

Exercício 4: Mistura de Gaussianas

Rúbia Reis Guerra
2013031143

3 de maio de 2017

1 Classificador Bayesiano: Mistura de Gaussianas

Nesta atividade, foi proposta a amostragem de dados do dataset BreastCancer, seguida da divisão em conjuntos de teste e treino e classificação bayesiana. Para obter as densidades de cada classe, foi utilizado o pacote *mclust*.

1.1 Pacotes utilizados

```
> rm(list=ls())
> library('MASS')
> library('mlbench')
> library('mclust')
```

1.2 Implementação

```
> #####
> # Dataset BreastCancer #
> data(BreastCancer)
> X<- data.matrix(BreastCancer[,2:10])
> X[is.na(X)] <- 0
> Y <- as.numeric(BreastCancer$Class)
> #####
> # Auxiliares #
> tp <- c()
> fp <- c()
> fn <- c()
> prec <- c()
> rec <- c()
> f1 <- c()
> error <- c()
> mse <- c()
> sde <- c()
```

```

> for(j in 1:10){
+ #####
+ # Amostrar dados #
+ index <- sample(2, nrow(BreastCancer), replace=TRUE, prob=c(0.70,0.30))
+
+ #####
+ # Conjunto de treinamento #
+ training <- X[which(index==1),]
+ trainingLabels <- as.matrix(Y[which(index==1)])
+
+ #####
+ # Conjunto de teste #
+ test <- X[which(index==2),]
+ testLabels <- as.matrix(Y[which(index==2)])
+
+ #####
+ # Probabilidades a priori #
+ pc1 <- length(Y[which(Y==1)])/(length(Y))
+ pc2 <- length(Y[which(Y==2)])/(length(Y))
+
+ #####
+ # Treinamento #
+ mod1 = densityMclust(training[which(trainingLabels==1),])
+ mod2 = densityMclust(training[which(trainingLabels==2),])
+
+ #####
+ # Teste #
+ pxc1 <- dens(modelName=mod1$modelName, data = test, parameters = mod1$parameters)
+ pxc2 <- dens(modelName=mod2$modelName, data = test, parameters = mod2$parameters)
+
+ #####
+ # Classificação #
+ Ntest <- dim(test)[1]
+ testY <- c()
+ for(i in 1:Ntest)
+ {
+   testY[i] <- ifelse(pxc1[i]/pxc2[i] >= pc2/pc1, 1, 2)
+   error[i] <- (testY[i]-testLabels[i])^2
+ }
+
+ # MSE e SD #
+ mse[j] <- mean(error)
+ sde[j] <- sd(error)
+
+ # Matriz de confusao #
+ testCM <- table(testY,testLabels)

```

```

+
+   # Precision, recall, F1 #
+   tp[j] <- sum((testY==1) & (testLabels==1)) # True positives
+   fp[j] <- sum((testY==1) & (testLabels==2)) # False positives
+   fn[j] <- sum((testY==2) & (testLabels==1)) # False negatives
+   prec[j] <- tp[j]/(tp[j] + fp[j]) # Precision
+   rec[j] <- tp[j]/(tp[j] + fn[j]) # Recall
+   f1[j] <- 2*prec[j]*rec[j]/(prec[j]+rec[j]) # F1 Score
+ }
> mean(mse) # MSE

[1] NA

> mean(sde) # SD

[1] NA

```

Por alguma razão ainda não identificada, a utilização de *densityMclust* para os dados da classe 2 resulta sempre em densidades nulas, que, por sua vez, força o classificador ter saída 1 para todos os dados amostrados.