14 - Quadratic Classification and Classification for Several Populations

Junvie Pailden

SIUE, F2017, Stat 589

October 19, 2017

Classification of Two Normal Populations When Covariances are Unequal

- Consider the multivariate normal densities with $\mathbf{\Sigma}_i, i = 1, 2$, replacing $\mathbf{\Sigma}$.
- Substituting multivariate normal densities with different covariance matrices gives

$$
\begin{aligned}
R_1 : -\frac{1}{2}\mathbf{x}'(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1})\mathbf{x} + (\boldsymbol{\mu}_1'\mathbf{\Sigma}_1^{-1} - \boldsymbol{\mu}_2'\mathbf{\Sigma}_2^{-1})\mathbf{x} - k& \\
\geq \ln\left(\frac{c(1|2)}{c(2|1)} \cdot \frac{p_2}{p_1}\right)& \\
R_2 : -\frac{1}{2}\mathbf{x}'(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1})\mathbf{x} + (\boldsymbol{\mu}_1'\mathbf{\Sigma}_1^{-1} - \boldsymbol{\mu}_2'\mathbf{\Sigma}_2^{-1})\mathbf{x} - k& \\
< \ln\left(\frac{c(1|2)}{c(2|1)} \cdot \frac{p_2}{p_1}\right)&
\end{aligned}
$$

- where

$$k = \frac{1}{2}\ln\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}\right) + \frac{1}{2}(\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2)$$

- The classification regions are define by quadratic functions of $\mathbf{x}$.
- When $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, the term $-\frac{1}{2}\mathbf{x}'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\mathbf{x}$ disappears.

Quadratic Classification Rule

- Allocate to $\mathbf{x}_0$ to π_1 if

$$-\frac{1}{2}\mathbf{x}_0'(\boldsymbol{S}_1^{-1} - \boldsymbol{S}_2^{-1})\mathbf{x}_0 + (\bar{\boldsymbol{x}}_1'\boldsymbol{S}_1^{-1} - \bar{\boldsymbol{x}}_2'\boldsymbol{S}_2^{-1})\mathbf{x}_0 - k \geq \ln\left(\frac{c(1|2)}{c(2|1)} \cdot \frac{p_2}{p_1}\right)$$

- Allocate to $\mathbf{x}_0$ to π_2 otherwise.

Play with the IRIS data

```r
library(dplyr) # for data manipulation
# select variables, filter species
iris23 <- iris %>%
  select(Sepal.Length, Petal.Length, Species) %>%
  filter(Species != "setosa") %>%
  droplevels() # drop empty level (e.g. setosa)
```

Using qda() for Quadratic Classification/Discriminant Analysis

```
library(MASS)
iris.qda <- qda(Species ~
                  Sepal.Length + Petal.Length,
                  data = iris23)
# use equal priors for comparison
iris.predict.qda <- predict(iris.qda)
table(iris.predict.qda$class, iris23$Species)
```

```
#
#               versicolor virginica
#   versicolor          47         3
#   virginica            3        47
```

Quadratic and Linear Discrimination produces similar results for the IRIS data.

Example 11.1: Discriminating owners from nonowners of riding mowers

```
mower <- read.table("T11-1.DAT",
          col.names = c("income", "size", "riding"))
mower$riding <- recode_factor(mower$riding, `1` = "owner",
str(mower)
```

```
# 'data.frame': 24 obs. of  3 variables:
#  $ income: num  90 115.5 94.8 91.5 117 ...
#  $ size  : num  18.4 16.8 21.6 20.8 23.6 19.2 17.6 22.4 2
#  $ riding: Factor w/ 2 levels "owner","nonowner": 1 1 1 1
```

Linear Classification/Discriminant Analysis

```r
mower.lda <- lda(riding ~ income + size,
                 data = mower)
mower.lda$means # means by owner type
```

```
#            income size
# owner         109   20
# nonowner       87   18
```

```r
mower.pred.lda <- predict(mower.lda)
table(mower.pred.lda$class, mower$riding)
```

```
#
#              owner nonowner
#   owner         11        2
#   nonowner       1       10
```

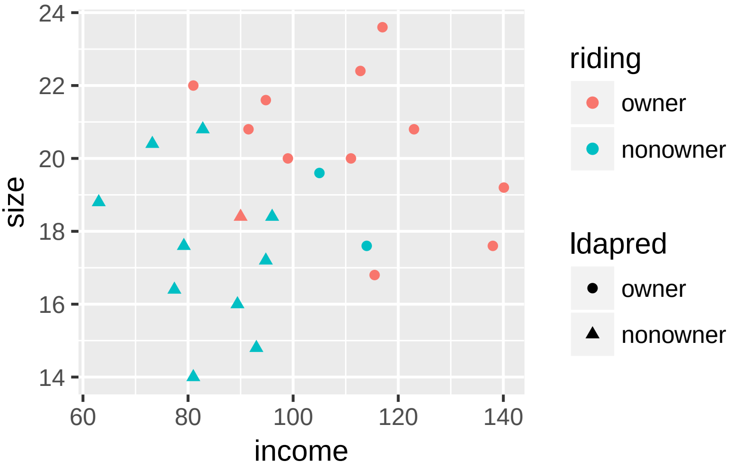
Add preicted class column to data

```
# add new predicted class column, save
mower <- mower %>%
  mutate(ldapred = mower.pred.lda$class)
head(mower)
```

```
#   income size riding  ldapred
# 1     90   18  owner nonowner
# 2    116   17  owner    owner
# 3     95   22  owner    owner
# 4     92   21  owner    owner
# 5    117   24  owner    owner
# 6    140   19  owner    owner
```

Plots

```
library(ggplot2)
# plot, add shape as predicted class
mower %>%
  ggplot(aes(x = income, y = size,
             colour = riding, shape = ldapred)) +
  geom_point()
```


size
24
22
20
18
16
14
60
80
100
120
140
income
riding
owner
nonowner
ldapred
owner
nonowner

Classification with Several Populations

- Let $f_i(\mathbf{x})$ be the density associated with population π_i, $i = 1, 2, \ldots, g$.

$$
\begin{aligned}
p_i &= \text{the prior prob'y of population } \pi_i, i = 1, \ldots, g \\
c(k|i) &= \text{cost of allocating an item to } \pi_k \text{ when it belongs to } \pi_i, k \neq i
\end{aligned}
$$

- Let R_k be the set of $\mathbf{x}$'s classified as π_k and

$$
\begin{aligned}
P(k|i) &= P(\text{classifying item as } \pi_k|\pi_i) = \int_{R_k} f_i(\mathbf{x})d\mathbf{x} \\
P(i|i) &= 1 - \sum_{\substack{k=1 \\ k \neq i}}^{g} P(k|i)
\end{aligned}
$$

Conditional Expected Cost of Misclassifying (CECM)

- The conditional expected cost of misclassifying an $\mathbf{x}$ from π_1 into π_2, or π_3,..., or π_k is

$$\begin{aligned} ECM(1) &= P(2|1)c(2|1) + P(3|1)c(3|1) + \cdots + P(g|1)c(g|1) \\ &= \sum_{k=2}^{g} P(k|1)c(k|1) \end{aligned}$$

- The conditional expected cost occurs with prior proby p_1, the proby of π_1.
- Similarly, we can obtain $ECM(2), \ldots, ECM(g)$.

Overall ECM

- The overall ECM is

$$
\begin{aligned}
ECM &= p_1 ECM(1) + \cdots + p_g ECM(g) \\
&= \sum_{i=1}^{g} p_i \left(\sum_{\substack{k=1 \\ k \neq i}}^{g} P(k|i) c(k|i) \right)
\end{aligned}
$$

- Choose mutually exclusive and exhaustive classification regions $R_1, \ldots, R_g$ such that ECM is a minimum.

Minimum ECM Classification Rule with Equal Misclassification Costs

- Suppose misclassification costs are equal (say all equal to 1).
- Allocate $\mathbf{x}_0$ to π_k if

$$p_k f_k(\mathbf{x}) > p_i f_i(\mathbf{x}) \quad \text{for all } i \neq k \tag{1}$$

or

$$\ln\, p_k f_k(\mathbf{x}) > \ln\, p_i f_i(\mathbf{x}) \tag{2}$$

- Note that this misclassification rule is identical to the one that maximizes the "posterior" probability

$$\begin{aligned} P(\pi_k|\mathbf{x}) &= P(\mathbf{x}\text{ comes from } \pi_k \text{ given that } \mathbf{x} \text{ was observed}) \\ &= \frac{p_k f_k(\mathbf{x})}{\sum_{i=1}^{g} p_i f_i(\mathbf{x})} = \frac{(\text{prior}) \times (\text{likelihood})}{\Sigma[(\text{prior}) \times (\text{likelihood})]} \end{aligned}$$

for $k = 1, \ldots, g$

Classification with Normal Populations

- Let $f_i(\mathbf{x}) \sim N_p(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, possibly unequal $\boldsymbol{\Sigma}_i$, $i = 1, \ldots, g$
- Allocate $\mathbf{x}$ to π_k if

$$\begin{aligned}\ln p_k f_k(\mathbf{x}) &= \ln p_k - \left(\frac{p}{2}\right)\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| \\ &\quad - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) \\ &= \max_i \ln p_i f_i(\mathbf{x})\end{aligned}$$

- Let $d_i^Q(\mathbf{x})$ be the **quadratic discriminant score** given by

$$d_i^Q(\mathbf{x}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) + \ln p_i,$$

where $i = 1, \ldots, g$

Quadratic Discriminant Analysis (QDA)

- Using Minimum Total Probability of Misclassification (TPM) Rule for Normal Populations
- (**QDA**) Allocate $\mathbf{x}$ to π_k if the **quadratic discriminant score**

$$d_k^Q(\mathbf{x}) = \max\left\{d_1^Q(\mathbf{x}), \ldots, d_g^Q(\mathbf{x})\right\}$$

where

$$d_i^Q(\mathbf{x}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln p_i,$$

Linear Discriminant Analysis (LDA)

- When $\mathbf{\Sigma}_i = \mathbf{\Sigma}$, for $i = 1, \ldots, g$, the quadratic discriminator score becomes

$$
\begin{aligned}
d_i^Q(\mathbf{x}) &= -\frac{1}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\mathbf{x}'\mathbf{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_i'\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i'\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln p_i \\
&= -\frac{1}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\mathbf{x}'\mathbf{\Sigma}^{-1}\mathbf{x} + d_i(\mathbf{x})
\end{aligned}
$$

where $d_i(\mathbf{x})$ is called the **linear discriminant score**,

$$
d_i(\mathbf{x}) = \boldsymbol{\mu}_i'\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i'\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln p_i
$$

- The first two terms are the same for all $d_i^Q(\mathbf{x})$ so we ignore these terms.
- (**LDA**) Allocate $\mathbf{x}$ to π_k if the **linear discriminant score**

$$
d_k(\mathbf{x}) = \max\{d_1(\mathbf{x}), \ldots, d_g(\mathbf{x})\}
$$

Sample Version of linear discriminant scores

- We estimate $d_i(\mathbf{x})$ by

$$\hat{d}_i(\mathbf{x}) = \bar{\boldsymbol{x}}_i' \boldsymbol{S}_p^{-1} \mathbf{x} - \frac{1}{2} \bar{\boldsymbol{x}}_i' \boldsymbol{S}_p^{-1} \bar{\boldsymbol{x}}_i + \ln p_i, \text{ for } i = 1, \ldots, g$$

- Equivalently, we can also consider the squared distances (estimate of the second term of $d_i^Q(\mathbf{x})$)

$$D_i^2(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}}_i)' \mathbf{S}_p^{-1} (\mathbf{x} - \bar{\mathbf{x}}_i)$$

Sample Linear Discriminant Analysis

- Estimated TPM Rule for Equal-Covariance Normal Populations
- Allocate $\mathbf{x}$ to π_k if the **estimated linear discriminant score**

$$\hat{d}_k(\mathbf{x}) = \max\left\{\hat{d}_1(\mathbf{x}), \ldots, \hat{d}_g(\mathbf{x})\right\}$$

- Or, (in terms of squared distances) allocate $\mathbf{x}$ to π_k if

$$-\frac{1}{2}D_k^2(\mathbf{x}) + \ln p_k \text{ is largest}$$

- Or, (when priors are all equal) allocate $\mathbf{x}$ to π_k if $D_k^2(\mathbf{x})$ is smallest (observation is assigned to closest population)

Example 11.11: Classifying a potential business-school graduate students

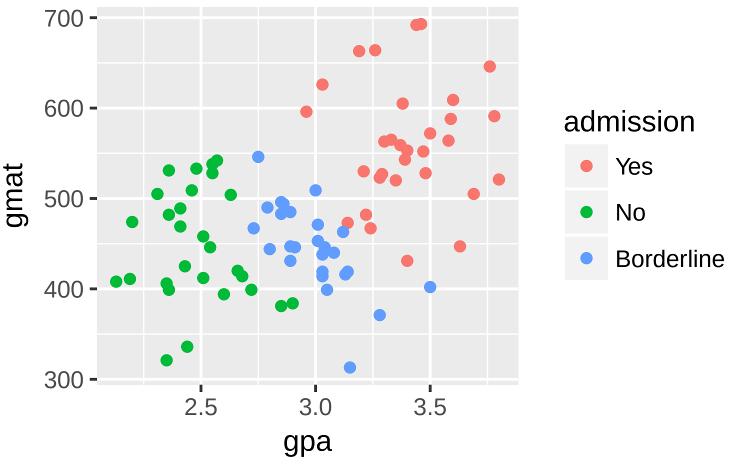
```
bsgrad.df <- read.table("T11-6.DAT", col.names = c("gpa", '
bsgrad.df$admission <- recode_factor(bsgrad.df$admission,

      `1` =  "Yes", `2` = "No", `3` =  "Borderline")
str(bsgrad.df)
```

```
# 'data.frame': 85 obs. of  3 variables:
#  $ gpa      : num  2.96 3.14 3.22 3.29 3.69 3.46 3.03 3.1
#  $ gmat     : int  596 473 482 527 505 693 626 663 447 58
#  $ admission: Factor w/ 3 levels "Yes","No","Borderline":
```

```r
head(bsgrad.df)
```

```
#   gpa gmat admission
# 1 3.0  596       Yes
# 2 3.1  473       Yes
# 3 3.2  482       Yes
# 4 3.3  527       Yes
# 5 3.7  505       Yes
# 6 3.5  693       Yes
```

```r
bsgrad.df %>%
  ggplot(aes(x = gpa, y = gmat, colour = admission)) +
  geom_point()
```



Linear Discriminant Analysis, Grad School

```
bs.lda <- lda(admission ~ gpa + gmat,
              data = bsgrad.df)
bs.pred.lda <- predict(bs.lda)
(cm.bs.lda <- table(bs.pred.lda$class,
                    bsgrad.df$admission))
```

```
#
#                Yes No Borderline
#   Yes           28  0          1
#   No             0 26          1
#   Borderline     3  2         24
```

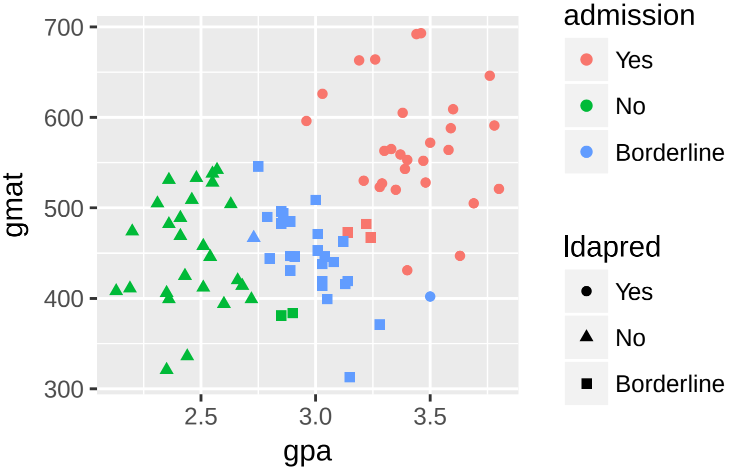
```
# accuracy rate
sum(diag(cm.bs.lda))/nrow(bsgrad.df)
```

```
# [1] 0.92
```

LDA, Business School

```r
# add new predicted class column
bsgrad.df <- bsgrad.df %>%
  mutate(ldapred = bs.pred.lda$class)

# plot, add shape as predicted class
bsgrad.df %>%
  ggplot(aes(x = gpa, y = gmat,
             colour = admission, shape = ldapred)) +
  geom_point()
```

LDA result, Business School



Quadratic Discriminant Analysis, Business School

```
bs.qda <- qda(admission ~ gpa + gmat,
                  data = bsgrad.df)
bs.pred.qda <- predict(bs.qda)
(cm.bs.qda <- table(bs.pred.lda$class,
                   bsgrad.df$admission))
```

```
#
#               Yes No Borderline
#   Yes          28  0          1
#   No            0 26          1
#   Borderline    3  2         24
```

```
# accuracy rate
sum(diag(cm.bs.qda))/nrow(bsgrad.df)
```

```
# [1] 0.92
```

LDA and QDA, Business School

Similar Result for LDA and QDA

```
cm.bs.lda
```

```
#
#                 Yes No Borderline
#   Yes            28  0          1
#   No              0 26          1
#   Borderline      3  2         24
```

```
cm.bs.qda
```

```
#
#                 Yes No Borderline
#   Yes            28  0          1
#   No              0 26          1
#   Borderline      3  2         24
```

QDA, Business School

```r
# add new predicted class column
bsgrad.df <- bsgrad.df %>%
  mutate(qdapred = bs.pred.qda$class)

# plot, add shape as predicted class
bsgrad.df %>%
  ggplot(aes(x = gpa, y = gmat,
             colour = admission, shape = qdapred)) +
  geom_point()
```

QDA result, Grad School