# 多元统计第二次作业

蒋文馨

2020/5/17

## 目录



# 1 LDA on Wine Data

8.1 Consider the wine data. Compute a LDA, draw a 2D-scatterplot of the first two LDF coordinates, and color-code the points by wine type. What do you notice?

```
library(MASS)
library(readr)
wine = read_table2("wine.train.txt")
wine.lda = lda(wine$type ~ ., wine)
LD = predict(wine.lda)
wine.lda
```

```
## Call:
## lda(wine$type ~ ., data = wine)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3220339 0.3898305 0.2881356
##
## Group means:
```

```
##    Alcohol MalicAcid      Ash   AlcAsh        Mg  Phenols      Flav
## 1 13.73763  1.933684 2.493684 17.00526 108.50000 2.845263 2.9697368
## 2 12.27739  1.976957 2.281739 20.69565  94.73913 2.250435 2.1556522
## 3 13.15059  3.458235 2.448235 21.44118  98.17647 1.725882 0.7864706
##   NonFlavPhenols     Proa    Color       Hue       OD   Proline
## 1      0.3050000 1.864474 5.554474 1.0831579 3.101579 1114.2105
## 2      0.3641304 1.721522 3.163043 1.0730435 2.838696  523.9348
## 3      0.4561765 1.137059 7.417059 0.6985294 1.687941  620.2941
##
## Coefficients of linear discriminants:
##                         LD1          LD2
## Alcohol        -0.511244928  0.969702370
## MalicAcid       0.258344142  0.292124971
## Ash            -1.025022033  2.660728316
## AlcAsh          0.137671902 -0.216784939
## Mg              0.005982794  0.001589583
## Phenols         0.825384675  0.204914606
## Flav           -1.497234903 -0.527624083
## NonFlavPhenols -0.782651707 -0.741267095
## Proa           -0.132131408 -0.563902276
## Color           0.332815833  0.167774503
## Hue            -1.230827739 -1.361751242
## OD             -1.264082609 -0.098889671
## Proline        -0.002683669  0.002759903
##
## Proportion of trace:
##    LD1    LD2
## 0.6837 0.3163
```

从图 1 右侧可以看出，利用 LD1 得分基本能将 3 种酒分开，只利用 LD2 得分几乎不能区分第 1 和第 3 种酒。但是由散点图可知，结合 LD1 和 LD2 可以很好的将 3 种酒分开。这说明虽然 LD1 是判别效率最高的函数，在 LD2 中仍然存在有助于判别的信息。注意到 LD1 的 Proportion of trace 为 0.6837，LD2 为 0.3163，也说明了这一点。

# 2 Coefficient of LDF

8.3 Suppose $\mathbf{X}_1 \sim \mathcal{N}_r(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{XX})$ and $\mathbf{X}_2 \sim \mathcal{N}_r(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{XX})$ are independently distributed. Consider the statistic

$$\frac{\{\mathrm{E}(\mathbf{a}'\mathbf{X}_1) - \mathrm{E}(\mathbf{a}'\mathbf{X}_2)\}^2}{\mathrm{var}(\mathbf{a}'\mathbf{X}_1 - \mathbf{a}'\mathbf{X}_2)}$$

as a function of a. Show that $\mathbf{a} \propto \Sigma_{XX}^{-1}(\mu_1 - \mu_2)$ maximizes the statistic by using a Lagrange multiplier approach.

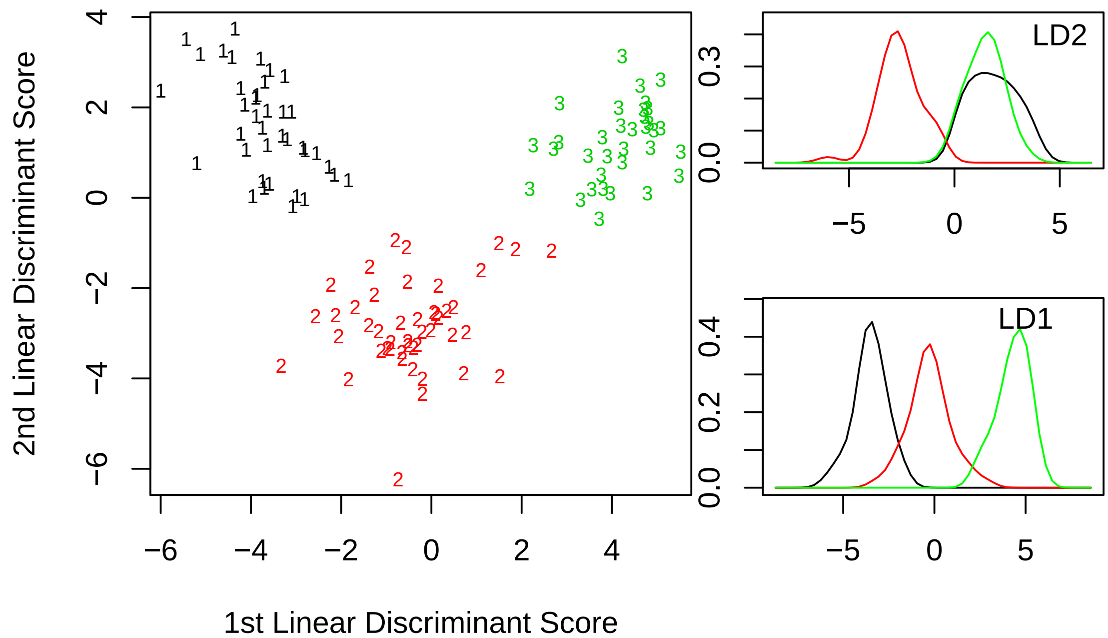



图 1: Wine

**解:**

$$
\begin{aligned}
T &= \frac{\{\mathrm{E}(\mathbf{a}'\mathbf{X}_1) - \mathrm{E}(\mathbf{a}'\mathbf{X}_2)\}^2}{\operatorname{var}(\mathbf{a}'\mathbf{X}_1 - \mathbf{a}'\mathbf{X}_2)} \\
&= \frac{(\mathbf{a}'(\mu_1 - \mu_2))^2}{\operatorname{var}(\mathbf{a}'\mathbf{X}_1) + \operatorname{var}(\mathbf{a}'\mathbf{X}_2)} \\
&= \frac{(\mathbf{a}'(\mu_1 - \mu_2))\,(\mathbf{a}'(\mu_1 - \mu_2))'}{2\mathbf{a}'\mathbf{\Sigma}_{XX}\mathbf{a}} \\
&= \frac{\mathbf{a}'(\mu_1 - \mu_2)(\mu_1 - \mu_2)'\mathbf{a}}{2\mathbf{a}'\mathbf{\Sigma}_{XX}\mathbf{a}}
\end{aligned}
$$

利用拉格朗日乘数法最大化 $T$, 等价于在约束 $\mathbf{a}'\mathbf{\Sigma}_{XX}\mathbf{a} = 1$ 下, 最大化 $\mathbf{a}'(\mu_1 - \mu_2)(\mu_1 - \mu_2)'\mathbf{a}$. 构造拉格朗日函数

$$
\begin{aligned}
L(\mathbf{a}, \lambda) &= \mathbf{a}'(\mu_1 - \mu_2)(\mu_1 - \mu_2)'\mathbf{a} - \lambda(\mathbf{a}'\mathbf{\Sigma}_{XX}\mathbf{a} - 1), \\
\frac{\partial L}{\partial \mathbf{a}} &= 2[(\mu_1 - \mu_2)(\mu_1 - \mu_2)' - \lambda\mathbf{\Sigma}_{XX}]\mathbf{a} =: 0, \\
\frac{\partial L}{\partial \lambda} &= 1 - \mathbf{a}'\mathbf{\Sigma}_{XX}\mathbf{a} =: 0.
\end{aligned}
$$

记 $b = (\mu_1 - \mu_2)'\mathbf{a}$, 注意 **b 不是向量**, 代入上式, 得

$$
\lambda = b^2, \quad (\mu_1 - \mu_2)b = \lambda\mathbf{\Sigma}_{XX}\mathbf{a}.
$$

故 $\mathbf{a} = b^{-1}\mathbf{\Sigma}_{XX}^{-1}(\mu_1 - \mu_2)$. 即 $\mathbf{a} \propto \Sigma_{XX}^{-1}(\mu_1 - \mu_2)$.

# 3 LDA and QDA on Transformed Iris Data

8.6 Try the following transformation on the iris data. Set X 5 = X 1 /X 2 and X 6 = X 3 /X 4 . Then, X 5 is a measure of sepal shape and X 6 is a measure of petal shape. Take logarithms of X 5 and of X 6 . Plot the transformed data, and carry out an LDA on X 5 and X 6 alone. Estimate the misclassification rate for the transformed data. Do the same for the QDA procedure.

```
library(MASS)
#transform data and plot----
iris.trsf = data.frame(
  cbind(
    iris$Sepal.Length / iris$Sepal.Width,
    iris$Petal.Length / iris$Petal.Width,
    iris$Species
  )
)
colnames(iris.trsf) = c('Sepal.Shape', 'Petal.Shape', 'Species')
iris.trsf[, -3] = apply(iris.trsf[, -3], 2, log)
```

```
plot(
  iris.trsf[, -3],
  col = iris.trsf$Species,
  pch = iris.trsf$Species,
  cex = .6
)
legend(
  'topright',
  c('setosa', 'versicolor', 'virginica'),
  col = c(1, 2, 3),
  pch = c(1, 2, 3)
)
```

```
#lda----
trsf.lda = lda(iris.trsf$Species ~ ., iris.trsf, CV = T)
paste(
  'misclassification rate of lda:',
  sum(trsf.lda$class != iris.trsf$Species) / dim(iris)[1]
)
```

```
## [1] "misclassification rate of lda: 0.18"
```

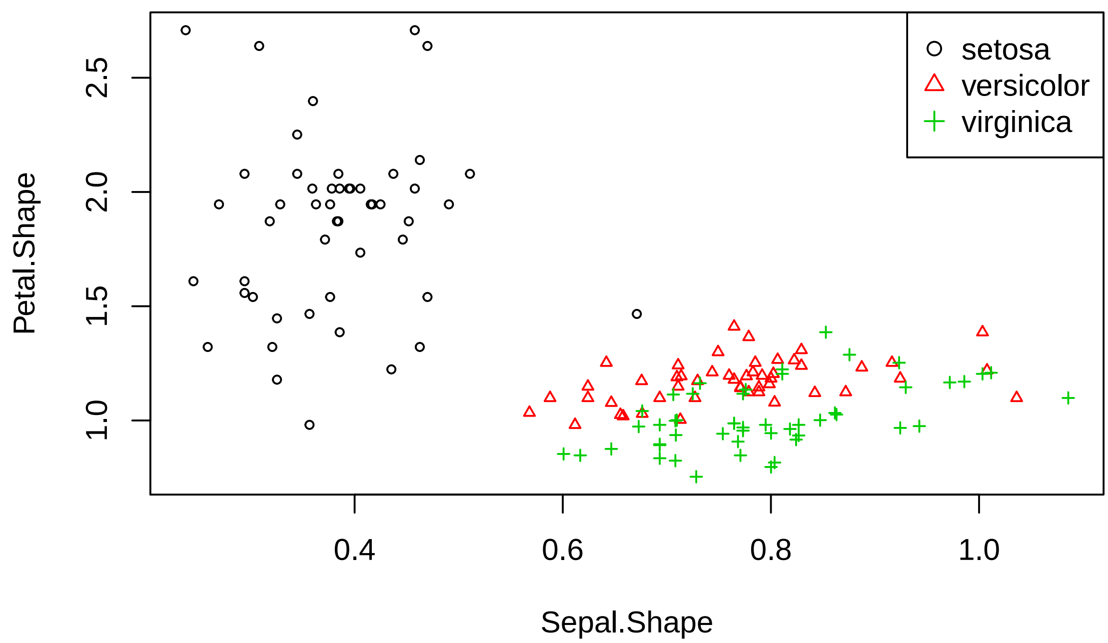



图 2: Iris Transformed Data

```
#qda----
trsf.qda = qda(iris.trsf$Species ~ ., iris.trsf, CV = T)
paste(
  'misclassification rate of qda:',
  sum(trsf.qda$class != iris.trsf$Species) / dim(iris)[1]
)
```

```
## [1] "misclassification rate of qda: 0.14"
```

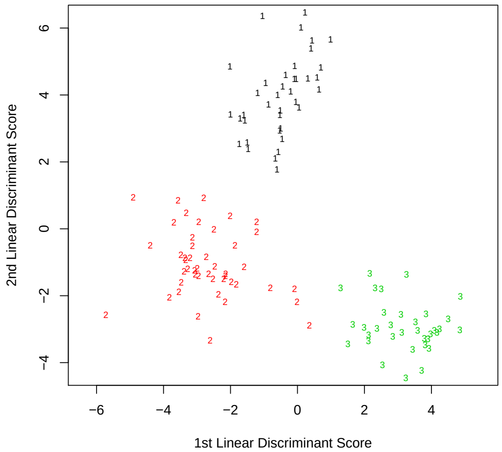
# 4 附录

## 4.1 关于 lda() 及先验概率的探讨

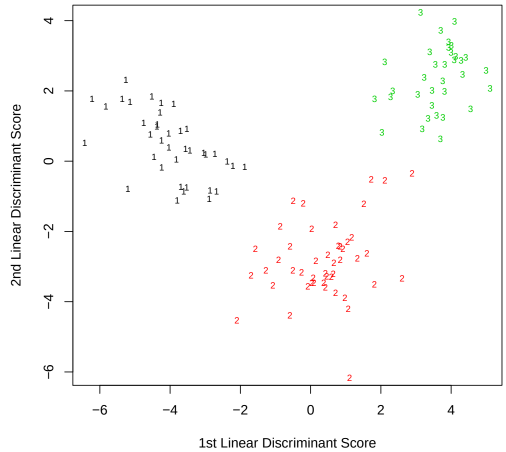
从图 3 中可以看出，先验概率的微小改变不会对分类结果造成很大影响。所以当样本分布相对均衡时，可以不考虑先验概率的影响。但如果样本分布极度不均衡，就应该考虑先验概率的影响。

## 4.2 实现高书版费舍尔判别

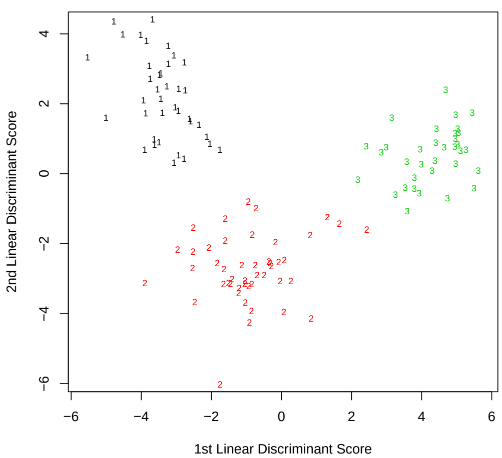
以下代码参考高慧璇版《应用多元统计分析》5.3 节费舍尔判别，即通过求 $A^{-1}B$ 的特征向量求解 LD 得分，并绘出类似作业第一题的散点图（见图 4 ）。**这里没有考虑先验概率。**



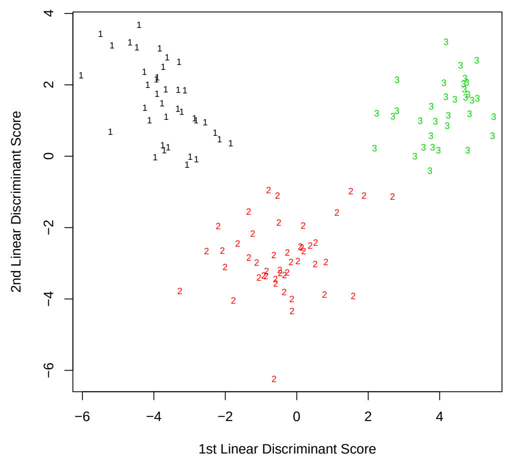

(a) pri = c(.001, .998, .001)



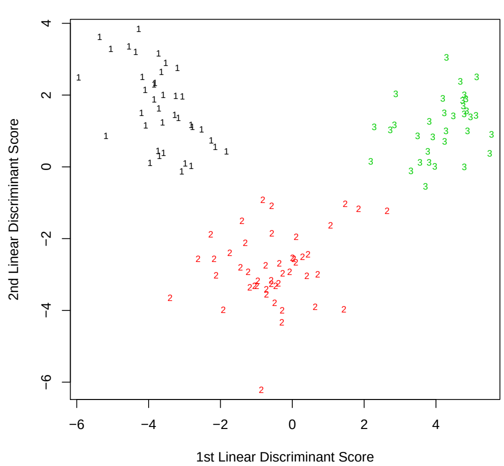

(b) pri = c(.998, .001, .001)



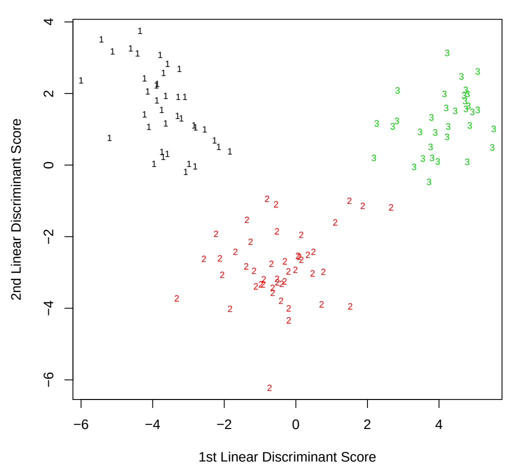

(c) pri = c(.2, .2, .6)




(d) pri = c(.4, .2, .4)




(e) pri = c(1, 1, 1) / 3




(f) pri = c(38, 46, 34) / 118

图 3: 不同先验对 lda 绘图结果的影响

```r
n = dim(wine)[1]#num of sample
ni = c(sum(wine$type == 1),
       sum(wine$type == 2),
       sum(wine$type == 3))
pri = ni / n#prior prob.
p = 13#num of feature
g = 3#num of group(or type)

data = apply(as.matrix(wine[, 1:p]), 2, scale)
type = wine$type

data.mean = as.matrix(colMeans(data[, 1:p]))

A = matrix(0, p, p)#合并的组内离差阵
B = A#组间离差阵
xbar_total = colMeans(data[, 1:p])
xbari = matrix(0, g, p)
for (i in 1:g) {
  xbari[i, ] = colMeans(data[which(type == i), 1:p])
}
for (i in 1:g) {
  xbar = xbari[i, ]
  B = B + ni[i] * (xbar - xbar_total) %*% t(xbar - xbar_total)
  A = A + t(data[which(type == i), 1:p]) %*%
    data[which(type == i), 1:p] - ni[i] * xbar %*% t(xbar)
}
eig = eigen(solve(A) %*% B)
head(eig$values)
```

```
## [1]  9.588416e+00+0.000000e+00i  4.435204e+00+0.000000e+00i
## [3] -5.356303e-16+8.207997e-16i -5.356303e-16-8.207997e-16i
## [5]  8.881784e-16+0.000000e+00i  7.532185e-16+0.000000e+00i
```

```r
fun.num = 2#判别函数数量
ld.score = data %*% Re(eig$vectors[, 1:fun.num]) * (-1)
#为了方便和lda函数的绘图结果比较，做一个翻转，乘系数-1
plot(
  ld.score,
  col = wine$type,
  type = 'n',
  xlab = "1st Linear Discriminant Score",
```

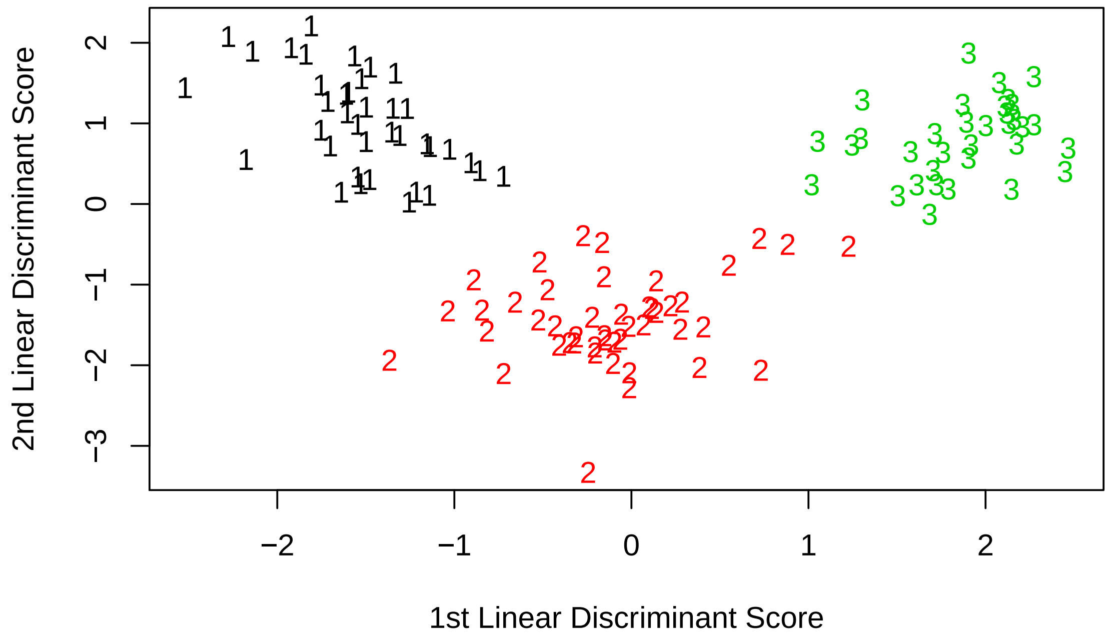



图 4: 高书版 FDA(不考虑先验)

```
  ylab = "2nd Linear Discriminant Score"
)
text(ld.score, labels = wine$type, col = wine$type)
```

### 4.3 利用教材 8.99-8.101 公式实现 LDA

这里考虑了先验概率。

```
#运行这一段代码之前要运行上一段代码
sigmaxx = A / (n - g)
sigmaxx.inv = solve(sigmaxx)
ij = combn(1:g, 2)#从g个分类中选择两个计算L
chs.num = choose(g, 2)#要比较的情况总数
L = matrix(0, n, chs.num + 1)#最后一列记录分类

for (ii in 1:n) {
  #对每一个样本
  for (jj in 1:chs.num) {
    #计算ii样本的L_ij
    i = ij[1, jj]
```

```
    j = ij[2, jj]
    b = (xbari[i,] - xbari[j,]) %*% sigmaxx.inv
    b0 = -1 / 2 * ((xbari[i,] - xbari[j,]) %*% sigmaxx.inv %*%
                     (xbari[i,] + xbari[j,])) + log(pri[i] / pri[j])
    L[ii, jj] = b %*% as.matrix(data[ii,]) + b0
  }
  tmp = which.max(abs(L[ii,]))
  L[ii, chs.num + 1] = if (L[ii, tmp] > 0)
    ij[1, tmp]
  else
    ij[2, tmp]
}
```

**注意：**虽然我确定 lda() 不是通过这种方法实现，而且这种方法是存在问题的（见习题 8.14）。但是我使用不同的先验，对比上述方法与 lda() 的结果，发现二者结果一致。所以我还是把这段代码放上来了。

## 4.4 lda() 代码

https://github.com/cran/MASS/blob/master/R/lda.R#L195