クレジット:
UTokyo Online Education 統計データ解析 Ⅱ 2018 小池祐太



東京大学 THE UNIVERSITY OF TOKYO

CC BY NC ND

# 統計データ解析 II (平成３０年度)

東京大学大学院数理科学研究科
統計データ解析教育研究グループ

村田　昇 (早稲田大学, 東京大学)
吉田朋広 (東京大学)
小池祐太 (東京大学)

# 第 7 章 判別分析

## 7.1. 目的

**判別分析** *(discriminant analysis)* とは, ある個体が $K(\geq 2)$ 個のクラスのいずれかに属するとき, その個体の属性 (特徴量) $X = (X_1, \ldots, X_q)$ からどのクラスに属するか予測するモデルを構築するための分析法である. 数学的には, クラスラベルを表す質的変数を $Y \in \{1, \ldots, K\}$ としたとき, $X = \boldsymbol{x}$ の下で $Y = k$ となる条件付き確率

$$p_k(\boldsymbol{x}) := P(Y = k | X = \boldsymbol{x})$$

に対するモデルを構築することが目的となる. ここで, 上式右辺の条件付き確率 $P(Y = k|X = \boldsymbol{x})$ は, $X$ が離散型の確率変数の場合

$$P(Y = k | X = \boldsymbol{x}) := \frac{P(Y = k, X = \boldsymbol{x})}{P(X = \boldsymbol{x})}$$

で定義されて, $X$ が連続型の確率変数の場合, 式

$$\frac{P(Y = k, x_1 - \varepsilon \leq X_1 \leq x_1 + \varepsilon, \ldots, x_q - \varepsilon \leq X_q \leq x_q + \varepsilon)}{P(x_1 - \varepsilon \leq X_1 \leq x_1 + \varepsilon, \ldots, x_q - \varepsilon \leq X_q \leq x_q + \varepsilon)} \tag{7.1}$$

において, $\varepsilon$ を 0 に近づけるときの極限として定義する. ここに, $x_j$ はベクトル $\boldsymbol{x}$ の第 $j$ 成分を表す.

観測データとしては, 組 $(Y, X_1, \ldots, X_q)$ に対する $n$ 個の観測データ

$$\{(y_i, x_{i1}, \ldots, x_{iq})\}_{i=1}^n$$

が与えられている状況を考える.

$p_k(\boldsymbol{x})$ をモデル化するアプローチとしては以下の 2 通りの方法がある:

(1) $p_k(\boldsymbol{x})$ を直接モデル化する (例: ロジスティック回帰).
(2) $Y = k$ の下での $X$ の条件付き確率質量関数もしくは条件付き確率密度関数 $f_k(\boldsymbol{x})$ のモデル化を通じて $p_k(\boldsymbol{x})$ をモデル化する.

本講義では後者のアプローチについて考察する. ここで, $X$ が離散型の場合, $f_k(\boldsymbol{x})$ は $Y = k$ の下での $X$ の条件付き確率質量関数を表し,

$$f_k(\boldsymbol{x}) := P(X = \boldsymbol{x} | Y = k)$$

で定義される. 他方, $X$ が連続型の場合, $f_k(\boldsymbol{x})$ は $Y = k$ の下での $X$ の条件付き確率密度関数, すなわちクラス $k$ に属するようなサンプルの場合に $X$ が従う確率分布の確率密度関数を表す. より厳密には, すべての $a_j \leq b_j$ $(j = 1, 2, \ldots, q)$ に対して

$$P(a_1 \leq X_1 \leq b_1, \ldots, a_q \leq X_q \leq b_q | Y = k) = \int_{a_1}^{b_1} \cdots \int_{a_q}^{b_q} f_k(x_1, \ldots, x_q) dx_1 \cdots dx_q$$

を満たすような非負の値をとる $q$ 変数関数 $f_k(x_1, \ldots, x_q)$ のことを指す. ここで, $P(a_1 \leq X_1 \leq b_1, \ldots, a_q \leq X_q \leq b_q | Y = k)$ は事象 $Y = k$ が起こった下で事象 $a_j \leq X_j \leq b_j$ $(j = 1, \ldots, q)$ が起こる条件付き確率を表す. すなわち,

$$P(a_1 \leq X_1 \leq b_1, \ldots, a_q \leq X_q \leq b_q | Y = k) = \frac{P(a_1 \leq X_1 \leq b_1, \ldots, a_q \leq X_q \leq b_q, Y = k)}{P(Y = k)}$$

である.

## 7.2. ベイズの公式

$f_k(\boldsymbol{x})$ のモデル化を通じて $p_k(\boldsymbol{x})$ のモデルが得られることの数学的原理は, 次の**ベイズの公式** *(Bayes' formula)* によって与えられる:

**定理 7.1 (ベイズの公式).**

$$P(Y=k|X=\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})P(Y=k)}{\sum_{l=1}^K f_l(\boldsymbol{x})P(Y=l)}.$$

ここでは $X$ が離散型の場合に上の公式が成り立つことを示す ($X$ が連続型の場合は極限操作を行うことで示すことができるが, 少し技術的となるためここでは省略する. 7.6 節参照). まず, 定義より

$$f_k(\boldsymbol{x}) = P(X=\boldsymbol{x}|Y=k) = \frac{P(X=\boldsymbol{x}, Y=k)}{P(Y=k)}$$

であるから,

$$P(X=\boldsymbol{x}, Y=k) = f_k(\boldsymbol{x})P(Y=k) \tag{7.2}$$

が成り立つ. これを $P(Y=k|X=\boldsymbol{x})$ の定義式に代入して

$$P(Y=k|X=\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})P(Y=k)}{P(X=\boldsymbol{x})} \tag{7.3}$$

を得る. 一方で, $Y$ は $1,\dots,K$ のうちいずれか一つの値のみ取ることに注意すると,

$$P(X=\boldsymbol{x}) = \sum_{l=1}^K P(X=\boldsymbol{x}, Y=l)$$

が成り立つ. 上式右辺の総和の各項に (7.2) 式を適用して

$$P(X=\boldsymbol{x}) = \sum_{l=1}^K f_l(\boldsymbol{x})P(Y=l) \tag{7.4}$$

を得る. この式を (7.3) 式に代入することで, 証明すべき等式が得られる. なお, (7.4) 式は**全確率の公式** *(formula of total probability)* と呼ばれることがある.

$Y=k$ となる確率を $\pi_k = P(Y=k)$ と書くことにすると, ベイズの公式より,

$$p_k(\boldsymbol{x}) = \frac{f_k(\boldsymbol{x})\pi_k}{\sum_{l=1}^K f_l(\boldsymbol{x})\pi_l}$$

が成り立つ. 従って, $\pi_1,\dots,\pi_K$ がわかっている, もしくはデータから推定できるのであれば, $f_k(\boldsymbol{x})$ をモデル化することで $p_k(\boldsymbol{x})$ のモデルが得られる. $\pi_1,\dots,\pi_K$ は**事前確率** *(prior probability)* と呼ばれ, 特徴量が与えられる前に予測できるそれぞれのクラスに属する確率である. 事前確率に関する特別な情報がない場合は, $\pi_k$ はデータから自然に決まる確率

$$\frac{Y=k \text{ であるサンプル数}}{\text{全サンプル数}}$$

で推定される. 一方で, 例えば日本人のサンプルから身長や体重などの特徴量を観測したデータから, その人が喫煙者か否かを判別するためのモデルを構築するといった状況の場合, 事前確率として日本人の喫煙者の割合といったデータを使うことも考えられる. すなわち, $Y=1$ が喫煙者を表し, $Y=2$ が非喫煙者を表す場合, $\pi_1$ として日本人の喫煙者の割合を使い, $\pi_2$ として日本人の非喫煙者の割合を使うということが考えられる.

実際に観測データに基づいて, 特徴量が $X = \boldsymbol{x}$ であるようなデータの属するクラスを判別する際には, $p_k(\boldsymbol{x})$ を最大にするようなクラス $k$ にデータを分類する. 従って, 関数 $\delta_k(\boldsymbol{x})$ $(k = 1, \ldots, K)$ で,

$$p_k(\boldsymbol{x}) < p_l(\boldsymbol{x}) \Leftrightarrow \delta_k(\boldsymbol{x}) < \delta_l(\boldsymbol{x})$$

を満たすようなものが存在すれば, $\delta_k(\boldsymbol{x})$ を最大化するようなクラス $k$ にそのデータを分類すればよいことになる. このような関数 $\delta_k(\boldsymbol{x})$ を**判別関数** *(discriminant function)* と呼ぶ.

## 7.3. 線形判別分析

**線形判別分析** *(linear discriminant analysis)* では, $f_k(\boldsymbol{x})$ をクラスごとに異なる平均ベクトル $\boldsymbol{\mu}_k$ をもつが, すべてのクラスで共通の共分散行列 $\Sigma$ をもつような $q$ 変量正規分布の確率密度関数としてモデル化する:

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{q/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right).$$

いま,

$$\begin{aligned}
& p_k(\boldsymbol{x}) < p_l(\boldsymbol{x}) \\
\Leftrightarrow\ & f_k(\boldsymbol{x})\pi_k < f_l(\boldsymbol{x})\pi_l \\
\Leftrightarrow\ & \log f_k(\boldsymbol{x}) + \log \pi_k < \log f_l(\boldsymbol{x}) + \log \pi_l \\
\Leftrightarrow\ & -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k) + \log \pi_k < -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_l)^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_l) + \log \pi_l \\
\Leftrightarrow\ & \boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k < \boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{\mu}_l - \frac{1}{2}\boldsymbol{\mu}_l^\top \Sigma^{-1}\boldsymbol{\mu}_l + \log \pi_l
\end{aligned}$$

が成り立つから, **線形判別関数** *(linear discriminant function)*

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^\top \Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

を最大化するようなクラス $k$ にデータを分類すればよい. 線形判別関数の計算のためには各クラスごとの特徴量の平均ベクトル $\boldsymbol{\mu}_k$ およびすべてのクラスで共通の特徴量の共分散行列 $\Sigma$ を計算する必要があるが, これらはそれぞれ

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i:y_i=k} \boldsymbol{x}_i, \qquad \hat{\Sigma} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^\top$$

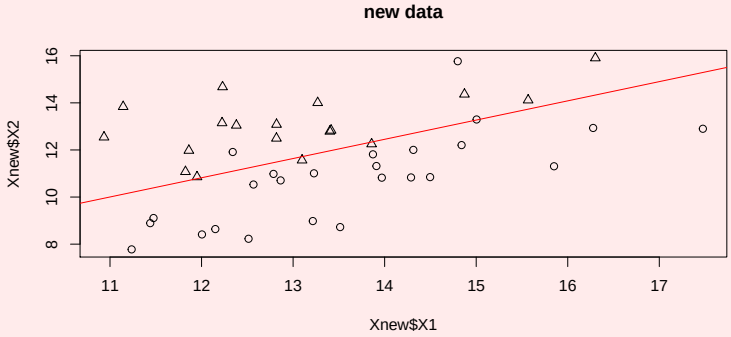
で推定すればよい.[1] ここに, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iq})^\top$ であり, $n_k$ は $y_i = k$ であるようなデータの総数を表す.

**7.3.1. R での実行.** パッケージ `MASS` には線形判別分析を実行するための関数 `lda()` が用意されている. 書式は関数 `lm()` とほとんど同じである (クラスラベルを目的変数, 特徴量を説明変数とする).

```
> # 人工データによる判別（２群の場合）
> # データの準備
> require(MASS)
> set.seed(123)
> mu1 <- c(14,11)
> mu2 <- c(13,13)
> Sigma <- matrix(c(1,0.7,0.7,1),2,2)*2.5
> n <- 30
> x1 <- mvrnorm(n,mu=mu1,Sigma=Sigma)
> x2 <- mvrnorm(n,mu=mu2,Sigma=Sigma)
```

[1] 以下簡単のために行列 $\hat{\Sigma}$ は正則であると仮定する.

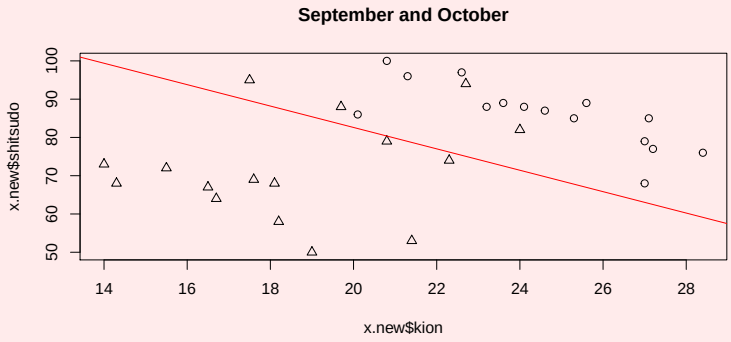
```
> X1 <- cbind(data.frame(x1),data.frame(cat=rep(0,n)))
> X2 <- cbind(data.frame(x2),data.frame(cat=rep(1,n)))
> X <- rbind(X1,X2)
> # plot(X[,1:2],pch=X[,3]+1)
> # 分析の開始：
> (mylda1 <- lda(cat~X1+X2,X))# トレーニングデータで判別関数を作る

Call:
lda(cat ~ X1 + X2, data = X)

Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
        X1       X2
0 13.82213 11.04054
1 13.09309 12.97810

Coefficients of linear discriminants:
          LD1
X1 -0.8319789
X2  1.0189425

> # 新しいデータを判別する：
> n1 <- 25
> n2 <- 18
> x1new <- mvrnorm(n1,mu=mu1,Sigma=Sigma)
> x2new <- mvrnorm(n2,mu=mu2,Sigma=Sigma)
> X1new <- cbind(data.frame(x1new),data.frame(cat=rep(0,n1)))
> X2new <- cbind(data.frame(x2new),data.frame(cat=rep(1,n2)))
> Xnew <- rbind(X1new,X2new)
> mypredict1<-predict(mylda1, newdata = Xnew[,1:2]) # Xnew を判別
> table(true = Xnew$cat, pred = mypredict1$class) # 真のクラスと予測されたクラスの比較

    pred
true  0  1
   0 22  3
   1  2 16

> mypredict1$class # 予測を真の分類と比較：

 [1] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 1 1 1
[39] 1 1 1 1 1
Levels: 0 1

> Xnew$cat

 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1

> #
> # compute the coefficients of the line
> myline <- function(z) {
+   a0<-as.vector(colMeans(z$means) %*% z$scaling)
+   a<-c(a0/z$scaling[2],-z$scaling[1]/z$scaling[2])
+   return(a)
+ }
> #  直線を引く：
> a <- myline(mylda1)
> plot(Xnew$X1,Xnew$X2,pch=Xnew[,3]+1,main="new data")
> abline(a,col="red")
```




(lda2g.r)

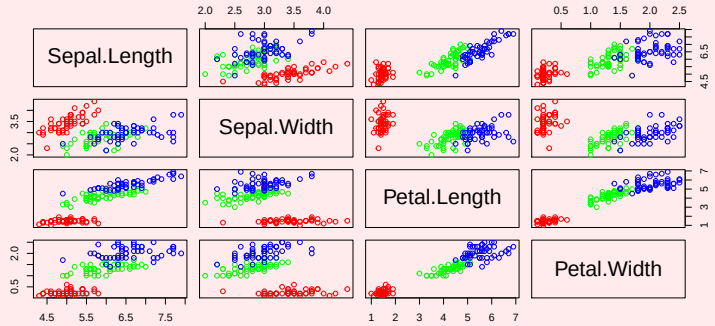
```
> # tokyo kion-shitsudo data
> # http://www.data.jma.go.jp/obd/stats/etrn/view/daily_s1.php
> # ?prec_no=44&block_no=47662&year=2016&month=09&day=1&view=p1
> require(MASS)
> x <- read.csv(file="SepOctTokyo_hkion_hshitsudo.csv",row.names=1, header = TRUE)
> # plot(x$kion,x$shitsudo,pch=x[,3]+1,main="September and October")
> idx <- seq(2,60,by = 2)
> x.learn <- x[idx,]# トレーニングデータ
> x.new <- x[-idx,]# 新しいデータ
> (mylda <- lda(cat~kion+shitsudo,x.learn))# トレーニングデータで判別関数を作る．等分散性は仮定する
Call:
lda(cat ~ kion + shitsudo, data = x.learn)

Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
      kion shitsudo
0 24.27333 85.73333
1 18.80667 70.86667

Coefficients of linear discriminants:
                LD1
kion     -0.21120310
shitsudo -0.07553831
> mypredict<-predict(mylda, newdata = x.new[,1:2]) # x.new を判別
> table(true = x.new$cat, pred = mypredict$class) # 真のクラスと予測されたクラスの比較
    pred
true  0  1
   0 15  0
   1  4 12
> mypredict$class # ９月／１０月の予測を比較
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1
Levels: 0 1
> x.new$cat # １０月はじめに誤判別が起きている
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
> # compute the coefficients of the line
> myline <- function(z) {
+   a0<-as.vector(colMeans(z$means) %*% z$scaling)
+   a<-c(a0/z$scaling[2],-z$scaling[1]/z$scaling[2])
```

```
+   return(a)
+ }
> (a <- myline(mylda))
[1] 138.525265  -2.795973
> plot(x.new$kion,x.new$shitsudo,pch=x.new[,3]+1,main="September and October")
> abline(a,col="red")
```




(tenki.r)

```
> ## データセット iris による例（３群の判別分析に lda を使ってみる）
> ## あやめの３品種 (setosa, versicolor, virginica) について，
> ## その萼片 (Sepal) および花弁 (Petal) の幅と長さを記録したデータセット
> ## 後者の情報から品種を判別することが目的
> pairs(subset(iris, select = -Species), col = rainbow(3)[iris$Species])
> ### 散布図 (Species ごとに色分け)
> ### 品種ごとに花弁・萼片の幅と長さの分布が異なるように見える
> ### 花弁の方が萼片より品種ごとの違いがある
```




```
> ## データをランダムに２分割して，一方を訓練データ，
> ## もう一方をテストデータとする
> set.seed(123)
> idx <- sample.int(nrow(iris), size = nrow(iris)/2)
> iris.train <- iris[idx, ] # 訓練データ
> iris.test <- iris[-idx, ] # テストデータ
> library(MASS) # パッケージのロード
> (mod1 <- lda(Species ~ Sepal.Length + Sepal.Width, data = iris.train))
Call:
lda(Species ~ Sepal.Length + Sepal.Width, data = iris.train)

Prior probabilities of groups:
    setosa versicolor  virginica
```

```
 0.3866667  0.2933333  0.3200000

Group means:
           Sepal.Length Sepal.Width
setosa         5.020690    3.482759
versicolor     5.850000    2.709091
virginica      6.558333    2.975000

Coefficients of linear discriminants:
                    LD1        LD2
Sepal.Length -2.469708 -0.8754951
Sepal.Width   3.233700 -1.8538021

Proportion of trace:
   LD1    LD2
0.9692 0.0308
> ### 萼片の長さ・幅を特徴量とする線形判別分析
> res1 <- predict(mod1) # 訓練データに対する予測結果
> head(res1$class) # 予測されたクラス (最初の 6 個)
[1] setosa     virginica  versicolor virginica  virginica  setosa
Levels: setosa versicolor virginica
> head(iris.train$Species) # 実際のクラス (最初の 6 個)
[1] setosa     virginica  versicolor virginica  virginica  setosa
Levels: setosa versicolor virginica
> table(true = iris.train$Species, pred = res1$class) # 真のクラスと予測されたクラスの比較
            pred
true         setosa versicolor virginica
  setosa         29          0         0
  versicolor      0         17         5
  virginica       0          7        17
> ### setosa は完全に判別できているが, versicolor と virginica の判別に少し誤りがある
> pred1 <- predict(mod1, newdata = iris.test) # テストデータに対する予測結果
> head(pred1$class) # 予測されたクラス (最初の 6 個)
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
> head(iris.test$Species) # 実際のクラス (最初の 6 個)
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
> table(true = iris.test$Species, pred = pred1$class) # 真のクラスと予測されたクラスの比較
            pred
true         setosa versicolor virginica
  setosa         20          1         0
  versicolor      0         17        11
  virginica       0          6        20
> ### setosa はほぼ判別できているが, versicolor と virginica の判別に少し誤りがある
> (mod2 <- lda(Species ~ Petal.Length + Petal.Width, data = iris.train))
Call:
lda(Species ~ Petal.Length + Petal.Width, data = iris.train)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3866667  0.2933333  0.3200000

Group means:
           Petal.Length Petal.Width
setosa         1.458621   0.2586207
versicolor     4.204545   1.3090909
virginica      5.433333   1.9291667
```

```
Coefficients of linear discriminants:
                 LD1       LD2
Petal.Length 1.737856 -1.973385
Petal.Width  2.738099  4.765673

Proportion of trace:
   LD1    LD2
0.9981 0.0019
> ### 花弁の長さ・幅を特徴量とする線形判別分析
> res2 <- predict(mod2) # 訓練データに対する予測結果
> head(res2$class) # 予測されたクラス (最初の 6 個)
[1] setosa     virginica  versicolor virginica  virginica  setosa
Levels: setosa versicolor virginica
> head(iris.train$Species) # 実際のクラス (最初の 6 個)
[1] setosa     virginica  versicolor virginica  virginica  setosa
Levels: setosa versicolor virginica
> table(true = iris.train$Species, pred = res2$class) # 真のクラスと予測されたクラスの比較
            pred
true         setosa versicolor virginica
  setosa         29          0         0
  versicolor      0         21         1
  virginica       0          0        24
> ### ほぼ完全に判別できている (蕚片による分類よりよい)
> pred2 <- predict(mod2, newdata = iris.test) # テストデータに対する予測結果
> head(pred2$class) # 予測されたクラス (最初の 6 個)
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
> head(iris.test$Species) # 実際のクラス (最初の 6 個)
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
> table(true = iris.test$Species, pred = pred2$class) # 真のクラスと予測されたクラスの比較
            pred
true         setosa versicolor virginica
  setosa         21          0         0
  versicolor      0         26         2
  virginica       0          2        24
> ### ほぼ完全に判別できている (蕚片による分類よりよい)
                                                                    (lda.r)
```

## 7.4. 2 次判別分析

**2 次判別分析** *(quadratic discriminant analysis)* では, $f_k(\boldsymbol{x})$ をクラスごとに異なる平均ベクトル $\boldsymbol{\mu}_k$ および共分散行列 $\Sigma_k$ をもつような $q$ 変量正規分布の確率密度関数としてモデル化する:

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{q/2}\sqrt{\det \Sigma_k}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right).$$

いま,

$$\begin{aligned} & p_k(\boldsymbol{x}) < p_l(\boldsymbol{x}) \\ \Leftrightarrow\ & f_k(\boldsymbol{x})\pi_k < f_l(\boldsymbol{x})\pi_l \\ \Leftrightarrow\ & \log f_k(\boldsymbol{x}) + \log \pi_k < \log f_l(\boldsymbol{x}) + \log \pi_l \\ \Leftrightarrow\ & -\frac{1}{2}\det \Sigma_k - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) + \log \pi_k \\ & < -\frac{1}{2}\det \Sigma_l - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_l)^\top \Sigma_l^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_l) + \log \pi_l \end{aligned}$$

が成り立つから, **2 次判別関数** *(quadratic discriminant function)*

$$\delta_k(\boldsymbol{x}) = -\frac{1}{2}\det \Sigma_k - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) + \log \pi_k$$

を最大化するようなクラス $k$ にデータを分類すればよい. 2 次判別関数の計算のためには各クラスごとの特徴量の平均ベクトル $\boldsymbol{\mu}_k$ および共分散行列 $\Sigma_k$ を計算する必要があるが, これらはそれぞれ

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i:y_i=k} \boldsymbol{x}_i, \qquad \hat{\Sigma} = \frac{1}{n_k - 1}\sum_{i:y_i=k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^\top$$

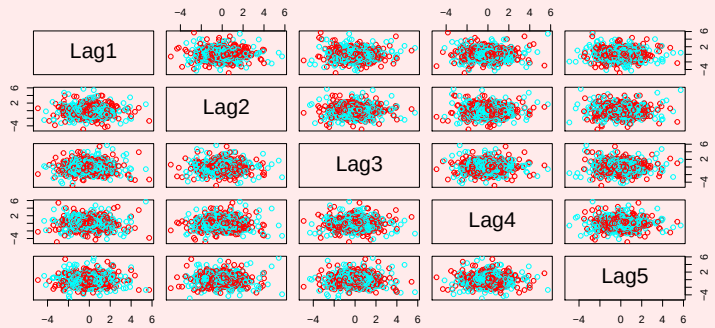
で推定すればよい. ここに, $n_k$ は $y_i = k$ であるようなデータの総数を表す.

**7.4.1. R での実行.** パッケージ `MASS` には 2 次判別分析を実行するための関数 `qda()` が用意されている. 書式は関数 `lda()`(従って関数 `lm()`) とほとんど同じである (クラスラベルを目的変数, 特徴量を説明変数とする).

```
> library(MASS) # パッケージのロード
> ## 人工データによる例 (2群の場合)
> set.seed(123)
> mu1 <- c(14,11)
> mu2 <- c(13,13)
> Sigma1 <- matrix(c(1,0.7,0.7,1),2,2)*2.5
> Sigma2 <- matrix(c(1,-0.3,-0.3,1),2,2)*0.5
> n <- 30
> x1 <- mvrnorm(n,mu=mu1,Sigma=Sigma1)
> x2 <- mvrnorm(n,mu=mu2,Sigma=Sigma2)
> X1 <- cbind(data.frame(x1),data.frame(cat=rep(0,n)))
> X2 <- cbind(data.frame(x2),data.frame(cat=rep(1,n)))
> X <- rbind(X1,X2)
> # plot(X[,1:2],pch=X[,3]+1,col=c("red","blue")[X$cat+1]) # データの分布
> # 分析の開始:
> (myqda1 <- qda(cat~X1+X2,X))# トレーニングデータで判別関数を作る
Call:
qda(cat ~ X1 + X2, data = X)

Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
        X1       X2
0 13.82213 11.04054
1 13.02535 13.05320
> # 新しいデータを判別する:
> n1 <- 25
> n2 <- 18
> x1new <- mvrnorm(n1,mu=mu1,Sigma=Sigma1)
> x2new <- mvrnorm(n2,mu=mu2,Sigma=Sigma2)
```

```
> X1new <- cbind(data.frame(x1new),data.frame(cat=rep(0,n1)))
> X2new <- cbind(data.frame(x2new),data.frame(cat=rep(1,n2)))
> Xnew <- rbind(X1new,X2new)
> mypredict1<-predict(myqda1, newdata = Xnew[,1:2]) # Xnew を判別
> table(true = Xnew$cat, pred = mypredict1$class) # 真のクラスと予測されたクラスの比較
    pred
true  0  1
   0 24  1
   1  0 18
> mypredict1$class # 予測を真の分類と比較：
 [1] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1
Levels: 0 1
> Xnew$cat
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1
> plot(Xnew[,1:2],pch=Xnew[,3]+1,col=c("red","blue")[Xnew$cat+1]) # データの分布
> ## 比較のため lda も実行
> (mylda1 <- lda(cat~X1+X2,X))
Call:
lda(cat ~ X1 + X2, data = X)

Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
        X1       X2
0 13.82213 11.04054
1 13.02535 13.05320

Coefficients of linear discriminants:
         LD1
X1 -0.832501
X2  1.125003
> mypredict1<-predict(mylda1, newdata = Xnew[,1:2]) # Xnew を判別
> table(true = Xnew$cat, pred = mypredict1$class) # 真のクラスと予測されたクラスの比較
    pred
true  0  1
   0 22  3
   1  0 18
> mypredict1$class # 予測を真の分類と比較：
 [1] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1
Levels: 0 1
> Xnew$cat
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1
> ## パッケージ ISLR のデータセット Smarket による例 (2 群の判別)
> ## S&P500 指数の 2001-2005 年の価格変動を記録したデータ
> ## ここでは価格の上昇・下降を直近何日かの収益率から予測するモデルを構築する
> # install.packages("ISLR") # パッケージのインストール (必要な場合)
> library(ISLR)
> head(Smarket)
  Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
1 2001  0.381 -0.192 -2.624 -1.055  5.010 1.1913  0.959        Up
2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032        Up
3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623      Down
```

```
4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614        Up
5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213        Up
6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392        Up
> pairs(subset(Smarket, select = paste("Lag", 1:5, sep = "")),
+       col = rainbow(2)[Smarket$Direction])
```




```
> Smarket.train <- subset(Smarket, Year < 2005) # 2004年までのデータを訓練データとする
> Smarket.test <- subset(Smarket, Year == 2005) # 2005年のデータをテストデータとする
> (mod.lda <- lda(Direction ~ Lag1 + Lag2, data = Smarket.train))
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket.train)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544

Coefficients of linear discriminants:
            LD1
Lag1 -0.6420190
Lag2 -0.5135293
> res.lda <- predict(mod.lda) # 訓練データに対する予測結果
> table(true = Smarket.train$Direction, pred = res.lda$class)
      pred
true   Down  Up
  Down  168 323
  Up    160 347
> (mod.qda <- qda(Direction ~ Lag1 + Lag2, data = Smarket.train))
Call:
qda(Direction ~ Lag1 + Lag2, data = Smarket.train)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544
> res.qda <- predict(mod.qda) # 訓練データに対する予測結果
> table(true = Smarket.train$Direction, pred = res.qda$class)
```

```
      pred
true   Down  Up
  Down  162 329
  Up    156 351
> res.lda <- predict(mod.lda, Smarket.test) # テストデータに対する予測結果
> table(true = Smarket.test$Direction, pred = res.lda$class)
      pred
true   Down  Up
  Down   35  76
  Up     35 106
> res.qda <- predict(mod.qda, Smarket.test) # テストデータに対する予測結果
> table(true = Smarket.test$Direction, pred = res.qda$class)
      pred
true   Down  Up
  Down   30  81
  Up     20 121
```

(qda.r)

## 7.5. 参考文献


1. T. Hastie, R. Tibshirani, J. Friedman 著「The Elements of Statistical Learning」, Springer (2009 年).
2. G. James, D. Witten, T. Hastie, R. Tibshirani 著「An Introduction to Statistical Learning」, Springer (2013 年).
3. 金明哲著「R によるデータサイエンス (第 2 版)」, 森北出版 (2017 年).
4. 東京大学教養学部統計学教室編「統計学入門」, 東京大学出版会 (1991 年).


## 7.6. 補足: $X$ が連続型の場合のベイズの公式の証明

$\varepsilon > 0$ を任意にとり,「$x_1 - \varepsilon \leq X_1 \leq x_1 + \varepsilon, \ldots, x_q - \varepsilon \leq X_q \leq x_q + \varepsilon$ が成り立つ」という事象を $A_\varepsilon$,「$Y = K$ が成り立つ」という事象を $B_k$ と書くことにする. 定義より,

$$P(Y = k|X = \boldsymbol{x}) = \lim_{\varepsilon \to 0} \frac{P(B_k \cap A_\varepsilon)}{P(A_\varepsilon)} \tag{7.5}$$

が成り立つ ($B_k \cap A_\varepsilon$ は事象 $B_k$ と $A_\varepsilon$ が同時に起こるという事象 (積事象) を表す). 一方で, $f_k(\boldsymbol{x})$ の定義より,

$$P(A_\varepsilon|B_k) = \int_{A_\varepsilon} f_k(\boldsymbol{y}) d\boldsymbol{y}$$

が成り立つから, 微分積分学の基本定理より

$$\lim_{\varepsilon \to 0} \frac{P(A_\varepsilon|B_k)}{(2\varepsilon)^q} = f_k(\boldsymbol{x})$$

が成り立つ. $P(A_\varepsilon|B_k) = P(A_\varepsilon \cap B_k)/P(Y = k)$ であったから, 両辺に $P(Y = k)$ をかけて

$$\lim_{\varepsilon \to 0} \frac{P(A_\varepsilon \cap B_k)}{(2\varepsilon)^q} = f_k(\boldsymbol{x}) P(Y = k) \tag{7.6}$$

を得る. いま, $Y$ は $1, \ldots, K$ のうちいずれか一つの値のみ取ることに注意すると,

$$P(A_\varepsilon) = \sum_{l=1}^{K} P(A_\varepsilon \cap B_l)$$

が成り立つ. 両辺を $(2\varepsilon)^q$ で割って極限をとると,

$$\lim_{\varepsilon\to 0}\frac{P(A_\varepsilon)}{(2\varepsilon)^q}=\sum_{l=1}^{K} f_l(\boldsymbol{x})P(Y=l) \tag{7.7}$$

を得る. (7.6)–(7.7) 式を (7.5) 式に代入して示すべき等式を得る.

## 7.7. 補足: 関数 `lda()` と正準判別分析との関連

本節では関数 `lda()` のアウトプットの 1 つである `scaling` の意味, およびその正準判別分析との関連について説明する.

まずいくつか記号を準備する. 群ごとの (標本) 平均ベクトル $\hat{\boldsymbol{\mu}}_k$ $(k=1,\dots,K)$ を横に並べて得られる行列を転置して得られる $(K,q)$ 行列を $M$ とする (関数 `lda()` で `mean` として出力される行列):

$$M=(\hat{\boldsymbol{\mu}}_1,\dots,\hat{\boldsymbol{\mu}}_K)^\top.$$

さらに,

$$\widetilde{M}=(\pi_1(\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}}),\dots,\pi_K(\hat{\boldsymbol{\mu}}_K-\hat{\boldsymbol{\mu}}))^\top$$

とおく. ただし, $\hat{\boldsymbol{\mu}}$ はクラスラベルを考慮せずに計算した特徴量の平均ベクトル

$$\hat{\boldsymbol{\mu}}=\sum_{k=1}^{K}\pi_k\hat{\boldsymbol{\mu}}_k$$

である. 関数 `lda()` のデフォルトでは $\pi_k=n_k/n$ $(k=1,\dots,K)$ であったので, その場合 $\hat{\boldsymbol{\mu}}$ は特徴量の観測データの (標本) 平均に一致する:

$$\hat{\boldsymbol{\mu}}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i.$$

行列 $\widetilde{M}$ の階数を $r$ とする. $\hat{\boldsymbol{\mu}}$ の定義から $\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}},\dots,\hat{\boldsymbol{\mu}}_K-\hat{\boldsymbol{\mu}}$ は一次従属なので, $r\le\min\{q,K-1\}$ となることに注意しておく. また, $B=\widetilde{M}^\top\widetilde{M}$ とおく. $B$ は群ごとの平均ベクトル $\hat{\boldsymbol{\mu}}_k$ $(k=1,\dots,K)$ の共分散行列に対応するため, **群間変動** *(between-class covariance)* と呼ばれる. 一方で, $\hat{\Sigma}$ は群ごとのデータ間の共分散行列に対応するので, **群内変動** *(within-class covariance)* と呼ばれる.

以上の準備の下で, `scaling` は $q\times r$ 行列 $S$ で次の 3 条件を満たすものとなる:

(i) $SS^\top\widetilde{M}=\hat{\Sigma}^{-1}\widetilde{M}$.
(ii) $S^\top\hat{\Sigma}S=E_r$.
(iii) $S$ の第 $k$ 列 $(k=1,\dots,r)$ は行列 $\hat{\Sigma}^{-1}B$ の $k$ 番目に大きい固有値に対する固有ベクトルである.[2]

$S$ の具体的な計算法を示す前に, 線形判別関数 $\delta_k(\boldsymbol{x})$ $(k=1,\dots,K)$ の比較は行列 $M,S$ および事前確率 $\pi_k$ $(k=1,\dots,K)$ (これらはすべて関数 `lda()` のアウトプットに含まれている) さえあれば実行できることに注意しておく. 実際, $\delta_k(\boldsymbol{x})$ は以下のように書き直せる:

$$\delta_k(\boldsymbol{x})=(\boldsymbol{x}-\hat{\boldsymbol{\mu}})^\top\hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})-\frac{1}{2}(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})^\top\hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})+\log\pi_k$$
$$+\boldsymbol{x}^\top\hat{\Sigma}^{-1}\hat{\boldsymbol{\mu}}-\frac{1}{2}\hat{\boldsymbol{\mu}}^\top\hat{\Sigma}^{-1}\hat{\boldsymbol{\mu}}.$$

上式 2 行目の項たちは $k$ に依存しないため, $\delta_k(\boldsymbol{x})$ の比較には無関係である. 一方で, 上式 1 行目の項は条件 (i) を用いて以下のように書き直せる:

$$(\boldsymbol{x}-\hat{\boldsymbol{\mu}})^\top SS^\top(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})-\frac{1}{2}(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})^\top SS^\top(\hat{\boldsymbol{\mu}}_k-\hat{\boldsymbol{\mu}})+\log\pi_k.$$

[2] $\hat{\Sigma}^{-1}B$ は非負定値対称行列 $\hat{\Sigma}^{-1/2}B\hat{\Sigma}^{-1/2}$ と共通の固有値をもつため, その固有値はすべて 0 以上の実数である. さらに, $\hat{\Sigma}^{-1}B$ の階数は $B$ の階数 $r$ と一致するため, ちょうど $r$ 個の正の固有値をもつ.

従って異なる $k$ に対する $\delta_k(\boldsymbol{x})$ の比較は $M, S, \pi_1, \ldots, \pi_K$ さえあれば実行できる.

次に, $S$ の具体的な計算方法を示す. そのためには, 以下で示す行列の特異値分解のコンパクト版が有用である:

**命題 7.1.** $A$ を $n \times q$ 行列とし, その階数を $r$ とする. このとき, $n \times q$ 行列 $U$, $q \times r$ 行列 $V$, $r$ 次対角行列 $D$ が存在して, $U^\top U = V^\top V = E_r$ を満たし, $D$ の対角成分は $A$ の正の特異値を降順に並べたものであり, かつ

$$A = UDV^\top$$

と書ける.

このとき, $S$ は以下の手順で計算できる:

(1) $n \times q$ 行列 $\widetilde{X} = \frac{1}{\sqrt{n-K}}(\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}_{y_1}, \ldots, \boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_{y_n})^\top$ の特異値分解 $\widetilde{X} = U_1 D_1 V_1^\top$ を計算する.[3]
(2) $K \times q$ 行列 $\widetilde{M} V_1 D_1^{-1}$ の階数が $r$ であることに注意して, $\widetilde{M} V_1 D_1^{-1}$ の「コンパクトな」特異値分解 $\widetilde{M} V_1 D_1^{-1} = U_2 D_2 V_2^\top$ を計算する.
(3) $S = V_1 D_1^{-1} V_2$ とおく.

**演習** 7.1. 上の手順で計算された行列 $S$ が実際に条件 (i)–(iii) を満たすことを確認せよ. さらに, 行列 $D_2$ の対角成分は $\hat{\Sigma}^{-1}B$ の正の固有値と一致することを確認せよ.

$\hat{\Sigma}$ の代わりに $S$ を計算しておくメリットの 1 つとして, $S$ の方が $\hat{\Sigma}$ よりサイズが小さいため計算効率の面で好ましい点がある. 別のメリットとして, $S$ が**正準判別分析** *(canonical discriminant analysis)* に応用できる点がある. 正準判別分析の目的は, 特徴量 $X_1, \ldots, X_q$ の 1 次結合 $a_1 X_1 + \cdots + a_q X_q$ で異なるクラス間の相違を最大限説明できるように係数ベクトル $\boldsymbol{a} = (a_1, \ldots, a_q)^\top$ を選ぶことである (対応する 1 次結合 $a_1 X_1 + \cdots + a_q X_q$ は**第 1 正準 (判別) 変数** *(first canonical (discriminant) variable)* と呼ばれる). そのためには, クラス間のばらつき具合を表す $\boldsymbol{a} \cdot \boldsymbol{\mu}_k$ $(k = 1, \ldots, K)$ の分散 $\boldsymbol{a}^\top B \boldsymbol{a}$ が大きく, クラス内のばらつきを表す $\boldsymbol{a} \cdot (\boldsymbol{x}_i - \boldsymbol{\mu}_{y_i})$ $(i = 1, \ldots, n)$ の分散 $\boldsymbol{a}^\top \hat{\Sigma} \boldsymbol{a}$ が小さければよい. 以上の考察より, 正準判別分析は $\boldsymbol{a}^\top B \boldsymbol{a}$ と $\boldsymbol{a}^\top \hat{\Sigma} \boldsymbol{a}$ の比

$$\frac{\boldsymbol{a}^\top B \boldsymbol{a}}{\boldsymbol{a}^\top \hat{\Sigma} \boldsymbol{a}}$$

を最大化するようなベクトル $\boldsymbol{a}$ を求める問題と定式化される. 尺度の不定性を取り除くために, 通常さらに $\boldsymbol{a}^\top \hat{\Sigma} \boldsymbol{a} = 1$ という制約を課す. このとき, Lagrange の乗数法によって, $\boldsymbol{a}$ は $\hat{\Sigma}^{-1}B$ の最大固有値に対応する固有ベクトルであることが確認できる. 従って, 条件 (ii)–(iii) より $S$ の第 1 列に対応するベクトルが求めるべきものである. 主成分分析の場合と同様の考え方で, 2 番目にクラス間の相違を説明できるような特徴量の 1 次結合 (**第 2 正準 (判別) 変数**) に対応する係数ベクトル $\boldsymbol{a}_2$, 3 番目にクラス間の相違を説明できるような特徴量の 1 次結合 (**第 3 正準 (判別) 変数**) に対応する係数ベクトル $\boldsymbol{a}_3, \ldots$ と拡張でき, $S$ の第 2 列, 第 3 列, ... がそれぞれ $\boldsymbol{a}_2, \boldsymbol{a}_3, \ldots$ に対応することが示せる. 構成法から, 異なるクラスに属するデータたちの正準変数たちは互いに大きく離れた値をとり, 同じクラスに属するデータたちの正準変数は近い値をとることが期待される. 従って, 正準変数をプロットすることで判別の様子を視覚化できる. R では, $r \geq 2$ の場合に関数 `lda()` のアウトプットに関数 `plot()` を適用することで, 第 1 正準変数と第 2 正準変数のバイプロットを描画する. また, 関数 `predict()` を適用した際, アウトプットのリストに `x` という名前で新規データに対する正準変数の計算結果を返す.[4]

正準判別分析は, 主成分分析と同様特徴量の次元縮約にも応用できる. 実際, 構成法から $\hat{\Sigma}^{-1}B$ の $k$ 番目に大きい固有値 $\lambda_k$ は第 $k$ 正準変数がクラス判別にどの程度有

[3] 行列 $\hat{\Sigma}$ の正則性の仮定の下で $\widetilde{X}$ の階数は $q$ となるため, ここでの特異値分解は 2 章の意味のものと上の意味のもので一致する.

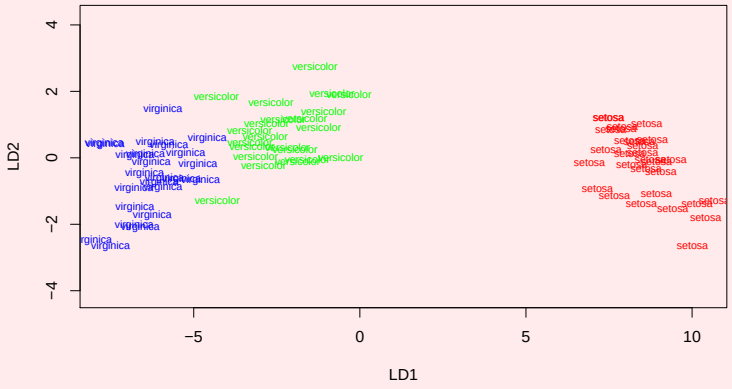
[4] R では, 主成分分析の場合と同様元のデータに対する正準変数たちが平均 0 となるように中心化されている (すなわち, 新規データ $\boldsymbol{x}$ に対して $(\boldsymbol{x} - \hat{\boldsymbol{\mu}})^\top S$ を計算する).

用かを表す指標だと考えられる. 主成分分析の場合にならって第 $k$ 正準変数の寄与率を $\lambda_k / \sum_{l=1}^{r} \lambda_l$ で定義する (関数 `lda()` をプリントした際に表示される “Proportion of trace” の欄で確認できる). 寄与率が 0 に近い正準変数は判別に不必要だと考えられるので, 寄与率が 0 に近い正準変数に対応する列を取り除いた, $S$ の最初の $m$ 列 $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m$ $(m \leq r)$ のみを残して判別を実行する方法が考えられる. この場合, 新規データ $\boldsymbol{x}$ が与えられたとき, 各クラス $k = 1, \ldots, K$ について, (正準変数に変換後の) 平均からの距離の二乗和

$$d_k = \sum_{j=1}^{m} (\boldsymbol{s}_j \cdot \boldsymbol{x} - \boldsymbol{s}_j \cdot \boldsymbol{\mu}_k)^2$$

を計算し, $d_k$ が最小となるようなクラス $k$ に $\boldsymbol{x}$ を分類すればよい.

```
> ## データセット iris による例
> library(MASS)
> n <- nrow(iris) # サンプル数
> ### データセットをランダムに 2 分割し, 一方を訓練データ,
> ### 他方をテストデータとする
> set.seed(123)
> idx <- sample.int(n, size = n/2)
> iris.train <- iris[idx, ] # 訓練データ
> iris.test <- iris[-idx, ] # テストデータ
> (res.lda <- lda(Species ~., data = iris.train))
Call:
lda(Species ~ ., data = iris.train)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3866667  0.2933333  0.3200000

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa         5.020690    3.482759     1.458621   0.2586207
versicolor     5.850000    2.709091     4.204545   1.3090909
virginica      6.558333    2.975000     5.433333   1.9291667

Coefficients of linear discriminants:
                    LD1        LD2
Sepal.Length  0.5045498  0.2587391
Sepal.Width   1.8254350 -2.5406901
Petal.Length -2.3722312  0.3895991
Petal.Width  -3.1321104 -2.1621864

Proportion of trace:
   LD1    LD2
0.9929 0.0071
> ### LD1 の寄与率が非常に高く, LD2 は判別にほとんど役立たないことが示唆される
> plot(res.lda, col = rainbow(3)[iris.train$Species])
> ### バイプロットからも LD1 の有効性と LD2 が判別に寄与していないことが確認できる
```




```
> # LD1 と LD2 の再現
> mu <- res.lda$means # クラスごとの平均ベクトル
> Mtilde <- sqrt(res.lda$prior) *
+   scale(mu, center = colSums(res.lda$prior %*% mu), scale = FALSE)
> Xtilde <-  (subset(iris.train, select = -Species) -
+                mu[iris.train$Species, ])/sqrt(n/2 - 3)
> s1 <- svd(Xtilde)
> s2 <- svd(Mtilde %*% s1$v %*% diag(1/s1$d))
> s1$v %*% diag(1/s1$d) %*% s2$v[ ,1:2]
           [,1]       [,2]
[1,]  0.5045498  0.2587391
[2,]  1.8254350 -2.5406901
[3,] -2.3722312  0.3895991
[4,] -3.1321104 -2.1621864
> res.lda$scaling
                    LD1        LD2
Sepal.Length  0.5045498  0.2587391
Sepal.Width   1.8254350 -2.5406901
Petal.Length -2.3722312  0.3895991
Petal.Width  -3.1321104 -2.1621864
> # 上の考察に基づき, LD1 のみによる正準判別分析を実行してみる
> mycda <- function(obj, newdata){
+
+   ybar <- drop(obj$means %*% obj$scaling[ ,1])
+   y <- predict(obj, newdata)$x[ ,1] + c(obj$prior %*% ybar)
+   d <- outer(y, ybar, FUN = "-")^2
+   # 別の書き方
+   # d <- matrix(0, length(y), length(ybar))
+   # for(k in 1:length(ybar)) d[ ,k] <- (y - ybar[k])^2
+   res <- apply(d, 1, "which.min")
+
+   return(rownames(obj$means)[res])
+ }
> # 訓練データの予測
> lda.train <- predict(res.lda)$class
> cda.train <- mycda(res.lda)
> table(true = iris.train$Species, pred = lda.train) # LDA の場合
            pred
true         setosa versicolor virginica
  setosa         29          0         0
  versicolor      0         21         1
  virginica       0          0        24
```

```
> table(true = iris.train$Species, pred = cda.train) # CDA の場合
            pred
true         setosa versicolor virginica
  setosa         29          0         0
  versicolor      0         20         2
  virginica       0          0        24
> ### 後者のパフォーマンスは前者と比べて遜色ない
> # テストデータの予測
> lda.test <- predict(res.lda, iris.test)$class
> cda.test <- mycda(res.lda, iris.test)
> table(true = iris.test$Species, pred = lda.test) # LDA の場合
            pred
true         setosa versicolor virginica
  setosa         21          0         0
  versicolor      0         25         3
  virginica       0          0        26
> table(true = iris.test$Species, pred = cda.test) # CDA の場合
            pred
true         setosa versicolor virginica
  setosa         21          0         0
  versicolor      0         25         3
  virginica       0          0        26
> ### 後者のパフォーマンスは前者と比べて遜色ない
```

(cda.r)

### 7.8. 補足: 関数 qda() の scaling の意味

関数 qda() のアウトプットの 1 つである scaling は, $(q, q, K)$ 次元の array であり, 各クラス $k = 1, \ldots, K$ について

$$\hat{\Sigma}_k^{-1} = S_k S_k^\top \tag{7.8}$$

を満たす $q$ 次上三角行列 $S_k$ が scaling[ , ,$k$] に格納されている.

(7.8) を満たす上三角行列 $S_k$ は以下のようにして計算できる. $\hat{\Sigma}_k$ は正定値対称行列なので, ある正則な $q$ 次上三角行列 $R_k$ が存在して

$$\hat{\Sigma}_k = R_k^\top R_k$$

を満たす (これを $\hat{\Sigma}_k$ の **Choleski 分解**と呼ぶ. 関数 chol() で計算できる). このとき,

$$\hat{\Sigma}_k^{-1} = R_k^{-1}(R_k^\top)^{-1}$$

となり, $R_k^{-1}$ は上三角行列となるから, $S_k = R_k^{-1}$ とおけばよい.[5]

[5] R の関数 qda() では, 実際には Choleski 分解は使わずに, 説明変数のデザイン行列を中心化した行列に対する **QR 分解** *(QR decomposition)* を使って計算している. おそらく数値計算の安定化のためと思われる.