why not change the world?®

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

Ahmed Eleish

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

October 21st 2025

Tetherless World Constellation
Rensselaer Polytechnic Institute

Definitions

- **Dimensions of a dataset:** While a tabular dataset has two dimensions (or axes), which are its rows (vertical) and its columns (horizontal), the term *dimension,* when applied to a dataset most commonly refers to its columns, also called variables, attributes, or features. The two axes are often referenced as n x d, where n denotes the number of rows and d (sometimes p) denotes the number of dimensions,
  - These are analogous to the axes of a matrix (usually m x n).

- **Dimensionality Reduction (of a dataset):** the process of reducing the number of features of a dataset through some transformation that preserves the patters or structure in the data.

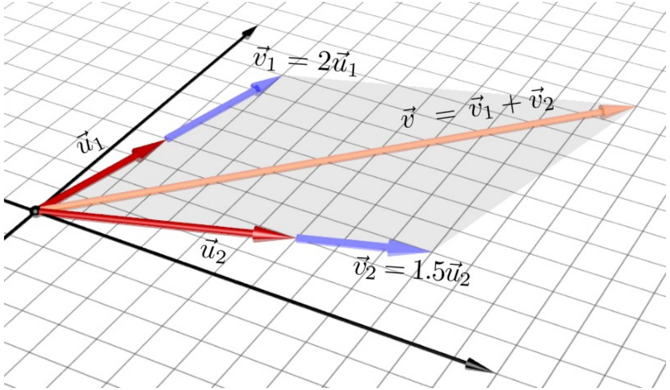
Definitions

- **Linear combination:** a mathematical expression where a set of terms of a vector are multiplied by a set of scalar constants and the results summed.

e.g.

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 + \cdots + a_n\mathbf{v}_n.$$

Where v_1 … v_n are vectors and a_1 … a_n are scalars



v is the linear combination of vectors u_1 and u_2 such that v = 2 * u_1 + 1.5 * u_2

https://en.wikipedia.org/wiki/Linear_combination

Dimensionality Reduction (DR)

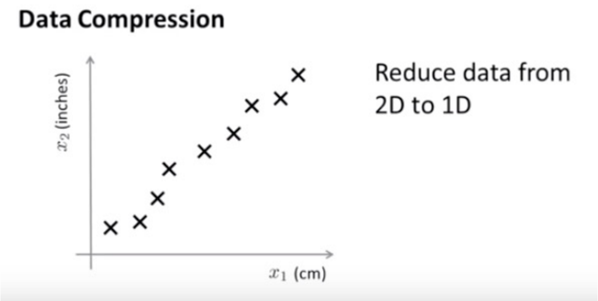
- There are multiple reasons that you want to do Dimensionality Reduction: one is to do compress data.

- Data compression not only saves memory space, it also speeds up learning algorithms.

Dimensionality Reduction (DR)

X1: distance measured in cm
X2: distance measured in inches

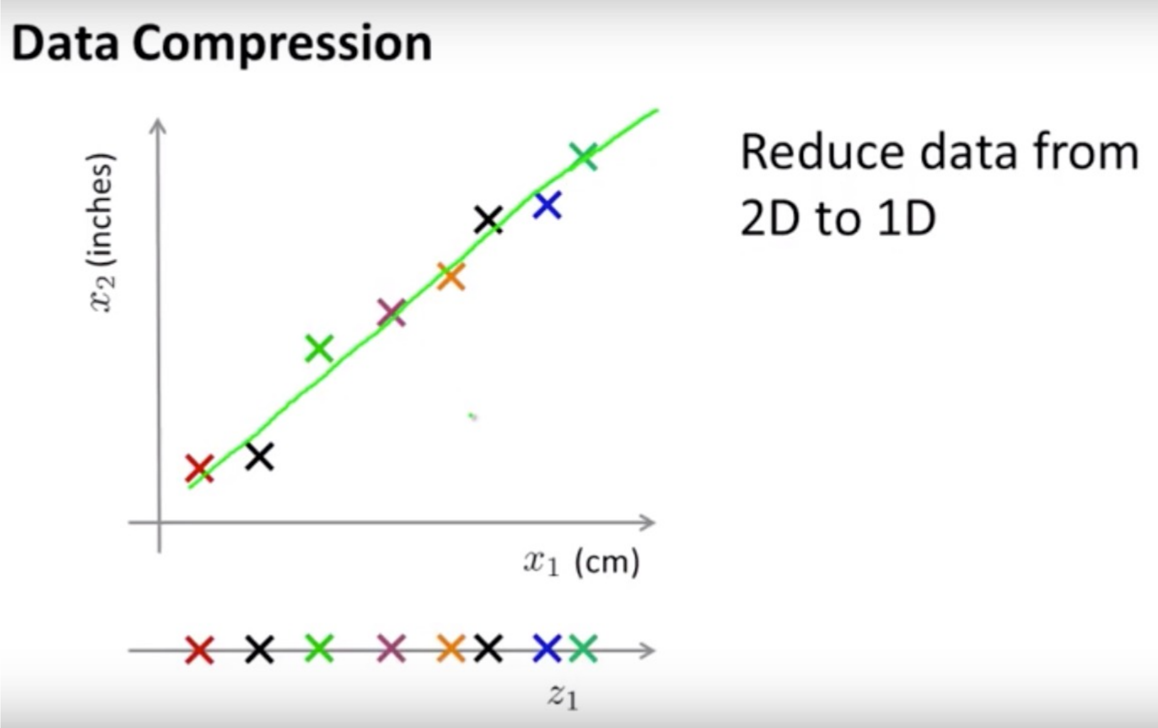
We want to reduce the data to one dimension



Length in cm is rounded off to the nearest cm and length in inches is rounded off to the nearest inch, that is why those examples do not perfectly lie on a straight line.

Image source; ML course Stanford University

Dimensionality Reduction (DR)



Dimensionality Reduction (DR)

- If we allow ourselves to approximate the original dataset by projecting all of the original examples onto the green line, then we need only one number to specify a point on the line.
- This way, we have reduced the problem from 2D to 1D.
- For this example this may not be a big deal, but if you have a dataset with large number of features with redundant information, it will take too much memory space and take more time to do the computations.
- In that case it's better to reduce the redundancy.

Dimensionality Reduction (DR)

- Let's imagine if you have hundreds of features, it is difficult to keep track of all those features of the dataset and sometimes we have redundant features such as the same measurement in both centimeters and inches like shown in the previous example.
- Therefore, DR used in feature selection, reduction
- Why?
    - **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

Methods:

- Principle component analysis (PCA)
- Singular Value Decomposition (SVD)

Principal Component Analysis (PCA)

Dimensionality Reduction with PCA

- Principal Component Analysis (PCA) is an Unsupervised Learning Technique.
- PCA is a popular approach for deriving a low-dimensional set of features from a large set of variables.
- A large part of the variation in the data can be explained in fewer variables called “Principal Components”.
- We will see how to implement PCA in R using the Iris dataset

Dimensionality Reduction with PCA

- (PCA) is a technique used to simplify a large and complex dataset by reducing its dimensionality while retaining as much information as possible.
- Imagine you have a large dataset with many variables (like age, height, weight, income, education level, etc.) for a large number of individuals.
- With so many variables, it can be difficult to understand the patterns and relationships between them.
- PCA can help by finding a smaller set of variables (called principal components) that explain the most variation in the data. In other words, it finds the most important aspects of the data that are responsible for most of its variation.

Covariance

- Variance and Covariance are a measure of the “spread” of a set of points around their center of mass (mean)
- **Variance** – measure of the deviation from the mean for points in one dimension e.g. heights
- **Covariance** as a measure of how much each of the dimensions vary from the mean with respect to each other.
- Covariance is measured between 2 dimensions to see if there is a relationship between the 2 dimensions e.g. number of hours studied & marks obtained.
- The covariance between one dimension and itself is the variance

Reference: https://en.wikipedia.org/wiki/Variance
https://en.wikipedia.org/wiki/Covariance

Variance

Fair die [edit]

A fair six-sided die can be modeled as a discrete random variable, X, with outcomes 1 through 6, each with equal probability 1/6. The expected value of X is $(1+2+3+4+5+6)/6 = 7/2$. Therefore, the variance of X is

$$\begin{aligned} \operatorname{Var}(X) &= \sum_{i=1}^{6} \frac{1}{6}\left(i - \frac{7}{2}\right)^2 \\ &= \frac{1}{6}\left((-5/2)^2 + (-3/2)^2 + (-1/2)^2 + (1/2)^2 + (3/2)^2 + (5/2)^2\right) \\ &= \frac{35}{12} \approx 2.92. \end{aligned}$$

https://en.wikipedia.org/wiki/Variance#Examples

Dimensionality Reduction with PCA

PCA is useful in many different scenarios, including:

- **Data exploration:** PCA can help you to visualize high- dimensional data in a lower dimensional space.
- **Data compression:** PCA can reduce the number of variables in a dataset, which can make 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

- These principal components are calculated by taking linear combinations of the original variables in such a way that each component is orthogonal (uncorrelated) to the others.
- Eliminating less significant principal components allows us to represent the data in a lower-dimensional space, which is easier to understand and analyze.

Principal Component Analysis

- Suppose that we wish to visualize *n* observations with measurements on a set of *p* features, *X1, X2, X3, ..., Xp* as a part of exploratory data analysis.

- We could do this by examining two-dimensional scatterplots of the data, which contains the n observations' measurements on two of the features, However, there are $C_2^p = \frac{p(p-1)}{2}$ of such scatterplots, for example with *p=10*, there are 45 plots!

- If p is large, then it will certainly not be possible to look at all of them. Moreover, most likely, many of them will not be informative since they each contain just a small fraction of the total information present in the dataset.

Principal Component Analysis

- We'll now explain the mathematics of PCA:
- 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.

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

By normalized, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. We refer to the elements φ11,...,φp1 as the loadings of the first principal component; together, the loadings make up the principal component loading vector, φ1 = (φ11 φ21 ... φp1)^T.

- Given a *n* × *p* dataset *X,*
  - Center the data (column means of *X* become zero)
  - We then look for the linear combination of the sample feature values of the form:

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

that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$

Principal Component Analysis

- To get the 1st PC, solve the optimization problem

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

$$\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$$

The objective that we are maximizing in is just the sample variance of the n values of z_{i1}

- We refer to *z11, . . . , zn1* as the scores of the first principal component.
- The above optimization problem can be solved via an **eigen decomposition***, a standard technique in linear algebra.
- After the first principal component *Z1* of the features has been determined, we can find the second principal component *Z2* the linear combination of *X1, . . . , Xp* that has maximal variance out of all linear combinations that are uncorrelated with *Z1*.
- The second principal component scores *z12, z22, ..., zn2* take the form:

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

Principal Component Analysis

- Steps:

1) Center dataset: subtract column means from each row such that the mean of each variable is 0: $X_c = X - \bar{x}$. Optionally scale variables by dividing each variable by its variance: $X_s = \frac{X_c}{var(x)}$

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

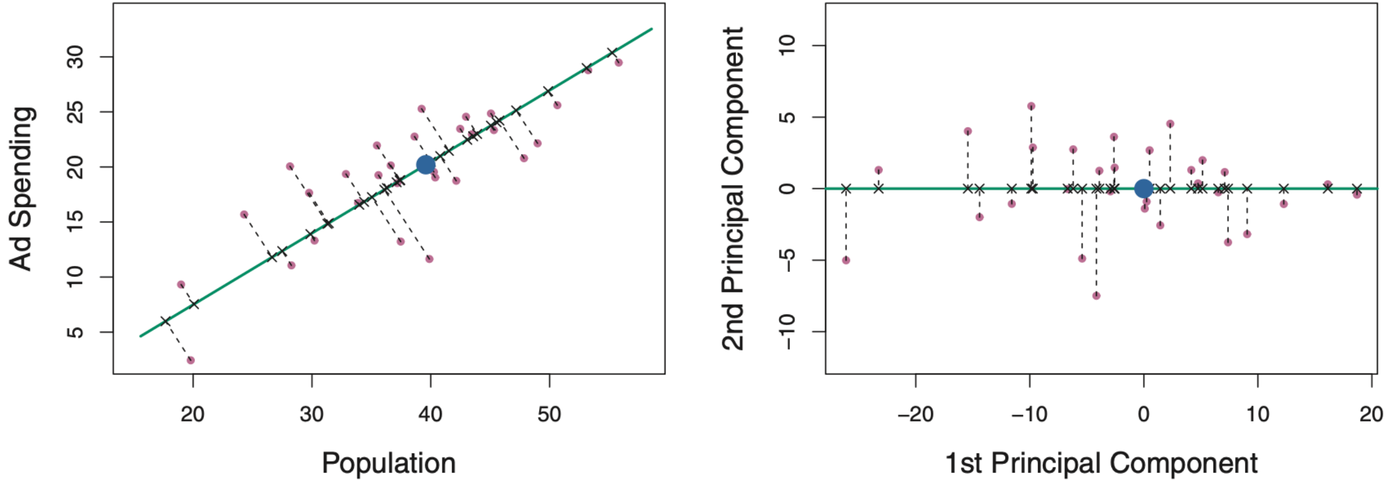
3) Eigen-decompose covariance matrix: $V^{-1}CV = D$

- V is the eigenvector matrix, D is the eigenvalue diagonal matrix.

4) Rearrange eigenvectors and eigenvalues descendingly by eigenvalues. The arranged eigenvectors are the principal components.

Principal Component Analysis

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



PC1: green
PC2: blue

In-Class Work examples

- PCA on Iris dataset.

  https://rpi.box.com/s/sma0oj43w4hgdzyxu4bn7z8dcn497873

PCA on Boston dataset

```
install.packages('MASS')
boston.df <- Boston
```

Do PCA!

Thanks!