Lista 2

César A. Galvão - 190011572

Gabriela Carneiro - 180120816

João Vitor Vasconcelos - 170126064

Índice

Questão 3 **2**
Item a 2
Item b 3
Item c 6
Item d 9

Questão 4 **12**

Questão 5 **13**

Referências **15**

Questão 3

Considerando duas classes com distribuição normal multivariada tal que $\omega_1 \sim N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ com

$$\boldsymbol{\mu} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}_2$$

e $\omega_2 \sim N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ com

$$\boldsymbol{\mu} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad \text{e} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}_2$$

Item a

Gere 100 valores para ω_1 e ω_2.

Os valores para as variáveis serão gerados utilizando o algoritmo de Cholesky disponibilizado na lista, comentada abaixo para facilitar a compreensão:

```
rmvn.cholesky <- function( n , mu , Sigma ) {
  p <- length(mu) # normal p-variada
  Q <- chol(Sigma) # {base} cholesky decomposition
  Z <- matrix(rnorm(n*p), nrow=n, ncol = p) # matriz nxp ~
↪  N_p(0,1)
  X <- Z %*% Q + # matriz nxp ~ N_p(0,Σ)
          matrix(mu, n, p, byrow=TRUE) #mu_1 e mu_2 em cada linha
  return(X)
}
```

A seguir, é escolhida uma semente para o gerador de números aleatórios e são geradas as variáveis. Uma prévia dos dados é exibida em seguida.

```
set.seed(11572)

n <- 100

mu1 <- c(1, 0)
Sigma1 <- diag(2)
omega1 <- rmvn.cholesky(n, mu1, Sigma1)

mu2 <- c(-1, 0)
```

Tabela 1: Primeiras linhas de ω_1 e ω_2.

V1	V2
0.4528201	0.2274710
1.2804296	1.3931905
-0.0467663	-1.2357199
1.7315127	-3.3383750
0.9339194	0.0419395
0.7733280	1.6790085

V1	V2
-1.3386455	-1.6352549
0.3181510	-1.3237408
-1.4029014	0.3951291
-0.2880302	-0.2085144
-2.2679073	-0.2438205
-0.4362316	0.6367181

```
Sigma2 <- diag(2)
omega2 <- rmvn.cholesky(n, mu2, Sigma2)
```

Item b

Verifique se os valores gerados seguem distribuição $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Lembre que neste caso, o par deve seguir uma distribuição χ^2 e cada variável deve ter distribuição Normal.

Considerando que cada variável deve seguir uma distribuição Normal univariada e que, neste caso, as matrizes de covariância são identidades — ou seja, as normais bivariadas são compostas por normais univariadas independentes entre si —, verificamos a normalidade da distribuição de cada componente de ω_1 e ω_2 utilizando gráficos quantil-quantil e testes Shapiro-Wilk.

O código a seguir exibe a função montada para gerar os gráficos.

```
plot_qq_mtvn <- function(x, var, i){
  x %>%
  as_tibble %>%
  ggplot(aes(sample = {{ var }})) +
  geom_qq(alpha = .4) +
  geom_qq_line() +
  labs(x = "Quantis teóricos",
       y = substitute(paste("Quantis observados -", omega[i],
       ↪  "-", var),
                      list(var = substitute(var),
                      i = substitute(i)))) +
  theme_bw()+
  theme(
    axis.text = element_text(size = 7.5),
```

```
    axis.title = element_text(size = 7.5)
  )
}
```

Com os gráficos gerados na Figura 1 a seguir vemos que a princípio não há motivos para visualmente rejeitar a normalidade dos dados. As caudas, como é de se esperar, são ou pouco mais pesadas.

```
plot_grid(plot_qq_mtvn(omega1, V1, 1),
          plot_qq_mtvn(omega1, V2, 1),
          plot_qq_mtvn(omega2, V1, 2),
          plot_qq_mtvn(omega2, V2, 2), nrow = 2)
```

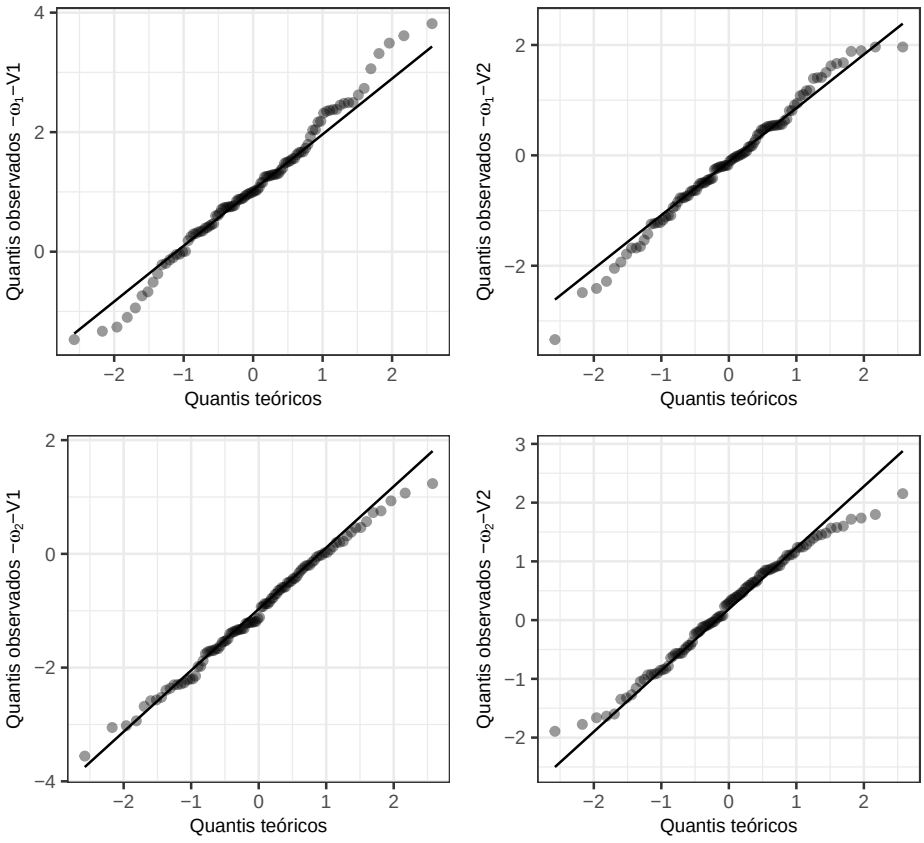

Figura 1: QQ-plot para ω_1 e ω_2 em relação à distribuição normal.

```r
bind_cols(omega1, omega2) %>%
  as_tibble() %>%
  summarise(across(everything(), ~ shapiro.test(.)$p.value)) %>%
  knitr::kable(
    col.names = c("omega1_V1", "omega1_V2", "omega2_V1",
    ↪  "omega2_V2")
  )
```

Tabela 4: Teste de Shapiro-Wilk para componentes de ω_1 e ω_2.

omega1_V1	omega1_V2	omega2_V1	omega2_V2
0.5506312	0.5300882	0.8559406	0.2021182

O teste Henze-Zirkler para normalidade multivariada aponta p-valores 0.58 e 0.29 para ω_1 e ω_2, respectivamente, não dando indícios de que se deva rejeitar a hipótese de normalidade.

Além disso, para uma visualização estilo QQ-plot, utiliza-se `heplots::cqplot`. A normalidade multivariada é avaliada utilizando a distância Malanobis ao quadrado, que conforme Artes e Barroso (2023), segue a forma:

$$D_M^2 = (\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \sim \chi_p^2 \tag{1}$$

Na Figura 2 não é possível identificar pontos fora da banda de confiança, enquanto na Figura 3 é possível ver alguns pontos de quantis inferiores fora da banda. No entanto, abos os conjuntos de dados serão considerados normais multivariadas.

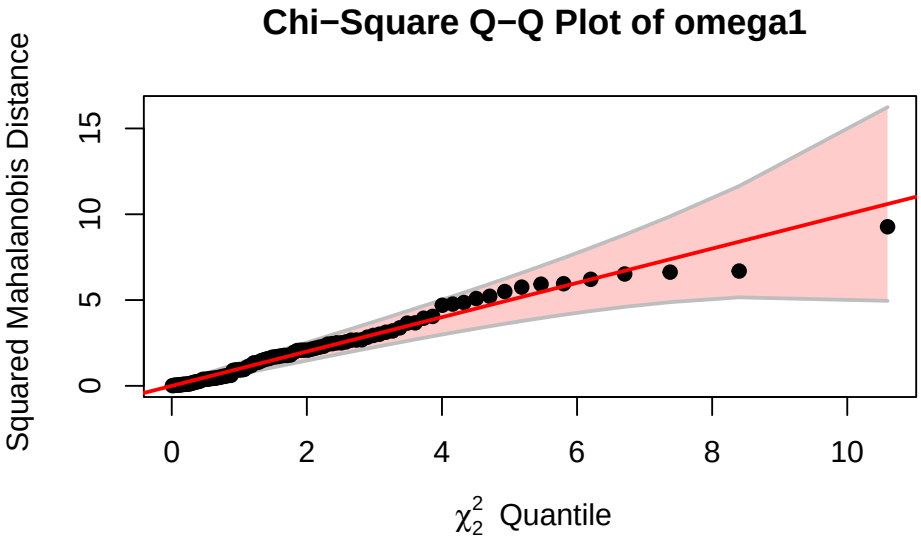

Figura 2: QQ-plot para ω_1 em relação à distribuição χ^2.

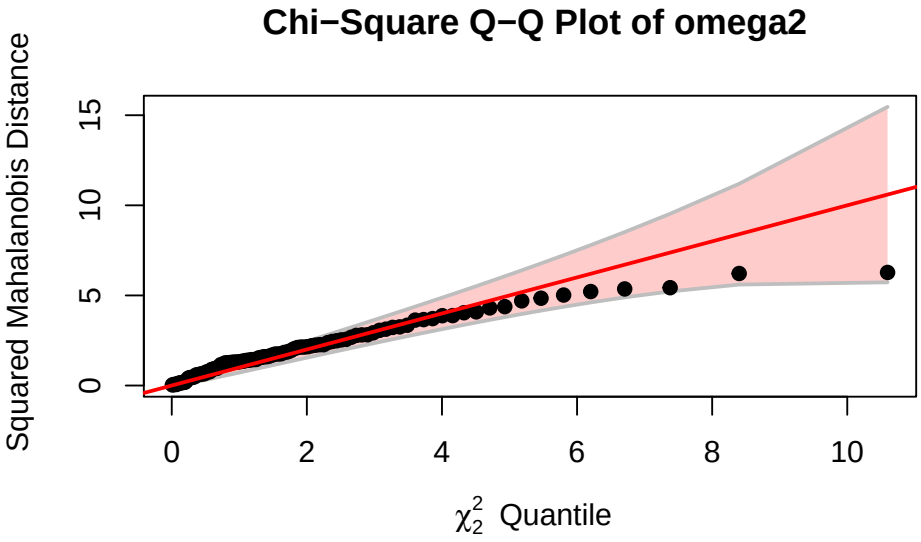

Figura 3: QQ-plot para ω_2 em relação à distribuição χ^2.

Item c

Para um determinado μ na razão de verossimilhança, determine as regiões Ω_1 e Ω_2 na regra de Neyman-Pearson.

A seguir é criada uma função para avaliar a razão de verossimilhanças $\frac{p(x|\omega_1)}{p(x|\omega_2)}$ para nossos conjuntos de dados.

```
p = 2
S <- diag(2)

# funcao para resolver a exponencial da verossimilhanca
expo_norm <- function(x, mu, S){
  part1 <- -t(t(x)-mu) %*% solve(S)
  part2 <- (x-mu)

  return(part1[,1]*part2[,1] + part1[,2]*part2[,2])
}

#funcao para calcular a razao de verossimilhancas
↪
razao_vero <- function(x, p, mu1, mu2, S){
  return(
  # verossimilhanca sob mu1 dividida por
   exp(expo_norm(x, mu1, S)/2)/
    # verossimilhanca sob mu2
   exp(expo_norm(x, mu2, S)/2)
  )
}
```

Classificaremos considerando as regiões em que a razão de verossimilhanças dá mais suporte a $p(x|\omega_1)$ ou $p(x|\omega_2)$.

```
razoes <- c(
  razao_vero(omega1, 2, mu1, mu2, S),
  razao_vero(omega2, 2, mu1, mu2, S))

# limite da regiao de classificaçao

limite <- quantile(razoes, probs = 0.5)

# tabelas com regioes corretas e classificacao

classificacoes <- tibble(
  regioes_corretas = rep(c("1", "2"), each = 100),
  x1 = c(omega1[,1], omega2[,1]),
  x2 = c(omega1[,2], omega2[,2]),
  razoes = razoes,
  classificacao = if_else(razoes > limite, "1", "2"),
  acertos = if_else(regioes_corretas == classificacao, 1, 0)
```

```
)
```

Quando a razão for superior à mediana das verossimilhanças, classificaremos como ω_1 e no complementar quando for inferior à mediana. Em outras palavras,

$$\boldsymbol{x} \in \Omega_1 \Rightarrow \frac{p(x|\omega_1)}{p(x|\omega_2)} > 1.1583663. \tag{2}$$

Dessa forma, há 84% de acertos. A Tabela 5 a seguir nos dá o desempenho da classificação:

Tabela 5: Tabela de contingências de classificações em ω_1 e ω_2 utilizando a regra de alocação de Neyman-Pearson.

	omega1	omega2
omega1	84	16
omega2	16	84

Avaliamos graficamente as classificações a seguir:

```
fig_corretas <- classificacoes %>%
  ggplot(aes(x = x1, y = x2, color = regioes_corretas))+
  geom_point()+
  theme_bw()+
  theme(legend.position = "bottom")

fig_classificadas <- classificacoes %>%
  ggplot(aes(x = x1, y = x2, color = classificacao))+
  geom_point()+
  theme_bw()+
  theme(legend.position = "bottom")

plot_grid(fig_corretas, fig_classificadas)
```

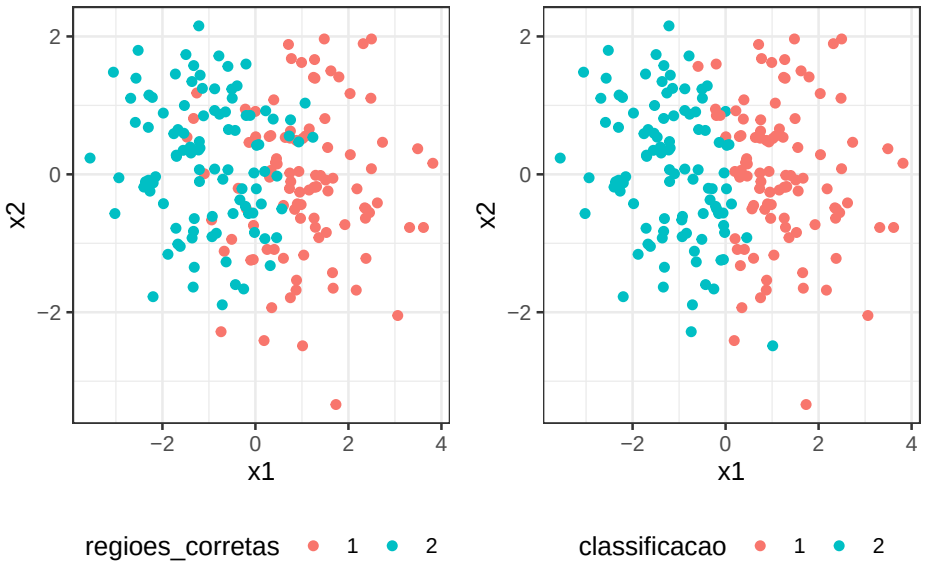

Figura 4: Grupos reais e preditos nas classes ω_1 e ω_2 utilizando regra de Neyman-Pearson.

Este comportamento é esperado, visto que há matrizes de covariância e μ_2 iguais, diferindo apenas em X_1.

Item d

Considere diferentes valores $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}]^\top$ e utilize a regra de decisão de Bayes para alocar estes valores em Ω_1 ou Ω_2.

De acordo com a regra de Bayes,

$$\boldsymbol{x} \in \Omega_1 \Rightarrow \frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{p(x|\omega_2)}{p(x|\omega_1)}. \tag{3}$$

Dessa forma, classificamos a seguir as observações:

```
razoes_bayes <- tibble( #monta as razoes de verossimilhanca
  grupos = rep(c("1", "2"), each = 100),
  vero1 = c(razao_vero(omega1, 2, mu1, mu2, S),
  ↪  razao_vero(omega2, 2, mu1, mu2, S)),
  vero2 = c(razao_vero(omega1, 2, mu2, mu1, S),razao_vero(omega2,
  ↪  2, mu2, mu1, S)),
```

```
  x1 = c(omega1[,1], omega2[,1]),
  x2 = c(omega1[,2], omega2[,2])
  ) %>%
  mutate(
    classificacao = case_when(
      vero1 > vero2 ~ "1",
      vero2 > vero1 ~ "2"
    ),
    acertos = if_else(grupos == classificacao, 1, 0)
  )
```

Nesse caso há 82% de acertos. A Tabela 6 a seguir nos dá o desempenho da classificação:

Tabela 6: Tabela de contingências de classificações em ω_1 e ω_2 utilizando regra de Bayes.

	omega1	omega2
omega1	84	16
omega2	16	84

Avaliamos graficamente as classificações a seguir:

```
fig_corretas_bayes <- razoes_bayes %>%
  ggplot(aes(x = x1, y = x2, color = grupos))+
  geom_point()+
  theme_bw()+
  theme(legend.position = "bottom")

fig_classificadas_bayes <- razoes_bayes %>%
  ggplot(aes(x = x1, y = x2, color = classificacao))+
  geom_point()+
  theme_bw()+
  theme(legend.position = "bottom")

plot_grid(fig_corretas_bayes, fig_classificadas_bayes)
```

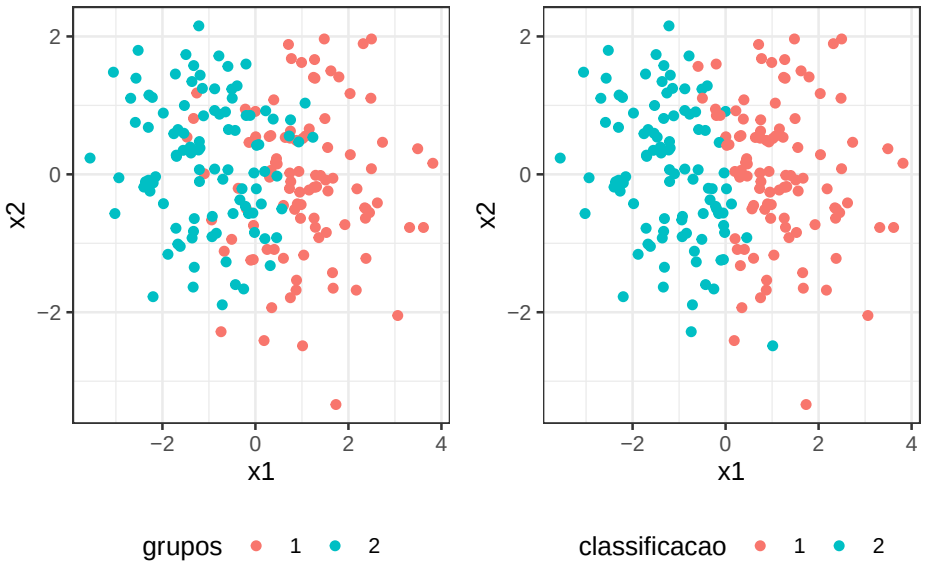

Figura 5: Grupos reais e preditos nas classes ω_1 e ω_2 utilizando regra de Bayes.

Neste caso, utilizando qualquer das regras de classificação se obtém os mesmos resultados.

Questão 4

Considere duas classes com distribuições multivariadas tal que $p(x|\omega_1) \sim N_p(\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$ e $p(x|\omega_2) \sim N_p(\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$. Mostre que o logaritmo da razão de verossimilhança é linear em relação ao vetor de características $\mathbf{x}$.

Considere a função densidade para a distribuição normal multivariada:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\} \tag{4}$$

A razão de verossimilhanças é dada já simplificada em relação às constantes de normalização por

$$\mathcal{L}(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} = \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1})\}}{\exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2})\}} \tag{5}$$

$$\begin{aligned}
\ell(\boldsymbol{x}) &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_1})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_1}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_2})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu_2}) \\
&= -\frac{1}{2}\left(\boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{x} - \boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{\mu_1} - \boldsymbol{\mu_1}^\top \Sigma^{-1}\boldsymbol{x} + \boldsymbol{\mu_1}^\top \Sigma^{-1}\boldsymbol{\mu_1}\right) \\
&+ \frac{1}{2}\left(\boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{x} - \boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{\mu_2} - \boldsymbol{\mu_2}^\top \Sigma^{-1}\boldsymbol{x} + \boldsymbol{\mu_2}^\top \Sigma^{-1}\boldsymbol{\mu_2}\right) \\
&= \frac{1}{2}\left(\boldsymbol{x}^\top \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{x} + \boldsymbol{\mu_2}^\top \Sigma^{-1}\boldsymbol{\mu_2} - \boldsymbol{\mu_1}^\top \Sigma^{-1}\boldsymbol{\mu_1}\right),
\end{aligned} \tag{6}$$

que é linear em relação a $\boldsymbol{x}$.

Questão 5

Pesquise sobre pacotes disponíveis no `R` para realizar análise de discriminantes e classificação. Verifique as regras de decisão utilizadas nestes pacotes. Compare os recursos do R com procedimento em outra linguagens de programação, como `SAS`, `Python`, `Matlab`.

O pacote MASS, disponível em R, oferece uma variedade de métodos para análise discriminante[1]:

1. Análise discriminante linear (LDA): Esta técnica utiliza combinações lineares de preditores para prever a classe de uma observação, assumindo distribuição normal para as variáveis preditoras e igualdade de variâncias entre as classes.
2. Análise discriminante quadrática (QDA): Mais flexível que a LDA, esta abordagem não assume que a matriz de covariância das classes seja a mesma.
3. Análise discriminante de mistura (MDA): Neste método, cada classe é considerada como uma mistura gaussiana de subclasses.
4. Análise discriminante flexível (FDA): Utiliza combinações não-lineares de preditores, como splines, para a classificação.
5. Análise discriminante regularizada (RDA): Aplica regularização para melhorar a estimativa das matrizes de covariância em situações onde o número de preditores é maior que o de amostras nos dados de treinamento, resultando em uma melhoria na análise discriminante.

Além disso, destaca-se o uso da função `MASS::lda()` que aplica o teorema de Bayes para calcular a probabilidade de cada classe com base nos valores dos preditores. Outro recurso interessante é o pacote `nproc`, que utiliza métodos de classificação de Neyman-Pearson para identificar regiões onde um método é mais eficaz que o outro (Tong, Feng, e Li 2018). O pacote `Rlda` também é relevante, especialmente para análise de agrupamentos em diferentes tipos de dados, como entradas multinomiais, Bernoulli e binomiais. Esse pacote é especialmente útil para o reconhecimento de padrões não supervisionados, sobretudo para análise de agrupamento de adesão mista de dados categóricos (Albuquerque, Valle, e Li 2019).

No ambiente SAS, está disponível o procedimento `DISCRIM`, que é utilizado para desenvolver um critério discriminante em conjuntos de observações contendo variáveis quantitativas e uma variável de classificação, permitindo assim a

[1] http://www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/

classificação de cada observação em grupos específicos[2].

Quando a distribuição dentro de cada grupo é considerada normal multivariada, emprega-se um método paramétrico para criar uma função discriminante. Essa função, também chamada de critério de classificação, é determinada por meio de uma medida de distância generalizada ao quadrado. O critério de classificação pode ser formulado com base nas matrizes de covariância dentro do grupo (resultando em uma função quadrática) ou na matriz de covariância agrupada (resultando em uma função linear), levando em conta as probabilidades anteriores dos grupos. As informações de calibração podem ser armazenadas em um conjunto de dados especial no SAS, sendo posteriormente aplicadas a outros conjuntos de dados.

Quando não é possível fazer suposições sobre a distribuição dentro de cada grupo, ou quando se presume que a distribuição não é normal multivariada, são utilizados métodos não paramétricos para estimar as densidades específicas do grupo. Esses métodos incluem técnicas como kernel e vizinho mais próximo. O procedimento `DISCRIM` emprega kernels uniformes, normais, Epanechnikov, biweight ou triweight para a estimativa de densidade. As distâncias de Mahalanobis ou Euclidiana podem ser utilizadas para avaliar a proximidade entre observações.

A distância de Mahalanobis pode ser calculada com base na matriz de covariância completa ou na matriz diagonal de variâncias. Com o método -nearest-neighbor, é usada a matriz de covariância agrupada para calcular as distâncias de Mahalanobis. Já com o método de kernel, tanto as matrizes de covariância dentro do grupo quanto a matriz de covariância agrupada podem ser empregadas para esse cálculo. Com as densidades específicas do grupo estimadas e as probabilidades anteriores associadas, é possível avaliar as estimativas de probabilidade posterior de pertencimento ao grupo para cada classe.

A análise discriminante canônica, relacionada à análise de componentes principais e correlação canônica, é uma técnica de redução de dimensão utilizada no `PROC DISCRIM`. Nesse procedimento, são derivadas variáveis canônicas que resumem a variação entre classes de maneira semelhante às componentes principais, resultando em um critério discriminante que é sempre obtido no `PROC DISCRIM`. Para realizar uma análise discriminante canônica sem a utilização do critério discriminante, recomenda-se o uso do procedimento `CANDISC`.

[2] https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_discrim_sect001.htm

Referências

Albuquerque, Pedro H. M., Denis Ribeiro do Valle, e Daijiang Li. 2019. «Bayesian LDA for Mixed-Membership Clustering Analysis: The Rlda Package». *Knowledge-Based Systems* 163 (janeiro): 988–95. https://doi.org/10.1016/j.knosys.2018.10.024.

Artes, Rinaldo, e Lucia Pereira Barroso. 2023. «Métodos multivariados de análise estatística».

Tong, Xin, Yang Feng, e Jingyi Jessica Li. 2018. «Neyman-Pearson Classification Algorithms and NP Receiver Operating Characteristics». *Science Advances* 4 (2). https://doi.org/10.1126/sciadv.aao1659.