Data Pre-Processing-III

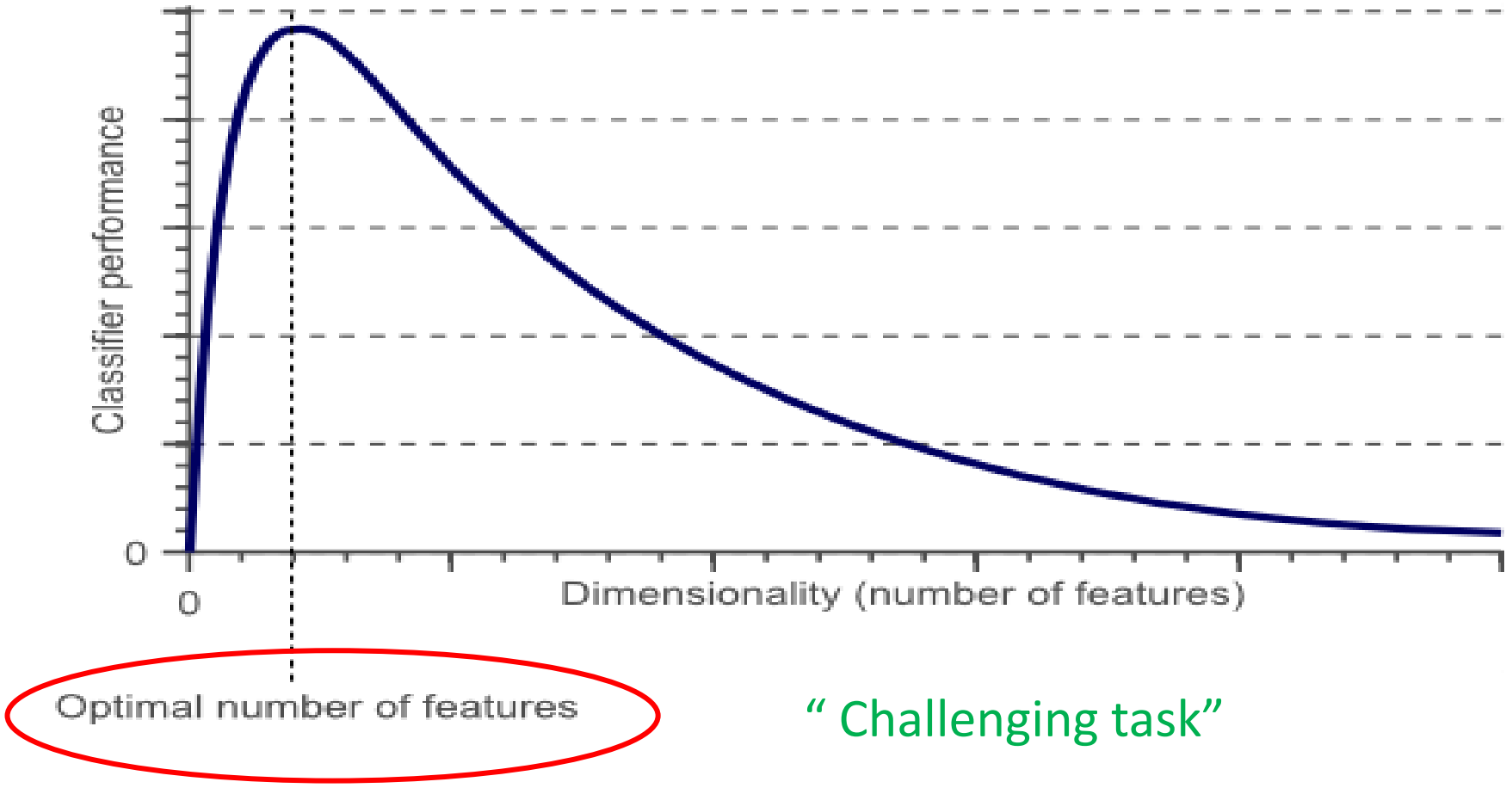
(Data Reduction)

TIET, PATIALA

Dimensionality/Data Reduction

- The number of input variables or features for a dataset is referred to as its dimensionality.
- **Dimensionality reduction** refers to techniques that reduce the number of input variables in a dataset.
- More input features often make a predictive modeling task more challenging to model, more generally referred to as the ***curse of dimensionality***.
- There exist a optimal number of feature in a feature set for corresponding Machine Learning task.
- Adding additional features than optimal ones (strictly necessary) results in a performance degradation (because of added noise).

Dimensionality/Data Reduction



“ Challenging task”

Dimensionality/Data Reduction

Benefits of data reduction

- Accuracy improvements.
- Over-fitting risk reduction.
- Speed up in training.
- Improved Data Visualization.
- Increase in explain ability of ML model.
- Increase storage efficiency.
- Reduced storage cost.

Data Reduction Techniques

Feature Selection –

find the best set of feature

- Filter methods
- Wrapper methods
- Embedded methods

Feature Extraction-

methods of constructing combinations of the variables to get around these problems while still describing the data.

- Principal Component Analysis
- Singular-Valued Decomposition
- Linear Discriminant Analysis

Feature Selection

- Feature selection in machine learning is to find the best set of features that allows one to build useful models of studied phenomena.
- The two key drivers used in feature selection are:
  - **Maximizing feature relevance**
    - Feature contributing significant information for the machine learning model – strongly relevant
    - Feature contributing little information for the machine learning model – weakly relevant
    - Feature contributing no information for the machine learning model – irrelevant
  - **Minimizing feature redundancy**
    - Information contributed by the feature is similar to the information contributed by one or more other features.

Feature Selection (Contd….)

Roll Number	Age	Height	Weight

- Let us consider a student database, with attributes Roll Number, Age , Height and Target Variable (Weight). The objective is to predict a weight for each new test case.
- Roll Number is irrelevant as it will not provide any information regarding weight of students.
- Age and Height are redundant as both provide same information.

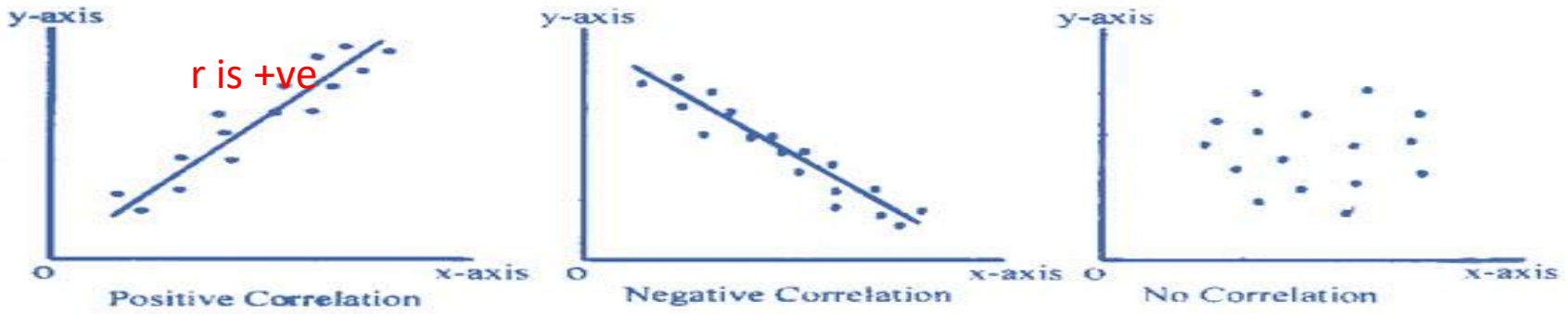
Feature Selection- Measuring Feature Redundancy

- Feature Redundancy is measured in terms of similarity information contributed by features.
- Similarity information is measured in terms of:
  - Correlation-based features.
  - Distance based features.

Feature Selection- Measuring Feature Redundancy

- To deal with redundant features correlation analysis is performed. Denoted by r.
- A threshold is decided to find redundant features.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$



Feature Selection- Measuring Feature Redundancy

Distance-based:

- The most commonly used distance metric is various forms of **Minkowski distance.**

$$d(F_1, F_2) = \sqrt[r]{\sum_{i=1}^{n} (F_{1i} - F_{2i})^r}$$

It takes the form of **Euclidian distance** when r =2 (L_2 norm) and **Manhattandistance** when r = 1 (L_1 norm).

- **Cosine similarity** is another important metric for computing similarity between features.

$$cos(F_1, F_2) = \frac{F_1 \,.\, F_2}{|F_1\,||F_2|}$$

Where F_1 and F_2 denote feature vectors.

Feature Selection- Measuring Feature Redundancy

For binary features, following metrics are useful:

1. Hamming distance: number of values which are different in two feature vectors.
2. Jaccard distance: 1- Jaccard Similarity

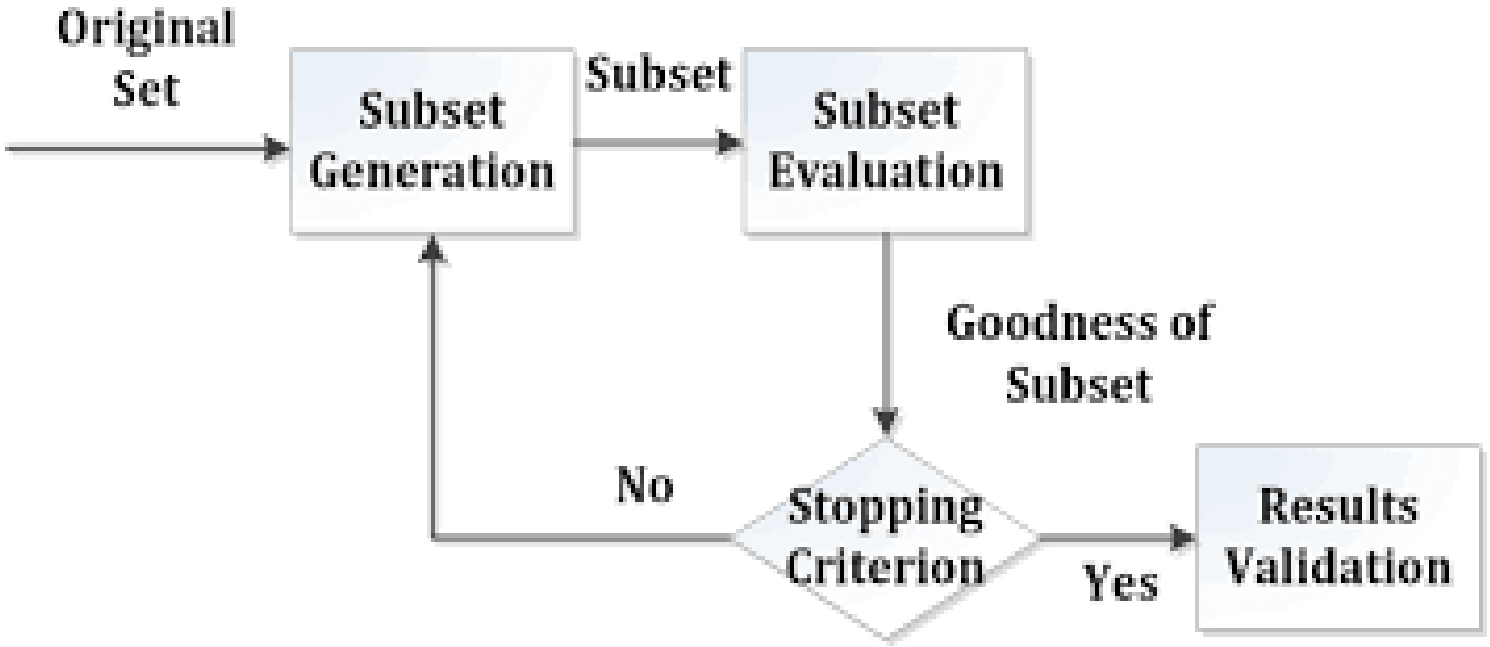
$$Jaccard\ Similarity = \frac{n_{11}}{n_{01} + n_{10} + n_{11}}$$

3. Simple Matching Coefficient (SMC):

$$SMC = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

Where n_{11}, n_{00} represent number of cases where both features have value 1 and 0 respectively
n_{10} denote cases where feature 1 has value 1 and feature 2 has value 0.
n_{01} denote cases where feature 1 has value 0 and feature 2 has value 1.

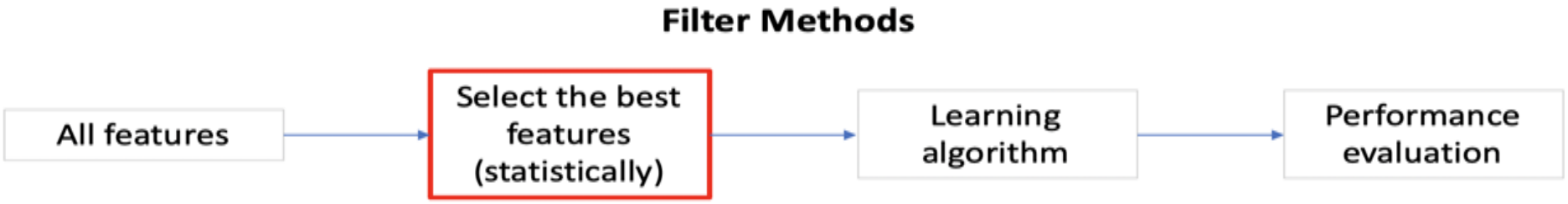
Overall Feature Selection Process



Feature Selection Approaches

Filter Approach:

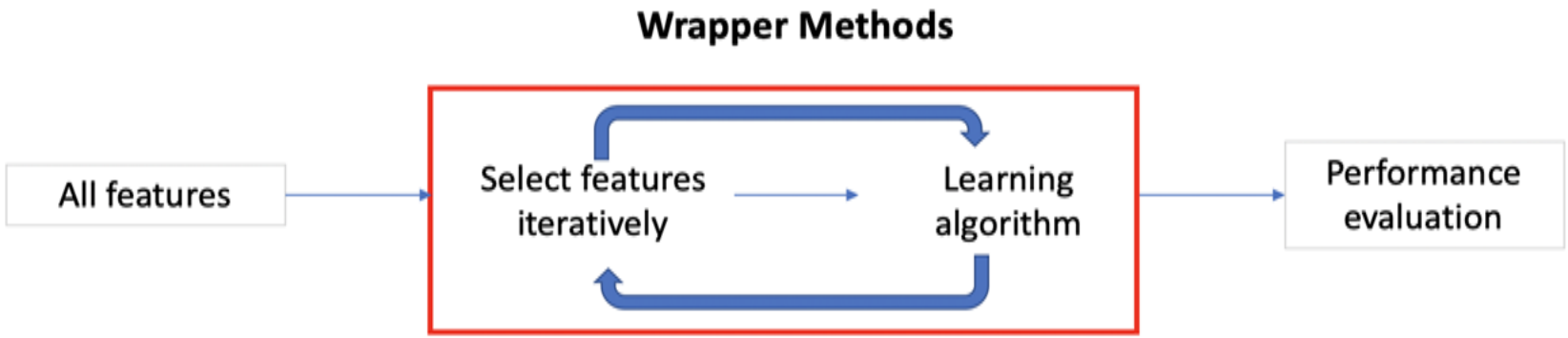
- In this approach, the feature subset is selected based on statistical measures.
- No learning algorithm is employed to evaluate the goodness of the feature selected.
- Commonly used metrics include correlation, chi square, Fisher score, ANOVA, Information Gain, etc.



Feature Selection Approaches

Wrapper Approach:

- In this approach, for every candidate subset, the learning model is trained and the result is evaluated by running the learning algorithm.
- Computationally very expensive but superior in performance.
- Requires some method to search the space of all possible subsets of features



Feature Selection Approaches

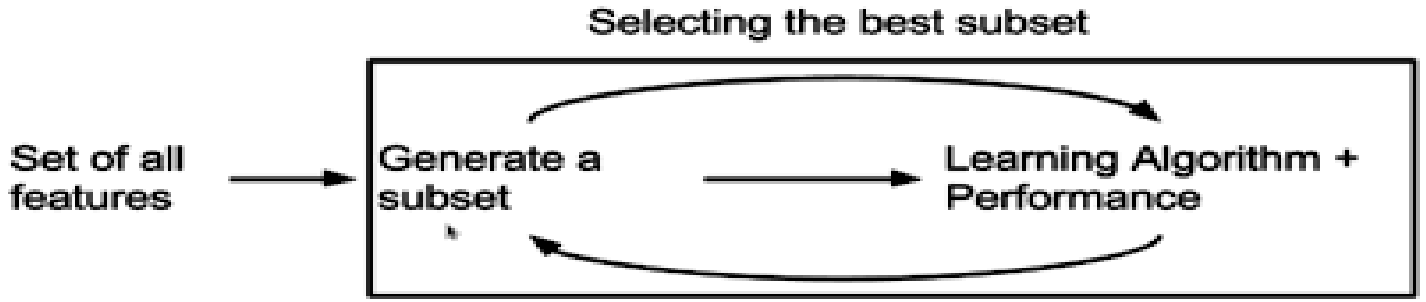
Wrapper Approach- Searching Methods:

- **Forward Feature Selection**
  - This is an iterative method wherein we start with the best performing variable against the target.
  - Next, we select another variable that gives the best performance in combination with the first selected variable.
  - This process continues until the preset criterion is achieved.
- **Backward Feature Elimination**
  - Here, we start with all the features available and build a model.
  - Next, we the variable from the model which gives the best evaluation measure value.
- **Exhaustive Feature Selection**
  - It tries every possible combination of the variables and returns the best performing subset.

Feature Selection Approaches

Embedded Approach

- These methods encompass the benefits of both the wrapper and filter methods.
- It includes interactions of features but also maintaining reasonable computational cost.
- Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extracts those features which contribute the most to the training for a particular iteration.

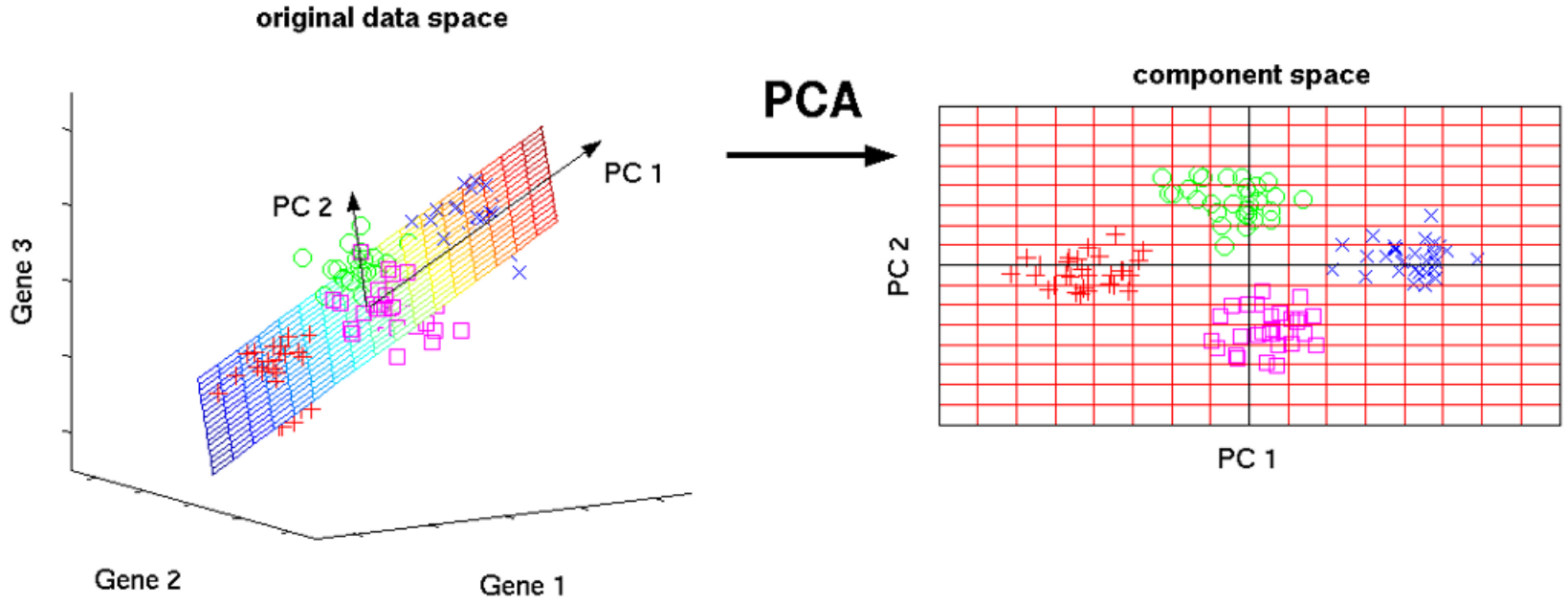


Feature Extraction

- Feature extraction, creates new features from a combination of original features.
- For a given Feature set F_i (F_1, F_2, F_3,……..F_n), feature extraction finds a mapping function that maps it to new feature set F_i' (F_1', F_2', F_3',…….F_m') such that F_i'=f(F_i) and m <n.
- For instance F_1'= k_1 F_1 + k_2F_2
- Some commonly used methods are:
  - Principal Component Analysis (PCA)
  - Singular Valued Decomposition (SVD)
  - Linear Discriminant Analysis (LDA)

Principal Component Analysis

Principal Component Analysis (PCA): It is a technique of dimensionality reduction which performs the said task by reducing the higher-dimensional feature-space to a lower-dimensional feature-space. It also helps to make visualization of large dataset simple.



Principal Component Analysis

Some of the major facts about PCA are:

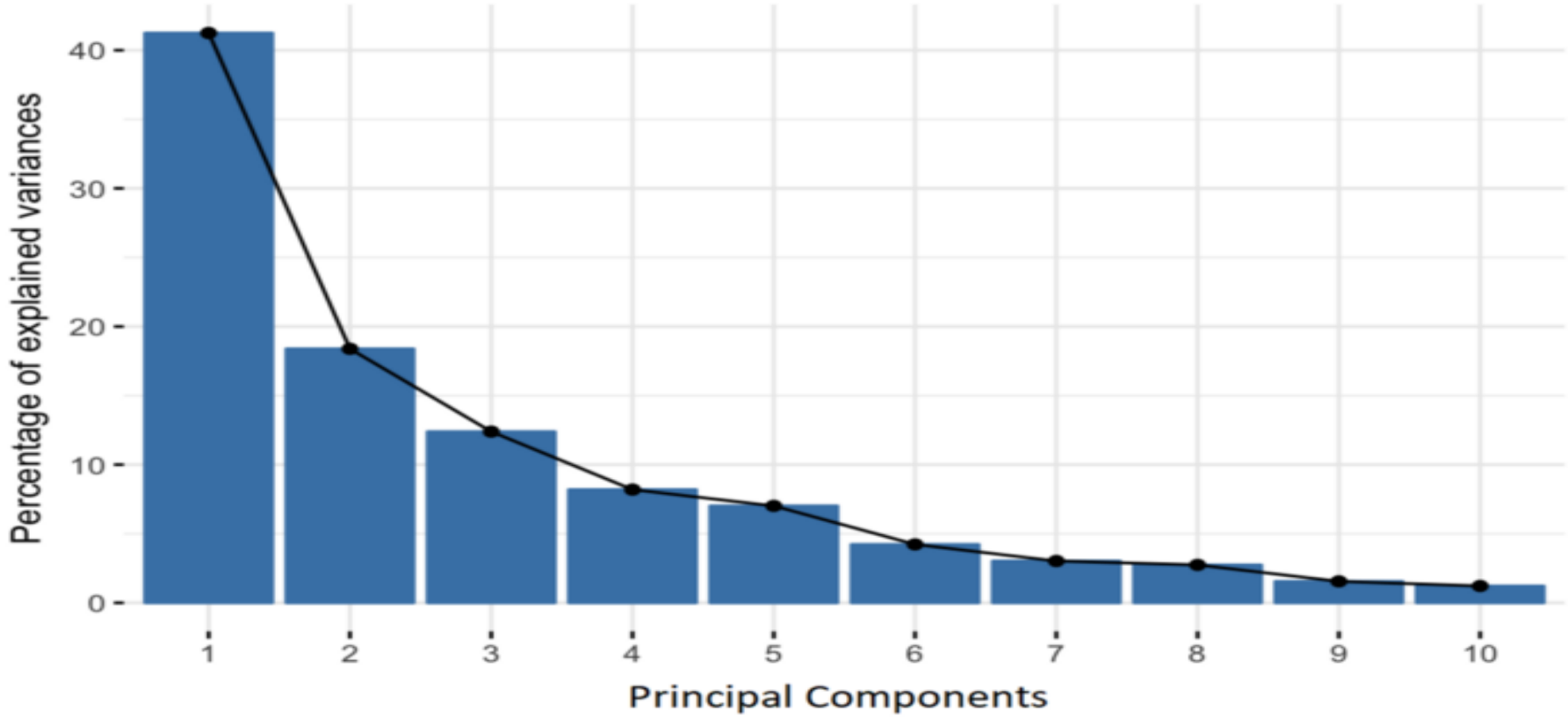
- Principal components are new features that are constructed as a linear combinations or mixtures of the initial feature set.
- These combinations is performed in such a manner that all the newly constructed principal components are uncorrelated.
- Together with reduction task, PCA also preserving as much information as possible of original data set.

Principal Component Analysis

Some of the major facts about PCA are:

- Principal components are usually denoted by PCi, where i can be 0, 1, 2, 3,. ….,n (depending on the number of feature in original dataset).
- The major proportion of information about original feature set can be alone explained by first principal component i.e. PC1.
- The remaining information can be obtained from other principal components in a decreasing proportion as per increase in value of i.

Principal Component Analysis



Principal Component Analysis

- **Geometrically ,** it can be said that principal components are lines pointing the directions that captures maximum amount of information about the data.
- Principal components also aims to minimize the error between the true location of the data points (in original feature space) and the projected location of the data points (in projected feature space).
- The larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has.

Simply, principal components are new axes to get better data visibility with clear difference in observations.

Principal Component Analysis

Stepwise working of PCA

Step 1: Construction of covariance matrix named as A.

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them.

$$Cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})$$

Step 2: Computation of eigenvalues for covariance matrix.

$$\det(A-\lambda I) = 0$$

The eigenvectors of the Covariance matrix are actually *the directions of the axes where there is the most variance* (most information) *and that we call Principal Components*

Principal Component Analysis

Stepwise working of PCA

Step 3: Compute eigenvectors corresponding to every eigenvalue obtained in step 2

$$[\boldsymbol{A} - \boldsymbol{\lambda I}]\, \boldsymbol{X} = \mathbf{O}$$

The eigenvalues are simply the coefficients attached to eigenvectors, which give the *amount of variance carried in each Principal Component*.

Step 4: Sort the eigenvectors in decreasing order of eigenvalues and choose k eigenvectors with the largest eigenvalues.

Step 5: Transform the data along the principal component axis.

Principal Component Analysis-Example

Example:

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Original dataset

Compute Covariance Matrix i.e. A

Cov(X,X)	Cov(X,Y)
Cov(Y, X)	Cov(Y,Y)

$$Cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})$$

Principal Component Analysis-Example

Example:

X	Y	$X_i - \bar{X}$	$(Y_i - \bar{Y})$	$(X_i-\bar{X})(X_i - \bar{X})$	$(Y_i-\bar{Y})(Y_i - \bar{Y})$	$(X_i-\bar{X})(Y_i - \bar{Y})$
2.5	2.4	0.69	0.49	0.4761	0.2401	0.3381
0.5	0.7	-1.31	-1.21	1.7161	1.4641	1.5851
2.2	2.9	0.39	0.99	0.1521	0.9801	0.3861
1.9	2.2	0.09	0.29	0.0081	0.0841	0.0261
3.1	3	1.29	1.09	1.6641	1.1881	1.4061
2.3	2.7	0.49	0.79	0.2401	0.6241	0.3871
2	1.6	0.19	-0.31	0.0361	0.0961	-0.0589
1	1.1	-0.81	-0.81	0.6561	0.6561	0.6561
1.5	1.6	-0.31	-0.31	0.0961	0.0961	0.0961
1.1	0.9	-0.71	-1.01	0.5041	1.0201	0.7171

$\bar{X}$=1.81

$\bar{Y} = 1.91$

Cov(X,X) = 0.6165

Cov(Y,Y) = 0.7165

Cov(X,Y)= Cov(Y,X) = 0.6154

Principal Component Analysis-Example

Example:

0.6165	0.6154
0.6154	0.7165

Compute eigenvalues

$\det(A-\lambda I) = 0$

0.6165	0.6154
0.6154	0.7165

$-\lambda$

1	0
0	1

0.6165-λ	0.6154
0.6154	0.7165-λ

Find determinate by equating to zero

$\lambda_1 = 1.284028$

$\lambda_2 = 0.049083$

Principal Component Analysis-Example

Example:

0.6165-λ_1	0.6154
0.6154	0.7165-λ_1

Compute eigenvectors

$$[\boldsymbol{A} - \boldsymbol{\lambda I}]\,\boldsymbol{X} = \boldsymbol{0}$$

0.6165-λ_2	0.6154
0.6154	0.7165-λ_2

$\lambda_1 = 1.284028$

$\lambda_2 = 0.049083$

-0.6675	0.6154
0.6154	-0.5675

V_1

x_1
x_2

$= 0$

0.5674	0.6154
0.6154	0.6674

V_2

x_1
x_2

$= 0$

Principal Component Analysis-Example

Example:

$V_1 =$

0.67787
0.73517

First Principal Component (PC1)

$V_2 =$

-0.73517
0.67787

Second Principal Component (PC2)

"vector corresponding to highest eigenvalue of considered as PC1 followed by other component as per their eigenvalue."

- To calculate the **percentage of information** explained by PC1 and PC2, divide each component by sum of eigenvalues

PC1 = 96%

PC2 = 4%

Principal Component Analysis-Example

Step 4 helps to reduce the dimension by discarding the components with very less percentage of information in a multi-dimensional space. The remaining ones form a matrix of vector known as feature vector. Each column correspond to one principal component.

Step 5 data transformation along principal component using

$$Final\ dataset = Feature\ vector^T * orignal\ dataset^T$$

Or

*Final Dataset= Original Dataset * Feature vector*

(every ith row now corresponds to new values of data points in the new feature space)