STAT 3690 Homework 4

zhiyanggeezhou.github.io

Zhiyang Zhou (zhiyang.zhou@umanitoba.ca)

Due at Apr 25 11:59 pm (Central Time)

Answers must be submitted electronically via Crowdmark. Please enclose your R code trunks (if applicable) as well.

1. We have information on $n = 138$ samples of Canadian hard red spring wheat and the flour made from these samples. The 5-dimensional vector $\mathbf{X}$ contains **standardized** wheat measurements on: (X_1) kernel texture, (X_2) test weight, (X_3) famaged kernels, (X_4) foreign material and (X_5) crude protein in the wheat. The 4-dimensional vector $\mathbf{Y}$ contains **standardized** flour measurements: (Y_1) wheat per barrel of flour, (Y_2) ash in flour, (Y_3) crude protein in flour, and (Y_4) gluten quality index. We are only given the sample correlation matrices:

$$R_X = \begin{pmatrix} 1.000 & 0.754 & -0.690 & -0.446 & 0.692 \\ & 1.000 & -0.712 & -0.515 & 0.412 \\ & & 1.000 & 0.323 & -0.444 \\ & & & 1.000 & -0.334 \\ & & & & 1.000 \end{pmatrix},$$

$$R_Y = \begin{pmatrix} 1.000 & 0.251 & -0.490 & 0.250 \\ & 1.000 & -0.434 & -0.079 \\ & & 1.000 & -0.163 \\ & & & 1.000 \end{pmatrix},$$

$$R_{XY} = \begin{pmatrix} -0.605 & -0.479 & 0.780 & -0.152 \\ -0.722 & -0.419 & 0.542 & -0.102 \\ 0.737 & 0.361 & -0.546 & 0.172 \\ 0.527 & 0.461 & -0.393 & -0.019 \\ -0.383 & -0.505 & 0.737 & -0.148 \end{pmatrix}.$$

   a. Use sequential tests (with the Holm-Bonferroni procedure) to determine the number of significant canonical correlations at level $\alpha = .05$.
   b. Compute sample canonical directions corresponding to the significant canonical correlations.

Answer to Q1a. The first two.

Answer to Q1b. Let $(\boldsymbol{a}_k, \boldsymbol{b}_k)$, $k = 1, 2$, be the first two (pairs of) sample canonical directions. Then

```
# [a_1, a_2] =
A
```

```
##           [,1]     [,2]
## [1,]  0.53501  1.01025
## [2,]  0.28768  0.02743
## [3,] -0.45690  0.97822
## [4,] -0.02504 -0.17963
```

```
# [b_1, b_2] =
B
```

```
##          [,1]    [,2]
## [1,] -0.2146  0.9232
## [2,] -0.1719 -0.5848
## [3,]  0.3297  0.6526
## [4,]  0.2638  0.3415
## [5,] -0.2976  0.5508
```

2. Consider the situation where you have two normal populations $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$. We observe independent samples $X_{1,1}, \ldots, X_{1,n_1} \sim N(\mu_1, \sigma_1^2)$ and $X_{2,1}, \ldots, X_{2,n_2} \sim N(\mu_2, \sigma_2^2)$ with means $\bar{X}_1$ and $\bar{X}_2$, respectively. We plan to use the following rule

   R : Classify a new X as coming from population 2 if $X > (\bar{X}_1 + \bar{X}_2)/2$ and population 1 otherwise.

   Assuming priors $\Pr(X \sim N(\mu_1, \sigma_1^2)) = \Pr(X \sim N(\mu_2, \sigma_2^2)) = 1/2$, please express the misclassification rate of rule R, i.e.,

   $$\text{err}(X) = \Pr(X > (\bar{X}_1 + \bar{X}_2)/2 \text{ and } X \sim N(\mu_1, \sigma_1^2)) + \Pr(X \leq (\bar{X}_1 + \bar{X}_2)/2 \text{ and } X \sim N(\mu_2, \sigma_2^2)),$$

   in terms of n_1, n_2, μ_1, μ_2, σ_1, σ_2 and the standard normal cumulative distribution function $\Phi(\cdot)$.

Answer to Q2. If $X \sim N(\mu_1, \sigma_1^2)$, the independence of X, $\bar{X}_1$ and $\bar{X}_2$ implies that $X - (\bar{X}_1 + \bar{X}_2)/2 \sim N((\mu_1 - \mu_2)/2, \sigma_1^2 + \sigma_1^2/(4n_1) + \sigma_2^2/(4n_2))$. That is,

$$\begin{aligned}
&\Pr\left(X > \frac{\bar{X}_1 + \bar{X}_2}{2} \mid X \sim N(\mu_1, \sigma_1^2)\right) \\
=&\Pr\left(\frac{X - (\bar{X}_1 + \bar{X}_2)/2 - (\mu_1 - \mu_2)/2}{\sqrt{\sigma_1^2 + \sigma_1^2/(4n_1) + \sigma_2^2/(4n_2)}} > \frac{-(\mu_1 - \mu_2)/2}{\sqrt{\sigma_1^2 + \sigma_1^2/(4n_1) + \sigma_2^2/(4n_2)}} \mid X \sim N(\mu_1, \sigma_1^2)\right) \\
=&\Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{(4 + n_1^{-1})\sigma_1^2 + n_2^{-1}\sigma_2^2}}\right).
\end{aligned}$$

Analogously,

$$\begin{aligned}
&\Pr\left(X \leq \frac{\bar{X}_1 + \bar{X}_2}{2} \mid X \sim N(\mu_2, \sigma_2^2)\right) \\
=&\Pr\left(\frac{X - (\bar{X}_1 + \bar{X}_2)/2 - (\mu_2 - \mu_1)/2}{\sqrt{\sigma_2^2 + \sigma_1^2/(4n_1) + \sigma_2^2/(4n_2)}} \leq \frac{-(\mu_2 - \mu_1)/2}{\sqrt{\sigma_2^2 + \sigma_1^2/(4n_1) + \sigma_2^2/(4n_2)}} \mid X \sim N(\mu_2, \sigma_2^2)\right) \\
=&\Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{(4 + n_1^{-1})\sigma_2^2 + n_2^{-1}\sigma_1^2}}\right).
\end{aligned}$$

So,

$$\text{err}(X) = \frac{1}{2}\Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{(4 + n_1^{-1})\sigma_1^2 + n_2^{-1}\sigma_2^2}}\right) + \frac{1}{2}\Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{(4 + n_1^{-1})\sigma_2^2 + n_2^{-1}\sigma_1^2}}\right).$$

3. Suppose there is a binary classification task: one would like to predict labels of n subjects, say $Y_1, \ldots, Y_n$, according to their independent p-dimensional observations $\mathbf{X}_1, \ldots, \mathbf{X}_n$. The two potential populations are assumed to be $MVN_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $MVN_p(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, i.e., $\mathbf{X}_i \mid Y_i = y_i \sim MVN_p(\boldsymbol{\mu}_{y_i}, \boldsymbol{\Sigma})$, $y_i = 1, 2$. Meanwhile, let $\Pr(Y_i = k) = \pi_k$ for all $k = 1, 2$ and $i = 1, \ldots, n$.
   a. Applying the linear discriminant analysis (LDA) to this problem, write down the mathematical expression of error rate in terms of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}$, π_1, π_2 and the standard normal cumulative distribution function $\Phi(\cdot)$.

b. There is a banknote authentication dataset (see below for the data import), where $n = 1,372$ data points consisted of features extracted (via the wavelet transformation) from images that were taken from genuine and forged banknotes. Specifically, the features are "variance" (the variance of wavelet-transformed image), "skewness" (the skewness of wavelet-transformed image), "curtosis" (the curtosis of wavelet-transformed image), and "entropy" (the entropy of image), all continuous. The authentication of banknote is indicated by "class" (0 for authentic and 1 for inauthentic). Figure out a parametric estimate for error rate of LDA by plugging estimates of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}$, π_1 and π_2 into the expression obtained in Q3a.

c. Apply LDA to the dataset in Q3b and utilize 5×8-fold cross validation to estimate the resulting error rate. Report this error rate.

d. Make a comment with one single sentence after comparing estimates given by Q3b and Q3c.

```
bn_df = read.table(
  "https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt",
  sep = ","
)
names(bn_df) = c("variance", "skewness", "curtosis", "entropy", "class")
```

Answer to Q3a.

$$\begin{aligned}
\text{error rate} &= \sum_{k=1}^{2} \Pr(\mathbf{X}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln \pi_k > \mathbf{X}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{3-k} - \frac{1}{2}\boldsymbol{\mu}_{3-k}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{3-k} + \ln \pi_{3-k} \text{ and } \mathbf{X} \sim MVN_p(\boldsymbol{\mu}_{3-k}, \boldsymbol{\Sigma}) \\
&= \sum_{k=1}^{2} \pi_{3-k} \Pr(\mathbf{X}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k}) - \frac{1}{2}(\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{3-k}) + \ln(\pi_k/\pi_{3-k}) > 0 \mid \mathbf{X} \sim MVN_p(\boldsymbol{\mu}_{3-k}, \boldsymbol{\Sigma})) \\
&= \sum_{k=1}^{2} \pi_{3-k} \Phi\left(\frac{\ln(\pi_k/\pi_{3-k}) - (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k})/2}{\sqrt{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{3-k})}} \right)
\end{aligned}$$

Answer to Q3b. Around 0.5%.

Answer to Q3c. Around 2.4%.

Answer to Q3d. Compared with the cross validation, the parametric estimation tends to underestimate the misclassification rate.

Appendix

```
options(digits = 4)
## Q1a
alpha = .05
p = 4
q = 5
n = 138
R_X <- matrix(
  c(1, 0.754, -0.690, -0.446, 0.692,
    0.754, 1, -0.712, -0.515, 0.412,
    -0.690, -0.712, 1, 0.323, -0.444,
    -0.446, -0.515, 0.323, 1, -0.334,
    0.692, 0.412, -0.444, -0.334, 1),
  ncol = 5)
R_Y <- matrix(
  c(1, 0.251, -0.490, 0.250,
    0.251, 1, -0.434, -0.079,
    -0.490, -0.434, 1, -0.163,
    0.250, -0.079, -0.163, 1),
```

```
  ncol = 4)
R_XY <- matrix(
  c(-0.605, -0.479, 0.780, -0.152,
    -0.722, -0.419, 0.542, -0.102,
    0.737, 0.361, -0.546, 0.172,
    0.527, 0.461, -0.393, -0.019,
    -0.383, -0.505, 0.737, -0.148),
  nrow = 5, ncol = 4, byrow = T)
R_X_sqrt = expm::sqrtm(R_X)
R_Y_sqrt = expm::sqrtm(R_Y)
M = solve(R_Y_sqrt) %*% t(R_XY) %*% solve(R_X_sqrt)
decomp = svd(M)
rhos = decomp$d
test.stats = rev(-n*cumsum(rev(log(1-rhos^2))))
pvals = numeric(length(test.stats))
for (k in 1:length(test.stats)){
  pvals[k] = 1-pchisq(test.stats[k], df=(p-k+1)*(q-k+1))
}
pvals
sort(pvals) < alpha/(p+1-(1:p))

# Q1b
## sample canonical directions corresponding to the first two canonical correlations
(A = solve(R_Y_sqrt) %*% decomp$u[,1:2])
(B = solve(R_X_sqrt) %*% decomp$v[,1:2])

# Q3b
bn_df = read.table(
  "https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt",
  sep = ","
)
names(bn_df) = c("variance", "skewness", "curtosis", "entropy", "class")
X = bn_df[, !(names(bn_df) %in% c("class"))]
Y = bn_df[, names(bn_df) %in% c("class")]
labels = unique(Y)
K = length(labels)
p = ncol(X)
n = nrow(X)
nks = numeric(K)
piks = numeric(K)
Muks = matrix(0, nrow = K, ncol = p)
Sigmaks = list()
for (k in 1:K){
  X_k = X[Y == labels[k],]
  nks[k] = nrow(X_k)
  piks[k] = nks[k]/n
  Muks[k,] = colMeans(X_k)
  Sigmaks[[k]] = cov(X_k)
  if (k==1){
    SigmaPool = Sigmaks[[k]] * (nks[k]-1)
  }else{
    SigmaPool = SigmaPool + Sigmaks[[k]] * (nks[k]-1)
  }
```

```
}
SigmaPool = SigmaPool/(n-1)
SigmaPoolInv = solve(SigmaPool)
err = 0
for (k in 1:K){
  quad = as.vector(t(Muks[k,]-Muks[3-k,])%*%SigmaPoolInv%*%(Muks[k,]-Muks[3-k,]))
  err = err + piks[3-k]*pnorm(
    (log(piks[k]/piks[3-k])-.5*quad)/sqrt(quad)
  )
}
err

# Q3c
set.seed(3690)
K = length(unique(bn_df$class))
M = 5; L = 8 # 5by8 fold CV
errLda = matrix(0, nrow = L, ncol = M)
for (l in 1:L){
  idx_new = sample(1:nrow(bn_df), size = nrow(bn_df))
  folds = cut(1:nrow(bn_df), breaks = M, labels=FALSE)
  for (m in 1:M){
    picked = idx_new[which(folds == m, arr.ind=TRUE)]
    train = bn_df[-picked,]
    Xtrain = train[, !(names(train) %in% c("class"))]
    Ytrain = train$class
    test = bn_df[picked,]
    Xtest = test[, !(names(test) %in% c("class"))]
    Ytest = test[, names(test) %in% c("class")]

    for (k in 1:K){
      objLda = MASS::lda(Xtrain, Ytrain, method = "moment")
    }
    errLda[l, m] = mean(Ytest != predict(objLda, Xtest)$class)
  }
}
(err = mean(errLda))
```