Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

18 junho 2024

Entrega 6

Prof. Dr. George von Borries

Aluno: Bruno Gondim Toledo

Matrícula: 15/0167636

Aluno: Stefan Zurman Gonçalves

Matrícula: 19/0116994

Tópicos 2

1º/2024

Observação: Nas questões seguintes, explicar cada passo dos exemplos apresentados.

14. Estudar o pacote rsample em https://rsample.tidymodels.org/ e apresentar um exemplo utilizando validação cruzada e Bootstrap.

15. Selecionar ou gerar um conjunto de dados e comparar a classificação após estimação de densidades utilizando os seguintes métodos:

1. **Método do Histograma**
2. **Estimação baseada em Núcleos**
3. **k-Vizinhos mais Próximos**

16. Estudar o pacote ks do R e apresentar um exemplo.

Instalando e carregando o pacote:

```
if (!require("ks")) install.packages("ks")
```

```
## Carregando pacotes exigidos: ks
```

O pacote *ks* (*kernel smoothing*) implementa a possibilidade de estudar distribuições provenientes de misturas, de até 6 dimensões.

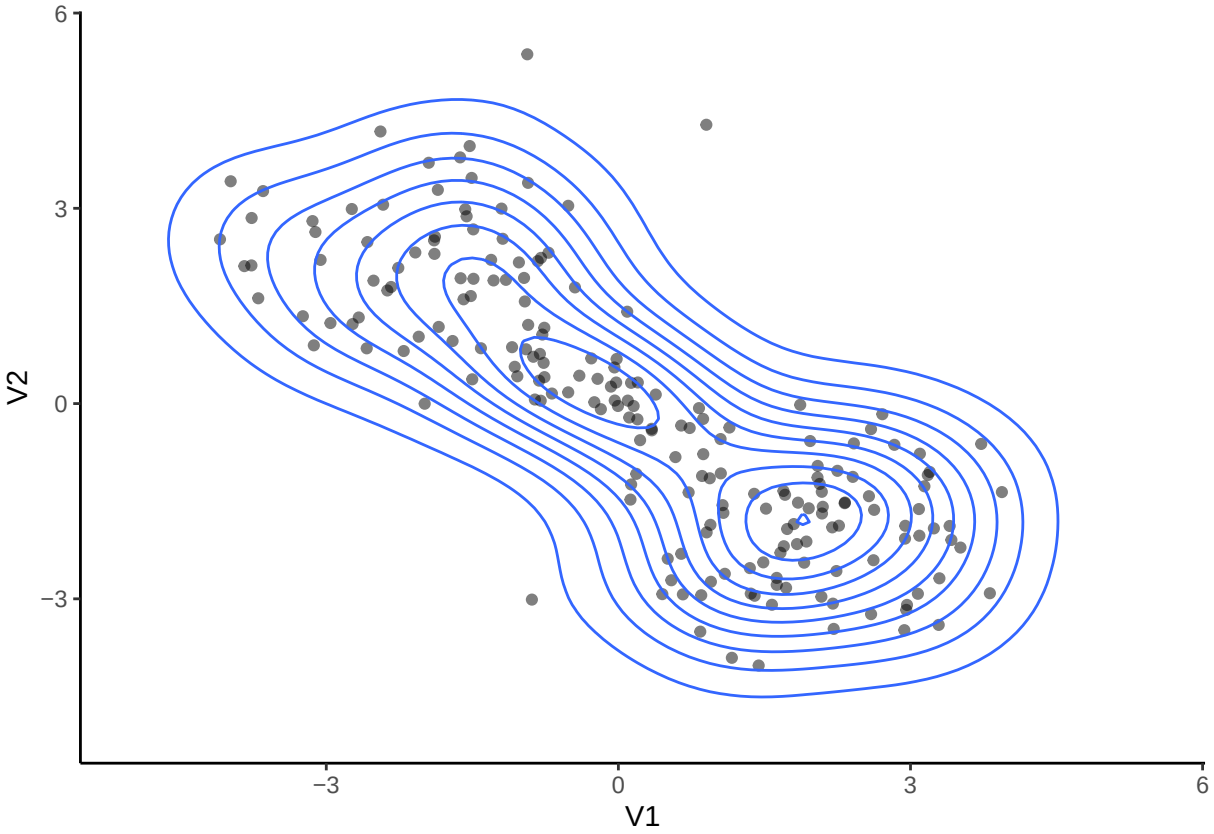
Em geral, o interesse será trabalhar com misturas de normais (2 ou 3). O pacote traz também um gerador de misturas, em que podemos passar os parâmetros de cada uma das normais, bem como a proporção de cada distribuição a amostra terá.

Este tipo de procedimento é interessante para dados multimodais, onde podemos tentar isolar grupos de acordo com misturas de normais. Em alguns casos, funciona bem também quando temos mistura de normal com distribuição de cauda pesada — neste caso, a distribuição de cauda pesada possivelmente será identificada pelo algoritmo como uma série de normais, algumas com variância bem alta — se o objetivo de interesse for modelar bem a normal.

```
# Definir uma seed para as amostras
set.seed(150167636)

# Definir tamanho da amostra
samp <- 200

# Definindo a matriz dos vetores de médias das normais
mus <- rbind(c(-2,2), c(0,0), c(2,-2))

# Definindo as matrizes de variância-covariância
Sigmas <- rbind(diag(2), matrix(c(0.8, -0.72, -0.72, 0.8), nrow=2), diag(2))

# Definindo as proporções das misturas
cwt <- 3/11
props <- c((1-cwt)/2, cwt, (1-cwt)/2)

# Gerando a amostra de uma mistura de normais com parâmetros definidos anteriormente
x <- rmvnorm.mixt(n=samp, mus=mus, Sigmas=Sigmas, props=props)
```

Gerada a amostra, podemos fazer o gráfico de contorno da amostra.

```
ggplot(as.data.frame(x), aes(x=V1, y=V2))+
  geom_point(alpha = .5) +
  geom_density_2d()+
  theme_classic()+
  ylim(-5,5.5)+
  xlim(-5,5.5)
```



Agora, vamos tentar entender esta amostra, como se não soubéssemos os parâmetros geradores. Para esta *seed*, é até difícil visualizar que são 3 normais, aparentando mais ser duas normais de alta variância.

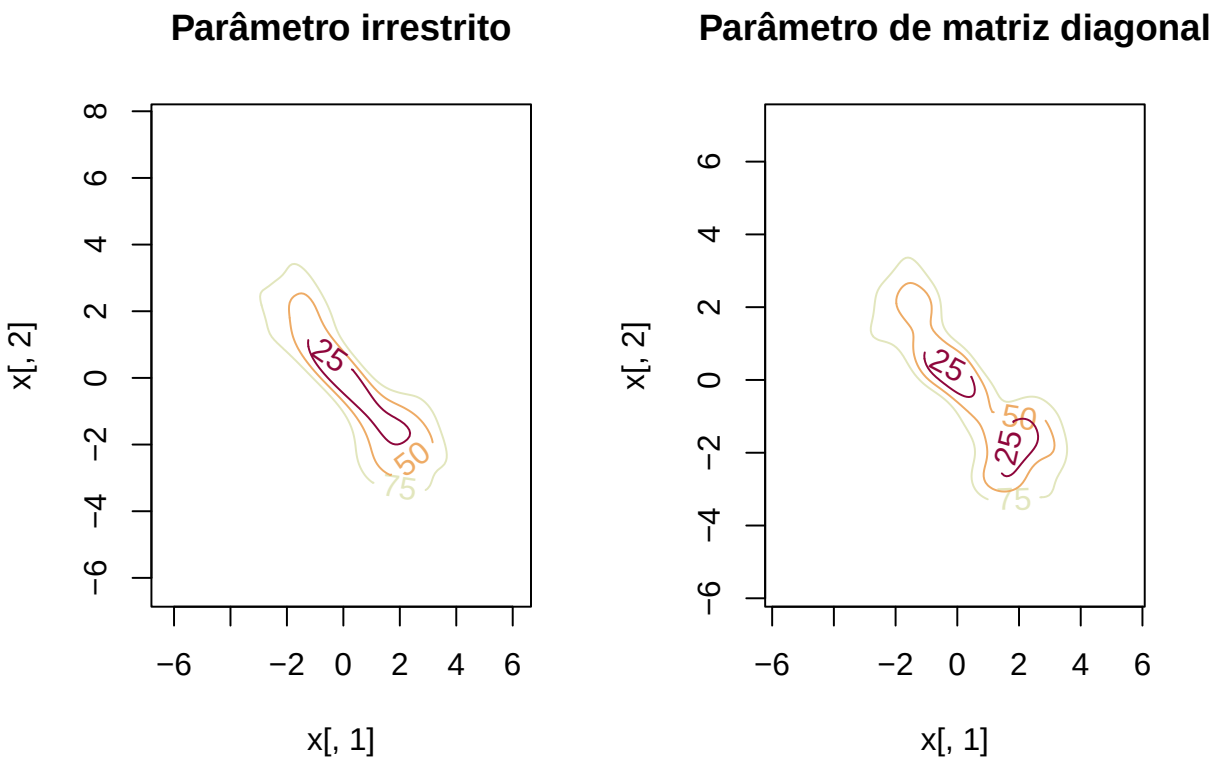
Para isto, o parâmetro **H**, que é a matriz (simétrica e positiva definida) de largura de banda. Este é o argumento mais crucial para estimação de densidade Kernel, segundo Chacón e Duong (2009)[1]. A função que encontra o melhor parâmetro **H** já está implementada no pacote *ks*, sendo a *Hpi* (ou *Hpi.diag*).

```
Hpi1 <- Hpi(x=x)
Hpi2 <- Hpi.diag(x=x)

fhat.pi1 <- kde(x=x, H=Hpi1) # Estima a densidade Kernel
fhat.pi2 <- kde(x=x, H=Hpi2)
```

Aqui a diferença entre as funções, é que a *Hpi* irá buscar um parâmetro irrestrito; enquanto a *Hpi.diag* irá buscar uma matriz diagonal de parâmetros (0 fora da diagonal principal).

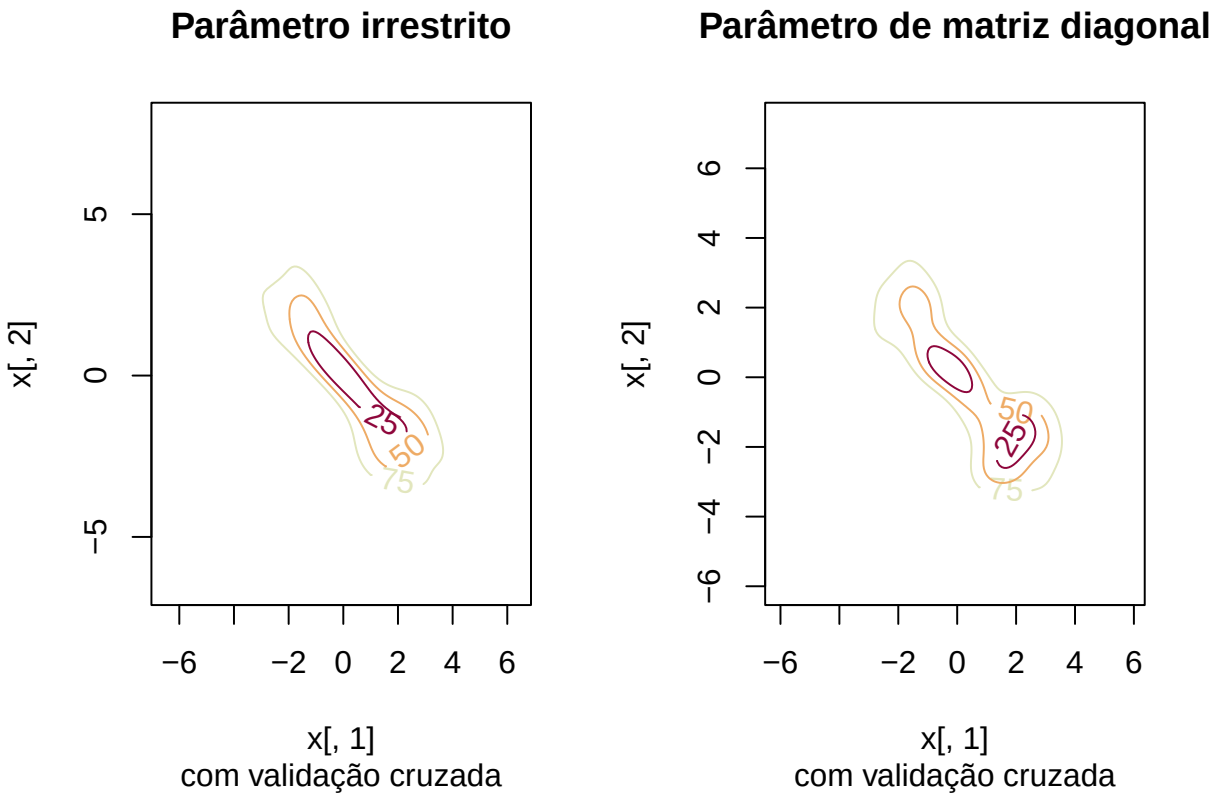
A diferença nesta escolha pode ser observada nos contornos:

```
par(mfrow = c(1, 2))
plot(fhat.pi1,main="Parâmetro irrestrito")
plot(fhat.pi2,main="Parâmetro de matriz diagonal")
```



O pacote tráz ainda a implementação das funções anteriores, mas com validação cruzada: *Hscv* e *Hscv.diag*.

```
Hscv1 = Hscv(x=x)
Hscv2 = Hscv.diag(x=x)

fhat.pi3 <- kde(x=x, H=Hscv1)
fhat.pi4 <- kde(x=x, H=Hscv2)
par(mfrow = c(1, 2))
plot(fhat.pi3,main="Parâmetro irrestrito",sub="com validação cruzada")
plot(fhat.pi4,main="Parâmetro de matriz diagonal",sub="com validação cruzada")
```



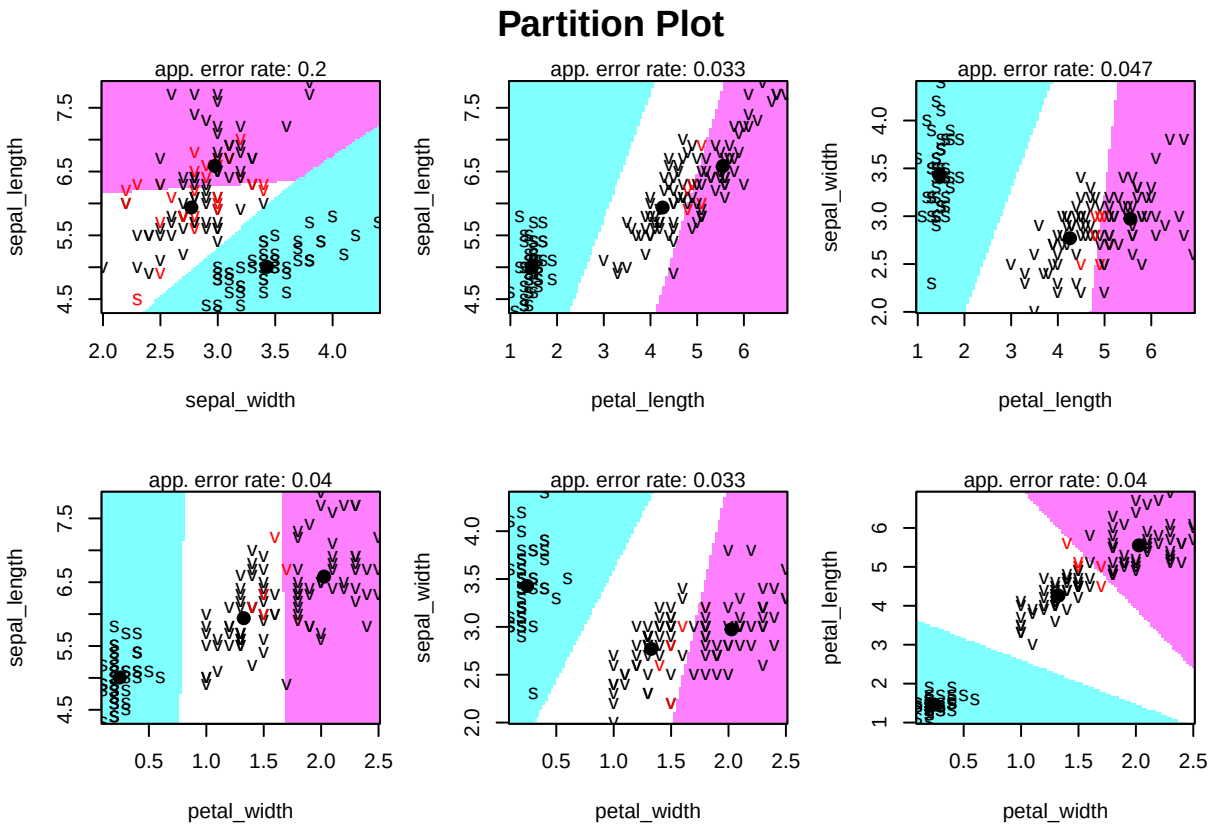
Vemos que os contornos não conseguiram distinguir bem as 3 normais para esta amostra, encontrando aparente normal "comportada" próxima à x = 2; y = -2 — E possivelmente entendendo se tratar de uma normal com alta variância ou distribuição de cauda pesada centrada em torno de x = y = 0.

Apesar disso, observando atentamente os parâmetros geradores da amostra, os resultados foram bastante satisfatórios, sendo possível modelar com alguma fidedignidade a amostra que foi utilizada como *input* para as funções.

17. Apresentar um exemplo com classificador LDA e QDA.

Análise discriminante linear (LDA)

Testaremos um LDA ao clássico conjunto de dados *Iris*. Iremos utilizar todas as variáveis do conjunto, plotar o comportamento dois-a-dois, calcular estatísticas relacionadas ao modelo, como matriz de confusão e proporção de acerto por classe do modelo.



Matriz de confusão:

	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	48	2
virginica	0	1	49

Proporção de classificações corretas em cada grupo:

	x
setosa	1.00
versicolor	0.96
virginica	0.98

Proporção total de classificação correta: 0.98

Com isso, temos que a taxa de erro do modelo pontual é de 2%.

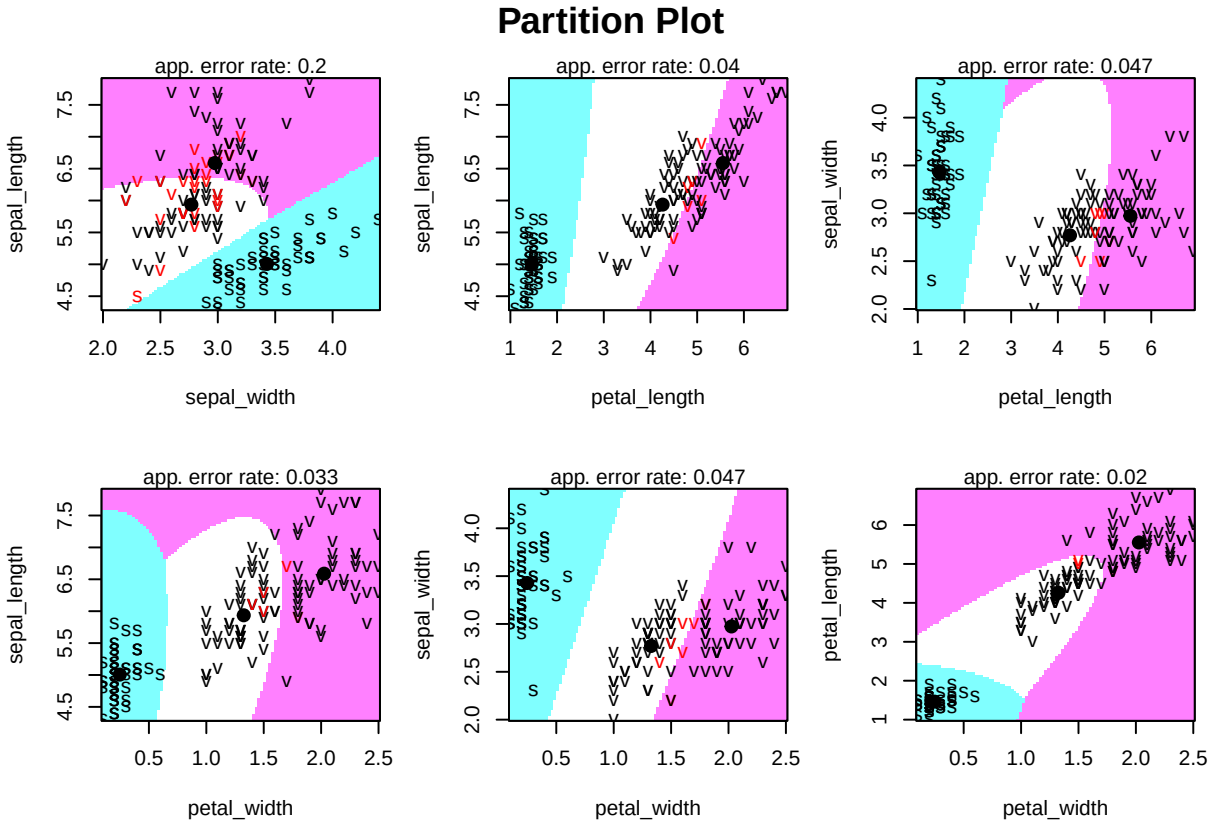
Análise discriminante quadrática (QDA)

Testaremos um QDA ao clássico conjunto de dados *Iris*. Iremos utilizar todas as variáveis do conjunto, plotar o comportamento dois-a-dois, testar a modelagem via validação cruzada, e por fim calcular estatís-

ticas relacionadas aos modelos, como matriz de confusão, proporção de acerto por classe dos modelos, e também o erro aparente e estimação do erro aparente para a modelagem com validação cruzada.

```
if (!require("MASS")) install.packages("MASS")
```

```
dados = iris %>% janitor::clean_names()
dados$species = factor(dados$species)
gqda <- qda(species~., data = dados)

gqdap1 <- predict(gqda)
gqctable1 <- table(dados$species, gqdap1$class)

klaR::partimat(species~., data=dados, method="qda",
         plot.matrix = F, imageplot = T,prec=100)
```




```
# Com validação cruzada
gqdaVC <- qda(species~., data = dados,CV=T)
```

Matriz de confusão:

	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	48	2
virginica	0	1	49

Proporção de classificações corretas em cada grupo:

	x
setosa	1.00
versicolor	0.96
virginica	0.98

Proporção total de classificação correta: 0.98

```
# Matrizes de confusão:
M <- table(dados$species, gqdap1$class)
MCV <- table(dados$species, gqdaVC$class)

# APER e \hat{E}APR:
APER <- (sum(M)-sum(diag(M)))/sum(M) # APER x_1,x_2
E_APR <- (sum(MCV)-sum(diag(MCV)))/sum(MCV) # \hat{E} APR x_1,x_2
```

O erro aparente (APER) deste conjunto foi calculado como sendo 0.02; enquanto que a estimação da taxa de erro aparente ($\hat{E}(AER)$) foi calculada como 0.0266667. Notamos que apesar de o erro estimado via validação cruzada Jackknife ter sido maior que o erro aparente, esta é uma estimativa mais robusta em comparação com o resultado sem validação cruzada.

Comentários

Para este conjunto de dados, LDA e QDA tiveram basicamente a mesma performance. Portanto, o ideal seria utilizar o modelo mais parsimonioso: LDA!

Referências

[1] J.E. Chacón.T. Duong. Multivariate plug-in bandwidth selectionwith unconstrained pilot bandwidth matrices. Sociedad de Estadística e Investigación Operativa. 2009.