PS5841

Data Science in Finance & Insurance

Dimension Reduction

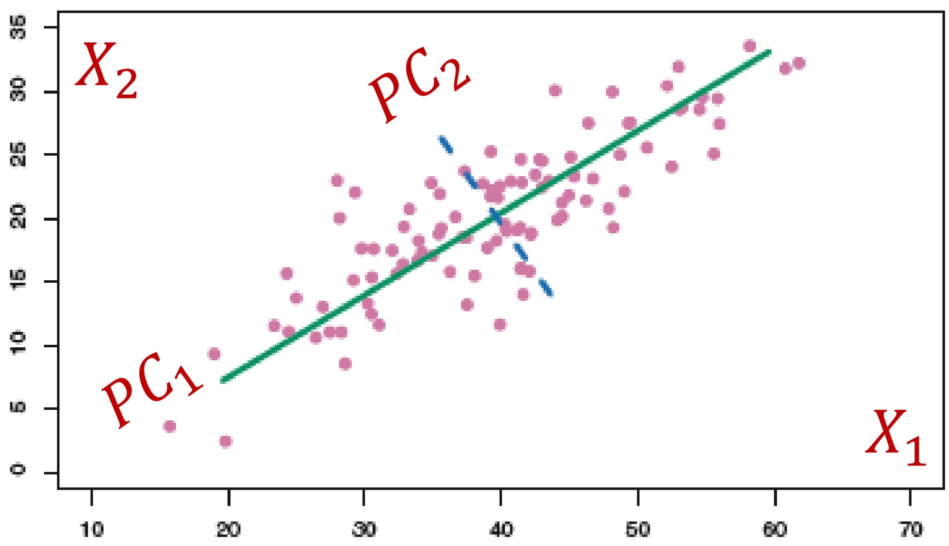
Yubo Wang

Autumn 2022

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Principal Component Analysis

At most $\min(n-1, p)$ principal components



notations

scores loadings

$$\underset{n\times q}{\boldsymbol{Z}} = \underset{n\times p}{\boldsymbol{X}}\underset{p\times q}{\boldsymbol{\Phi}}$$

$$(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_q) = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_p)(\boldsymbol{\Phi}_1, \dots, \boldsymbol{\Phi}_q)$$

$$\begin{bmatrix} z_{11} & \cdots & z_{1q} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nq} \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \phi_{11} & \cdots & \phi_{1q} \\ \vdots & \ddots & \vdots \\ \phi_{p1} & \cdots & \phi_{pq} \end{bmatrix}$$

$$\boldsymbol{X}_j = (x_{1j}, \dots, x_{nj})^T, \qquad \boldsymbol{\Sigma} = Var(\boldsymbol{X}), \qquad \boldsymbol{\Phi}_k = (\phi_{1k}, \dots, \phi_{pk})^T$$

$$\boldsymbol{Z}_k = (z_{1k}, \dots, z_{nk})^T = \sum_{j=1}^{p} \phi_{jk}\boldsymbol{X}_j = \phi_{1k}\boldsymbol{X}_1 + \cdots + \phi_{pk}\boldsymbol{X}_p$$

$$z_{ik} = \sum_{j=1}^{p} x_{ij}\phi_{jk} = \phi_{1k}x_{i1} + \cdots + \phi_{pk}x_{ip}$$

$$\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I}$$

Principal Components

$$\max_{\boldsymbol{\Phi}_k} Var(\boldsymbol{Z}_k) = \max_{\boldsymbol{\Phi}_k} \boldsymbol{\Phi}_k^T \boldsymbol{\Sigma} \boldsymbol{\Phi}_k$$

such that

$$\boldsymbol{\Phi}_k^T \boldsymbol{\Phi}_k = 1$$

$$Cov(\boldsymbol{Z}_{k'}, \boldsymbol{Z}_k) = \boldsymbol{\Phi}_{k'}^T \boldsymbol{\Sigma} \boldsymbol{\Phi}_k = 0, \qquad k' = k-1, \ldots, 1$$

- Solution:

$\boldsymbol{\Phi}_k = \boldsymbol{e}_k$, the k-th eigenvector of $\boldsymbol{\Sigma}$

$Var(\boldsymbol{Z}_k) = \lambda_k$, the k-th eigenvalue of $\boldsymbol{\Sigma}$

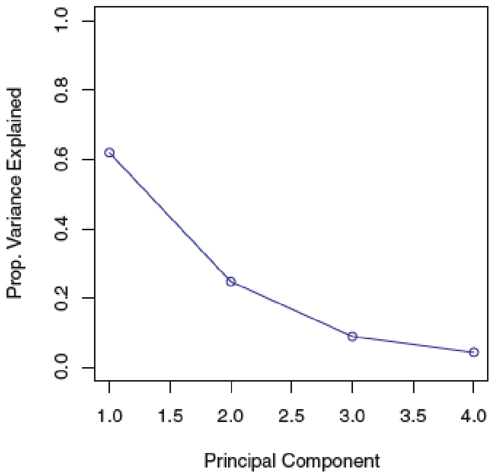
- In practice, use sample variance

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-1} \boldsymbol{X}_{centered}^T \boldsymbol{X}_{centered}$$

Proportion of Variance Explained

- Total Variance of $\boldsymbol{X}$

$$trace(\boldsymbol{\Sigma}) = \sum_{j=1}^{p} \lambda_j$$



- PVE

$$PVE = \frac{Var(\boldsymbol{Z}_k)}{trace(\boldsymbol{\Sigma})} = \frac{\lambda_k}{\sum_{j=1}^{p} \lambda_j}$$

Uniqueness

- Loading vectors are unique up to a sign flip
  - sign flip does not alter the coordinate system
- Score vectors are unique up to a sign flip
  - Variance of $\boldsymbol{Z}_k$ and $-\boldsymbol{Z}_k$ are the same
- But the right sign may improve interpretability

Coding (PCA)

- R

  base::prcomp()

- Python

  sklearn.decomposition.PCA()

Dimension Reduction

- From p predictors to $M < p$ predictors
  - Standardizing the predictors necessary to have predictors on the same scale

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

$$= \theta_0 + \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} x_{ij}\phi_{jm} + \varepsilon_i$$

$$= \theta_0 + \sum_{m=1}^{M} \sum_{j=1}^{p} x_{ij}\phi_{jm}\theta_m + \varepsilon_i$$

$$= \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

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

$$\beta_0 = \theta_0$$

~ a representation of the original regression
recall $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Phi}^T$

Principal Components Regression (PCR)

- A small number of PCs may be able to explain most of the variability in data, as well as the relationship with the response (no guarantee)
- Assumption: the directions in which the predictors show the most variation are the direction that are associated with the response
- Not a variable selection method
  - Each PC is a linear combination of all original predictors

Coding (PCR)

- R

  pls::pcr()

- Python

  sklearn.decomposition.PCA()

  sklearn.linear_model.LineaRegression()

Partial Least Squares Regression (PLS)

- $\boldsymbol{Z}_1 = (z_{11}, \dots, z_{n1})^T = \sum_{j=1}^{p} \phi_{j1} \boldsymbol{X}_j$

where ϕ_{j1} is the SLR slope of Y on $\boldsymbol{X}_j$

- $\boldsymbol{Z}_2 = (z_{12}, \dots, z_{n2})^T = \sum_{j=1}^{p} \phi_{j2} \boldsymbol{X}_j$

where ϕ_{j1} is the SLR slope of the residual

(of Y on $\boldsymbol{Z}_1$) on $\boldsymbol{X}_j$

- …

Coding (PLS)

- R

  pls::plsr()

- Python

  sklearn.decomposition.PLSRegression()

That was