多元统计第一次作业

蒋文馨

2020/5/17

目录

1 PCA on Random Data

Generate a random sample of size n = 100 from a three-dimensional (r = 3) Gaussian distribution, where one of the variables has very high variance (relative to the other two). Carry out PCA on these data using the covariance matrix and the correlation matrix. In each case, find the eigenvalues and eigenvectors, draw the scree plot, compute the PC scores, and plot all pairwise PC scores in a matrix plot. Compare results.

```
library(MASS)
n = 100
sigma = matrix(1, 3, 3)
sigma = sigma + diag(c(2, 2, 100))
```

```
mu = c(1, 2, 3)
set.seed(0)# 设置种子, 使实验结果可重复
data = mvrnorm(n = n, mu = mu, Sigma = sigma)
data.mean = apply(data, 2, mean)
data.sigma = apply(data, 2, sd)
data.sc = apply(data, 2, scale)
```

1.1 Using Covariance Matrix

```
#cov
covm = cov(data)
covm_ = (t(data) %*% data - n * data.mean %*% t(data.mean)) / (n - 1)#=covm
eigv = eigen(covm)
eigv
```

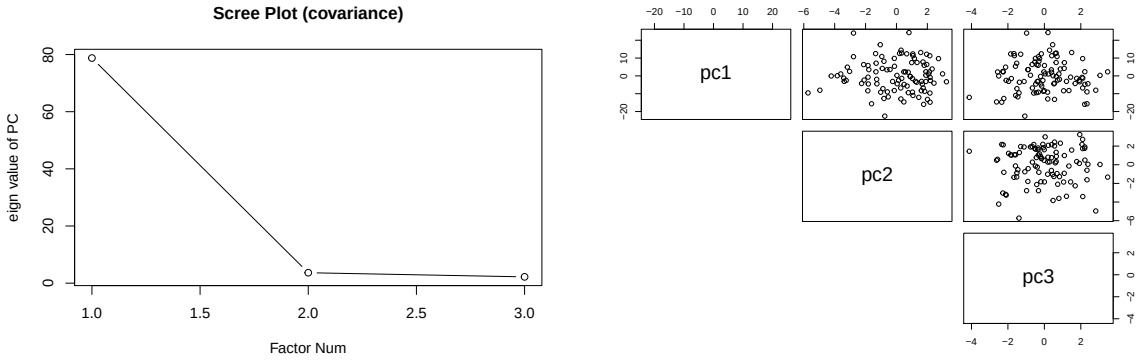
```
## eigen() decomposition
## $values
## [1] 78.774497  3.648625  2.202538
##
## $vectors
##             [,1]       [,2]        [,3]
## [1,] -0.00218733 0.73149218  0.68184632
## [2,] -0.01802631 0.68170831 -0.73140196
## [3,]  0.99983512 0.01389099 -0.01169498
```

```
cov.pcs = (data - matrix(1, n, 1) %*% data.mean) %*% eigv$vectors
head(cov.pcs)
```

```
##           [,1]       [,2]       [,3]
## [1,]  12.49003 -1.3495599 -1.6674500
## [2,]  -3.56246  1.3094367 -1.3696307
## [3,]  13.11774  1.5704388  1.4989423
## [4,]  12.56885  0.2000336  0.5973495
## [5,]   3.88655  2.3092195  0.8776273
## [6,] -15.63876 -1.6127618  2.3409119
```

```
plot(
  eigv$values,
  type = "b",
  xlab = "Factor Num",
  ylab = "eign value of PC",
```

```
  main = "Scree Plot (covariance)"
)
pairs(cov.pcs, labels = c("pc1","pc2","pc3"), lower.panel = NULL)
```



(a) 碎石图 (b) 主成分得分图

图 1: 使用协方差矩阵生成的碎石图和主成分得分图

1.2 Using Correlation Matrix

```
#cor
corm = cor(data)
corm_ = t(data.sc) %*% data.sc / (n - 1)#=corm
eigr = eigen(corm)
eigr
```

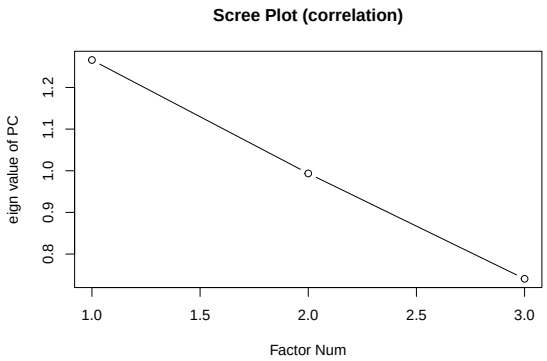
```
## eigen() decomposition
## $values
## [1] 1.2659036 0.9935529 0.7405435
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] -0.6610650 0.3435997  0.6670325
## [2,] -0.7024662 0.0290105 -0.7111256
## [3,]  0.2636935 0.9386680 -0.2221894
```

```
cor.pcs = data.sc %*% eigr$vectors
head(cor.pcs)
```

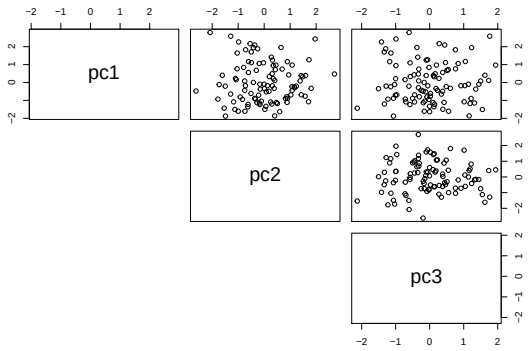
```
##            [,1]       [,2]       [,3]
## [1,]  1.1647491  0.8938086 -1.1755626
```

```
## [2,] -0.9250051 -0.3334491 -0.7173694
## [3,] -0.3227409  1.8099115  0.6091921
## [4,]  0.3891585  1.4246211  0.1090044
## [5,] -1.1128473  0.8819205  0.4231896
## [6,]  0.4048243 -1.6125567  1.6234230
```

```
plot(
  eigr$values,
  type = "b",
  xlab = "Factor Num",
  ylab = "eign value of PC",
  main = "Scree Plot (correlation)"
)
pairs(cor.pcs,labels = c("pc1","pc2","pc3"), lower.panel = NULL)
```



(a) 碎石图



(b) 主成分得分图

图 2: 使用相关系数阵生成的碎石图和主成分得分图

对比结果: 使用相关系数阵和协方差阵的结果有很大的不同。从碎石图看：使用协方差阵会使得第一主成分的方差与其余主成分方差之比很大；从得分图看：注意到使用协方差阵时生成的得分图中第一主成分的坐标轴，数据点在 pc1 轴上相当分散。如果设置横纵坐标轴等比例，两种方法绘制的得分图将有显著不同：协方差阵绘制的得分图数据点几乎分布在一条直线上。以上结果说明主成分分析对是否标准化敏感。因此，如果想要消除原始数据量纲的影响，在进行主成分分析之前，应将数据标准化。

2 Handwritten Digit

In the pen-based handwritten digit recognition (pendigits) example of Section 7.2.1, compute the variance of each of the 16 variables and show that they are very similar. Then, carry out PCA using the covariance matrix. How many PCs explain 80% and 90% of the total variation in the data? Display the first three PCs using pairwise scatterplots as in Figure 7.1. Do you see any differences between a covariance-based and a correlation-based PCA for this example?

```r
library(scatterplot3d)
library(RColorBrewer)
library(readr)
data = read_csv("pendigits.txt", col_names = FALSE)
n = dim(data)[2]
#data[,-1] is the num it belongs to
data.sc = apply(data[, 1:16], 2, scale)
data.mean = apply(data[, 1:16], 2, mean)
data.sigma = apply(data[, 1:16], 2, sd)
data.sigma^2 #variance of variables which show that they are very similar
```

```
##        X1        X2        X3        X4        X5        X6        X7        X8
## 1173.5957  263.0420  693.9528  367.2453 1162.8451  728.8211  934.9457  894.3750
##        X9       X10       X11       X12       X13       X14       X15       X16
## 1165.2344  742.6469 1390.4013  735.4934  498.8763 1099.2847 1743.9310 1280.0720
```

```r
# PCA
covm = cov(data[, 1:16])
eig = eigen(covm)
eig.cumpercent = cumsum(eig$values) / sum(eig$values)
paste(min(which(eig.cumpercent >= 0.8)),"PCs explain 80% of the total variation")
```

```
## [1] "5 PCs explain 80% of the total variation"
```

```r
paste(min(which(eig.cumpercent >= 0.9)),"PCs explain 90% of the total variation")
```

```
## [1] "7 PCs explain 90% of the total variation"
```

```r
pc.s = as.matrix(data[, 1:16] - matrix(1, n, 1) %*% data.mean) %*% eig$vectors
```

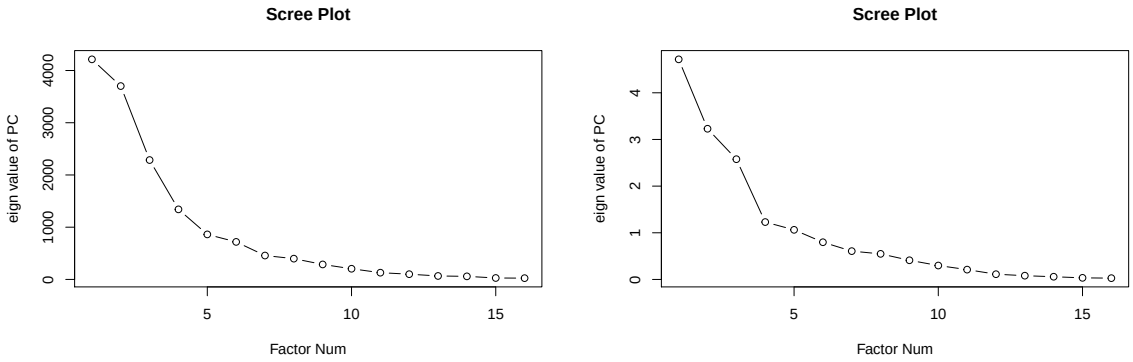
图 3 中使用两种方法生成的碎石图变化趋势相近（和第一题相比），由此也可知 16 个变量的方差相似。

和教材图 7.6 对比：注意到图 4 中的坐标轴。与进行标准化相比，前 3 个主成分的方差都相对更大。“数字” 的分布也有一些不同。但是，和习题 7-1 相比，由于本题原始数据各个特征的方差相差不多，原始数据量纲的影响不大，两种方法绘制的得分图基本相似。

注：7-4 绘图代码略。

3 PCA of Fisher's Iris Data

Carry out a PCA of Fisher's iris data. These data consist of 50 observations on each of three species of iris: Iris setosa, Iris versicolor, and Iris virginica. The four measured variables are sepal length, sepal width, petal length, and petal width. Ignore the species labels. Compute the PC scores and plot all pairwise sets of PC scores in a matrix plot. Explain your results, taking into consideration the species labels.



(a) 基于协方差阵的碎石图

(b) 基于相关系数阵的碎石图

图 3: 手写数字碎石图

3.1 不调用 PCA 函数

```
library(RColorBrewer)
data = iris
#pca using correlation matrix----
data.sc = apply(data[, 1:4], 2, scale)
corm = cor(data.sc)
eig = eigen(corm)
pcs = data.sc %*% eig$vectors
sqrt(eig$values)
```

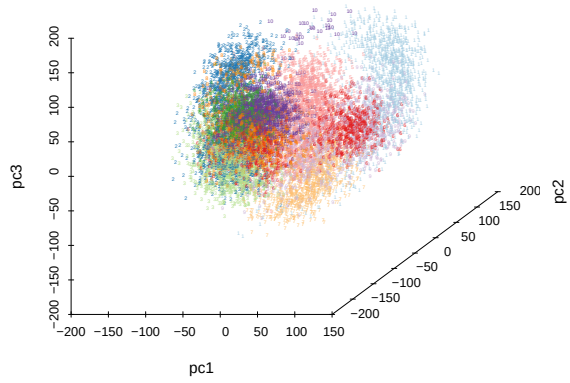
```
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
```

```
eig$vectors
```

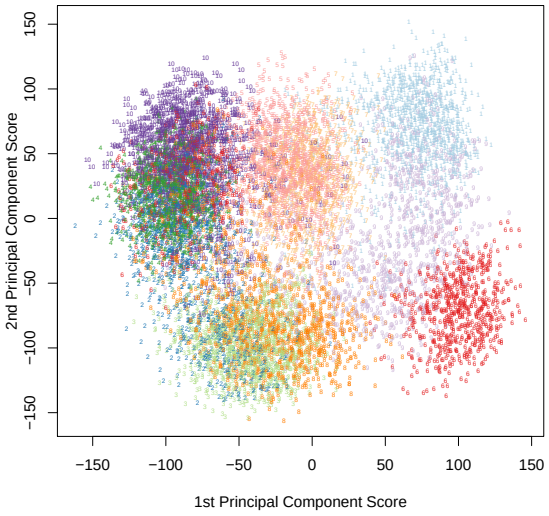
```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.5210659 -0.37741762  0.7195664  0.2612863
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
## [3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
## [4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

```
head(pcs)#score
```

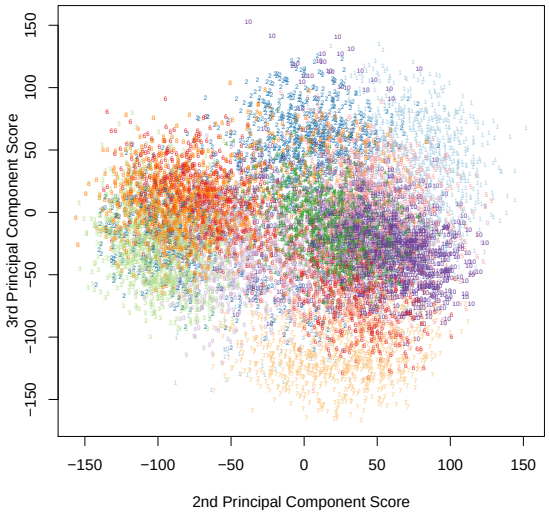
```
##           [,1]       [,2]        [,3]         [,4]
## [1,] -2.257141 -0.4784238  0.12727962  0.024087508
## [2,] -2.074013  0.6718827  0.23382552  0.102662845
## [3,] -2.356335  0.3407664 -0.04405390  0.028282305
## [4,] -2.291707  0.5953999 -0.09098530 -0.065735340
## [5,] -2.381863 -0.6446757 -0.01568565 -0.035802870
```



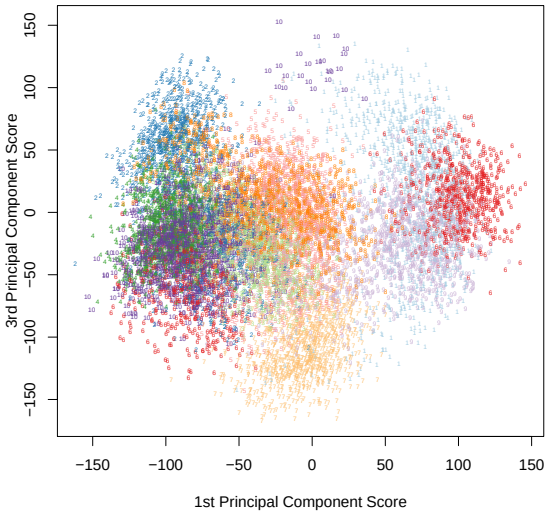
(a) 3d 图



(b) 1vs2



(c) 2vs3



(d) 1vs3

图 4: 手写数字主成分得分 (放大图片查看标签)

```
## [6,] -2.068701 -1.4842053 -0.02687825  0.006586116
```

```
# 使用/n来计算协方差
n = dim(iris)[1]
cormn = corm * (n - 1) / n
eig.n = eigen(cormn)
sqrt(eig.n$values)
```

```
## [1] 1.7026571 0.9528572 0.3818095 0.1434459
```

```
eig.n$vectors
```

```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.5210659 -0.37741762  0.7195664  0.2612863
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
## [3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
## [4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

3.2 使用 princomp 函数

```
#try to use princopm----
pca.pri.sc = princomp(data.sc)
head(pca.pri.sc$scores)# ==pcs
```

```
##         Comp.1     Comp.2      Comp.3       Comp.4
## [1,] -2.257141  0.4784238  0.12727962  0.024087508
## [2,] -2.074013 -0.6718827  0.23382552  0.102662845
## [3,] -2.356335 -0.3407664 -0.04405390  0.028282305
## [4,] -2.291707 -0.5953999 -0.09098530 -0.065735340
## [5,] -2.381863  0.6446757 -0.01568565 -0.035802870
## [6,] -2.068701  1.4842053 -0.02687825  0.006586116
```

```
summary(pca.pri.sc)# sdev!=sqrt(eig$values)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3      Comp.4
## Standard deviation     1.7026571 0.9528572 0.38180950 0.143445939
## Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
## Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000
```

```
pca.pri.sc$loadings#eign$vector
```

```
##
## Loadings:
##              Comp.1 Comp.2 Comp.3 Comp.4
## Sepal.Length  0.521  0.377  0.720  0.261
## Sepal.Width  -0.269  0.923 -0.244 -0.124
## Petal.Length  0.580         -0.142 -0.801
## Petal.Width   0.565         -0.634  0.524
##
##                Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

3.3 使用 prcomp 函数

```
#try to use prcomp----
pca.pr.sc = prcomp(data.sc)
pca.pr.sc$sdev # ==sqrt(eig$values)
```

```
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
```

```
head(pca.pr.sc$x) #score ==pcs
```

```
##            PC1        PC2         PC3          PC4
## [1,] -2.257141 -0.4784238  0.12727962  0.024087508
## [2,] -2.074013  0.6718827  0.23382552  0.102662845
## [3,] -2.356335  0.3407664 -0.04405390  0.028282305
## [4,] -2.291707  0.5953999 -0.09098530 -0.065735340
## [5,] -2.381863 -0.6446757 -0.01568565 -0.035802870
## [6,] -2.068701 -1.4842053 -0.02687825  0.006586116
```

```
head(pca.pr.sc$rotation)# eign$vector
```

```
##                    PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

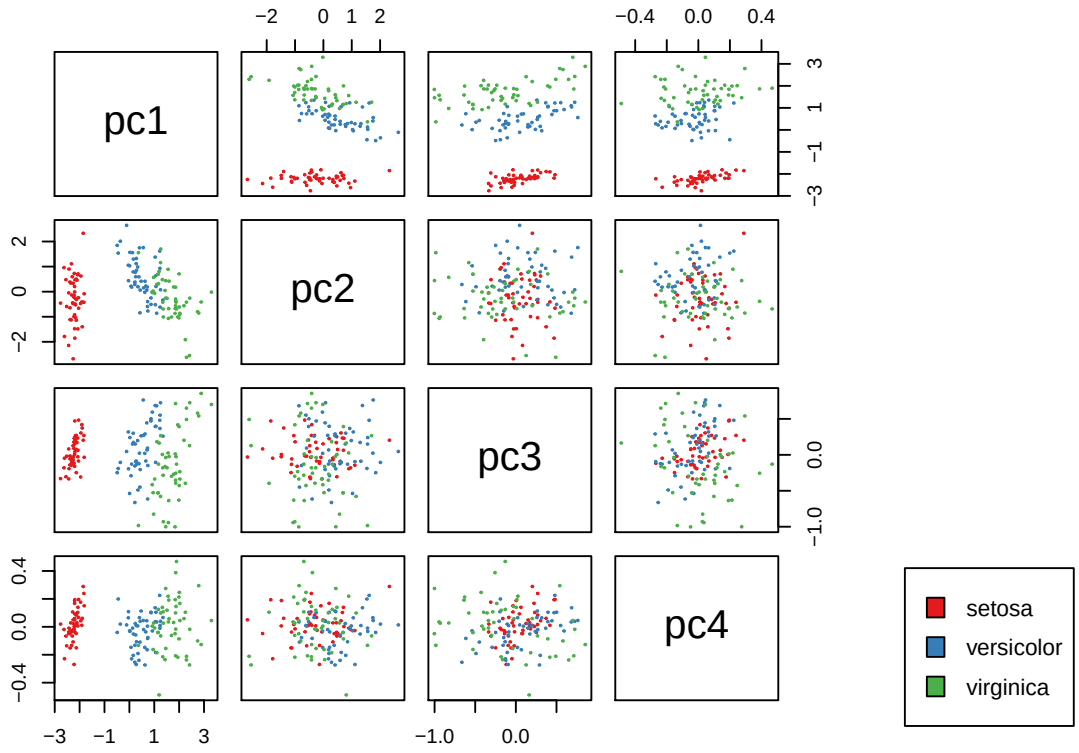


图 5: Iris 主成分得分图

```
summary(pca.pr.sc)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

以上一段代码说明：虽然 princomp 在计算协方差阵时 “/n” 而不是 “/(n-1)”，但是这对算出来的结果影响不大，特别是特征向量和主成分得分。

解释：根据 PCA 的结果，无论使用何种方法，第一主成分的方差在总体方差中的占比都很大。从得分图可以看出，用第一主成分可以对不同品种的 iris 实现较好的分类。同时，结合图例得知，virginica 和 versicolor 特征较为相似，而二者与 setosa 都有较大区别，但 versicolor 较 virginica 与 setosa 更为相似。

注：7-9 绘图代码略。

4 CVA under equicorrelation model

Let $r = s = 2$ and suppose the equicorrelation model holds for $\mathbf{X}$ and $\mathbf{Y}$. Then , $\mathbf{\Sigma}_{XX} = \mathbf{\Sigma}_{YY} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ and $\mathbf{\Sigma}_{XY} = \begin{pmatrix} \rho & \rho \\ \rho & \rho \end{pmatrix}$. Find the canonical correlations and the canonical

variates. Generalize your results to general r and s. Find the matrix $\mathbf{R}$ and the RRR solutions for $t = 1, 2$.

4.1 r = s = 2 :

1. 求 $\mathbf{\Sigma}_{XX}^{-1}$ 和 $\mathbf{\Sigma}_{YY}^{-1}$:

$$\mathbf{\Sigma}_{XX}^{-1} = \mathbf{\Sigma}_{YY}^{-1} = \frac{1}{1-\rho^2}\begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}.$$

2. 求 $\mathbf{\Sigma}_{XX}^{-1/2}$ 和 $\mathbf{\Sigma}_{YY}^{-1/2}$:

以求 $\mathbf{\Sigma}_{XX}^{-1/2}$ 为例, 首先需要利用特征值分解实现 $\mathbf{\Sigma}_{XX} = \mathbf{PDP'}$, 其中 $\mathbf{P}$ 为正交矩阵, $\mathbf{D}$ 为对角阵.

为此, 先求 $\mathbf{\Sigma}_{XX}$ 的特征值和单位特征向量:

$$|\mathbf{\Sigma}_{XX} - \lambda\mathbf{I}| = \begin{vmatrix} 1-\lambda & \rho \\ \rho & 1-\lambda \end{vmatrix} = \lambda^2 - 2\lambda + (1-\rho^2) =: 0 \quad \Rightarrow \lambda = 1 \pm \rho.$$

故,

$$\mathbf{P} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1-\rho & 0 \\ 0 & 1+\rho \end{bmatrix}.$$

所以,

$$\mathbf{\Sigma}_{XX}^{-1/2} = \mathbf{PD}^{-1/2}\mathbf{P'} = \mathbf{P}\begin{bmatrix} \frac{1}{\sqrt{1-\rho}} & 0 \\ 0 & \frac{1}{\sqrt{1+\rho}} \end{bmatrix}\mathbf{P'} = \frac{1}{2}\begin{bmatrix} \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix}.$$

其中,

$$\sigma_1 = \frac{1}{\sqrt{1+\rho}} + \frac{1}{\sqrt{1-\rho}}, \quad \sigma_2 = -\frac{1}{\sqrt{1+\rho}} + \frac{1}{\sqrt{1-\rho}}.$$

同理可求得

$$\mathbf{\Sigma}_{YY}^{-1/2} = \frac{1}{2}\begin{bmatrix} \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix}.$$

3. 计算 $\mathbf{R}$ 和 $\mathbf{R}^*$:

$$\mathbf{R} = \mathbf{\Sigma}_{YY}^{-1/2}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1/2} = \sigma_3\mathbf{J}_2$$

其中,

$$\sigma_3 = \frac{2\rho^2}{(\rho+1)^2}, \quad \mathbf{J_i} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}_{i\times i}.$$

同理可得

$$\mathbf{R}^* = \mathbf{\Sigma}_{XX}^{-1/2}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1/2} = \mathbf{R} = \sigma_3\mathbf{J}_2$$

4. 求典型相关系数和典型变量:

为此, 先求 $\mathbf{R}$ 的特征值和单位特征向量:

可得:

$$\mathbf{P} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \left(\frac{2\rho}{\rho+1}\right)^2 & 0 \\ 0 & 0 \end{bmatrix}.$$

所以典型相关系数为 $\frac{2\rho}{\rho+1}, 0$.

第 j 对典型相关变量由下式给出:

$$\xi_j = \mathbf{g}_j^\tau \mathbf{X}, \quad \omega_j = \mathbf{h}_j^\tau \mathbf{Y},$$

其中

$$\mathbf{g}_j = \lambda_j^{-1} \mathbf{\Sigma}_{XX}^{-1} \mathbf{\Sigma}_{XY} \mathbf{\Sigma}_{YY}^{-1/2} \mathbf{v}_j = \mathbf{\Sigma}_{XX}^{-1/2} \mathbf{u}_j$$
$$\mathbf{h}_j = \mathbf{\Sigma}_{YY}^{-1/2} \mathbf{v}_j$$

由此可得,

$$\mathbf{g_1'} = \mathbf{h_1'} = \left(\frac{1}{\sqrt{2(\rho+1)}}, \frac{1}{\sqrt{2(\rho+1)}} \right), \quad \mathbf{g_2'} = \mathbf{h_2'} = \left(-\frac{1}{\sqrt{2(1-\rho)}}, \frac{1}{\sqrt{2(1-\rho)}} \right).$$

4.2 r, s 未知:

对于未指定大小的 r 和 s, 进行同样的处理:

1. 求 $\mathbf{\Sigma}_{XX}^{-1}$ 和 $\mathbf{\Sigma}_{YY}^{-1}$:

以 $\mathbf{\Sigma}_{XX}^{-1}$ 为例:

考虑到 $\mathbf{\Sigma}_{XX} = \rho\mathbf{J} + (1-\rho)\mathbf{I}$, 假设 $\mathbf{\Sigma}_{XX}^{-1} = a\mathbf{I} + b\mathbf{J}$.$a, b$ 为与 ρ 相关的常数.

故

$$\mathbf{\Sigma}_{XX}^{-1} \mathbf{\Sigma}_{XX} = (a\mathbf{I} + b\mathbf{J}) (\rho\mathbf{J} + (1-\rho)\mathbf{I}) = a\rho\mathbf{J} + a(1-\rho)\mathbf{I} + br\rho\mathbf{J} + b(1-\rho)\mathbf{J} = \mathbf{I}$$

解得

$$\mathbf{\Sigma}_{XX}^{-1} = \frac{1}{1-\rho}\mathbf{I} + \frac{\rho}{(\rho-1)[(r-1)\rho+1]}\mathbf{J}.$$

同理

$$\mathbf{\Sigma}_{YY}^{-1} = \frac{1}{1-\rho}\mathbf{I} + \frac{\rho}{(\rho-1)([s-1)\rho+1]}\mathbf{J}.$$

2. 求 $\mathbf{\Sigma}_{XX}^{-1/2}$ 和 $\mathbf{\Sigma}_{YY}^{-1/2}$:

以 $\mathbf{\Sigma}_{XX}^{-1/2}$ 为例:

利用和上一步类似的思路, 假设 $\mathbf{\Sigma}_{XX}^{-1/2} = a\mathbf{I} + b\mathbf{J}$, 注意到 $\mathbf{\Sigma}_{XX}^{-1/2}$ 可能不唯一.

求出 $\mathbf{\Sigma}_{XX}^{-1/2}$ 的一个解为:

$$\frac{1}{\sqrt{1-\rho}}\mathbf{I} + \frac{1}{r}\left(\frac{1}{\sqrt{(r-1)\rho+1}} - \frac{1}{\sqrt{1-\rho}} \right)\mathbf{J}.$$

或者利用 $r = s = 2$ 时的特征值分解法, 也可以求出这个答案。

$$|\mathbf{\Sigma}_{XX} - \lambda\mathbf{I}| = \begin{vmatrix} 1-\lambda & & & & \\ & 1-\lambda & & \rho & \\ & & \ddots & & \\ & \rho & & & \\ & & & & 1-\lambda \end{vmatrix} = [1-\lambda+(r-1)\rho](1-\lambda-\rho)^{r-1}$$

在求行列式时，可以利用初等列变换把第二列到最后一列全部加到第一列上，再把系数 $1-\lambda+(r-1)\rho$ 提出来。之后利用初等行变换，把第二到最后一行全部减去第一行。再按第一列展开，可以得到上三角矩阵。

同理可得

$$\mathbf{\Sigma}_{YY}^{-1/2} = \frac{1}{\sqrt{1-\rho}}\mathbf{I} + \frac{1}{s}\left(\frac{1}{\sqrt{(s-1)\rho+1}} - \frac{1}{\sqrt{1-\rho}}\right)\mathbf{J}.$$

3. 计算 $\mathbf{R}$ 和 $\mathbf{R}^*$:

将以上结果带入 $\mathbf{R}$ 和 $\mathbf{R}^*$ 的计算公式, 得

$$\mathbf{R} = \frac{r\rho^2}{[(r-1)\rho+1][(s-1)\rho+1]}\mathbf{J_s}, \quad \mathbf{R}^* = \frac{s\rho^2}{[(r-1)\rho+1][(s-1)\rho+1]}\mathbf{J_r}.$$

4. 求典型相关系数和典型变量:

易得,

$$\rho_1 = \sqrt{\frac{rs\rho^2}{[(r-1)\rho+1][(s-1)\rho+1]}}, \quad \rho_2 = \rho_3 = \cdots = 0.$$

$$\mathbf{G} = \begin{bmatrix} \frac{1}{\sqrt{r}\sqrt{(r-1)\rho+1}} & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} & \cdots & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} \\ \frac{1}{\sqrt{r}\sqrt{(r-1)\rho+1}} & \frac{1}{\sqrt{2}\sqrt{1-\rho}} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ \frac{1}{\sqrt{r}\sqrt{(r-1)\rho+1}} & \vdots & \cdots & \frac{1}{\sqrt{2}\sqrt{1-\rho}} & 0 \\ \frac{1}{\sqrt{r}\sqrt{(r-1)\rho+1}} & 0 & \cdots & 0 & \frac{1}{\sqrt{2}\sqrt{1-\rho}} \end{bmatrix}_{r\times r},$$

$$\mathbf{H} = \begin{bmatrix} \frac{1}{\sqrt{s}\sqrt{(s-1)\rho+1}} & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} & \cdots & -\frac{1}{\sqrt{2}\sqrt{1-\rho}} \\ \frac{1}{\sqrt{s}\sqrt{(s-1)\rho+1}} & \frac{1}{\sqrt{2}\sqrt{1-\rho}} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ \frac{1}{\sqrt{s}\sqrt{(s-1)\rho+1}} & \vdots & \cdots & \frac{1}{\sqrt{2}\sqrt{1-\rho}} & 0 \\ \frac{1}{\sqrt{s}\sqrt{(s-1)\rho+1}} & 0 & \cdots & 0 & \frac{1}{\sqrt{2}\sqrt{1-\rho}} \end{bmatrix}_{s\times s}.$$

注: 考虑到本题的计算量，在计算 r、s 未知的情况时，我先用 MATLAB 符号运算对给定的 r、s 进行模拟。具体实现代码见附录。之后，根据模拟结果，补充计算过程。

5 Face Recognition Using Eigenfaces

使用 svd 复现教材上的例子。

```
library(bmp)
library(OpenImageR)
filename = list.files(pattern = "f*.bmp")
n = length(filename)
imag = read.bmp(filename[2])
d = dim(imag)
data = matrix(0, n, d[1] * d[2])
for (i in 1:n) {
  data[i, ] = t(as.vector(read.bmp(filename[i])[, , 1]))
```

```
}
data.scaled = apply(data, 2, scale)
data.mean = apply(data, 2, mean)
data.sigma = apply(data, 2, sd)
#数据中存在完全相同的列，scale函数会输出NaN
data.scaled[which(data.scaled == 'NaN')] = 0

#--------------------------
# 注意：关键步骤！数据量太大，计算cor会报错！
usv = svd(data.scaled)
eignv = (usv$d) ^ 2
pc = data.scaled %*% usv$v
#--------------------------
#或者直接调用prcomp函数可以得到一样的结果
#prcomp也是使用奇异值分解
pca = prcomp(data.scaled)
usv_v = pca$rotation
pc.score = pca$x
#--------------------------

#A test for recover to origin img
imgid = 4#the id of img which prepare to show
npc = 10# num of principle component
imgrecover = pc[imgid, 1:npc] %*% t(usv$v[, 1:npc]) * data.sigma + data.mean
imr2ims = array(imgrecover, d[1:2])
imageShow(imr2ims)

#save img
setwd("./img")
#save ghostlyplot
bmp(file="ghostly.bmp",width=d[2], height=d[1])
ghost = array(data.mean, d[1:2])
imageShow(ghost)
dev.off()
#save recimg with increased num of pc
imgid=2
for (npc in 1:9){
  fileName=paste("fr",npc,'.bmp',sep = "")
  bmp(file=fileName,width=d[2], height=d[1])
  imgrecover = pc[imgid, 1:npc] %*% t(usv$v[, 1:npc]) * data.sigma + data.mean
  imr2ims = array(imgrecover, d[1:2])
```

图 6: The cumulative effect of the nine principal components, adding one PC at a time, for eigenface 10 (“calm”). The calm face starts to appear by the fifth PC. The average eigenface is given in the left panel.

```
  imageShow(imr2ims)
  dev.off()
}
```

6 附录

6.1 7-12 使用的 MATLAB 代码

```
syms rho
assume((-1 < rho) & (rho < 1))
r = 3
s = 5
sxx = rho * ones(r) + diag(ones([1, r]) * (1 - rho))
syy = rho * ones(s) + diag(ones([1, s]) * (1 - rho))
sxy = rho * ones([r, s])
syx = rho * ones([s, r])

xx1 = sxx ^ (-1)
xx2 = simplify(sxx ^ (-0.5))
yy1 = simplify(syy ^ (-1))
yy2 = simplify(syy ^ (-0.5))
```

```
rstar = simplify(xx2 * sxy * yy1 * syx * xx2)
r = simplify(yy2 * syx * xx1 * sxy * yy2)

[u, lam] = eig(rstar)
[v, lam_] = eig(r)
u = normalize(double(u), "norm")
v = normalize(double(v), "norm")
g = simplify((u'*xx2)')
h = simplify((v'*yy2)')
```