STAT 3690 Lecture Note

Part VII: Principal component analysis

Zhiyang Zhou (zhiyang.zhou@umanitoba.ca, zhiyanggeezhou.github.io)

2023/Mar/22 12:31:21

Principal component analysis (PCA)

Population PCA

- Population PCA based upon covariance matrix $\boldsymbol{\Sigma}$
  - Random p-vector $\boldsymbol{X} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - Looking for (nonrandom) p-vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_p \in \mathbb{R}^p$ such that, given $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{j-1}$,

    $$\boldsymbol{w}_j^\top \boldsymbol{w}_j = 1 \quad \text{AND}$$

    $$\boldsymbol{X}^\top \boldsymbol{w}_j \text{ has the maximal variance and is uncorrelated with } \boldsymbol{X}^\top \boldsymbol{w}_1, \ldots, \boldsymbol{X}^\top \boldsymbol{w}_{j-1},$$

    i.e.,

    $$\boldsymbol{w}_1 = \arg\max_{\boldsymbol{w} \in \mathbb{R}^p} \operatorname{var}(\boldsymbol{X}^\top \boldsymbol{w}) \text{ subject to } \boldsymbol{w}_1^\top \boldsymbol{w}_1 = 1$$

    and, for $j \geq 2$,

    $$\boldsymbol{w}_j = \arg\max_{\boldsymbol{w} \in \mathbb{R}^p} \operatorname{var}(\boldsymbol{X}^\top \boldsymbol{w})$$

    $$\text{subject to} \qquad \boldsymbol{w}_j^\top \boldsymbol{w}_j = 1 \text{ and } \operatorname{cov}(\boldsymbol{X}^\top \boldsymbol{w}_j, \boldsymbol{X}^\top \boldsymbol{w}_{j'}) = 0 \text{ for } j' = 1, \ldots, j-1$$
  - (PCA Theorem) Let $\lambda_1 \geq \cdots \geq \lambda_p$ be eigenvalues of $\boldsymbol{\Sigma}$. Then the above $\boldsymbol{w}_j$ is the eigenvector corresponding to λ_j.
  - Vocabulary
    - $\boldsymbol{w}_j$: the jth vector of loadings
    - $Z_j = (\boldsymbol{X} - \boldsymbol{\mu})^\top \boldsymbol{w}_j \sim (0, \lambda_j)$: the jth principal component (PC) of $\boldsymbol{X}$
  - Representation of/approximation to $\boldsymbol{X}$ in terms of loadings and PCs

    $$\boldsymbol{X} = \boldsymbol{\mu} + \sum_{j=1}^{p} Z_j \boldsymbol{w}_j \approx \boldsymbol{\mu} + \sum_{j=1}^{s} Z_j \boldsymbol{w}_j$$
  - Identities
    - $\boldsymbol{w}_j^\top \boldsymbol{w}_{j'} = 1$ if $j = j'$ and 0 otherwise, i.e., $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_p\}$ is an orthogonal basis of $\mathbb{R}^p$
    - $\operatorname{cov}(Z_j, Z_{j'}) = \boldsymbol{w}_j^\top \boldsymbol{\Sigma} \boldsymbol{w}_{j'} = \lambda_j$ if $j = j'$ and 0 otherwise
    - $\sum_{j=1}^p \operatorname{var}(Z_j) = \sum_{j=1}^p \lambda_j = \operatorname{tr}(\boldsymbol{\Sigma}) = \sum_{j=1}^p \operatorname{var}(X_j)$
    - Z_j contributes $\lambda_j / \sum_{j=1}^p \lambda_j \times 100\%$ of the overall variance
      - Scree plot: displaying the amount of variation in each PC
      - Stopping rule (to determine s)

        $$s = \min\left\{k \in \mathbb{Z}^+ : \sum_{j=1}^{k} \lambda_j \Big/ \sum_{j=1}^{p} \lambda_j \geq 90\% \text{ (or another preset threshhold)}\right\}$$

- Population PCA based upon correlation matrix
  - (Pearson) correlation matrix

$$\mathbf{R} = [\text{corr}(X_i, X_j)]_{p\times p} = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_p}\right)\boldsymbol{\Sigma}\text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_p}\right)$$

    - $\sigma_j = \sqrt{\text{var}(X_j)}$, i.e., the root of the (j, j)-th entry of $\boldsymbol{\Sigma}$
  - Loadings and PCs from $\mathbf{R}$ NOT identical to those obtained from $\boldsymbol{\Sigma}$
    - Vectors of loadings $\boldsymbol{w}_j$: eigenvectors of $\mathbf{R}$
    - PCs $Z_j = \boldsymbol{X}_{\text{sd}}^\top \boldsymbol{w}_j$
      - $\boldsymbol{X}_{\text{sd}} = \text{diag}\left(\sigma_1^{-1}, \ldots, \sigma_p^{-1}\right)(\boldsymbol{X} - \boldsymbol{\mu}) = [(X_1 - \mu_1)/\sigma_1, \ldots, (X_p - \mu_p)/\sigma_p]^\top$
  - General advice: $\boldsymbol{\Sigma}$ is superior when entries of $\boldsymbol{X}$ are of the same units and comparable; otherwise $\mathbf{R}$ is preferred.
    - Using $\mathbf{R}$ rather than $\boldsymbol{\Sigma}$ $\Leftrightarrow$ standardizing entries of $\boldsymbol{X}$ before carrying out PCA
    - Without standardizing, the component with the "smallest" units (e.g., centimeter vs. meter) could be driving most of overall variance.

```r
options(digits = 2)
(Sigma <- matrix(
  c(10, 5, 1,
    5, 6, 5,
    1, 5, 8),
  ncol = 3))

# pca based upon covariance matrix
pca1 = eigen(Sigma, symmetric = T)
pca1$vectors # vectors of loadings
(variation1 = data.frame(
  idx = 1:length(pca1$values),
  var = pca1$values
))
plot(variation1, type='b') # scree plot
cumsum(pca1$values)/sum(pca1$values) # cummulative contribution of PCs

# pca based upon correlation matrix
pca2 = eigen(cov2cor(Sigma), symmetric = T)
pca2$vectors # vectors of loadings
(variation2 = data.frame(
  idx = 1:length(pca2$values),
  var = pca2$values
)); plot(variation2, type='b') # scree plot
cumsum(pca2$values)/sum(pca2$values) # cummulative contribution of PCs
```

Sample PCA

- $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]_{n\times p}^\top$
  - Assuming $\boldsymbol{X}_i \overset{\text{iid}}{\sim} (\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - $\boldsymbol{X}_i^\top$ is the ith row of $\boldsymbol{X}$
- Estimate the loadings $\boldsymbol{w}_j$ through the eigenvectors of sample covariance matrix $\mathbf{S}$ or sample correlation matrix $\hat{\mathbf{R}}$

- Score matrix of the first s PCs

$$\boldsymbol{Z} = [Z_{ij}]_{n\times s} = \boldsymbol{X}_{\mathrm{c}}\widehat{\mathbf{W}}$$

  - $\boldsymbol{X}_{\mathrm{c}} = [\boldsymbol{X}_1 - \bar{\boldsymbol{X}}, \ldots, \boldsymbol{X}_n - \bar{\boldsymbol{X}}]^\top_{n\times p}$: row-centered $\boldsymbol{X}$ (i.e. the sample mean has been subtracted from each row of $\boldsymbol{X}$)
    - $\bar{\boldsymbol{X}} = n^{-1}\sum_{i=1}^{n}\boldsymbol{X}_i$
  - $\widehat{\mathbf{W}} = [\hat{\boldsymbol{w}}_1, \ldots, \hat{\boldsymbol{w}}_s]_{p\times s}$: $\hat{\boldsymbol{w}}_j$ is the estimate of $\boldsymbol{w}_j$, i.e., eigenvectors of $\mathbf{S}$ or $\hat{\mathbf{R}}$
  - $Z_{ij} = (\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\top\hat{\boldsymbol{w}}_j$: the jth PC score for the ith observation

```
options(digits = 4)
set.seed(1)
Mu <- c(1, 2, 2)
(Sigma <- matrix(
  c(100000, 500, 100,
    500, 6, 5,
    100, 5, 8),
  ncol = 3))
n = 100
X = MASS::mvrnorm(n, Mu, Sigma)
axis_lim = range(X)
rgl::plot3d(X[,1], X[,2], X[,3], col = "red", size = 6,
            xlim = axis_lim, ylim = axis_lim, zlim = axis_lim)

# pca based upon sample covariance matrix
pca3 = eigen(cov(X), symmetric = T)
pca3$vectors # vectors of loadings
variation3 = data.frame(
  idx = 1:length(pca3$values),
  var = pca3$values
); plot(variation3, type='b') # scree plot
cumsum(pca3$values)/sum(pca3$values) # cummulative contribution of PCs
Z3 = scale(X, center = T, scale = F) %*% pca3$vectors # PC scores

pca4 = prcomp(X)
pca4$rotation # vectors of loadings
screeplot(pca4, type = 'l') # scree plot
cumsum((pca4$sdev)^2)/sum((pca4$sdev)^2) # cummulative contribution of PCs
Z4 = pca4$x # PC scores

# pca based upon sample correlation matrix
pca5 = eigen(cor(X), symmetric = T)
pca5$vectors # loadings
cumsum(pca5$values)/sum(pca5$values) # cummulative contribution of PCs
Z5 = scale(X, center = T, scale = F) %*% pca5$vectors # PC scores

pca6 = prcomp(X, scale. = T)
pca6$rotation
cumsum((pca6$sdev)^2)/sum((pca6$sdev)^2) # cummulative contribution of PCs
Z6 = pca6$x # PC scores

pca7 = prcomp(scale(X))
pca7$rotation
cumsum((pca7$sdev)^2)/sum((pca7$sdev)^2) # cummulative contribution of PCs
```

```
Z7 = pca7$x # PC scores

pca8 = prcomp(scale(X), scale. = T)
pca8$rotation
cumsum((pca8$sdev)^2)/sum((pca8$sdev)^2) # cummulative contribution of PCs
Z8 = pca8$x # PC scores
```

Geometric interpretation of (sample) PCA (optional)

- The definition of PCA as a linear combination that maximises variance is due to H. Hotelling (1933, Journal of Educational Psychology, 24, 417–441).
- PCA was introduced earlier by K. Pearson (1901, Philosophical Magazine, Series 6, 2(11), 559–572) to minimize the overall error in reconstructing data points

$$(\bar{\boldsymbol{X}}, \widehat{\mathbf{W}}, \boldsymbol{Z}_{i\cdot}) = \arg\min_{\boldsymbol{\theta},\mathbf{A},\mathbf{B}_i} \sum_{i=1}^{n} (\boldsymbol{X}_i - \boldsymbol{\theta} - \mathbf{A}\mathbf{B}_i)^\top (\boldsymbol{X}_i - \boldsymbol{\theta} - \mathbf{A}\mathbf{B}_i)$$

  - $\boldsymbol{Z}_{i\cdot} = [Z_{i1}, \ldots, Z_{is}]$: the ith row of score matrix $\boldsymbol{Z}$

Application of (sample) PCA

- Image compression: `mnist` is a list with two components: `train` and `test`. Each of these is a list with two components: images and labels.
  - The `images` component is a matrix with each row for one image consisting of $28 \times 28 = 784$ entries (pixels). Their value are integers between 0 and 255 representing grey scale.
  - The `labels` components is a vector representing the digit shown in the image.

```
library(tidyverse)
mnist <- dslabs::read_mnist()
dim(mnist$train$images)
dim(mnist$test$images)

# The i0-th image in the training set
i0 = 2023
matrix(mnist$train$images[i0,], ncol = 28) %>%
  image(col = gray.colors(12, rev = TRUE), axes = FALSE)

# The digit for the i0-th image
mnist$train$labels[i0]

# PCA for training images
decomp <- prcomp(mnist$train$images)

# Plot the first 9 loadings
par(mfrow = c(3, 3))
for (i in seq_len(9)) {
  matrix(decomp$rotation[,i], ncol = 28) %>%
    image(col = gray.colors(12, rev = TRUE), axes = FALSE, main = paste0("PC", i))
}

# Plot training images according to their 1st and 2nd PC scores
decomp$x[,1:2] %>%
```

```r
  as.data.frame() %>%
  mutate(label = factor(mnist$train$labels)) %>%
  ggplot(aes(PC1, PC2, colour = label)) +
  geom_point(alpha = 0.5) +
  theme_minimal()

# Plot testing images according to their 1st and 2nd PC scores
decomp %>%
  predict(newdata = mnist$test$images) %>%
  as.data.frame() %>%
  mutate(label = factor(mnist$test$labels)) %>%
  ggplot(aes(PC1, PC2, colour = label)) +
  geom_point(alpha = 0.5) +
  theme_minimal()

# Figure out the termination point of PCA
s = which(cumsum((decomp$sdev)^2)/sum((decomp$sdev)^2)>=.9)[1]

# Approximating the 2022nd image in the training set with s PC scores
x.bar.train = colMeans(mnist$train$images)
approx_mnist <- x.bar.train + decomp$rotation[, seq_len(s)] %*% decomp$x[i0, seq_len(s)]
par(mfrow = c(1, 2))
matrix(approx_mnist, ncol = 28) %>%
  image(col = gray.colors(12, rev = TRUE), axes = FALSE, main = "Approx")
matrix(mnist$train$images[i0,], ncol = 28) %>%
  image(col = gray.colors(12, rev = TRUE), axes = FALSE, main = "Original")

# Approximating the 2022nd image in the testing set with s PC scores
PCscores = t(mnist$test$images[i0,] - x.bar.train) %*% decomp$rotation
approx_mnist <- x.bar.train + decomp$rotation[, seq_len(s)] %*% PCscores[1:s]
par(mfrow = c(1, 2))
matrix(approx_mnist, ncol = 28) %>%
  image(col = gray.colors(12, rev = TRUE), axes = FALSE, main = "Approx")
matrix(mnist$test$images[i0,], ncol = 28) %>%
  image(col = gray.colors(12, rev = TRUE), axes = FALSE, main = "Original")
```

- PC regression (PCR): regression on PC scores
    1. Perform PCA on the observed data matrix of explanatory variables, usually centered
    2. Regress the outcome vector(s) on the selected PCs as covariates using linear regression to get a vector of estimated regression coefficients
    3. Transform this coefficient vector back to the scale of the actual covariates
- Note that the prediction of PCR is identical to that of linear regression, when all the PCs are included.

- Example of PCR: dataset `Prostate` comes from a study that examined the correlation between the level of prostate-specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy; see Stamey et al, 1989, Journal of Urology 141(5), 1076–1083.
    - `lcavol`: log(cancer volume)
    - `lweight`: log(prostate weight)
    - `age`: patient age
    - `lbph`: log(benign prostatic hyperplasia amount)
    - `svi`: seminal vesicle invasion

- `lcp`: log(capsular penetration)
- `gleason`: Gleason score
- `pgg45`: percentage Gleason scores 4 or 5
- `lpsa`: log(prostate specific antigen)

Summary of PCA

- Procedure
    1. Create PCs which are weighted sums of (centered) explanatory variables, with eigenvectors of (sample) correlation/covariance matrix taken as weights.
    2. Take PCs as surrogates of (centered) explanatory variables for various techniques
- Pros and cons
    - Doable without strong distribution assumption
    - Uninterpretable PCs
    - Not involving response; abandoned PCs possibly related to response