Lecture 05. Principal Components Analysis (Sections 12.2-3&Section 6.3)

Ping Yu

HKU Business School
The University of Hong Kong

History of PCA

Karl Pearson (1857-1936), UCL

Harold Hotelling (1895-1973), Columbia and UNC-Chapel Hill[1]

[1]Kenneth Arrow was supervised by him at Columbia!

Principal Components Analysis

(Section 12.2)

What Are Principal Components?

- PCA produces a low-dimensional representation of a dataset with a large set of correlated variables. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- The first principal component of a set of features $X_1, X_2, \cdots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

  that has the largest variance. By normalized, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- We refer to the elements $\phi_{11}, \cdots, \phi_{p1}$ as the loading of the first principal component; together, the loadings make up the first principal component loading vector, $\phi_1 = (\phi_{11}, \cdots, \phi_{p1})^T$, or the first principal component direction.
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

PCA: Example

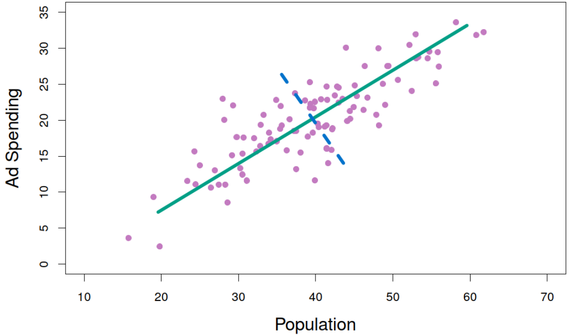


FIGURE 6.14. *The population size (*`pop`*) and ad spending (*`ad`*) for* 100 *different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

Computation of Principal Components

- Suppose we have a $n \times p$ data set $\mathbf{X}$. Since we are only interested in variance, we always assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero below (i.e., the column means of $\mathbf{X}$ are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \cdots + \phi_{p1} x_{ip}$$

  for $i = 1, 2, \cdots, n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- Since $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, so does $\frac{1}{n}\sum_{i=1}^{n} z_{i1}$ (for any ϕ_1). Hence the sample variance of $\{z_{i1}\}_{i=1}^{n}$ can be written as $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$.
- In other words, the first PC loading vector solves the optimization problem

$$\max_{\phi_{11},\cdots\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

- We refer to $z_{11}, \cdots, z_{n1}$ and $Z_1 := (z_{11}, \cdots, z_{n1})^T$ as the scores and score vector of the first PC.[2]

[2]We are abusing notations here: Z_1 represents both the population random variable and the sample vector for the first PC.

Further Principal Components

- The second PC is the linear combination of $X_1, X_2, \cdots, X_p$ that has maximal variance among all linear combinations that are uncorrelated with Z_1.
- The second principal component scores $z_{12}, \cdots, z_{n2}$ take the form

$$z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{p2} x_{ip},$$

  where ϕ_2 is the second PC loading vector, with elements $\phi_{12}, \cdots, \phi_{p2}$.
- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1. So the optimization problem for the second PC is

$$\begin{array}{c} \max_{\phi_{12},\cdots\phi_{p2}} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j2} x_{ij} \right)^2 \\ \text{subject to } \sum_{j=1}^{p} \phi_{j2}^2 =: \|\phi_2\|^2 = 1, \\ \text{and } \sum_{j=1}^{p} \phi_{j2}\phi_{j1} =: \phi_2^T \phi_1 = 0. \end{array}$$

- How about the third PC? ϕ_3 satisfies $\|\phi_3\|^2 = 1$, $\phi_3^T \phi_1 = 0$, and $\phi_3^T \phi_2 = 0$.
- There are at most $\text{rank}(\mathbf{X}) = \min(n-1, p)$ PCs.

(**) Technicalities

- The optimization problems in PCA can be solved via a Singular Value Decomposition (SVD) of the matrix $\mathbf{X}$, standard in linear algebra.
- From SVD, $\mathbf{X} = \mathbf{UDV}^T$, where $\mathbf{U} := [\mathbf{u}_1, \cdots, \mathbf{u}_p] \in \mathbb{R}^{n \times p}$ such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$, $\mathbf{V} := [\mathbf{v}_1, \cdots, \mathbf{v}_p] \in \mathbb{R}^{p \times p}$ such that $\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$, and $\mathbf{D} = \text{diag}\{d_1, d_2, \cdots, d_p\}$ with $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ called the singular values of $\mathbf{X}$.[3]
- If $\mathbf{Z} := [Z_1, \cdots, Z_p] = \mathbf{X}\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} = [\phi_1, \cdots, \phi_p]$, then PCA requires

$$\text{Var}(\mathbf{Z}) = \text{Var}(\mathbf{X}\boldsymbol{\Phi}) = \boldsymbol{\Phi}^T \text{Var}(\mathbf{X})\boldsymbol{\Phi} = \boldsymbol{\Phi}^T \frac{\mathbf{X}^T\mathbf{X}}{n}\boldsymbol{\Phi} = \boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_p\}$$

  with $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$.
- So $\phi_1, \cdots, \phi_p$ must be the eigenvectors of $\text{Var}(\mathbf{X})$, which are also $\mathbf{v}_1, \cdots, \mathbf{v}_p$.
  - Why? $\frac{\mathbf{X}^T\mathbf{X}}{n} = \frac{1}{n}\mathbf{VDU}^T\mathbf{UDV}^T = \mathbf{V}\frac{\mathbf{D}^2}{n}\mathbf{V}^T$, so $\boldsymbol{\Phi} = \mathbf{V}$ and $\boldsymbol{\Lambda} = \frac{\mathbf{D}^2}{n} = \text{diag}\left\{\frac{d_1^2}{n}, \cdots, \frac{d_p^2}{n}\right\}$.
- Now,

$$\begin{aligned} Z_1 &= \mathbf{X}\mathbf{v}_1 = \mathbf{UDV}^T\mathbf{v}_1 = [\mathbf{u}_1, \cdots, \mathbf{u}_p]\,\text{diag}\{d_1, \cdots, d_p\}\,[\mathbf{v}_1, \cdots, \mathbf{v}_p]^T\mathbf{v}_1 \\ &= [\mathbf{u}_1, \cdots, \mathbf{u}_p]\,\text{diag}\{d_1, \cdots, d_p\}\,\mathbf{e}_1 = \mathbf{u}_1 d_1, \end{aligned}$$

  where $\mathbf{e}_1$ is the first base vector of $\mathbb{R}^p$.
  - Similarly, $Z_j = \mathbf{X}\mathbf{v}_j = \mathbf{u}_j d_j$.

[3] If $\mathbf{X}$ is full column rank, then $d_p > 0$.

Geometry of PCA

- The loading vector $\phi_1 = \left(\phi_{11}, \cdots, \phi_{p1}\right)$ defines a direction in feature space along which the data vary the most.
- If we project the n data points $x_1, ..., x_n$ onto this direction, the projected values are the principal component scores $z_{11}, \cdots, z_{n1}$ themselves.
- Can you tell what are ϕ_1 and $\{z_{i1}\}_{i=1}^n$ in Figure 6.14?
  - $\phi_{11} = 0.839$, and $\phi_{21} = 0.544$, so

$$Z_1 = 0.839 \times \left(\text{pop} - \overline{\text{pop}}\right) + 0.544 \times \left(\text{ad} - \overline{\text{ad}}\right).$$

- In Figure 6.14, ϕ_2 has only one choice since $p = 2$: $\phi_{12} = 0.544$ and $\phi_{22} = -0.839$.
- Generally, we can plot Z_1 against Z_2, Z_1 against Z_3 etc.; this amounts to projecting the original data onto the subspace spanned by ϕ_1, ϕ_2, and ϕ_3, and plotting the projected points.

Illustration: USArrests Data

- USArrests data: For each of the fifty states in the United States, the data set contains the number of arrests per $100,000$ residents for each of three crimes: Assault, Murder, and Rape. We also record UrbanPop (the percent of the population in each state living in urban areas).
- The PC score vectors have length $n = 50$, and the PC loading vectors have length $p = 4$.
- PCA was performed after standardizing each variable to have mean zero and standard deviation one.
- In Figure 12.1, the blue state names represent the scores for the first two PCs.
  - Mathematically, $(Z_1, Z_2) \in \mathbb{R}^{50\times 2}$.
- The orange arrows indicate the first two PC loading vectors (with axes on the top and right). For example, the loading for Rape on the first PC is 0.54, and its loading on the second PC 0.17 [the word Rape is centered at the point $(0.54, 0.17)$].
  - Mathematically, $(\phi_1, \phi_2) \in \mathbb{R}^{4\times 2}$.
- This figure is known as a biplot, because it displays both the PC scores and the PC loadings.

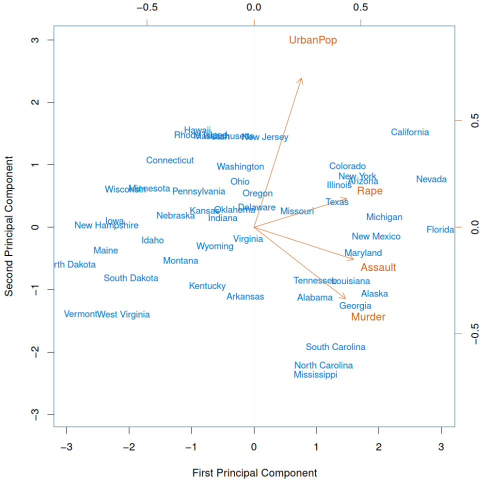


FIGURE 12.1. *The first two principal components for the* USArrests *data.*

PCA Loadings

	PC1	PC2
Murder	0.5358995	−0.4181809
Assault	0.5831836	−0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

TABLE 12.1. *The principal component loading vectors, ϕ_1 and ϕ_2, for the* USArrests *data. These are also displayed in Figure 12.1.*

- ϕ_1 places approximately equal weight on Assault, Murder, and Rape, but with much less weight on UrbanPop, so it is roughly a measure of overall rates of serious crimes. [refer to Figure 12.1]
- ϕ_2 places most of its weight on UrbanPop, but much less weight on the other three features, so it is roughly a measure of urbanization. [refer to Figure 12.1]
- In summary, PCA groups **X** into two uncorrelated parts of information, one corresponds to crime and the other corresponds to urbanization.
- Now, crimes rates: California, Nevada and Florida – high, North Dakota – low; urbanization: California – high, Mississippi – low; Indiana – average in both.

Another Interpretation of Principal Components

- The PC directions are directions in feature space along which the original data are highly variable.
- These directions also define lines and subspaces that are as close as possible to the data cloud.
- Specifically, the first PC loading vector defines the line in p-dimensional space that is closest to the n observations (using average squared Euclidean distance as a measure of closeness). [see Figure 6.15]
- The notion of PCs as the dimensions that are closest to the n observations extends beyond just the first PC.
- For instance, the first two PCs of a data set span the plane that is closest to the n observations, in terms of average squared Euclidean distance. [see Figure 12.2]

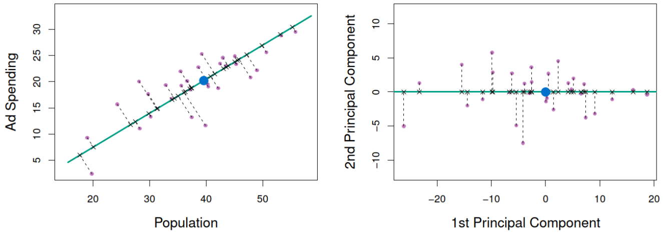
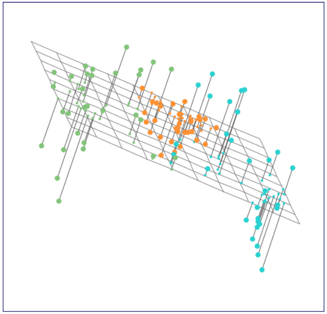


FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all n of the observations. The distances from each observation to the principal component are represented using the black dashed line segments. The blue dot represents* $(\overline{\texttt{pop}}, \overline{\texttt{ad}})$. Right: *The left-hand panel has been rotated so that the first principal component direction coincides with the x-axis.*



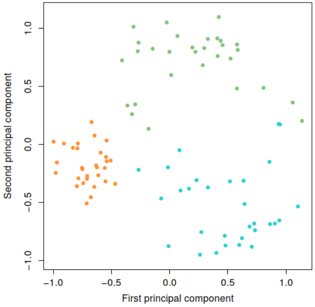


FIGURE 12.2. *Ninety observations simulated in three dimensions. The observations are displayed in color for ease of visualization.* Left: *the first two principal component directions span the plane that best fits the data. The plane is positioned to minimize the sum of squared distances to each point.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane.*

Technicalities

- Generally, x_{ij} can be approximated by M PCs:

$$x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm}.$$

- $\{z_m\}_{m=1}^M$ and $\{\phi_m\}_{m=1}^M$ can be found by the following minimization problem:

$$\begin{aligned} &\min_{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}} \sum_{j=1}^{p}\sum_{i=1}^{n}\left(x_{ij}-\sum_{m=1}^{M}a_{im}b_{jm}\right)^2 \\ = &\min_{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}} \text{trace}\left[\left(\mathbf{X}-\mathbf{A}\mathbf{B}^T\right)\left(\mathbf{X}-\mathbf{A}\mathbf{B}^T\right)^T\right], \end{aligned}$$

  where $\mathbf{A}=(a_{im})$, $\mathbf{B}=(b_{jm})$, and $\{b_m\}_{m=1}^M$ are normalized as $\|b_m\|^2=1$ for any m, and $b_m^Tb_k=0$ for any $m\neq k$.
  - (*) Unique up to a sign flip: no effect on the problem if $(\phi_m, z_m)\to(-\phi_m,-z_m)$; intuitively, ϕ_m and $-\phi_m$ spans the same line, and Var$(Z_m)=$Var$(-Z_m)$.
- (*) When $M=\min(n-1,p)$, we get a perfect fit: $x_{ij}=\sum_{m=1}^M z_{im}\phi_{jm}$.

The Proportion of Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.
- The total variance present in a data set is defined as

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2,$$

  and the variance explained by the *m*th PC is

$$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2.$$

- It can be shown that $\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(Z_m)$ with $M = \min(n-1, p)$.

Continued

- Therefore, the PVE of the mth PC is given by the positive quantity in $(0,1)$:

$$\mathrm{PVE}_m = \frac{\sum_{i=1}^n z_{im}^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2}.$$

- $\sum_{m=1}^{\min(n-1,p)} \mathrm{PVE}_m = 1$, so we also display the cumulative PVEs. [see Figure 12.3]
- It can be shown that

$$\underbrace{\sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n x_{ij}^2}_{\text{Var. of data}} = \underbrace{\sum_{m=1}^M \frac{1}{n} \sum_{i=1}^n z_{im}^2}_{\text{Var. of first } M \text{ PCs}} + \underbrace{\frac{1}{n} \sum_{j=1}^p \sum_{i=1}^n \left(x_{ij} - \sum_{m=1}^M z_{im}\phi_{jm} \right)^2}_{\text{MSE of } M\text{-dimensioanl approximation}}.$$

- Since the LHS is fixed, maximizing the variance and minimizing the approximation error are equivalent.
- It is also obvious from this decomposition that

$$\sum_{m=1}^M \mathrm{PVE}_m = 1 - \frac{\sum_{j=1}^p \sum_{i=1}^n \left(x_{ij} - \sum_{m=1}^M z_{im}\phi_{jm} \right)^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}},$$

  i.e., a kind of R^2, but different from least squares, we are now explaining p (instead of 1) columns using (the same unknown) M columns.

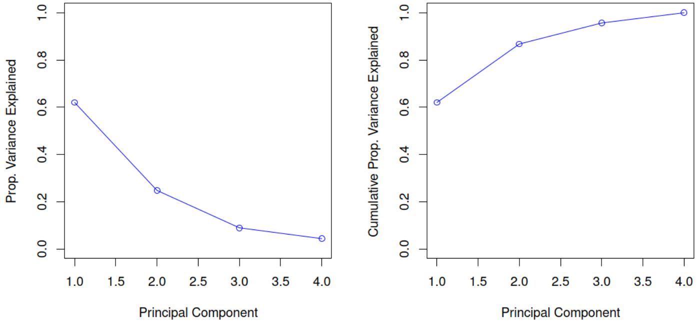


FIGURE 12.3. Left: *a scree plot depicting the proportion of variance explained by each of the four principal components in the* USArrests *data.* Right: *the cumulative proportion of variance explained by the four principal components in the* USArrests *data.*

- $\text{PVE}_1 = 62.0\%$, $\text{PVE}_2 = 24.7\%$, so $\text{PVE}_1 + \text{PVE}_2 = 87\%$.
- The left panel is known as a scree plot.[4]

[4] Scree means a mass of small loose stones that form or cover a slope on a mountain.

Principal Components Regression (Section 6.3.1)

- After the first M PCs are obtained, we can fit the linear regression model

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \varepsilon_i,\ i = 1, \cdots, n,$$

  using least squares, where

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j.$$

  - Different from subset selection and shrinkage methods where the original predictors, $X_1, \cdots, X_p$, are used, we here use transformed variables to fit least squares.
  - Because this approach reduces the dimension of unknown parameters from $p+1$ to $M+1$ with $M < p$, it is referred to as a dimension reduction method.
- PCR assumes that the directions in which $X_1, \cdots, X_p$ show the most variation are also the directions that are associated with Y.
- PCR is not a feature selection (but a feature averaging) method since all original features are used in Z_m.
  - (*) In this sense, PCR is more closely related to ridge regression than to the lasso, and the ridge regression can be treated as a continuous version of PCR. [figure here]

Comparison of PCR and RR

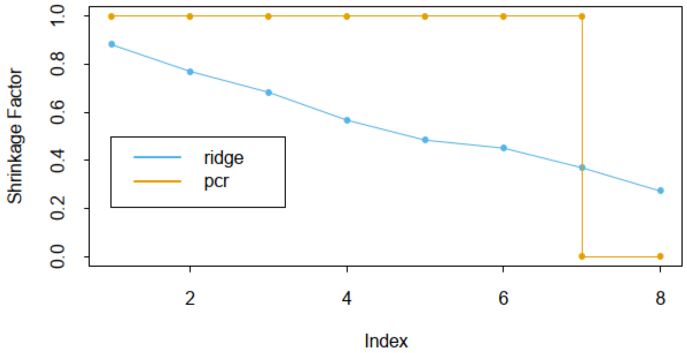


Figure: RR shrinks the regression coefficients of the PC while PCR truncates them.

(**) Continued: Technicalities

- Given the SVD of $\mathbf{X}$, $\mathbf{X} = \mathbf{UDV}^T$, it is not hard to see that the fitted values of OLS is

$$\hat{\mathbf{y}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{UDV}^T\left(\mathbf{VDU}^T\mathbf{UDV}^T\right)^{-1}\mathbf{VDU}^T\mathbf{y} = \mathbf{UU}^T\mathbf{y},$$

  where $\mathbf{U}^T\mathbf{y} = \left[\mathbf{u}_1^T\mathbf{y}, \cdots, \mathbf{u}_p^T\mathbf{y}\right]^T$ is the coordinates of $\mathbf{y}$ projected on the basis vectors $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_p]$.
- Now, the fitted values of RR is

$$\begin{aligned}\mathbf{X}\hat{\beta}_\lambda^R &= \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{UD}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{DU}^T\mathbf{y} \\ &= \sum_{j=1}^{p}\mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y},\end{aligned}$$

  so RR shrinks the OLS coordinate of $\mathbf{u}_j$ by a factor $d_j^2/\left(d_j^2+\lambda\right) \le 1$.
  - A greater amount of shrinkage is applied to the coordinates of basis vectors with smaller d_j^2.

Constrained Least Squares Interpretation

- Note that

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij},$$

  where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}, \tag{1}$$

  so PCR can be treated as a constrained least squares regression since β_j must take the form of (1).

- The constraints (1) would bias the coefficient estimates, but may significantly reduce the variance of fitted coefficients especially when p is large relative to n.
  - If $M = p$, and all the Z_m are linear independent, then the PCR fitting is equivalent to the original least squares.

- PCR mitigates overfitting by estimating only $M \ll p$ coefficients.

Application to PCR

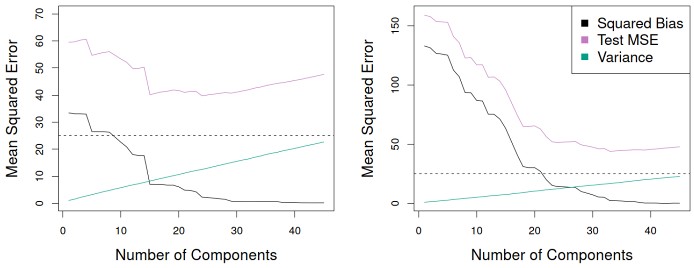


FIGURE 6.18. *PCR was applied to two simulated data sets. In each panel, the horizontal dashed line represents the irreducible error.* Left: *Simulated data from Figure 6.8.* Right: *Simulated data from Figure 6.9.*

- In the simulation, $n = 50$, and $p = 45$. y in the first dataset involves all predictors, but in the second dataset only two.
- PCR performs better than LS ($M = 45$) especially in the first dataset, but worse than the ridge or lasso regression.
- PCR tends to do well when the first few PCs are sufficient to capture most of the variation in X_j's as well as their relationship with Y.

A Design Favorable to PCR

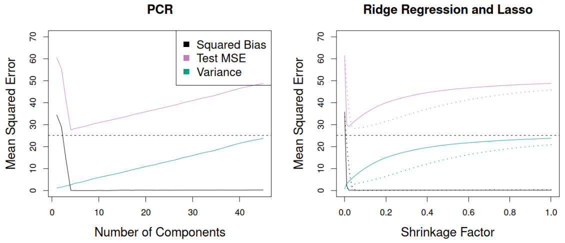


FIGURE 6.19. *PCR, ridge regression, and the lasso were applied to a simulated data set in which the first five principal components of X contain all the information about the response Y. In each panel, the irreducible error $Var(\epsilon)$ is shown as a horizontal dashed line.* Left: *Results for PCR.* Right: *Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the ℓ_2 norm of the shrunken coefficient estimates divided by the ℓ_2 norm of the least squares estimate.*

- Compared with Figure 6.18, the bias of PCR drops to zero rapidly with M and the MSE displays a clear minimum at $M = 5$.
- PCR, ridge and lasso all perform better than LS with PCR and ridge slightly outperforming lasso.

Choosing the Number of Directions M

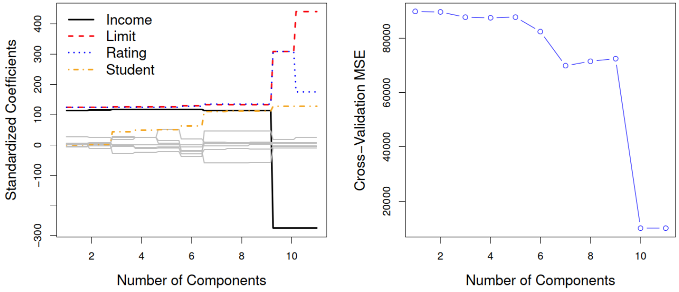


FIGURE 6.20. Left: *PCR standardized coefficient estimates on the* Credit *data set for different values of M.* Right: *The ten-fold cross-validation MSE obtained using PCR, as a function of M.*

- The Credit dataset contains credit card debt information for 10,000 customers.
- The CV $M = 10$, almost no dimension reduction given that $p = 11$.
- The left panel reports four β_j's in (1) as a function of M, where "standardized" is discussed below.

Partial Least Squares (Section 6.3.2)

- PCR identifies linear combinations, or directions, that best represent the predictors $X_1, \cdots, X_p$.
- These directions are identified in an unsupervised way, since the response Y is not used to help determine the principal component directions.
- That is, the response does not supervise the identification of the principal components.
- Consequently, PCR suffers from a potentially serious drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.
- Here, we introduce an alternative of PCR – PLS.

History of PLS

Herman Wold (1908-1992), Uppsala U of Sweden[5]

[5]He is also famous for the Cramér–Wold theorem characterizing the normal distribution and the Wold decomposition in time series analysis.

Continued

- Like PCR, PLS is a dimension reduction method, which first identifies a new set of features $Z_1, \cdots, Z_M$ that are linear combinations of the original features, and then fits a linear model via OLS using these M new features.
- But unlike PCR, PLS identifies these new features in a supervised way – that is, it makes use of the response Y in order to identify new features that not only approximate the old features well, but also that are related to the response.
- Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.
- The details of PLS are described in the next slide.
- Here, we explain some notations in the algorithm:

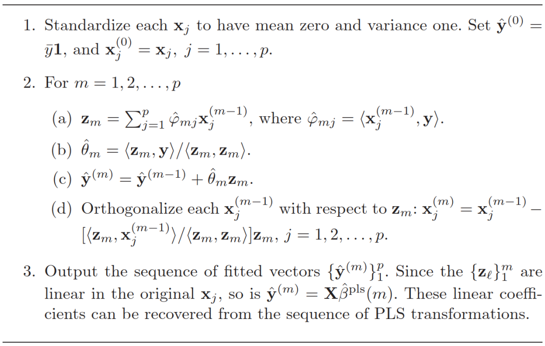
$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p), \mathbf{y} = (y_1, \cdots, y_n)^T, \mathbf{1} = (1, 1, \cdots, 1)^T,$$

  i.e., $\mathbf{x}_j$ is the jth column of $\mathbf{X}$, $\mathbf{y}$ collects all n observations of the response Y, and $\mathbf{1}$ is an $n \times 1$ column of ones;

$$\langle \mathbf{x}_j, \mathbf{x}_k \rangle := \mathbf{x}_j^T \mathbf{x}_k := \sum_{i=1}^{n} x_{ij} x_{ik}$$

  is the Euclidean inner product between $\mathbf{x}_j$ and $\mathbf{x}_k$.
- PLS assigns weights on $\mathbf{x}_j$ not solely based $\mathbf{X}$ as in PCR, but places the highest weight on the variables that are most strongly related to $\mathbf{y}$.

[Algorithm] Partial Least Squares



1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j$, $j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots, p$
   (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y}\rangle$.
   (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y}\rangle / \langle \mathbf{z}_m, \mathbf{z}_m\rangle$.
   (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m\mathbf{z}_m$.
   (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)}\rangle / \langle \mathbf{z}_m, \mathbf{z}_m\rangle]\mathbf{z}_m$, $j = 1, 2, \ldots, p$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

- (*) Note that in Step 2(d), although the mean of $\mathbf{x}_j^{(m)}$ is zero, we did not normalize its variance to be one. As a result, $\hat{\varphi}_{mj}$ in Step 2(a) is not the OLS coefficient in regressing $\mathbf{y}$ on $\mathbf{x}_j^{(m-1)}$, but the covariance between them. Anyway, $\hat{\varphi}_{mj}$ for any j is the scaled coefficient (with the same scale) in regressing $\mathbf{x}_j^{(m-1)}$ on $\mathbf{y}$ and this scale will not affect $\widehat{\theta}_m\mathbf{z}_m$.
- The ultimate number of $\mathbf{z}_m$ used in PLS, M, can be chosen by CV as in PCR.

[Example] Sales

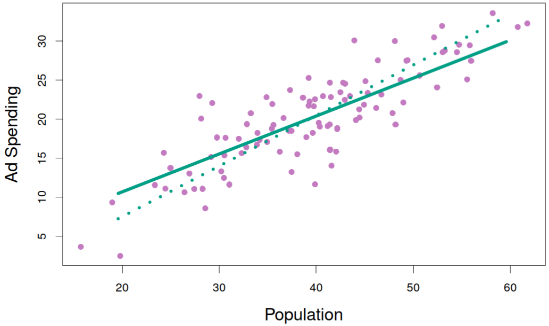


FIGURE 6.21. *For the advertising data, the first PLS direction (solid line) and first PCR direction (dotted line) are shown.*

- $\mathbf{z}_1$ in PLS puts a larger (relative) weight on pop than in PCR, implying pop is more highly correlated with the sales than is ad.

A Comparison of the Selection and Shrinkage Methods

- In the following two slides, we compare the performance of six methods we have learned until now in a simple setup.
- There are only two features, X_1 and X_2, with correlation $\rho = 0.5$ or -0.5, $\beta_1 = 4$, and $\beta_2 = 2$.
- The tuning parameters for ridge and lasso vary over a continuous range, while best subset, PLS and PCR take just two discrete steps to the LS solution.
- When $\rho = 0.5$, ridge shrinks the coefficients together until it finally converges to LS. PLS and PCR show similar behavior to ridge, although are discrete and more extreme. Best subset overshoots the solution and then backtracks. The behavior of the lasso is intermediate to the other methods.
- When $\rho = -0.5$, again PLS and PCR roughly track the ridge path, while all of the methods are more similar to one another.
- PLS often performs no better than ridge regression or PCR – while the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance.
  - (*) Like ridge and PCR, PLS also tends to shrink the low-variance directions, but can actually inflate some of the higher variance directions, which makes it less stable.

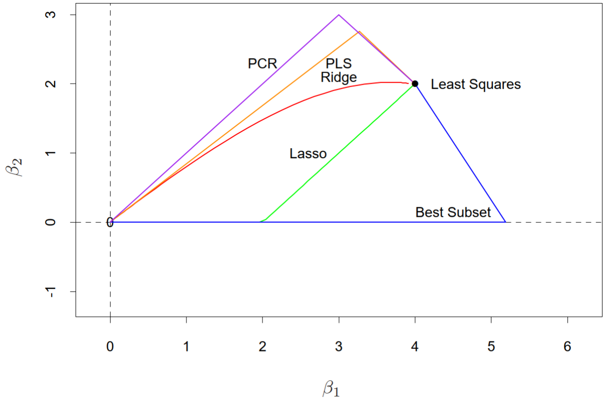


Figure: Coefficient profiles from different methods for a simple problem: two inputs with correlation $\rho = 0.5$, and the true regression coefficients $\beta = (4,2)^T$.

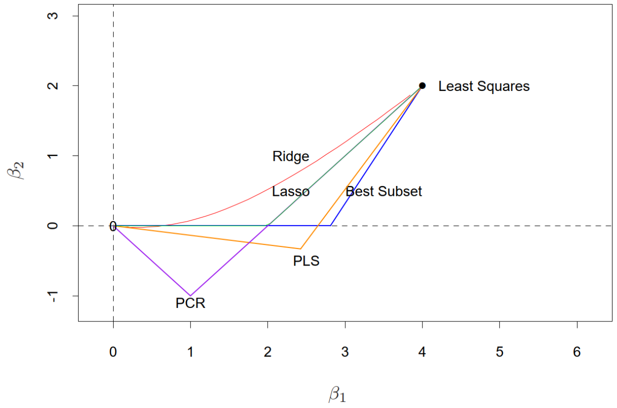


Figure: $\rho = -0.5$

- To summarize, PLS, PCR and ridge regression tend to behave similarly. Ridge regression may be preferred because it shrinks smoothly, rather than in discrete steps. Lasso falls somewhere between ridge regression and best subset regression, and enjoys some of the properties of each.

Scaling the Variables

- Principal components are sensitive to scaling of **X**, so it is recommended to first standardize the columns of **X** to have mean 0 and variance 1. [see Figure 12.4]
  - If they are in the same units (say, kilograms, or inches), you might or might not scale the variables.

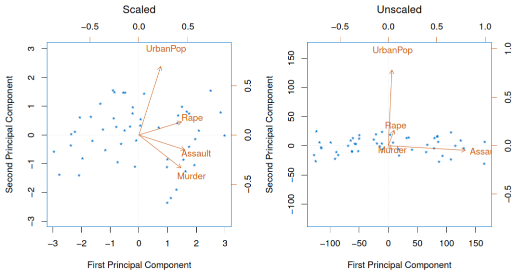


FIGURE 12.4. *Two principal component biplots for the* `USArrests` *data.* Left: *the same as Figure 12.1, with the variables scaled to have unit standard deviations.* Right: *principal components using unscaled data.* `Assault` *has by far the largest loading on the first principal component because it has the highest variance among the four variables. In general, scaling the variables to have standard deviation one is recommended.*

Deciding How Many PCs to Use

- We would like to use the smallest number of PCs required to get a good understanding of the data.
- No single (or simple) answer to this question, as cross-validation is not available for this purpose.
  - Why not? No object like Y is supervised!
  - When could we use cross-validation to select the number of components? Think about PCR.
- The scree plot in Figure 12.3 can be used as a guide: we look for an “elbow”.
  - In this example, $M = 2$ seems enough.
  - Quite subjective!
- In practice, we can try the first few PCs to check whether there are interesting patterns; if YES, then continue until NO; if NO, then stop directly.

Missing Values and Matrix Completion

(Section 12.3)

Missing Values

- The missing values phenomenon is everywhere, e.g., the USArrests data contain 20 (out of 200) missing values.
  - Missing can be necessary: a customer of Netflix cannot watch all movies in its catalog; actually, most data of ratings (1 to 5) are missing [see the example below].
- Possible Solutions:
  - Remove the rows containing missing values: too wasteful, and unrealistic if the number of missing rows is too big.
  - Replace the missing x_{ij} by the mean of jth column (using non-missing entries): does not exploit the correlation with other columns (i.e., other variables).
- A process called matrix completion, which is based on PCA, is more successful in imputing the missing values.
  - Key Assumption: missing at random (MAR), e.g., the missing of a patient's weight is due to a low battery of the scale rather than that the patient is too heavy.
- The completed matrix can then be used in statistical learning as in other lectures, e.g., a completed rating matrix can be used to recommend unseen movies to a customer.

PCA with Missing Values

- We first approximate the observed data by M PCs:

$$\min_{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}\left\{\sum_{(i,j)\in\mathscr{O}}\left(x_{ij}-\sum_{m=1}^{M}a_{im}b_{jm}\right)^2\right\}, \tag{2}$$

  where $\mathscr{O}$ is the set of indices for all observed data, so $\text{card}(\mathscr{O})<np$ with $\text{card}(\cdot)$ meaning cardinality of a set.
- We can then approximate the missing x_{ij} by

$$\hat{x}_{ij}=\sum_{m=1}^{M}\hat{a}_{im}\hat{b}_{jm}.$$

- It is hard to solve (2) exactly, but the following iterative "hard-impute" algorithm works out at least a local optimum.
  - Note that (12.14) checks the fitting of observed data, and it will decrease in each iteration [Why? In (12.13), $\sum_{j=1}^{p}\sum_{i=1}^{n}=\sum_{(i,j)\in\mathscr{O}}+\sum_{(i,j)\notin\mathscr{O}}$. If use the $\hat{a}_{im}$ and $\hat{b}_{jm}$ in the previous step, $\sum_{(i,j)\notin\mathscr{O}}=0$ and $\sum_{(i,j)\in\mathscr{O}}$ remains the same. Since we minimize (12.13), $\sum_{j=1}^{p}\sum_{i=1}^{n}$ is smaller while $\sum_{(i,j)\notin\mathscr{O}}>0$, so $\sum_{(i,j)\in\mathscr{O}}$ must be smaller].

Algorithm 12.1 *Iterative Algorithm for Matrix Completion*

1. Create a complete data matrix $\tilde{\mathbf{X}}$ of dimension $n \times p$ of which the (i,j) element equals

$$\tilde{x}_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in \mathcal{O} \\ \bar{x}_j & \text{if } (i,j) \notin \mathcal{O}, \end{cases}$$

   where $\bar{x}_j$ is the average of the observed values for the jth variable in the incomplete data matrix $\mathbf{X}$. Here, $\mathcal{O}$ indexes the observations that are observed in $\mathbf{X}$.

2. Repeat steps (a)–(c) until the objective (12.14) fails to decrease:

   (a) Solve

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times M},\mathbf{B}\in\mathbb{R}^{p\times M}}{\text{minimize}} \left\{ \sum_{j=1}^{p}\sum_{i=1}^{n}\left(\tilde{x}_{ij} - \sum_{m=1}^{M} a_{im}b_{jm}\right)^2 \right\} \qquad (12.13)$$

   by computing the principal components of $\tilde{\mathbf{X}}$.

   (b) For each element $(i,j) \notin \mathcal{O}$, set $\tilde{x}_{ij} \leftarrow \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm}$.

   (c) Compute the objective

$$\sum_{(i,j)\in\mathcal{O}}\left(x_{ij} - \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm}\right)^2. \qquad (12.14)$$

3. Return the estimated missing entries $\tilde{x}_{ij}$, $(i,j) \notin \mathcal{O}$.

(**) Regularization Using Nuclear Norm

- Like RR or lasso, it is better to regularize a_{im} and b_{jm} when $\dim(\mathbf{X})$ is large.
- We can restrict the rank of the completed matrix, but the rank of a matrix is like the number of nonzeros in a vector, i.e., ℓ_0 norm, so it is better to consider its convex relaxation, the nuclear norm, which can be treated as the ℓ_1 norm of a matrix.
- The nuclear norm of a matrix $\mathbf{A}$, $\|\mathbf{A}\|_N$, is the sum of its singular values.
- The matrix completion problem reduces to

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_N \text{ subject to } \mathbf{Z} \text{ and } \mathbf{X} \text{ share the same known entries.}$$

- Candès and Recht (2009) show that for an $n \times n$ matrix of rank r, if the number of known entries $K > Cn^{5/4} r \log n$ for some positive constant C, then with high probability, $\mathbf{X}$ can be perfectly recovered.
- In algorithm, we solve the following minimization problem as in Mazumder et al. (2010):

$$\min_{\mathbf{Z}} \frac{1}{2} \left\|(\mathbf{X}-\mathbf{Z})_{\mathscr{O}}\right\|_F^2 + \lambda \|\mathbf{Z}\|_N, \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm or Euclidean norm of a matrix, and the subscript $\mathscr{O}$ means we consider only the observed entries.
- Using matrix notation, (2) can be re-expressed as

$$\min_{\mathbf{A},\mathbf{B}} \frac{1}{2} \left\|\left(\mathbf{X}-\mathbf{A}\mathbf{B}^T\right)_{\mathscr{O}}\right\|_F^2. \tag{4}$$

(**) Relationship with (4)

- Like RR, we can regularize **A** and **B** in (4),
$$\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\left\|\left(\mathbf{X}-\mathbf{A}\mathbf{B}^T\right)_{\mathscr{O}}\right\|_F^2+\frac{\lambda}{2}\left(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2\right), \quad (5)$$
resulting in the maximum-margin factorization of Srebro et al. (2005).
- If **X** is fully known, and we restrict **Z** in (3) to have rank r, then the solution to (3) is
$$\widehat{\mathbf{Z}}=\mathbf{U}_r S_\lambda\left(\mathbf{D}_r\right)\mathbf{V}_r^T,$$
where $\mathbf{U}_r$, $\mathbf{D}_r$ and $\mathbf{V}_r$ are inherited from the SVD of **X** but retain only the first r rows and/or columns, and $S_\lambda\left(\mathbf{D}_r\right)=\text{diag}\left\{(d_1-\lambda)_+,\cdots,(d_r-\lambda)_+\right\}$.
- This solution is the same as the solution to (5), i.e., $\widehat{\mathbf{Z}}=\widehat{\mathbf{A}}\widehat{\mathbf{B}}^T$, where
$$\widehat{\mathbf{A}}=\mathbf{U}_r S_\lambda\left(\mathbf{D}_r\right)^{1/2} \text{ and } \widehat{\mathbf{B}}=\mathbf{V}_r S_\lambda\left(\mathbf{D}_r\right)^{1/2}.$$
- When **X** is not fully known, we can complete it as in Algorithm 12.1.
  - Specifically, we estimate **A** and **B** alternately (because (5) is convex in **A** and **B** separately but not jointly) like in RR, and then complete **X** as $\mathbf{X}_{\mathscr{O}}+(\widehat{\mathbf{A}}\widehat{\mathbf{B}}^T)_{\overline{\mathscr{O}}}=(\mathbf{X}-\widehat{\mathbf{A}}\widehat{\mathbf{B}}^T)_{\mathscr{O}}+\widehat{\mathbf{A}}\widehat{\mathbf{B}}^T$, which produces an efficient sparse plus low rank representation of **X**, where the subscript $\overline{\mathscr{O}}$ means entries unobserved.
  - This is the Soft-impute Alternating Least Squares algorithm of Hastie et al. (2015).

[Example] USArrests

- $n = 50$ (states), $p = 4$, and discard 20 observations by randomly choose 20 states and then randomly discard one variable for each chosen state, so $\text{card}(\mathscr{O}) = 180$.
  - Standardize the data with mean 0 and variance 1 before discarding.
- Apply Algorithm 12.1 with $M = 1$.
- Figure 12.5 shows that the imputed values match the original values very well.
- In 100 random runs, the average correlation between the true and imputed values is 0.63, with a sd 0.11.
- If we estimate the 20 missing values using the first PC from the complete data (rather than incomplete data as in Algorithm 12.1) in each run, then the correlation is 0.79, with a sd 0.08, which is the infeasible benchmark for comparison.
- Figure 12.6 shows more details of the comparison: the left is the average imputed $\hat{a}_{i1}$ (with a sd bar) against true $\hat{a}_{i1}$, $i = 1, \cdots, 50$, and the right is the average imputed $\hat{b}_{j1}$ (with a sd bar) against true $\hat{b}_{j1}$, $j = 1, \cdots, 4$.
- Comments: (i) Since $p = 4$ is small, it is hard to borrow information from other variables, which is why we discard only one variable per state and set $M = 1$. (ii) How to choose M? We can randomly leave out a few observed elements as the validation set to see which M performs the best.

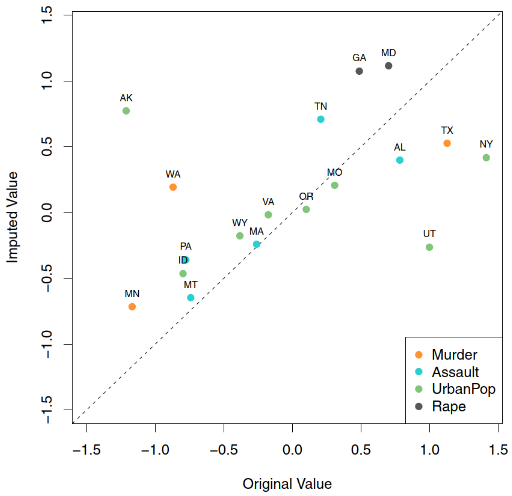


FIGURE 12.5. *Missing value imputation on the* USArrests *data.*

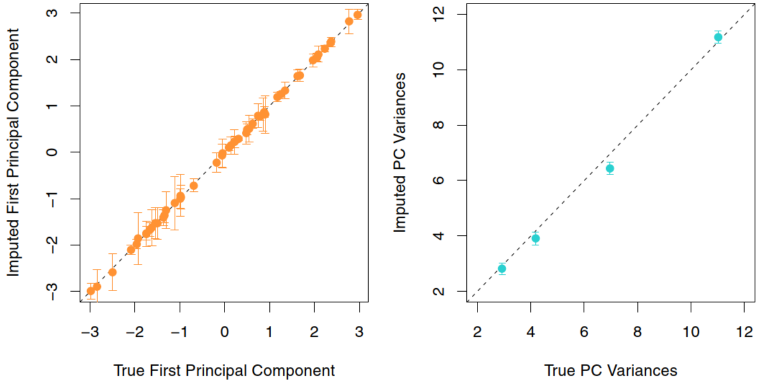
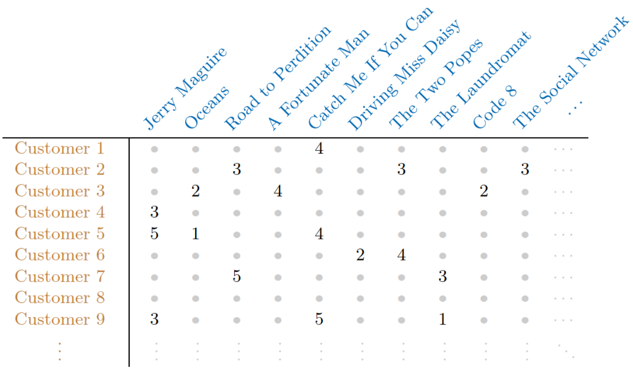


FIGURE 12.6. *As described in the text, in each of 100 trials, we left out 20 elements of the* USArrests *dataset. In each trial, we applied Algorithm 12.1 with* $M = 1$ *to impute the missing elements and compute the principal components.*

[Example] Recommender Systems of Netflix



	Jerry Maguire	Oceans	Road to Perdition	A Fortunate Man	Catch Me If You Can	Driving Miss Daisy	The Two Popes	The Laundromat	Code 8	The Social Network	...
Customer 1	●	●	●	●	4	●	●	●	●	●	...
Customer 2	●	●	3	●	●	●	3	●	●	3	...
Customer 3	●	2	●	4	●	●	●	●	2	●	...
Customer 4	3	●	●	●	●	●	●	●	●	●	...
Customer 5	5	1	●	●	4	●	●	●	●	●	...
Customer 6	●	●	●	●	●	2	4	●	●	●	...
Customer 7	●	●	5	●	●	●	●	3	●	●	...
Customer 8	●	●	●	●	●	●	●	●	●	●	...
Customer 9	3	●	●	●	5	●	●	1	●	●	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

TABLE 12.2. *Excerpt of the Netflix movie rating data.*

- $n = 480,189$, $p = 17,770$, and each rating between 1 and 5. Averagely, each customer watches 200 movies, so $1 - 200/17,770 = 99\%$ data are missing.
- We can use Algorithm 12.1 to impute $\hat{x}_{ij} = \sum_{m=1}^{M} \hat{a}_{im}\hat{b}_{jm}$, where $\hat{a}_{im}$ is the strength with which the ith customer belongs to the mth clique (a group of customers that enjoys movies of the mth genre), and $\hat{b}_{jm}$ is the strength with which the jth movie belongs to the mth genre.

The Netflix Prize

- Competition started in October 2006. Training data are as above.
- The objective is to predict the rating for a set of 2,817,131 customer-movie pairs that are missing in the training data.
- Netflix's original algorithm Cinematch achieved a root MSE of 0.9525. The first team to achieve a 10% improvement wins one million dollars.
- The leaderboard

▸ Netflix Prize leaderboard

- BellKor's Pragmatic Chaos wins, beating The Ensemble by a narrow margin.

Conclusions

- Unsupervised learning is important for understanding the variation and grouping structure of a set of unlabeled data, and can be a useful pre-processor for supervised learning.
- It is intrinsically more difficult than supervised learning because there is no gold standard (like an outcome variable) and no single objective (like test set accuracy).
- It is an active field of research, with many recently developed tools such as self-organizing maps, independent components analysis and spectral clustering; see Chapter 14 of ESL.

Lab: Principal Components Analysis

(Section 12.5.1-2 and 6.5.3)

- Principal Components Analysis
- Matrix Completion
- PCR and PLS Regression