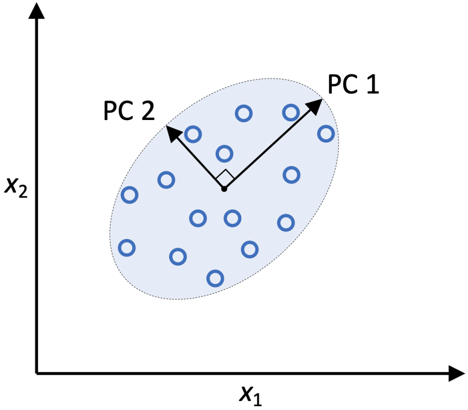
Principal Component Analysis

Marcin Kuta

Introduction

- linear transformation of the original variables into new, uncorrelated variables
- PCA finds the best low-dimensional representation that captures most of the variation

Principal Component Analysis

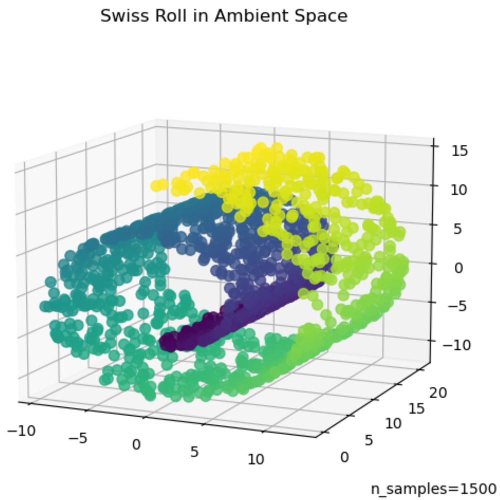


Assumptions

Assumptions

- correlation between features
- linear relationship between features
- no missing values
- not robust against outliers
- sensitive to the scale of features

Principal Component Analysis



Computing PCA

Algorithms for PCA

- Eigendecomposition of XX^T
- Singular Value Decomposition of X
- NIPALS
- ...

SVD is preferred over forming covariance matrix

- Singular Value Decomposition approach is more general
- SVD is numerically more stable
  Läuchli matrix

$$\begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$$

Eigendecompostion

Eigendecomposition of covariance matrix

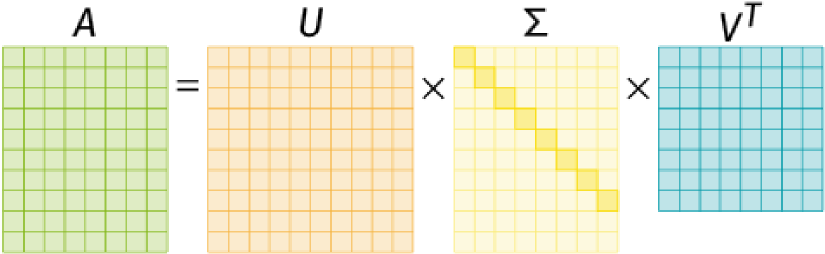
$$X^T X = W^T \Sigma^2 W \tag{1}$$

W is orthonormal (rotation) matrix
Σ is diagonal (scaling) matrix
Vectors of W are called eigenvectors
Values of Σ are called eigenvalues

Projection:

$$x' = \underbrace{\Sigma}_{\text{scaling}} \underbrace{W}_{\text{rotation}} x \tag{2}$$

Full Singular Value Decompostion



Applications of SVD

- Principal Component Analysis
- Latent Semantic Indexing
- Spectral clustering
- Word embeddings [1]
- Transformer networks (LightFormer, SIngularFormer)
  - decomposition of self-attention
- Data compression
- Noise removal
- Recommendation systems, market data analysis
- Numerical methods (total least squares, Lanczos method)

Full Singular Value Decompostion

$X \in \mathbb{R}^{n \times m}$

We assume X contains centered data.

$$\underset{n\times m}{X} = \underset{n\times n}{U} \; \underset{n\times m}{\Sigma} \; \underset{m\times m}{V^T} \tag{3}$$

U is an orthogonal matrix of left singular vectors

Σ is a diagonal matrix of singular values

V is an orthogonal matrix of right singular vectors

U and V are orthogonal matrices:

$U^T U = UU^T = I_n$

$V^T V = VV^T = I_m$

Reduced Singular Value Decompostion

$X \in \mathbb{R}^{n\times m}$

$$\underset{n\times m}{X} = \underset{n\times m}{U} \underset{m\times m}{\Sigma} \underset{m\times m}{V^T} \tag{4}$$

U and V are semi-orthogonal matrices:

$U^T U = I_m, \; UU^T = I_n$
$V^T V = VV^T = I_m$

Truncated Singular Value Decompostion

$X \in \mathbb{R}^{n \times m}$
$k < m$

$$\underset{n\times m}{X_k} = \underset{n\times k}{U_k}\ \underset{k\times k}{\Sigma_k}\ \underset{k\times m}{V_k^T} \tag{5}$$

U_k and V_k are semi-orthogonal matrices:

$U_k^T U_k = I_k,\ \ U_k U_k^T = I_n$
$V_k^T V_k = I_k,\ \ V_k V_k^T = I_m$

Principal Components Analysis

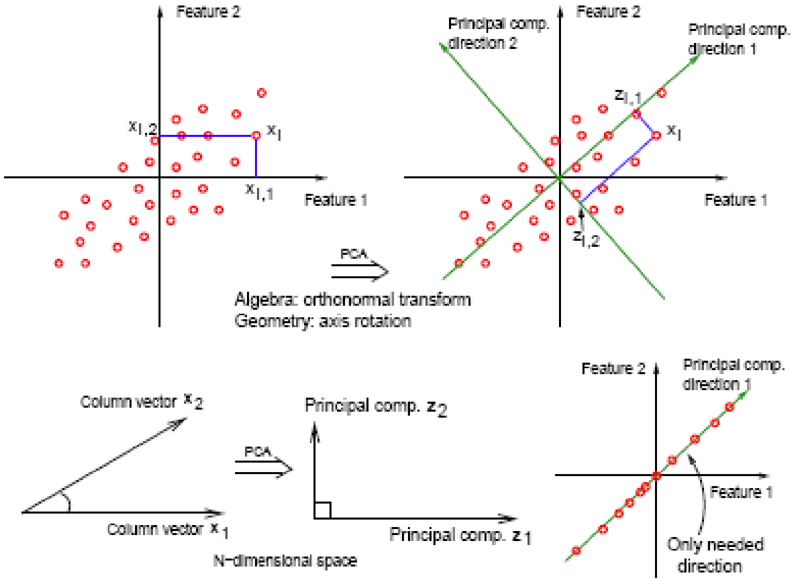
Principal Component Analysis is the result of the best k-rank SVD after centering the data.

$$X = \sum_{i=1}^{\text{rank}(X)} \sigma_i u_i v_i^T = \sum_{i=1}^{\text{rank}(X)} \sigma_i A_i \quad (6)$$

Rank-k approximation of X

$$\hat{X}_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = \sum_{i=1}^{k} \sigma_i A_i \quad (7)$$

Principal Components



Principal Components

Principal axes Columns of V are the *principal axes/directions (eigenvectors)* of X. They define the new coordinate system. Elements of principal axes are the cosines of rotation.

Principal Components (scores) Scores are the projections of the data on the principal axes. They define coordinates of data in the new basis.

$$\underset{n\times m}{Z} = \underset{n\times m}{X}\ \underset{m\times m}{V} \tag{8}$$

$$Z = U\Sigma V^T V = U\Sigma \tag{9}$$

Variables in Z are also called *latent variables*. Original data can be reconstructed from latent variables:

$$X = ZV^T \tag{10}$$

Principal Components

Loadings Columns of L are the PCA *loadings*.

$$L = \frac{1}{\sqrt{N-1}} V\Sigma \tag{11}$$

Singular values The diagonal of Σ contains positive values sorted from largest to smallest, called *singular values*.

Principal Component Analysis Learning

Algorithm 1: Learning the PCA model

Data: Training data $\{ x^{(i)} \}_{i=1}^{n}$
Result: Principal axes V and scores Z

Compute the mean vector: $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$
Center the data:
for $i = 1$ *to* n **do**
  $x^{(i)} \leftarrow x^{(i)} - \bar{x}$
Feature scaling
Construct the data matrix: $X = [x^{(1)} \dots x^{(n)}]^T$
Perform SVD on X to obtain factorization $X = U \Sigma V^T$
Compute principal components: $Z = U\Sigma$

Sample Covariance Matrix

Covariance matrix

$$\frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\bar{x})(x^{(i)}-\bar{x})^T = \frac{1}{n}X^TX = \tag{12}$$

$$= \frac{1}{n}V\Sigma^TU^TU\Sigma V^T = V(\frac{1}{n}\Sigma^T\Sigma)V^T = V\Lambda V^T$$

$\Lambda_{ii} = \sigma_i^2/n$

Explained Variance

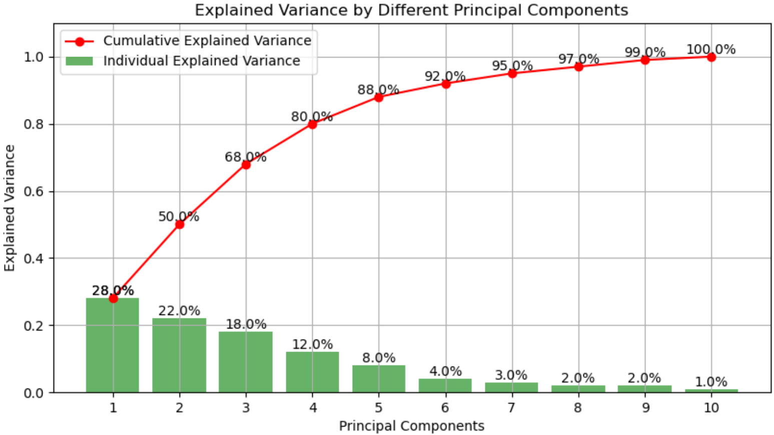
Cumulative explained variance

$$\frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_m}, \;\; k \in \{1, \ldots, m\} \tag{13}$$

Individual explained variance

$$\frac{\lambda_k}{\lambda_1 + \cdots + \lambda_m}, \tag{14}$$

Explained Variance



Biplot


CHAS
AGE
NOX
INDUS
B
RM
LSTAT
TAX
RAD
PTRATIO
CRIM
ZN
DIS
PC2
PC1
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
0.6

Variants of PCA

- Kernel PCA
- Incremental PCA
- Randomized PCA
  - PCA: $O(nm^2 + m^3)$
  - Randomized PCA: $O(nd^2 + d^3)$

References

[1] https://online.stat.psu.edu/stat508/book/export/html/639

[2] https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

[3] https://github.com/rasbt/machine-learning-book/tree/main/ch05

[1] O. Levy and Y. Goldberg.
Neural word embedding as implicit matrix factorization.
In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 2177–2185, 2014.