
    for(i in 1:precision)
    {
      surfacel[i,1]<-xmin+1 + i*((xmax-xmin)/precision )
      surfacel[i+precision,1]<-xmin+1 + i*((xmax-xmin)/precision )
      a<-coefy2
      b<-coefy + coefxy*surfacel[i,1]
    }
  }
}
```

```
c<-coefx2*surfacel[i,1]^2 + coefx*surfacel[i,1] +coef0
delt = b*b - 4*(a*c)
if( delt >=0 )
{
  y1 = ( (-b + sqrt(delt)) / (2*a) )
  y2 = ( (-b - sqrt(delt)) / (2*a) )
  surfacel[i,2] <- y1
  surfacel[i+precision,2] <- y2
}
}
else
{
  surfacel<-matrix(ncol=2,nrow=precision)
  for(i in 1:precision)
  {
    surfacel[i,1]<-xmin+1 + i*((xmax-xmin)/precision )
    b<-coefy + coefxy*surfacel[i,1]
    c<-coefx2*surfacel[i,1]^2 + coefx*surfacel[i,1] +coef0
    surfacel[i,2] <- (-c/b)
  }
}

lines(surfacel)

s=paste(coefx2,"*x^2 +(",coefx,")*x +(",coefy2,")*y^2 +(",coefy,")*y
+(",coefxy,")*x*y +(",coef0,")\n")
cat(s)
return(s)
}
```

Arquivo da Função

[bayes.r](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2010:alunos:trabalho_final:saulo.oliveira:start 

Last update: **2020/08/12 06:04**