Rensselaer

why not change the world?®

Dimension Reduction (DR), Principal Component Analysis (PCA) Review

Ahmed Eleish

Data Analytics ITWS/MGMT 4600/6600 – CSCI/BCBP 4600

March 13th 2026

Tetherless World Constellation
Rensselaer Polytechnic Institute

Dimensionality Reduction (DR)

- Reducing the number of dimensions (variables, features) of a dataset while preserving the patterns

- Why?

    - Reduce data volume for storage/processing/transfer.

    - For easier exploration/visualization of high dimensional data

    - **Curse of dimensionality** – challenges of learning from (very) high-dimensional data, or datasets with a large number of dimensions relative to the number of observations*.

* These considerations are generalizations and are more narrowly defined per domain/dataset/analysis

Dimensionality Reduction Methods

- Linear:
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Factor Analysis (FA)
- Non-linear:
  - Uniform Manifold Approximation and Projection (UMAP)
  - Autoencoders (Neural Networks)
  - Kernel PCA

Principal Component Analysis (PCA)

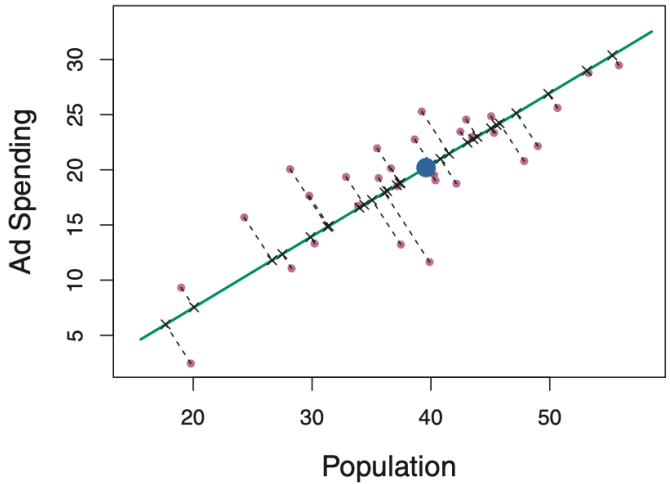
Dimensionality Reduction with PCA

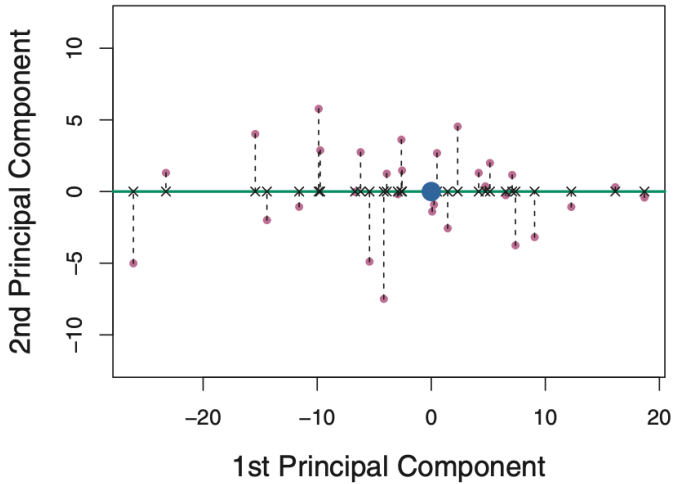
PCA is useful in many different scenarios, including:

- **Data exploration:** PCA can help to visualize high-dimensional data in a lower dimensional space.
- **Data compression:** PCA can reduce the number of variables in a dataset, making it easier to work with.
- **Feature selection:** PCA can help to identify the most important variables in a dataset.
- **Data pre-processing:** PCA can be used to remove noise from a dataset and to standardize variables so that they have a similar scale.
- **Downstream Machine learning:** PCA can be used as a pre-processing step before applying machine learning algorithms to a dataset, to improve their performance and reduce overfitting.

Principal Component Analysis

- The dataset is projected onto the Principal Components for visualization or further analysis:





Principal Components

- The principal components of a dataset are calculated by taking **linear combinations of the original variables** in such a way that *each component is orthogonal (uncorrelated) to the others*.

Finding the Principal Components

The **first principal component** of a set of features X1, X2, . . . , Xp is the normalized linear combination of the features that has the largest **variance**.

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p \qquad \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

We refer to the elements $\varphi_{11},\ldots,\varphi_{p1}$ as the loadings of the first principal component, φ_1

After the loading vectors for all principal components are obtained and combined into a matrix of loading vectors $(\varphi_1, \varphi_2, \ldots, \varphi_p)^T$ or *V*

The dataset can be transformed (projected/rotated) onto the new axes (PCs):

$$Z = XV$$

where *Z* is the transformed matrix, *X* is the original matrix, *V* is the matrix of loading vectors $(\varphi_1, \ldots \varphi_p)$

Finding the Principal Components

1) Center dataset: subtract column means from each row such that the mean of each variable is 0.

$$X_c = X - \bar{X}$$

Optionally standardize variables by dividing each variable by its standard deviation: $X_s = \frac{X_c}{SD(x)}$

2) Find covariance matrix: $C = \frac{1}{N-1} X_s * X_s$

3) Factorize the covariance matrix using eigen-decomposition to find the its eigenvectors and eigenvalues, analogous to factoring an integer into 2 integers, e.g. 12 = 3 * 4

$$C = V \Lambda V^T$$

C is a square matrix (the covariance matrix), ***V*** is a square matrix of *eigenvectors* and **Λ** is a diagonal matrix of *eigenvalues* ($\lambda_{1, \ldots,} \lambda_n$)

➔ *The (sorted) eigenvector matrix is equivalent to the loadings for the principal components of the original matrix*

4) To obtain the transformed/rotated data:

$$Z = XV$$

In-Class example

- PCA on Iris dataset.

https://rpi.box.com/s/xut9h86qwsehlry41dq3eagu3v5p5o1p

Thanks!