StatMod 2 - Exe 5 - PCA

MAURICE
DIESENDRUCK

1 SVD of Y is $Y = U\Sigma V^T$, where: $\underset{n\times p}{Y}$, $\underset{n\times n}{U}$, $\underset{n\times p}{\Sigma}$, $\underset{p\times p}{V^T}$

and $\Sigma = \begin{bmatrix} d_1 & & & 0 \cdots \\ & \ddots & & \vdots \\ & & d_p & 0 \cdots \end{bmatrix}$ for case $p > n$, is pseudo-diagonal of singular values.

Let $S = \text{cov}(Y) = \frac{1}{n} Y^T Y$, when Y is mean-centered and normalized by column.

Therefore, substituting in the SVD of Y into the definition of S, get:

$$S = \frac{1}{n}(U\Sigma V^T)^T(U\Sigma V^T) = \frac{1}{n} V\Sigma^T U^T U \Sigma V^T = \frac{1}{n} V \Sigma^T \Sigma V^T$$

$$= V \frac{\Sigma^T \Sigma}{n} V^T$$, since $U^T U = I$ by orthonormality of columns.

This is the eigenvalue decomposition for S, where $\Lambda = \Sigma^T \Sigma$ and $\left[\frac{\Lambda}{n}\right]_{kk}$ is the eigenvalue of S associated with v_k.

$\boxed{2}$ Characterize the vector w_1, which maximizes projection variance.

$$V_w = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^T w_1 - \bar{z}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^T w_1 - \bar{y}^T w_1\right)^2 = w_1^T\left[\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})(y_i-\bar{y})'\right]w_1$$

$= w_1^T S w_1$. Now $\max_{w_1} V_w$ subject to constraint $w_1^T w_1 \leq 1$ using Lagrangian multipliers.

$$L(w_1) = w_1^T S w_1 - \lambda_1(w_1^T w_1 - 1)$$

$$\Rightarrow \frac{\partial L}{\partial w_1} = 2Sw_1 - 2\lambda_1 w_1 = 0 \Rightarrow Sw_1 = \lambda_1 w_1 \text{ must hold.}$$

$\Rightarrow V_w = w_1^T S w_1 = w_1^T \lambda_1 w_1 = \lambda_1 w_1^T w_1$ is maximized when $w_1^T w_1 = 1$

and when λ_1 is the largest eigenvalue of S (with $w_1 \doteq v_1$*, the eigenvector of S associated with its largest eigenvalue).

* Recall that eigenvectors of S are columns of V.

$$V_w^{max} = \lambda_1 = \left[\frac{\Lambda}{n}\right]_{11} = \frac{d_1^2}{n},$$ where d_1 is the first/largest singular value of Y.

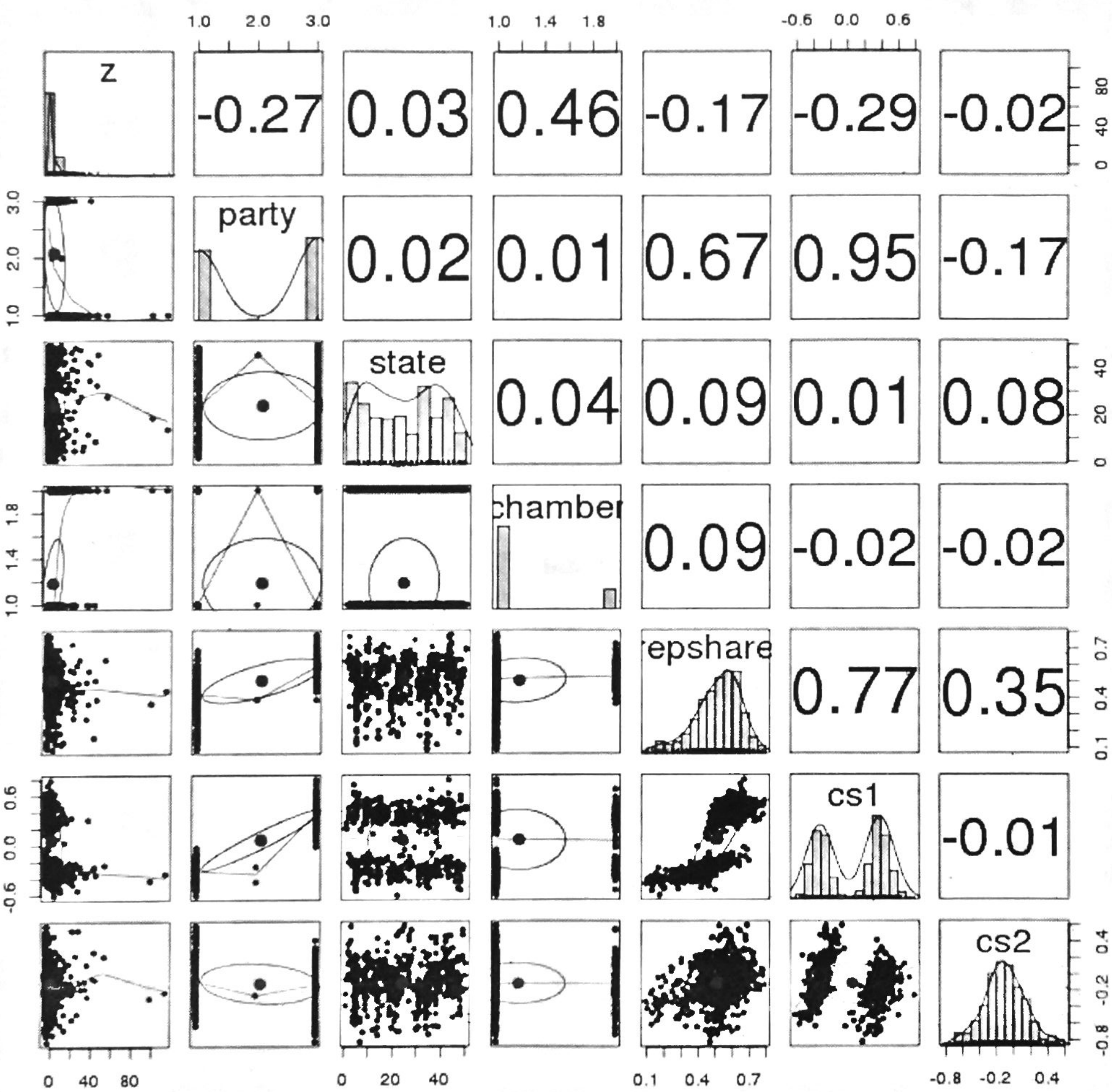
StatMod2 - Exercises 5 - PCA - Question 3 - Congress Data

Maurice Diesendruck

May 5, 2015

3.1 Evaluation of First Principal Component

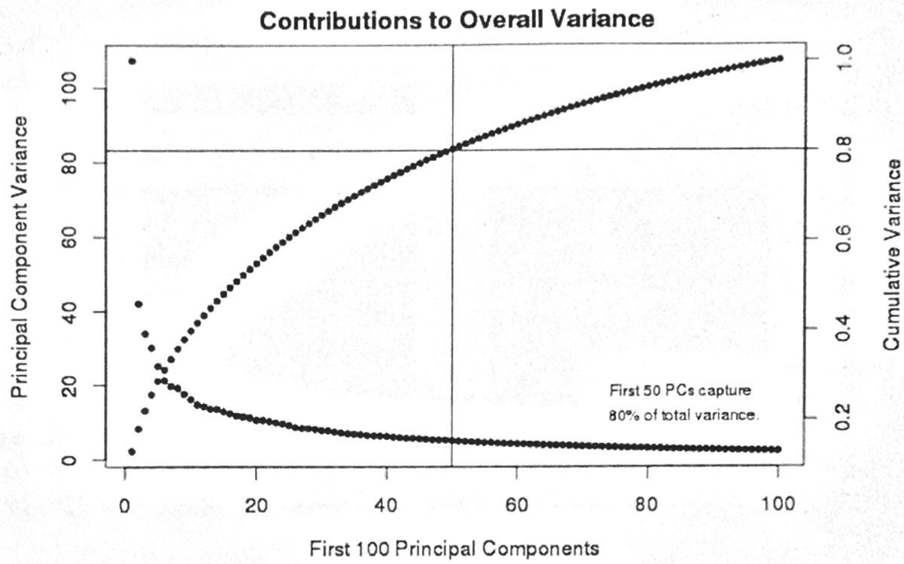
Using "congress109.csv" and "congress109members.csv", project data onto first principal component (i.e. get vector $Z = Yv_1$, where v_1 is the first column of the right-singular matrix of $Y = U\Sigma V'$). Then, merge with data about people, and identify relationships. A `pairs.panels` output is displayed.



The first principal component appears to be most strongly related to `chamber`, where a higher z is associated with being in the smaller chamber (presumed to be the "senate", versus the "house").

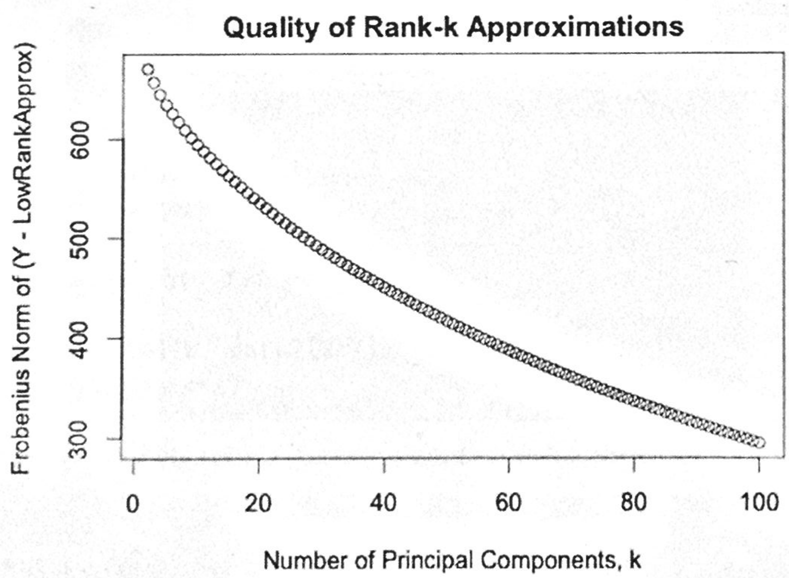
3.2 Rank k Approximation. Which k?

Each principal component accounts for a portion of the overall variance. What should k be for a rank-k approximation? One heuristic is to choose k so that the cumulative sum of those k variances is 80% of the total variance. The following graph shows that for this data set, $k = 50$.



3.3 Comparing Lower-Rank Approximations Using Frobenius Norm

The Frobenius norm can measure how close the rank-k approximation is to the original data matrix. Results are shown below, demonstrating that including more principal components generally improves "closeness" to the original data.



3.4 Full R Code

```
# StatMod2 - Latent Feature Models (PCA)

library(Matrix)
library(psych)

data <- read.csv("congress109.csv")
data2 <- read.csv("congress109members.csv")

# Remove column of names.
Y <- as.matrix(data[,-1])
n <- dim(Y)[1]
p <- dim(Y)[2]
names(Y) <- seq(1:dim(Y)[2]) # Enables compact printing.
# Standardize columns of Y to be mean=0, sd=1.
for (c in 1:p) {
  Y[,c] <- (Y[,c] - mean(Y[,c]))/sqrt(var(Y[,c]))
}

# Do SVD of Y.
decomp.y <- svd(Y)
U.y <- as.matrix(decomp.y$u)
D.y <- as.matrix(diag(decomp.y$d))
V.y <- as.matrix(decomp.y$v)

# Do eigenvalue decomposition of S=(1/n)*t(Y)*Y
S1 <- cov(Y)
S <- (1/n)*t(Y)%*%Y
colnames(S) <- seq(1:dim(S)[2]) # Enables compact printing.
rownames(S) <- seq(1:dim(S)[2])
decomp.s <- eigen(S)
vals.s <- decomp.s$values
vecs.s <- decomp.s$vectors

# Project all values onto 1st column of V.
z <- Y%*%vecs.s[,1]
pc1.in.context <- cbind(z, data2[2:7])
pairs.panels(pc1.in.context)

# Test variance of projections using first 100 columns of V.
range <- 1:100
t <- NULL
cumulative.vars <- NULL
for (i in range) {
  t <- c(t, var(Y%*%vecs.s[,i]))
  cumulative.vars[i] <- sum(t[1:i])
```

```
}
cumulative.vars.scaled <- cumulative.vars/max(cumulative.vars)
pc.cutoff <- min(which(cumulative.vars.scaled>0.8))

# Plot cumulative variance.
par(mar = c(5,5,2,5))
plot(range, t, ylab="Principal Component Variance",
     xlab=bquote("First"~.(length(range))~"Principal Components"),
     main="Contributions to Overall Variance")
text(85, 15, cex=.75,
     bquote(atop("First"~.(pc.cutoff)~"PCs capture",
                 "  80% of total variance.")))
par(new = T)
plot(range, cumulative.vars.scaled,
     col = "blue", axes = F, xlab = NA, ylab = NA)
axis(side = 4)
mtext(side = 4, line = 3, "Cumulative Variance")
abline(h=0.8)
abline(v=pc.cutoff)

# Get a lower-rank approximation of Y, using the top eigenvalues that describe
# 80% of the total variance.
LowRankApprox <- function(U.y, D.y, V.y, pc.cutoff) {
  lr.U <- U.y[,1:pc.cutoff]
  lr.D <- D.y[1:pc.cutoff,1:pc.cutoff]
  lr.V <- V.y[,1:pc.cutoff]
  lr.Y <- lr.U%*%lr.D%*%t(lr.V)
  return (lr.Y)
}

# Compute difference between original and low-rank approximation of Y.
frob.dist <- NULL
for (c in 2:100) {
  lr.approx <- LowRankApprox(U.y, D.y, V.y, c)
  frob.dist[c] <- norm(Y-lr.approx)
}
plot(2:100, frob.dist[2:100], xlab="Number of Principal Components, k",
     ylab="Frobenius Norm of (Y - LowRankApprox)",
     main="Quality of Rank-k Approximations")
```

StatMod 2 - Exe 5 - PCA

MD

4 Show induction argument for principal components $2, \dots, K$.

w_2 maximizes variance after accounting for w_1, and is the right-singular vector of Y, corresponding to the second largest singular value, d_2.

Given $z_i = y_i^T w_1$, all z_i's are $\underline{Z} = Y w_1$, and residuals are $R = Y - Z w_1^T$.

Residuals can alternatively be defined as the variance of projections of Y onto w_2, such that $w_2 \perp w_1$ and $w_2^T w_2 = 1$, the space in the orthogonal complement of $C(w_1)$ onto the unit vector w_2. Again, Lagrangian multipliers are used.

$$L(w_2) = w_2^T S w_2 - \lambda_2 (w_2^T w_2 - 1) - \lambda_1 (w_2^T w_1 - 0)$$

Unit vector w_2 — Orthogonal to earlier vector

$$\frac{\partial L}{\partial w_2} = 2 S w_2 - 2 \lambda_2 w_2 - \lambda_1 w_1 = 0$$

Note: Left-multiply by w_1^T.

$$\Rightarrow 2 w_1^T S w_2 - 2\lambda_2 \underbrace{w_1^T w_2}_{\text{Equals } 0,\text{ by assumption } w_1 \perp w_2} - \lambda_1 \underbrace{w_1^T w_1}_{\text{Equals } 1,\text{ by assumption } w_1^T w_1 = 1} = 0$$

$$2 w_1^T S w_2 - \lambda_1 = 0 \Rightarrow 2 w_2^T S w_1 - \lambda_1 = 0$$

$$\Rightarrow 2 w_2^T (\lambda_1 w_1) - \lambda_1 = 0 \Rightarrow 2 \lambda_1 w_2^T w_1 - \lambda_1 = 0$$

$$\Rightarrow 0 = \lambda_1$$

$$\Rightarrow S w_2 = \lambda_2 w_2$$

Thus, projection variance $w_2^T S w_2 = w_2^T \lambda_2 w_2 = \lambda_2 w_2^T w_2$ is maximized when $w_2^T w_2 = 1$ and with λ_2 and w_2 as the second-largest eigenvalue and eigenvec of S, respectively.

$\boxed{4}$ (continued) Step K. Use similar Lagrangian setup.

$$L(w_k) = w_k^T S w_k - \underbrace{\lambda_k(w_k^T w_k - 1)}_{\uparrow\, w_k \text{ as unit vector}} - \underbrace{\lambda_{k-1}(w_k^T w_{k-1} - 0) - \ldots - \lambda_1(w_k^T w_1 - 0)}_{\uparrow\, w_k \text{ as orthogonal to all previous vectors}}$$

$$\frac{\partial L}{\partial w_k} = 2Sw_k - 2\lambda_k w_k - \lambda_{k-1} w_{k-1} - \ldots - \lambda_1 w_1 = 0$$

Note: Left-multiply by w_{k-1}^T.

$$\Rightarrow 2 w_{k-1}^T S w_k - 2\lambda_k w_{k-1}^T w_k - \lambda_{k-1} w_{k-1}^T w_{k-1} - \ldots - \lambda_1 w_{k-1}^T w_1 = 0$$

$$\Rightarrow 2 w_k^T S w_{k-1} - 0 - \lambda_{k-1} \cdot I - 0 = 0$$

$$\Rightarrow 2 w_k^T (\lambda_{k-1} w_{k-1}) = \lambda_{k-1}$$

$$\Rightarrow 2\lambda_{k-1} \underbrace{w_k^T w_{k-1}}_{=0} = \lambda_{k-1} \quad \Rightarrow \quad 0 = \lambda_{k-1}$$

$$\Rightarrow 2Sw_k - 2\lambda_k w_k - \underbrace{0 - \ldots - 0}_{\text{Induction step demonstrates that all previous } \lambda\text{'s} = 0.} = 0$$

$$\Rightarrow Sw_k = \lambda_k w_k$$

Thus, projection variance $w_k^T S w_k = w_k^T \lambda_k w_k = \lambda_k w_k^T w_k$ is maximized when $w_k^T w_k = 1$ and with λ_k and w_k as the k^{th} largest eigenvalue and eigen vector of S, respectively.

S