Canonical Correlation Analysis

Max Turgeon

STAT 7200–Multivariate Statistics

Objectives

- Introduce Canonical Correlation Analysis
    - Both the population and sample models
- Discuss generalizations of correlation coefficients
- Give a geometric interpretation of CCA
- Explain the relationship between CCA and the likelihood ratio test for independence
- Introduce reduced-rank regression

Introduction

- Canonical Correlation Analysis (CCA) is a dimension reduction method that is similar to PCA, but where we simultaneously reduce the dimension of **two** random vectors $\mathbf{Y}$ and $\mathbf{X}$.
- Instead of trying to explain overall variance, we try to explain the correlation $\mathrm{Corr}(\mathbf{Y}, \mathbf{X})$.
    - Note that this is a measure of **association** between $\mathbf{Y}$ and $\mathbf{X}$.
- Examples include:
    - Arithmetic speed and power ($\mathbf{Y}$) and reading speed and power ($\mathbf{X}$)
    - College performance metrics ($\mathbf{Y}$) and high-school achievement metrics ($\mathbf{X}$)

Population model i

- Let $\mathbf{Y}$ and $\mathbf{X}$ be p- and q-dimensional random vectors, respectively.
  - We will assume that $p \leq q$.
- Let μ_Y and μ_X be the mean of $\mathbf{Y}$ and $\mathbf{X}$, respectively.
- Let Σ_Y and Σ_X be the covariance matrix of $\mathbf{Y}$ and $\mathbf{X}$, respectively, and let $\Sigma_{YX} = \Sigma_{XY}^T$ be the covariance matrix $\mathrm{Cov}(\mathbf{Y}, \mathbf{X})$.
  - Assume Σ_Y and Σ_X are positive definite.
- Note that Σ_{YX} has pq entries, corresponding to all covariances between a component of $\mathbf{Y}$ and a component of $\mathbf{X}$.
- **Goal of CCA**: Summarise Σ_{YX} with p numbers.
  - These p numbers will be called the *canonical correlations*.

Dimension reduction i

- Let $U = a^T\mathbf{Y}$ and $V = b^T\mathbf{X}$ be linear combinations of $\mathbf{Y}$ and $\mathbf{X}$, respectively.
- We have:
  - $\mathrm{Var}(U) = a^T\Sigma_Y a$
  - $\mathrm{Var}(V) = b^T\Sigma_X b$
  - $\mathrm{Cov}(U, V) = a^T\Sigma_{YX} b$.
- Therefore, we can write the correlation between U and V as follows:

$$\mathrm{Corr}(U, V) = \frac{a^T\Sigma_{YX} b}{\sqrt{a^T\Sigma_Y a}\sqrt{b^T\Sigma_X b}}.$$

- We are looking for vectors $a \in \mathbb{R}^p, b \in \mathbb{R}^q$ such that $\mathrm{Corr}(U, V)$ is **maximised**.

Definitions

- The *first pair of canonical variates* is the pair of linear combinations U_1, V_1 with unit variance such that $\mathrm{Corr}(U_1, V_1)$ is maximised.
- The k-**th pair of canonical variates** is the pair of linear combinations U_k, V_k with unit variance such that $\mathrm{Corr}(U_k, V_k)$ is maximised among all pairs that are uncorrelated with the previous $k-1$ pairs.
- When U_k, V_k is the k-th pair of canonical variates, we say that $\rho_k = \mathrm{Corr}(U_k, V_k)$ is the k-th *canonical correlation*.

Derivation of canonical variates i

- Make a change of variables:
    - $\tilde{a} = \Sigma_Y^{1/2} a$
    - $\tilde{b} = \Sigma_X^{1/2} b$
- We can then rewrite the correlation:

$$\begin{aligned}\mathrm{Corr}(U, V) &= \frac{a^T \Sigma_{YX} b}{\sqrt{a^T \Sigma_Y a}\sqrt{b^T \Sigma_X b}} \\ &= \frac{\tilde{a}^T \Sigma_Y^{-1/2} \Sigma_{YX} \Sigma_X^{-1/2} \tilde{b}}{\sqrt{\tilde{a}^T \tilde{a}}\sqrt{\tilde{b}^T \tilde{b}}}.\end{aligned}$$

- Let $M = \Sigma_Y^{-1/2} \Sigma_{YX} \Sigma_X^{-1/2}$. We have

$$\max_{a,b} \mathrm{Corr}(a^T \mathbf{Y}, b^T \mathbf{X}) \iff \max_{\tilde{a}, \tilde{b} : \|\tilde{a}\| = 1, \|\tilde{b}\| = 1} \tilde{a}^T M \tilde{b}$$

Derivation of canonical variates ii

- As we will see, the solution to this maximisation problem involves the **singular value decomposition** of M.
- Equivalently, it involves the **eigendecomposition** of MM^T, where

$$MM^T = \Sigma_Y^{-1/2}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}\Sigma_Y^{-1/2}.$$

CCA: Main theorem i

- Let $\lambda_1 \geq \cdots \geq \lambda_p$ be the eigenvalues of $\Sigma_Y^{-1/2}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}\Sigma_Y^{-1/2}$.
    - Let $e_1, \ldots, e_p$ be the corresponding eigenvector with unit norm.
- Note that $\lambda_1 \geq \cdots \geq \lambda_p$ are also the p largest eigenvalues of

$$M^T M = \Sigma_X^{-1/2}\Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}\Sigma_X^{-1/2}.$$

    - Let $f_1, \ldots, f_p$ be the corresponding eigenvectors with unit norm.
- Then the k-th pair of canonical variates is given by

$$U_k = e_k^T \Sigma_Y^{-1/2}\mathbf{Y}, \qquad V_k = f_k^T \Sigma_X^{-1/2}\mathbf{X}.$$

CCA: Main theorem ii

- Moreover, we have

$$\rho_k = \mathrm{Corr}(U_k, V_k) = \sqrt{\lambda_k}.$$

Proof i

First, we write

$$\rho_1 = \frac{\tilde{a}^T M \tilde{b}}{\sqrt{\tilde{a}^T \tilde{a}} \sqrt{\tilde{b}^T \tilde{b}}}.$$

Applying the Cauchy-Schwartz inequality to the numerator of ρ_1^2, we have

$$\left(\tilde{a}^T M \tilde{b}\right)^2 \leq \left(\tilde{a}^T \tilde{a}\right) \left(\tilde{b}^T M^T M \tilde{b}\right),$$

with equality if there exists a scalar C such that

$$\tilde{a} = C M \tilde{b}.$$

Proof ii

We now have

$$\rho_1^2 \le \frac{\left(\tilde{a}^T\tilde{a}\right)\left(\tilde{b}^T M^T M\tilde{b}\right)}{\left(\tilde{a}^T\tilde{a}\right)\left(\tilde{b}^T\tilde{b}\right)}$$
$$= \frac{\left(\tilde{b}^T M^T M\tilde{b}\right)}{\tilde{b}^T\tilde{b}}.$$

From our discussion on PCA, we know that we can maximise the ratio $\frac{\left(\tilde{b}^T M^T M\tilde{b}\right)}{\tilde{b}^T\tilde{b}}$ by taking $\tilde{b}$ to be the eigenvector corresponding to the largest eigenvalue λ_1 of $M^T M$.

Proof iii

In turn, this gives us

$$\begin{aligned} MM^T\tilde{a} &= MM^T\left(CM\tilde{b}\right) \\ &= CM\left(M^TM\tilde{b}\right) \\ &= CM\left(\lambda_1\tilde{b}\right) \\ &= \lambda_1\left(CM\tilde{b}\right) \\ &= \lambda_1\tilde{a}. \end{aligned}$$

In other words, when ρ_1^2 attains its maximum, $\tilde{a}$ is equal to the eigenvector corresponding to the largest eigenvalue λ_1 of MM^T.

Proof iv

Finally, we simply note that if $\tilde{a} = e_1$ and $\tilde{b} = f_1$, then we have

$$a = \Sigma_Y^{-1/2} e_1, \qquad b = \Sigma_X^{-1/2} f_1.$$

The next canonical variates are obtained by imposing an orthgonality constraint and repeating this analysis. $\square$

Some vocabulary

1. Canonical directions: $(e_k^T \Sigma_Y^{-1/2}, f_k^T \Sigma_X^{-1/2})$
2. Canonical variates: $(U_k, V_k) = \left(e_k^T \Sigma_Y^{-1/2} \mathbf{Y}, f_k^T \Sigma_X^{-1/2} \mathbf{X}\right)$
3. Canonical correlations: $\rho_k = \sqrt{\lambda_k}$

Example i

```r
Sigma_Y <- matrix(c(1, 0.4, 0.4, 1), ncol = 2)
Sigma_X <- matrix(c(1, 0.2, 0.2, 1), ncol = 2)
Sigma_YX <- matrix(c(0.5, 0.3, 0.6, 0.4), ncol = 2)
Sigma_XY <- t(Sigma_YX)

rbind(cbind(Sigma_Y, Sigma_YX),
      cbind(Sigma_XY, Sigma_X))
```

Example ii

```
##      [,1] [,2] [,3] [,4]
## [1,]  1.0  0.4  0.5  0.6
## [2,]  0.4  1.0  0.3  0.4
## [3,]  0.5  0.3  1.0  0.2
## [4,]  0.6  0.4  0.2  1.0
```

Example iii

```r
library(expm)
sqrt_Y <- sqrtm(Sigma_Y)
sqrt_X <- sqrtm(Sigma_X)
M1 <- solve(sqrt_Y) %*% Sigma_YX %*% solve(Sigma_X)%*%
  Sigma_XY %*% solve(sqrt_Y)

(decomp1 <- eigen(M1))
```

Example iv

```
## eigen() decomposition
## $values
## [1] 0.5457180317 0.0009089525
##
## $vectors
##            [,1]       [,2]
## [1,] -0.8946536  0.4467605
## [2,] -0.4467605 -0.8946536
```

```
decomp1$vectors[,1] %*% solve(sqrt_Y)
```

Example v

```
##              [,1]       [,2]
## [1,] -0.8559647 -0.2777371
```

```
M2 <- solve(sqrt_X) %*% Sigma_XY %*% solve(Sigma_Y)%*%
  Sigma_YX %*% solve(sqrt_X)

decomp2 <- eigen(M2)
decomp2$vectors[,1] %*% solve(sqrt_X)
```

```
##             [,1]      [,2]
## [1,] 0.5448119 0.7366455
```

Example vi

```r
sqrt(decomp1$values)
```

```
## [1] 0.73872731 0.03014884
```

Sample CCA

- Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ and $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be random samples, and arrange them in $n \times p$ and $n \times q$ matrices $\mathbb{Y}, \mathbb{X}$, respectively.
  - Note that both sample sizes are equal.
  - Indeed, we assume that $(\mathbf{Y}_i, \mathbf{X}_i)$ are sampled jointly, i.e. on the **same** experimental unit.
- Let $\bar{\mathbf{Y}}$ and $\bar{\mathbf{X}}$ be the sample means.
- Let S_Y and S_X be the sample covariances.
- Define

$$S_{YX} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\mathbf{Y}_i - \bar{\mathbf{Y}}\right) \left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T.$$

Sample CCA: Main theorem i

- Let $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p$ be the eigenvalues of $S_Y^{-1/2} S_{YX} S_X^{-1} S_{XY} S_Y^{-1/2}$.
  - Let $\hat{e}_1, \ldots, \hat{e}_p$ be the corresponding eigenvector with unit norm.
- Note that $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p$ are also the p largest eigenvalues of

$$S_X^{-1/2} S_{XY} S_Y^{-1} S_{YX} S_X^{-1/2}.$$

  - Let $\hat{f}_1, \ldots, \hat{f}_p$ be the corresponding eigenvectors with unit norm.
- Then the k-th pair of *sample* canonical variates is given by

$$\hat{U}_k = \mathbb{Y} S_Y^{-1/2} \hat{e}_k, \qquad \hat{V}_k = \mathbb{X} S_X^{-1/2} \hat{f}_k.$$

Sample CCA: Main theorem ii

- Moreover, we have that $\hat{\rho}_k = \sqrt{\hat{\lambda}_k}$ is the sample correlation of $\hat{U}_k$ and $\hat{V}_k$.

Example (cont'd) i

```
# Let's generate data
library(mvtnorm)
Sigma <- rbind(cbind(Sigma_Y, Sigma_YX),
               cbind(Sigma_XY, Sigma_X))

YX <- rmvnorm(100, sigma = Sigma)
Y <- YX[,1:2]
X <- YX[,3:4]

decomp <- stats::cancor(x = X, y = Y)
```

Example (cont'd) ii

```
U <- Y %*% decomp$ycoef
V <- X %*% decomp$xcoef

diag(cor(U, V))
```

```
## [1] 0.789215963 0.005973183
```

```
decomp$cor
```

```
## [1] 0.789215963 0.005973183
```

Example i

```r
library(tidyverse)
library(dslabs)

str(olive)
```

```
## 'data.frame':    572 obs. of  10 variables:
##  $ region     : Factor w/ 3 levels "Northern Italy",..
##  $ area       : Factor w/ 9 levels "Calabria","Coast-S
##  $ palmitic   : num  10.75 10.88 9.11 9.66 10.51 ...
##  $ palmitoleic: num  0.75 0.73 0.54 0.57 0.67 0.49 0.6
##  $ stearic    : num  2.26 2.24 2.46 2.4 2.59 2.68 2.64
```

Example ii

```
##  $ oleic      : num  78.2 77.1 81.1 79.5 77.7 ...
##  $ linoleic   : num  6.72 7.81 5.49 6.19 6.72 6.78 6.1
##  $ linolenic  : num  0.36 0.31 0.31 0.5 0.5 0.51 0.49
##  $ arachidic  : num  0.6 0.61 0.63 0.78 0.8 0.7 0.56 0
##  $ eicosenoic : num  0.29 0.29 0.29 0.35 0.46 0.44 0.2
```

```r
# X contains the type of acids
X <- select(olive, -area, -region) %>%
  as.matrix

# Y contains the information about regions
count(olive, region)
```

Example iii

```
## # A tibble: 3 x 2
##   region              n
##   <fct>           <int>
## 1 Northern Italy    151
## 2 Sardinia           98
## 3 Southern Italy    323
```

```
Y <- select(olive, region) %>%
  model.matrix(~ region - 1, data = .)
```

```
# We get three dummy variables
head(unname(Y))
```

Example iv

```
##      [,1] [,2] [,3]
## [1,]    0    0    1
## [2,]    0    0    1
## [3,]    0    0    1
## [4,]    0    0    1
## [5,]    0    0    1
## [6,]    0    0    1
```

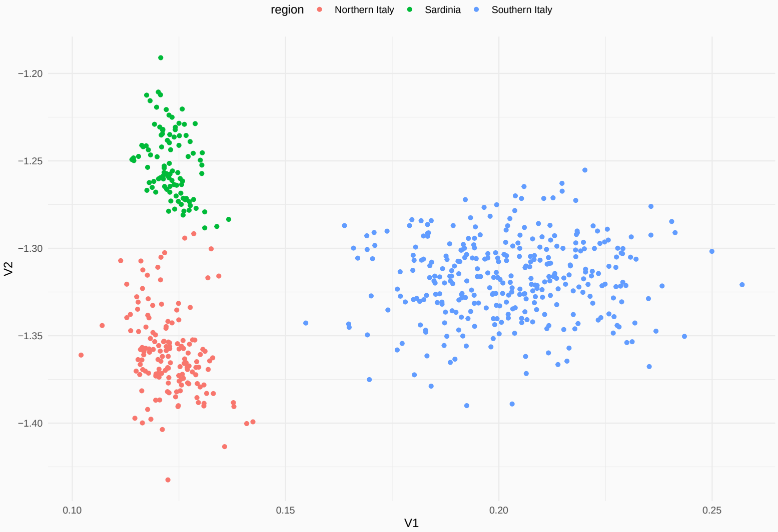
```
decomp <- cancor(X, Y)

V <- X %*% decomp$xcoef
```

Example v

```
data.frame(
  V1 = V[,1],
  V2 = V[,2],
  region = olive$region
) %>%
  ggplot(aes(V1, V2, colour = region)) +
  geom_point() +
  theme_minimal() +
  theme(legend.position = 'top')
```

Example vi



region
Northern Italy
Sardinia
Southern Italy
−1.20
−1.25
−1.30
−1.35
−1.40
V2
0.10
0.15
0.20
0.25
V1

Comments i

- The main difference between CCA and Multivariate Linear Regression is that CCA treats $\mathbb{Y}$ and $\mathbb{X}$ *symmetrically*.
- As with PCA, you can use CCA and the covariance matrix or the correlation matrix.
    - The latter is equivalent to performing CCA on the standardised variables.
- Note that sample CCA involves inverting the sample covariance matrices S_Y and S_X:
    - This means we need to assume $p, q < n$.
    - In general, this is what drives most of the performance (or lack thereof) of CCA.

Comments ii

    - There may be gains in efficiency by directly estimating the inverse covariance.
- When one of the two datasets $\mathbb{Y}$ or $\mathbb{X}$ represent indicators variables for a categorical variables (cf. the olive dataset), CCA is equivalent to **Linear Discriminant Analysis**.
    - To learn more about this method, see a course/textbook on Statistical Learning.

Proportions of Explained Sample Variance i

- Just like in PCA, there is a notion of *proportion of explained variance* that may be helpful in determining the number of canonical variates to retain.
- Assume that $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ and $\mathbf{X}_1, \ldots, \mathbf{X}_n$ have been **standardized**.
- Recall that
    - $\mathrm{tr}\left(\mathrm{Corr}(\mathbb{Y})\right) = p$
    - $\mathrm{tr}\left(\mathrm{Corr}(\mathbb{X})\right) = q$

Proportions of Explained Sample Variance ii

- We define the following quantities:
  - Proportion of total standardized sample variance in $\mathbb{Y} = \begin{pmatrix} \mathbb{Y}_1 & \cdots & \mathbb{Y}_p \end{pmatrix}$ explained by $\hat{U}_1, \ldots, \hat{U}_r$:

$$R^2(\mathbf{Y} \mid \hat{U}_1, \ldots, \hat{U}_r) = \frac{\sum_{i=1}^{r} \sum_{j=1}^{p} \mathrm{Corr}\left(\hat{U}_i, \mathbb{Y}_j\right)^2}{p}$$

  - Proportion of total standardized sample variance in $\mathbb{X} = \begin{pmatrix} \mathbb{X}_1 & \cdots & \mathbb{X}_q \end{pmatrix}$ explained by $\hat{V}_1, \ldots, \hat{V}_r$:

$$R^2(\mathbf{X} \mid \hat{V}_1, \ldots, \hat{V}_r) = \frac{\sum_{i=1}^{r} \sum_{j=1}^{q} \mathrm{Corr}\left(\hat{V}_i, \mathbb{X}_j\right)^2}{q}$$

Example i

```r
# Olive data--Standardize
X_sc <- scale(X)
Y_sc <- scale(Y)
decomp_sc <- cancor(X_sc, Y_sc)

# Extract Canonical variates
V_sc <- X_sc %*% decomp_sc$xcoef
colnames(V_sc) <- paste0("CC", seq_len(ncol(V_sc)))
```

```r
(prop_X <- rowMeans(cor(V_sc, X_sc)^2))
```

Example ii

```
##   CC1   CC2   CC3   CC4   CC5   CC6   CC7   CC8
## 0.340 0.153 0.124 0.081 0.134 0.039 0.067 0.061
```

```r
cumsum(prop_X)
```

```
##  CC1  CC2  CC3  CC4  CC5  CC6  CC7  CC8
## 0.34 0.49 0.62 0.70 0.83 0.87 0.94 1.00
```

Example iii

```r
# But since we are dealing with correlations
# We get the same with unstandardized variables
decomp <- cancor(X, Y)
V <- X %*% decomp$xcoef
colnames(V) <- paste0("CC", seq_len(ncol(V)))

(prop_X <- rowMeans(cor(V, X)^2))
```

```
##   CC1   CC2   CC3   CC4   CC5   CC6   CC7   CC8
## 0.340 0.153 0.124 0.081 0.134 0.039 0.067 0.061
```

Example iv

```r
cumsum(prop_X)
```

```
##  CC1  CC2  CC3  CC4  CC5  CC6  CC7  CC8
## 0.34 0.49 0.62 0.70 0.83 0.87 0.94 1.00
```

Interpreting the population canonical variates i

- To help interpretating the canonical variates, let's go back to the population model.
- Define

$$A = \begin{pmatrix} e_1^T \Sigma_Y^{-1/2} & \cdots & e_p^T \Sigma_Y^{-1/2} \end{pmatrix}^T,$$
$$B = \begin{pmatrix} f_1^T \Sigma_X^{-1/2} & \cdots & f_p^T \Sigma_X^{-1/2} \end{pmatrix}^T.$$

- In other words, the *rows* of A and B are the canonical directions.

Interpreting the population canonical variates ii

- Using this notation, we can get all canonical variates using one linear transformation:

$$\mathbf{U} = A\mathbf{Y}, \qquad \mathbf{V} = B\mathbf{X}.$$

- We then have

$$\mathrm{Cov}(\mathbf{U}, \mathbf{Y}) = \mathrm{Cov}(A\mathbf{Y}, \mathbf{Y}) = A\Sigma_Y.$$

- Since $\mathrm{Cov}(\mathbf{U}) = I_p$, we have

$$\mathrm{Corr}(U_k, Y_i) = \mathrm{Cov}(U_k, \sigma_i^{-1} Y_i),$$

where σ_i^2 is the variance of Y_i.

Interpreting the population canonical variates iii

- If we let D_Y be the diagonal matrix whose i-th diagonal element is $\sigma_i = \sqrt{\mathrm{Var}(Y_i)}$, we can write

$$\mathrm{Corr}(\mathbf{U}, \mathbf{Y}) = A\Sigma_Y D_Y^{-1}.$$

- Using similar computations, we get

$$\begin{aligned} \mathrm{Corr}(\mathbf{U}, \mathbf{Y}) &= A\Sigma_Y D_Y^{-1}, & \mathrm{Corr}(\mathbf{V}, \mathbf{Y}) &= B\Sigma_{XY} D_Y^{-1}, \\ \mathrm{Corr}(\mathbf{U}, \mathbf{X}) &= A\Sigma_{YX} D_X^{-1}, & \mathrm{Corr}(\mathbf{V}, \mathbf{X}) &= B\Sigma_X D_X^{-1}. \end{aligned}$$

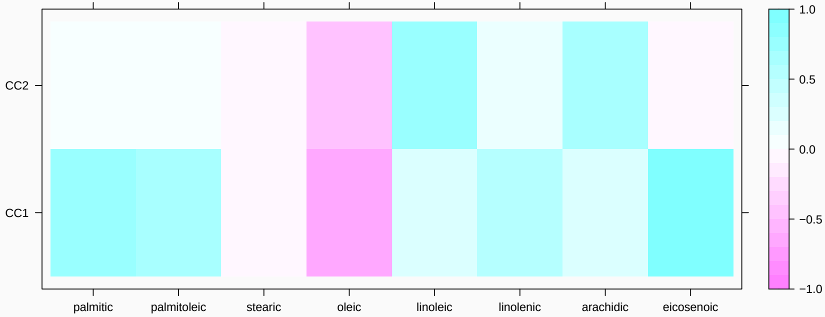
- **These quantities** (and their sample counterparts) **give us information about the contribution of the original variables to the canonical variates**.

Example i

```
# Let's go back to the olive data
decomp <- cancor(X, Y)
V <- X %*% decomp$xcoef
colnames(V) <- paste0("CC", seq_len(8))

library(lattice)
levelplot(cor(X, V[,1:2]),
          at = seq(-1, 1, by = 0.1),
          xlab = "", ylab = "")
```

Example ii

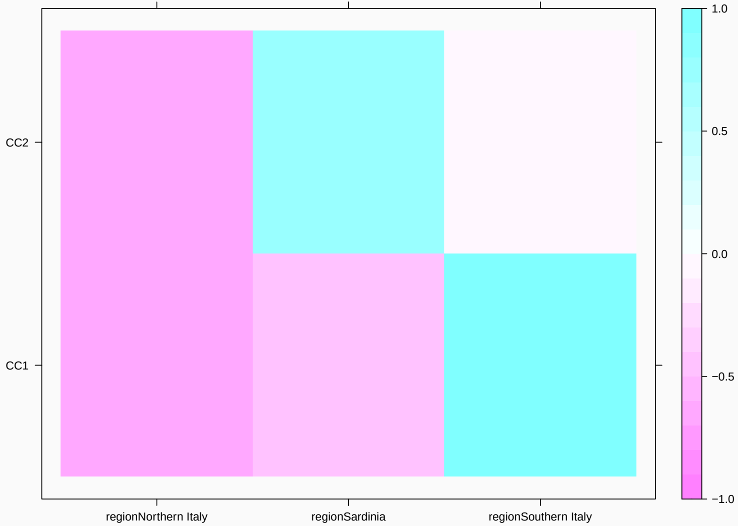


CC2
CC1
palmitic
palmitoleic
stearic
oleic
linoleic
linolenic
arachidic
eicosenoic
1.0
0.5
0.0
−0.5
−1.0

Example iii

```
levelplot(cor(Y, V[,1:2]),
          at = seq(-1, 1, by = 0.1),
          xlab = "", ylab = "")
```

Example iv


CC2
CC1
regionNorthern Italy
regionSardinia
regionSouthern Italy
1.0
0.5
0.0
−0.5
−1.0

Generalization of correlation coefficients i

- The canonical correlations can be seen as a generalization of many notions of "correlation".
- If both $\mathbf{Y}, \mathbf{X}$ are one dimensional, then

$$\mathrm{Corr}(a^T\mathbf{Y}, b^T\mathbf{X}) = \mathrm{Corr}(\mathbf{Y}, \mathbf{X}), \quad \text{for all } a, b.$$

- In other words, the canonical correlation generalizes the **univariate correlation coefficient**.
- Then assume $\mathbf{Y}$ is one-dimensional, but $\mathbf{X}$ is q-dimensional. Then CCA is equivalent to (univariate) linear regression, and the first canonical correlation is equal to the **multiple correlation coefficient**.

Generalization of correlation coefficients ii

- Now, let's go back to full-generality: $\mathbf{Y} = (Y_1, \ldots, Y_p)$, $\mathbf{X} = (X_1, \ldots, X_q)$. Let a be all zero except for a one in position i, and let b be all zero except for a one in position j. We have

$$\begin{aligned} |\mathrm{Corr}(Y_i, X_j)| &= |\mathrm{Corr}(a^T\mathbf{Y}, b^T\mathbf{X})| \\ &\leq \max_{a,b} \mathrm{Corr}(a^T\mathbf{Y}, b^T\mathbf{X}) \\ &= \rho_1. \end{aligned}$$

- In other words, the **first canonical correlation is larger than any entry** (in absolute value) **in the matrix** $\mathrm{Corr}(\mathbf{Y}, \mathbf{X})$.

Generalization of correlation coefficients iii

- Finally, the k-th canonical correlation ρ_k can be interpreted as the **multiple correlation coefficient** of two different univariate linear regression model:
  - U_k against $\mathbf{X}$;
  - V_k against $\mathbf{Y}$.

Example (cont'd) i

```r
# Canonical correlations
decomp$cor
```

```
## [1] 0.95 0.84
```

```r
# Maximum value in correlation matrix
max(abs(cor(Y, X)))
```

```
## [1] 0.89
```

Example (cont'd) ii

```r
# Multiple correlation coefficients
sqrt(summary(lm(V[,1] ~ Y))$r.squared)
```

```
## [1] 0.95
```

```r
sqrt(summary(lm(V[,2] ~ Y))$r.squared)
```

```
## [1] 0.84
```

Geometric interpretation i

- Let's look at a geometric interpretation of CCA.
- First, some notation:
    - Let A be the matrix whose k-th row is the k-th canonical direction $e_k^T\Sigma_Y^{-1/2}$.
    - Let E be the matrix whose k-th *column* is the eigenvector e_k. Note that $E^TE = I_p$.
    - We thus have $A = E^T\Sigma_Y^{-1/2}$.
- We get all canonical variates U_k by transforming $\mathbf{Y}$ using A:

$$\mathbf{U} = A\mathbf{Y}.$$

Geometric interpretation ii

- Now, using the spectral decomposition of Σ_Y, we can write

$$A = E^T \Sigma_Y^{-1/2} = E^T P_Y \Lambda_Y^{-1/2} P_Y^T,$$

  where P_Y contains the eigenvectors of Σ_Y and Λ_Y is the diagonal matrix with its eigenvalues.
- Therefore, we can see that

$$\mathbf{U} = A\mathbf{Y} = E^T P_Y \Lambda_Y^{-1/2} P_Y^T \mathbf{Y}.$$

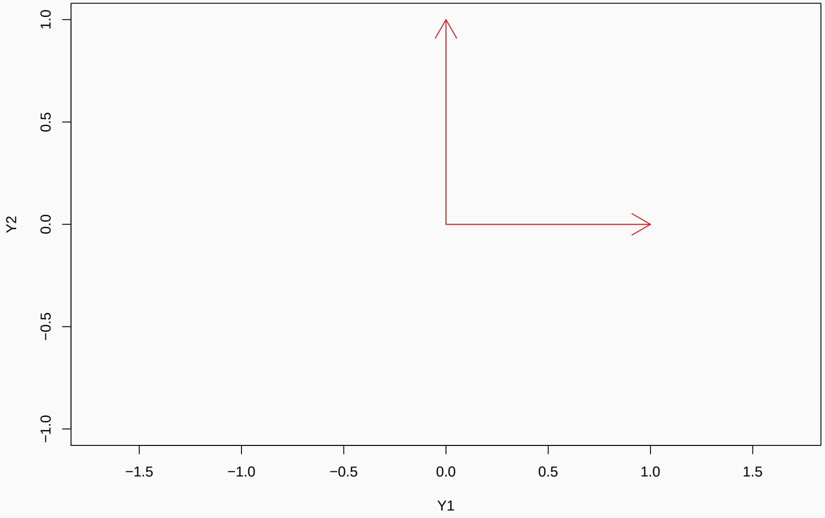
Geometric interpretation iii

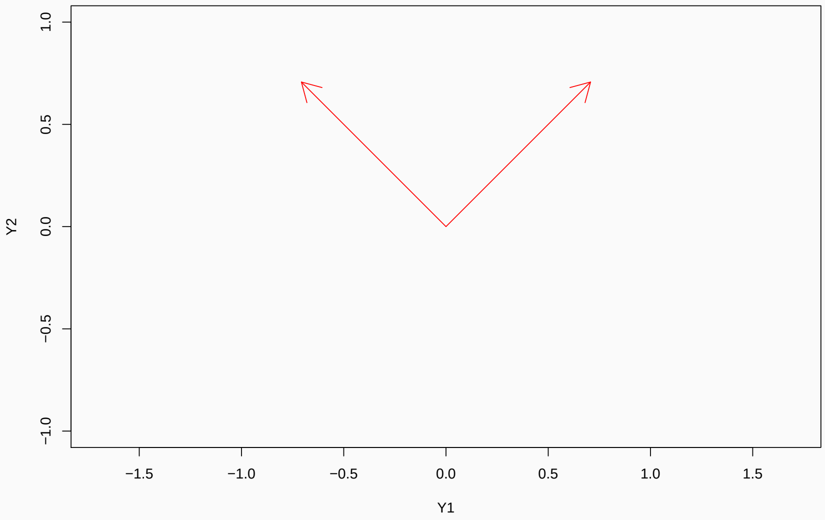
- Let's look at this expression in stages:
    - $P_Y^T \mathbf{Y}$: This is the matrix of **principal components** of $\mathbf{Y}$.
    - $\Lambda_Y^{-1/2}\left(P_Y^T \mathbf{Y}\right)$: We standardize the principal components to have unit variance.
    - $P_Y\left(\Lambda_Y^{-1/2} P_Y^T \mathbf{Y}\right)$: We rotate the standardized PCs using a transformation that **only involves** Σ_Y.
    - $E^T\left(P_Y \Lambda_Y^{-1/2} P_Y^T \mathbf{Y}\right)$: We rotate the result using a transformation that **involves the whole covariance matrix** Σ.

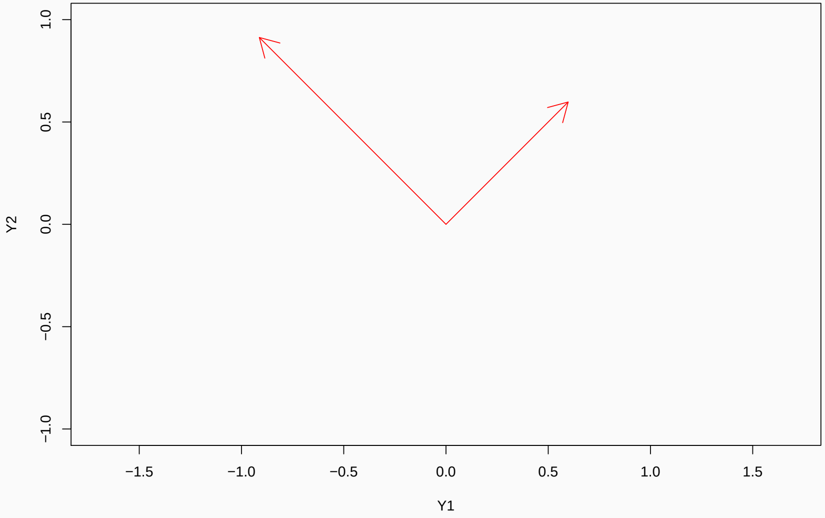
Example i

- Let's go back to the covariance matrix at the beginning of this slide deck:

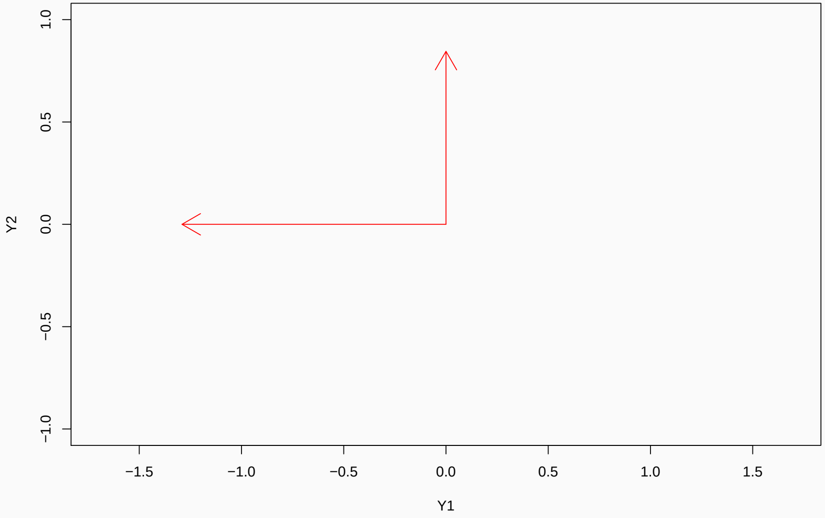
$$\Sigma = \begin{pmatrix} 1.0 & 0.4 & 0.5 & 0.6 \\ 0.4 & 1.0 & 0.3 & 0.4 \\ 0.5 & 0.3 & 1.0 & 0.2 \\ 0.6 & 0.4 & 0.2 & 1.0 \end{pmatrix}.$$

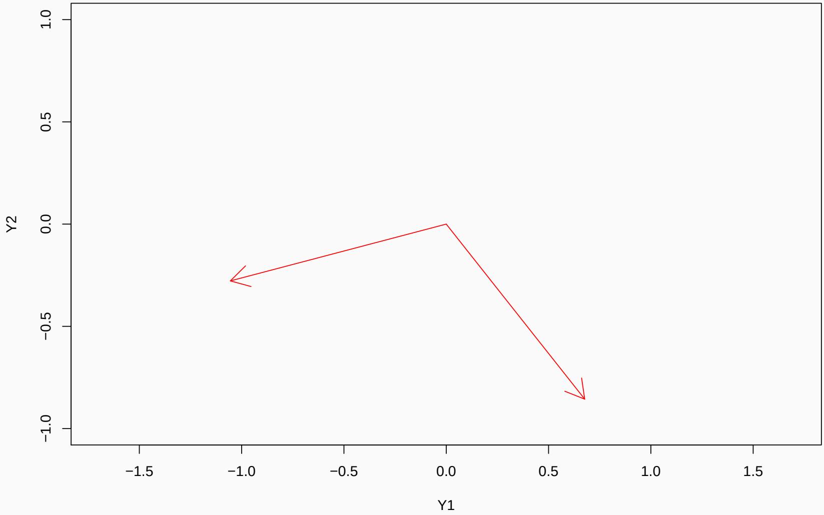

1.0
0.5
0.0
−0.5
−1.0
Y2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y1


1.0
0.5
0.0
−0.5
−1.0
Y2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y1



Y2
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y1


1.0
0.5
0.0
−0.5
−1.0
Y2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y1



1.0
0.5
0.0
−0.5
−1.0
Y2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Y1

Large sample inference

Test of independence i

- Recall what we said at the outset: CCA trys to explain the covariance $\mathrm{Cov}(\mathbf{Y}, \mathbf{X})$.
- If there is no correlation between $\mathbf{Y}, \mathbf{X}$, then $\Sigma_{YX} = 0$.
  - In particular, $a^T \Sigma_{YX} b = 0$ for any choice of $a \in \mathbb{R}^p, b \in \mathbb{R}^q$, and therefore all canonical correlations are equal to 0.
- To test for independence between $\mathbf{Y}$ and $\mathbf{X}$, we can use a **likelihood ratio test**.
  - Recall our discussion of tests for covariance matrices.

LRT for $\Sigma_{YX} = 0$ i

Let $(\mathbf{Y}_i, \mathbf{X}_i)$, $i = 1, \ldots, n$, be a random sample from a normal distribution $N_{p+q}(\mu, \Sigma)$, with

$$\Sigma = \begin{pmatrix} \Sigma_Y & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_X \end{pmatrix}.$$

Let S_Y, S_X be the sample covariances of $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ and $\mathbf{X}_1, \ldots, \mathbf{X}_n$, respectively, and let S_n be the $p+q$-dimensional sample covariance of $(\mathbf{Y}_i, \mathbf{X}_i)$.

Then the likelihood ratio test for $H_0 : \Sigma_{YX} = 0$ rejects H_0 for large values of

$$-2 \log \Lambda = n \log \left(\frac{|S_Y||S_X|}{|S_n|} \right) = -n \log \prod_{i=1}^{p} (1 - \hat{\rho}_i^2),$$

LRT for $\Sigma_{YX} = 0$ ii

where $\hat{\rho}_1, \ldots, \hat{\rho}_p$ are the sample canonical correlations.

Let's prove the second equality: first, note that this is equivalent to showing

$$\Lambda^{2/n} = \frac{|S_n|}{|S_Y||S_X|} = \prod_{i=1}^{p}(1 - \hat{\rho}_i^2).$$

Also, note that we can decompose S_n into a block matrix:

$$S_n = \begin{pmatrix} S_Y & S_{YX} \\ S_{XY} & S_X \end{pmatrix}.$$

LRT for $\Sigma_{YX} = 0$ iii

We can then use the formula for the determinant of block matrix:

$$|S_n| = |S_X| \cdot |S_Y - S_{YX} S_X^{-1} S_{XY}|.$$

LRT for $\Sigma_{YX} = 0$ iv

We can thus write

$$\begin{aligned}\Lambda^{2/n} &= \frac{|S_n|}{|S_Y||S_X|} \\ &= \frac{|S_X| \cdot |S_Y - S_{YX}S_X^{-1}S_{XY}|}{|S_Y||S_X|} \\ &= \frac{|S_Y - S_{YX}S_X^{-1}S_{XY}|}{|S_Y|} \\ &= |I_p - S_{YX}S_X^{-1}S_{XY}S_Y^{-1}| \\ &= |I_p - S_Y^{-1/2}S_{YX}S_X^{-1}S_{XY}S_Y^{-1/2}| \quad = |I_p - \hat{M}\hat{M}^T|,\end{aligned}$$

where

$$\hat{M}\hat{M}^T = S_Y^{-1/2}S_{YX}S_X^{-1}S_{XY}S_Y^{-1/2}.$$

But we know that the eigenvalues of $\hat{M}\hat{M}^T$ are $\hat{\rho}_1^2 > \ldots > \hat{\rho}_p^2$, and therefore we can write

$$\Lambda^{2/n} = \prod_{i=1}^{p}(1 - \hat{\rho}_i^2).$$

$\square$

Null distribution

1. For large n, the statistic $-2\log\Lambda$ is approximately chi-square with degrees of freedom equal to

$$\left(\frac{(p+q)(p+q+1)}{2}\right)-\left(\frac{p(p+1)}{2}+\frac{q(q+1)}{2}\right)=pq.$$

2. Bartlett's correction uses a different statistic (but the same null distribution):

$$-\left(n-1-\frac{1}{2}(p+q+1)\right)\log\prod_{i=1}^{p}(1-\hat{\rho}_i^2).$$

Example i

- We will look at a different example, this time from the field of vegetation ecology.
- We have two datasets:
  - `varechem`: 14 chemical measurements from the soil.
  - `varespec`: 44 estimated cover values for lichen species.
- The data has 24 observations.
- For more details, see Väre, H., Ohtonen, R. and Oksanen, J. (1995) *Effects of reindeer grazing on understorey vegetation in dry Pinus sylvestris forests.* Journal of Vegetation Science 6, 523–530.

Example ii

```r
library(vegan)

data(varespec)
data(varechem)

# There are too many variables in varespec
# Let's pick first 10
Y <- select(varespec, Callvulg:Diphcomp) %>%
  as.matrix
```

Example iii

```r
# The help page in `vegan` suggests a better
# chemical model
X <- model.matrix( ~ Al + P*(K + Baresoil) - 1,
                  data = varechem)
colnames(X)[1:4]
```

```
## [1] "Al"       "P"        "K"        "Baresoil"
```

```r
colnames(X)[5:6]
```

```
## [1] "P:K"        "P:Baresoil"
```

Example iv

```
decomp <- cancor(x = X, y = Y)

n <- nrow(X)
(LRT <- -n*log(prod(1 - decomp$cor^2)))
```

```
## [1] 156
```

```
p <- min(ncol(X), ncol(Y))
q <- max(ncol(X), ncol(Y))
LRT > qchisq(0.95, df = p*q)
```

Example v

```
## [1] TRUE
```

```
LRT_bart <- -(n - 1 - 0.5*(p + q + 1)) *
  log(prod(1 - decomp$cor^2))

c("Large Sample" = LRT,
  "Bartlett" = LRT_bart)
```

```
## Large Sample     Bartlett
##          156           94
```

```
LRT_bart > qchisq(0.95, df = p*q)
```

Example vi

```
## [1] TRUE
```

Sequential inference i

- The LRT above was for independence, i.e. $\Sigma_{YX} = 0$.
- Given our description of CCA above, this test is equivalent to having all canonical correlations being equal to 0.

$$\Sigma_{YX} = 0 \iff \rho_1 = \cdots = \rho_p = 0.$$

- If we reject the null hypothesis, it is natural to ask how many canonical correlations are nonzero.
- Recall that by design $\rho_1 \geq \cdots \geq \rho_p$. We thus get a sequence of null hypotheses:

$$H_0^k : \rho_1 \neq 0, \ldots, \rho_k \neq 0, \rho_{k+1} = \cdots = \rho_p = 0.$$

Sequential inference ii

- We can test the k-th hypothesis using a *truncated* version of the likelihood ratio test statistic:

$$LRT_k = -\left(n - 1 - \frac{1}{2}(p + q + 1)\right) \log \prod_{i=k+1}^{p} (1 - \hat{\rho}_i^2),$$

where its null distribution is approximately chi-square on $(p - k)(q - k)$ degrees of freedom.

Example (cont'd) i

```r
# We can get the truncated LRTs in one go
(log_ccs <- rev(log(cumprod(1 - rev(decomp$cor)^2))))
```

```
## [1] -6.513 -4.002 -2.259 -1.011 -0.262 -0.073
```

```r
(LRTs <- -(n - 1 - 0.5*(p + q + 1)) * log_ccs)
```

```
## [1] 94.4 58.0 32.7 14.7  3.8  1.1
```

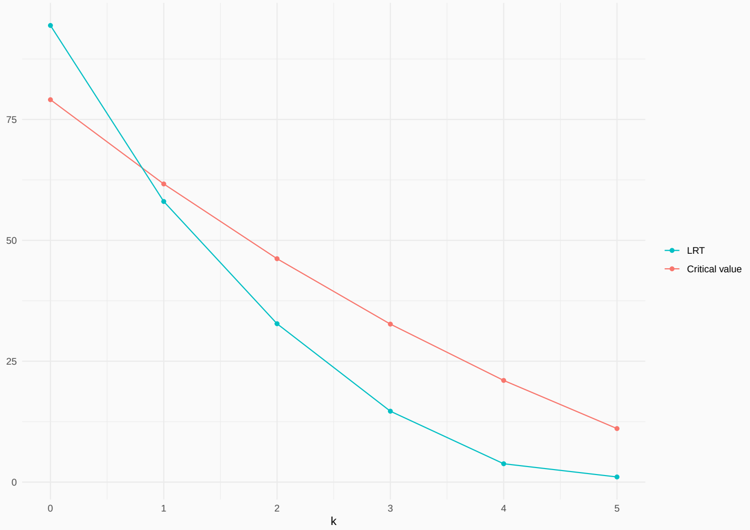
Example (cont'd) ii

```r
k_seq <- seq(0, p - 1)
LRTs > qchisq(0.95,
              df = (p - k_seq)*(q - k_seq))
```

```
## [1]  TRUE FALSE FALSE FALSE FALSE FALSE
```

```r
# We only reject the first null hypothesis
# of independence
```

Example (cont'd) iii



Reduced-Rank Regression

Multivariate Linear Regression

- Recall the setup for MLR: Let $\mathbf{Y}_1 \ldots, \mathbf{Y}_n$ be a random sample of size n, and let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be the corresponding sample of covariates.
- We assume a **linear relationship**:

$$E(\mathbf{Y}_i \mid \mathbf{X}_i) = B^T \mathbf{X}_i,$$

  where B is a $q \times p$ matrix of *regression coefficients*.
- We write $\mathbb{Y}$ and $\mathbb{X}$ for the matrices whose i-th row is $\mathbf{Y}_i$ and $\mathbf{X}_i$, respectively.
- The OLS estimator is then given by

$$\hat{B}_{OLS} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}.$$

Reduced-Rank Regression—Motivation i

- Two important observations:
  - The OLS estimate is equivalent to p independent univariate regressions. In other words, **no sharing of information across outcome variables.**
  - There are pq regression coefficients to estimate. **Every time we had an outcome variable, we need to estimate q new parameters.**

Reduced-Rank Regression—Motivation ii

- One way to mitigate both effects is to impose a rank restriction on B:
  - $\mathrm{rank}(B) = k$ is equivalent to having $p - k$ linear constraints

$$\ell_j^T B = 0, \qquad j = 1, \ldots, p - k.$$

  - $\mathrm{rank}(B) = k$ is also equivalent to writing $B^T = UV$, where U is $p \times k$, V is $k \times q$, and both are of rank k. This means that we have at most $(p + q)k$ regression coefficients to estimate.

Assume $\mathbf{X}_i, \mathbf{Y}_i$ have mean zero. Define $\Sigma_Y = \mathrm{Cov}(\mathbf{Y}_i)$, $\Sigma_X = \mathrm{Cov}(\mathbf{X}_i)$, and $\Sigma_{YX} = \mathrm{Cov}(\mathbf{Y}_i, \mathbf{X}_i)$, and assume that Σ_X is invertible. Finally, let Γ be a $p \times p$ positive-definite weight matrix. The $p \times k$ and $k \times q$ matrices U, V of rank k that minimize

$$\mathrm{tr}\left(E\left(\Gamma^{1/2}(\mathbf{Y}_i - UV\mathbf{X}_i)(\mathbf{Y}_i - UV\mathbf{X}_i)^T\Gamma^{1/2}\right)\right)$$

are given by

$$\hat{U} = \Gamma^{-1/2}W_k,$$
$$\hat{V} = W_k^T\Gamma^{1/2}\Sigma_{YX}\Sigma_X^{-1},$$

where the columns of W_k are the normalized eigenvectors corresponding to the k largest eigenvalues of $\Gamma^{1/2}\Sigma_{YX}\Sigma_X^{-1}\Sigma_{YX}^T\Gamma^{1/2}$.

Comments i

- This theorem can be proven using the Eckart-Young theorem (see lectures on PCA).
- When $p \leq q$ and we choose $k = p$, we recover the OLS estimate:
    - $\hat{B} = \hat{V}^T \hat{U}^T = \Sigma_X^{-1} \Sigma_{YX}^T$
- When $\Gamma = \Sigma_Y^{-1}$, the columns of U are the **canonical directions** for $\mathbf{Y}_i$
- The term *reduced-rank regression* is typically reserve for the case when $\Gamma = I_p$, i.e. the weight matrix is the identity matrix.

Comments ii

- At the sample level, the result becomes

$$\hat{U} = W_k,$$
$$\hat{V} = W_k^T \mathbb{Y}^T \mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1},$$

where the columns of W_k are the normalized eigenvectors corresponding to the k largest eigenvalues of $\mathbb{Y}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$.

  - This gives

$$\hat{B}_{RR} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}W_kW_k^T = \hat{B}_{OLS}W_kW_k^T$$

Example i

```r
# Recall the plastic film data
library(heplots)

fit <- lm(cbind(tear, gloss, opacity) ~ rate + additive,
          data = Plastic)
coef(fit)
```

```
##             tear gloss opacity
## (Intercept) 6.30  9.40    3.29
## rateHigh    0.59 -0.51    0.29
## additiveHigh 0.39  0.35    0.99
```

Example ii

```r
Y <- Plastic %>%
  select(tear, gloss, opacity) %>%
  as.matrix
X <- model.matrix(~ rate + additive, data = Plastic)

# We get the same as OLS
(beta_ols <- solve(crossprod(X), crossprod(X, Y)))
```

```
##               tear gloss opacity
## (Intercept)  6.29  9.39    3.29
## rateHigh     0.59 -0.51    0.29
## additiveHigh 0.39  0.35    0.99
```

Example iii

```r
# Reduced-Rank regression
M <- crossprod(Y, X) %*% beta_ols
decomp <- eigen(M)

# Take rank = 1
W <- decomp$vectors[,1, drop=FALSE]
rownames(W) <- colnames(Y)
(beta_rrr <- beta_ols %*% tcrossprod(W))
```

Example iv

```
##                 tear gloss opacity
## (Intercept)  6.551 8.990   3.811
## rateHigh     0.018 0.025   0.011
## additiveHigh 0.449 0.616   0.261
```

```
# Note that rank 1 means rows are colinear
beta_rrr[1,]/beta_rrr[2,]
```

```
##    tear   gloss opacity
##     359     359     359
```

Selecting the rank i

- Of course, the rank k is a *tuning parameter* that we need to select.
- One approach is to use sequential inference (see Section 2.6 of Reinsel and Velu).
- Another approach is to choose k that minimises the cross-validated MSE (cf. Lectures on Regularized Regression).
- In this lecture, we will focus on **Information Criteria**.
    - Recall the general form of Akaike's information criterion:

    $$-2\log L(\hat{B}, \hat{\Sigma}) + 2d,$$

    where d is the number of parameters to estimate.

Selecting the rank ii

- On the other hand, if we restrict B to have rank k, there are only $d = (p + q - k)k$ free parameters.
    - kq free parameters for the column space of B
    - $k(p - k)$ free parameters for the remaining columns
- However, a careful analysis shows that this is actually an *underestimate* of the true degrees of freedom
    - If $\lambda_1, \ldots, \lambda_p$ are the eigenvalues of $\mathbb{Y}^T\mathbb{X}(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$, then

$$d = (p + q - k)k + 2\sum_{\ell=1}^{k}\sum_{j=k+1}^{p}\frac{\lambda_j}{\lambda_\ell - \lambda_j}.$$

    - See for example Yuan (2016) *Degrees of freedom in low rank matrix estimation*

Selecting the rank iii

- The function `rrpack::rrr` calls the first type of degrees of freedom *naive*, and the second type, *exact*.
    - By default, it uses the exact degrees of freedom.

Example (cont'd) i

```
# Let's create a function
redrank <- function(Y, X, rank = 1) {
  beta_ols <- solve(crossprod(X), crossprod(X, Y))
  M <- crossprod(Y, X) %*% beta_ols
  decomp <- eigen(M)
  W <- decomp$vectors[,seq_len(rank),drop=FALSE]
  rownames(W) <- colnames(Y)
  return(beta_ols %*% tcrossprod(W))
}
```

Example (cont'd) ii

```
all.equal(beta_rrr, redrank(Y, X))
```

```
## [1] TRUE
```

```
# First the log likelihoods
loglik <- sapply(c(1, 2, 3), function(k) {
  beta_rrr <- redrank(Y, X, k)
  resids <- Y - X %*% beta_rrr
  -2*sum(dmvnorm(resids, log = TRUE,
                 sigma = crossprod(resids)/nrow(resids)))
})
```

Example (cont'd) iii

```
# With naive degrees of freedom
2*seq_len(3)*(ncol(X) + ncol(Y) -
                seq_len(3)) + loglik
```

```
## [1] 139 133 126
```

```r
# With exact degrees of freedom
dfs <- sapply(seq_len(3), function(k) {
  total <- 0
  lambdas <- decomp$values[seq(k+1, ncol(Y))]
  for (ell in seq(1, k)) {
    total <- sum(lambdas/(decomp$values[ell] - lambdas))
  }
  if (k == ncol(Y)) return(0) else return(2*total)
})
```

Example (cont'd) v

```r
2*seq_len(3)*(ncol(X) + ncol(Y) -
                seq_len(3)) + 2*dfs + loglik
```

```
## [1] 139.4238 134.8934 125.9592
```

```r
# Both approaches select the full-rank model
```

```r
# Constrast this with rrpack::rrr
# Which uses a different AIC
rrpack::rrr(Y, X, ic.type = "AIC")
```

Example (cont'd) vi

```
## Call:
## rrpack::rrr(Y = Y, X = X, ic.type = "AIC")
##
## Estimated Rank: 1
```

Example 2 i

```
# Tobacco dataset
tobacco_y <- as.matrix(rrr::tobacco[,1:3])
tobacco_x <- as.matrix(rrr::tobacco[,4:9])

dim(tobacco_x)
```

```
## [1] 25  6
```

```
dim(tobacco_y)
```

```
## [1] 25  3
```

Example 2 ii

```
(rr_fit <- rrpack::rrr(tobacco_y, tobacco_x))
```

```
## Call:
## rrpack::rrr(Y = tobacco_y, X = tobacco_x)
##
## Estimated Rank: 1
```

```
library(lattice)
coef <- rr_fit$coef
colnames(coef) <- colnames(tobacco_y)
levelplot(coef)
```

Example 2 iii