Data Pre-Processing-III

(Data Reduction)

Dr. JASMEET SINGH

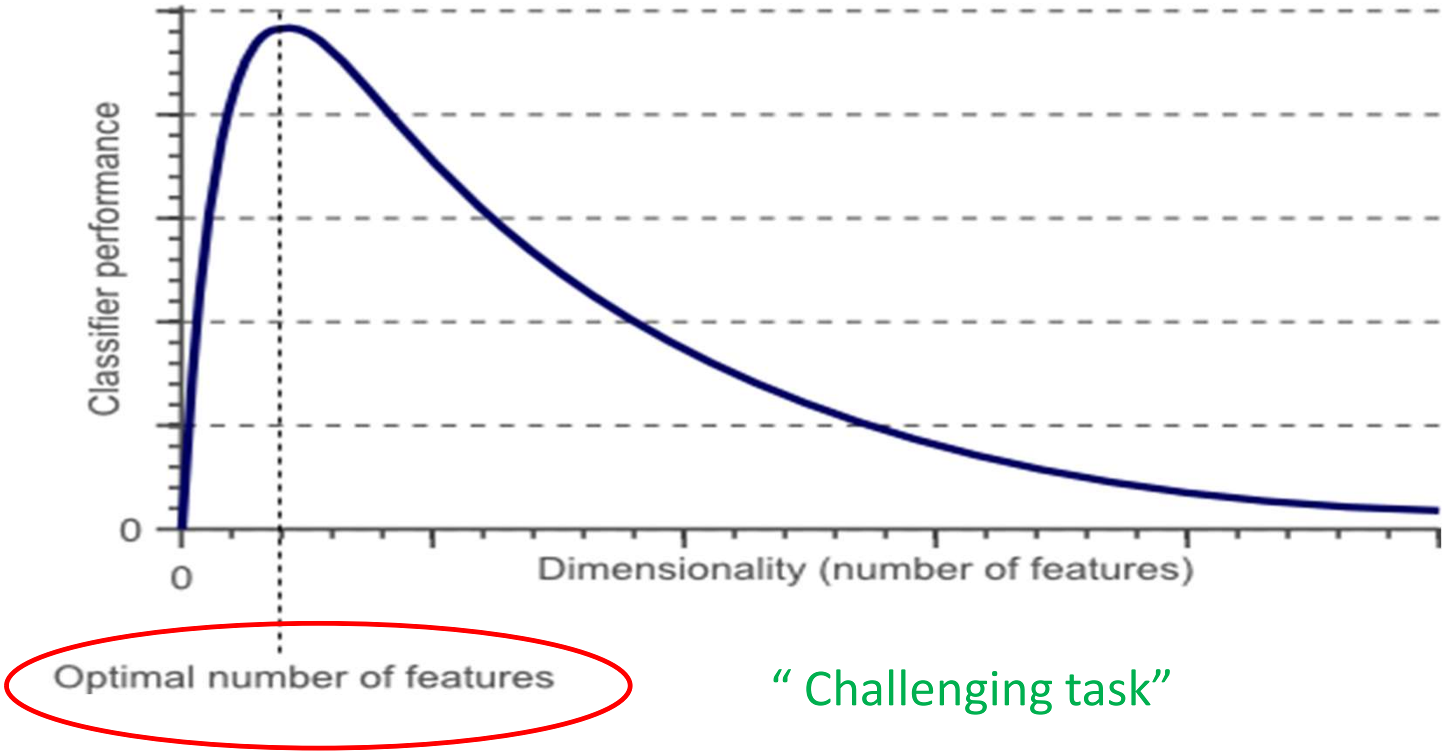
ASSISTANT PROFESSOR, CSED

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



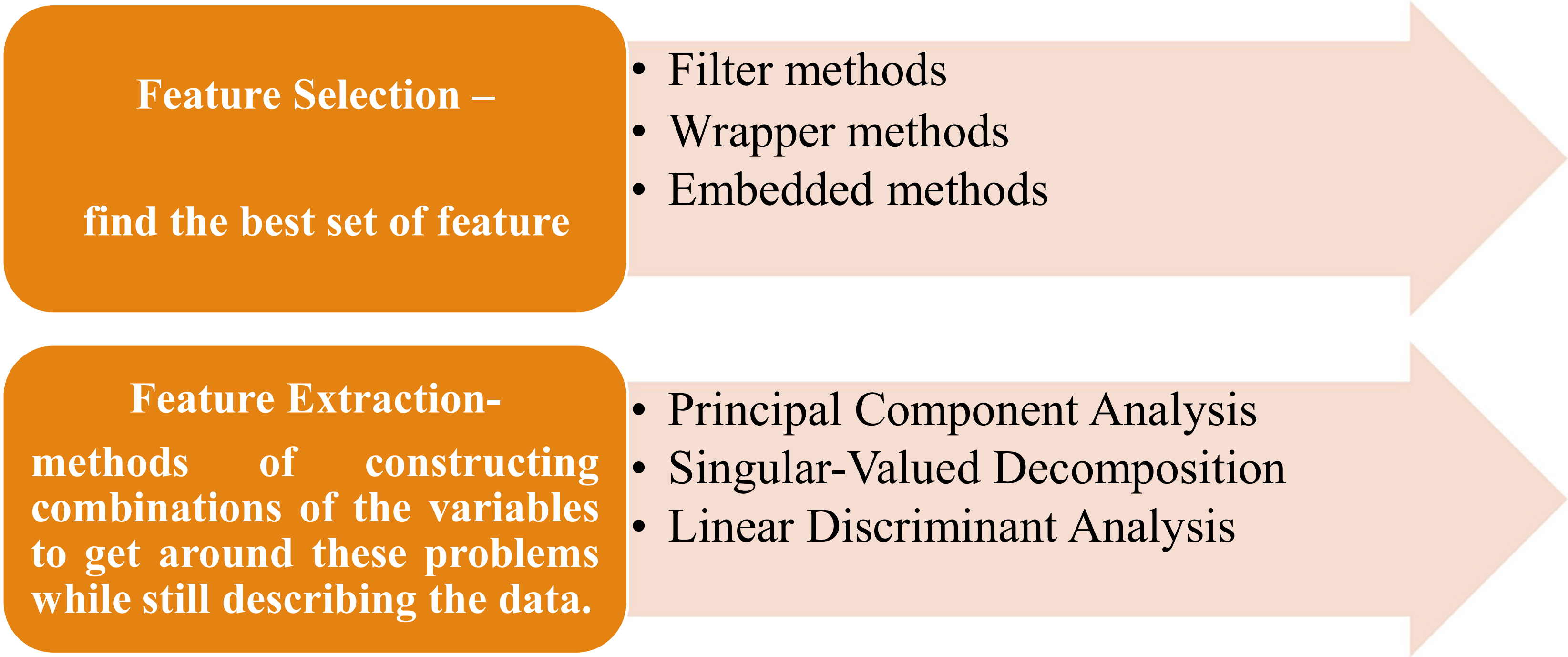
“ Challenging task”

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

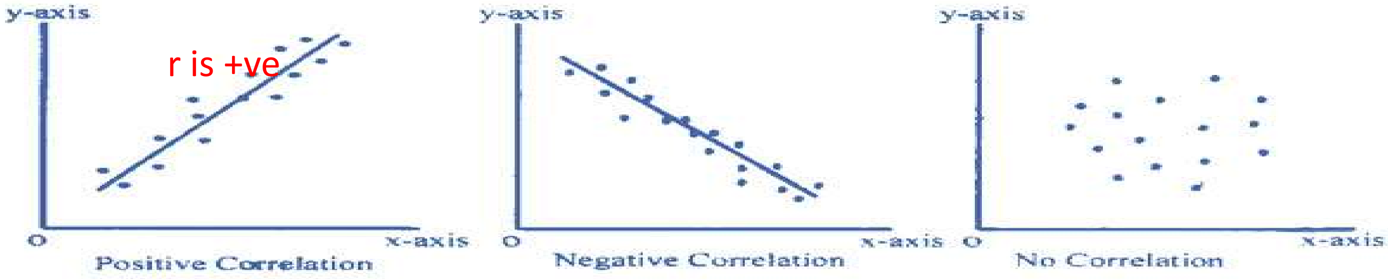
- Feature Redundancy is measured in terms of similarity information contributed by features.

- 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.

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



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 **Manhattan distance** 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

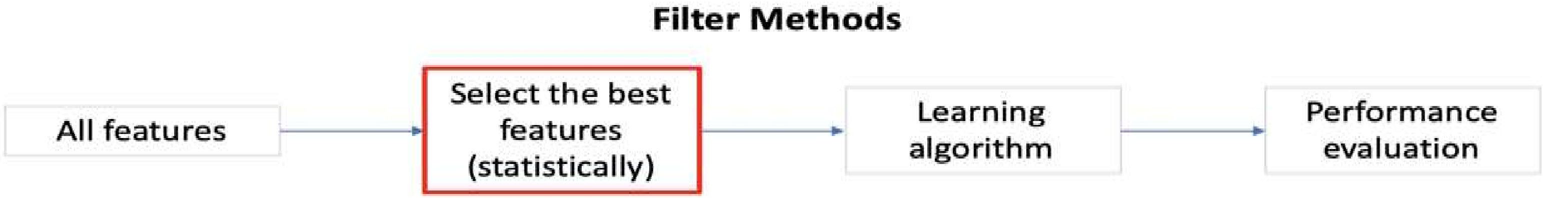
n_{10} denote cases where feature 1 has value 1 and feature 2 has value 0.

n_{01} denote cases where feature 1 has value 0 and feature 2 has value 1.

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.



Chi-Square Test for Feature Selection

- A chi-square test is used in statistics to test the independence of two events.
- Given the data of two variables, we can get observed count O and expected count E.
- Chi-Square measures how expected count E and observed count O deviates each other.

$$\chi_c^2 = \sum \frac{(O_i^2 - E_i^2)}{E_i}$$

where c is the degree of freedom, O is the observed value and E is the expected value.

- When two features are independent, the observed count is close to the expected count, thus we will have smaller Chi-Square value.
- Higher the Chi-Square value the feature is more dependent on the response and it can be selected for model training.

Chi-Square Test for Feature Selection (Contd....)

- Steps to perform the Chi-Square Test:

1. Define Hypothesis.
2. Build a Contingency table.
3. Find the expected values.
4. Calculate the Chi-Square statistic.
5. Accept or Reject the Null Hypothesis.

Chi-Square Test for Feature Selection (Contd….)

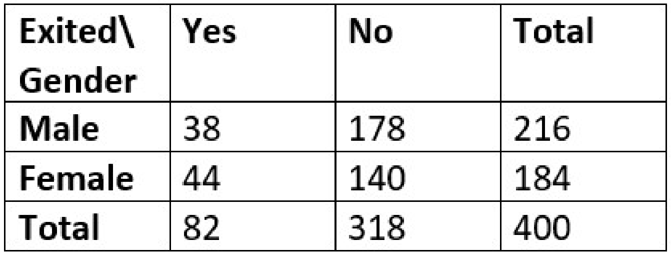
- Consider a data-set where we have to determine why customers are leaving the bank, let's perform a Chi-Square test for two variables.
- ***Gender*** of a customer with values as Male/Female as the predictor and ***Exited*** describes whether a customer is leaving the bank with values Yes/No as the response.
- In this test we will check *is there any relationship between Gender and Exited*.
- **Step 1: Define Hypothesis**

Null Hypothesis (H0): Two variables are independent.

Alternate Hypothesis (H1): Two variables are not independent.

Chi-Square Test for Feature Selection (Contd….)

- **Step 2: Contingency Table**



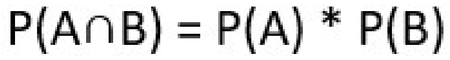
Exited\ Gender	Yes	No	Total
Male	38	178	216
Female	44	140	184
Total	82	318	400

Degrees of freedom for contingency table is given as (r-1) * (c-1) where r,c are rows and columns. Here df = (2–1) * (2–1) = 1.

Chi-Square Test for Feature Selection (Contd….)

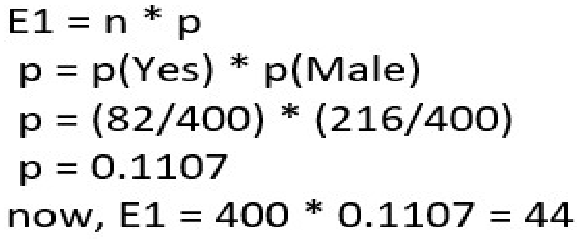
- **Step 3. Find the Expected Value**

Based on the null hypothesis that the two variables are independent. We can say if A, B are two independent events



$$P(A \cap B) = P(A) * P(B)$$

Let’s calculate the expected value for the first cell that is those who are Males and are Exited from the bank.



$E1 = n * p$

$p = p(Yes) * p(Male)$

$p = (82/400) * (216/400)$

$p = 0.1107$

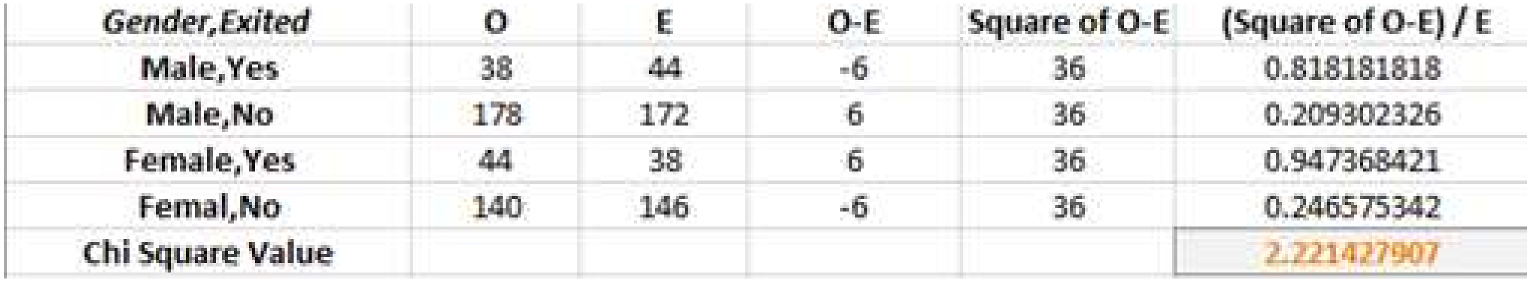
now, $E1 = 400 * 0.1107 = 44$

Chi-Square Test for Feature Selection (Contd….)

- In similar, we calculate E2, E3, E4 and get the following results.

Exited\Gender	Yes	No
Male	44	172
Female	38	146

- Step 4 **Calculate Chi-Square value**: Summarizing the observed values and calculated expected values into a table and determine the Chi-Square value.

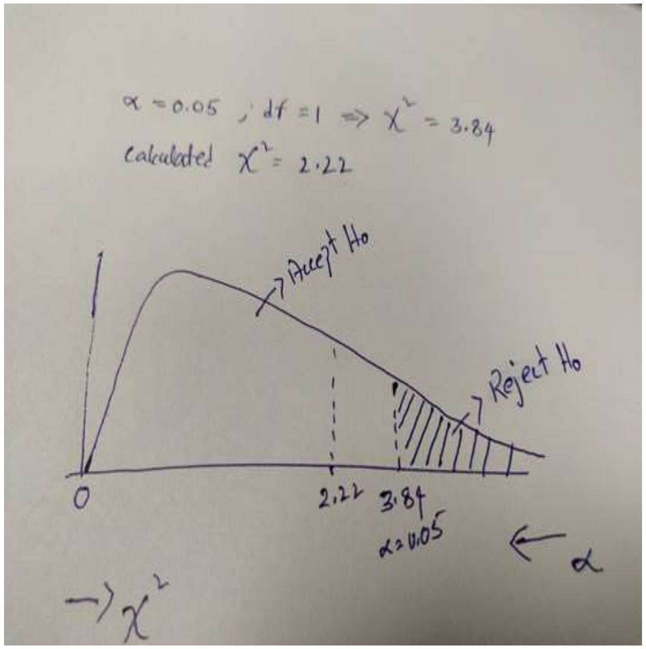


Gender,Exited	O	E	O-E	Square of O-E	(Square of O-E) / E
Male,Yes	38	44	-6	36	0.818181818
Male,No	178	172	6	36	0.209302326
Female,Yes	44	38	6	36	0.947368421
Femal,No	140	146	-6	36	0.246575342
Chi Square Value					2.221427907

Chi-Square Test for Feature Selection (Contd….)

Step 5. Accept or Reject the Null Hypothesis

- With 95% confidence that is alpha = 0.05, we will check the calculated Chi-Square value falls in the acceptance or rejection region.
- Having degrees of freedom =1(calculated with contingency table) and alpha =0.05 the Chi-Square value is 3.84.
- In the fig, we can see Chi-Square ranges from 0 to inf and alpha ranges from 0 to 1 in the opposite direction. We will reject the Null hypothesis if Chi-Square value falls in the error region (alpha from 0 to 0.05).
- So here we are accepting the null hypothesis since the Chi-Square value is less than the critical Chi-Square value.
- *To conclude the two variables are independent, Gender variable cannot be selected for training the model.*



Feature Selection using Information Gain

- Information Gain = Entropy of Parent – sum (weighted % * Entropy of Child)

Weighted % = Number of observations in particular child/sum (observations in all child nodes)

- In particular, Information Gain for a feature column **A** is calculated as:

$$Information\ Gain(S,A) = Entropy(S) - \sum_{v=1}^{|v|} \frac{|S_v|}{|S|} Entropy(S_v)$$

where **S_v** is the set of rows in **S** for which the feature column **A** has value **v**, |**S_v**| is the number of rows in **S_v** and likewise |**S**| is the number of rows in **S.**

- **Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes.**
- **The feature with the highest Information Gain is selected as the best one.**

Feature Selection using Gain Ratio

- ID3 algorithm's Information Gain metric is biased towards a feature with large number of distinct values.
- This limitation of ID3 algorithm is handled by normalizing the *Information Gain* metric using a parameter called *SplitInfo.* The normalized Information Gain is called *Gain Ratio.*
- *Gain Ratio of an attribute A for a given dataset is computed as:*

$$Gain\ Ratio(S, A) = \frac{Information\ Gain(S, A)}{SplitInfo(S, A)}$$

$$Information\ Gain(S, A) = Entropy(S) - \sum_{v=1}^{|v|} \frac{|S_v|}{|S|} Entropy(S_v)$$

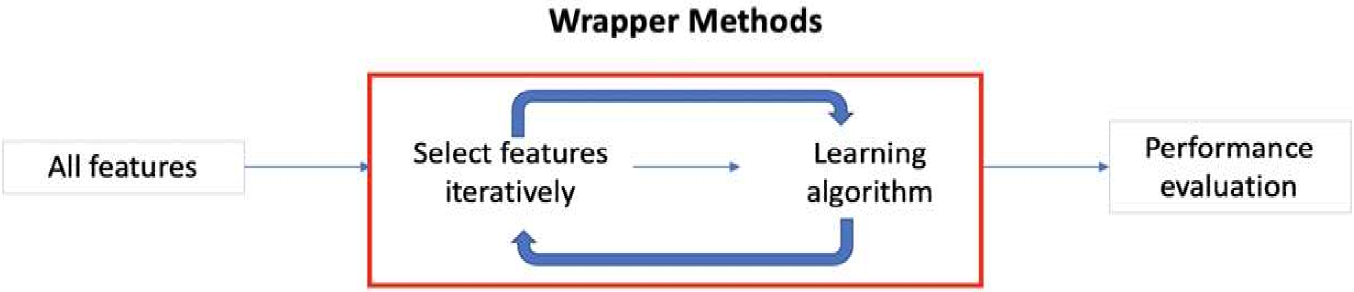
$$SplitInfo(S, A) = -\sum_{v=1}^{|v|} \frac{|S_v|}{|S|} log_2\left(\frac{|S_v|}{|S|}\right)$$

where $\mathbf{S_v}$ is the set of rows in **S** for which the feature column **A** has value **v**, $|\mathbf{S_v}|$ is the number of rows in $\mathbf{S_v}$ and likewise $|\mathbf{S}|$ is the number of rows in **S.**

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

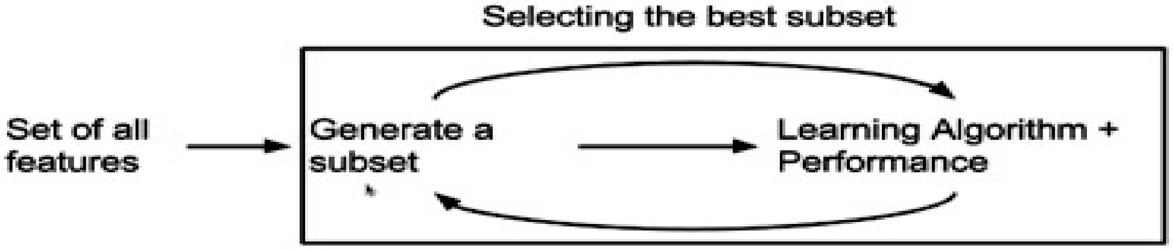
Wrapper Approach- Searching Methods:

- **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.



Embedded Approach

Random forests uses ***embedded approach*** to rank the importance of variables in a regression or classification problem in a natural way.

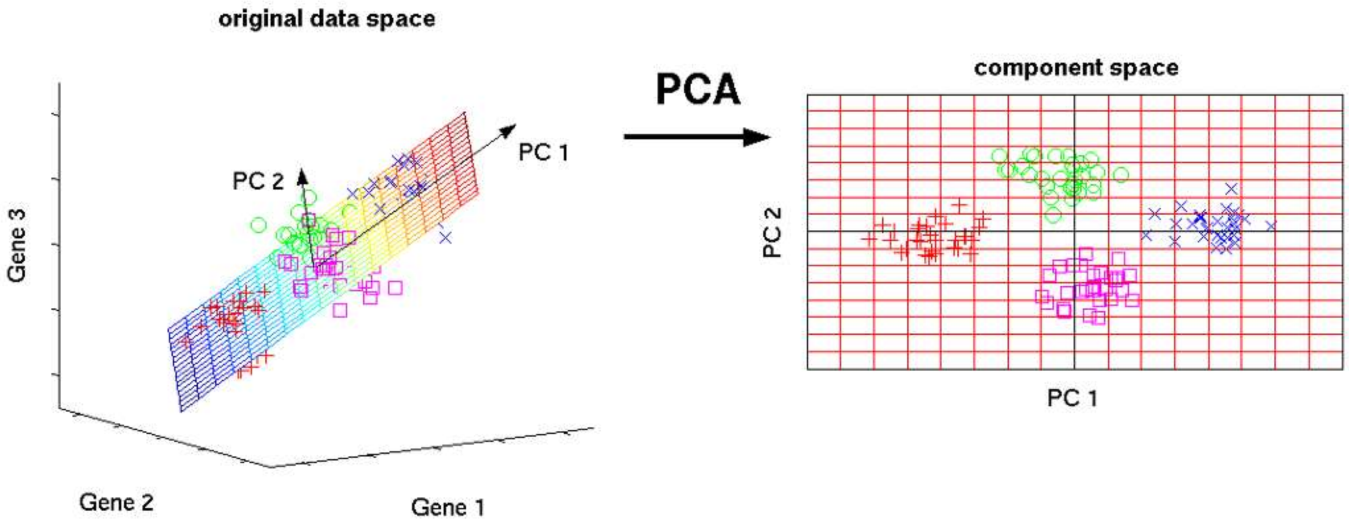
- The first step in measuring the feature importance in a data set is to fit a random forest to the data.
- During the fitting process the out-of-bag error for each data point is recorded and averaged over the forest.
- To measure the importance of the j-th feature after training, the values of the j-th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set.
- The importance score for the j-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees.
- The score is normalized by the standard deviation of these differences.

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:

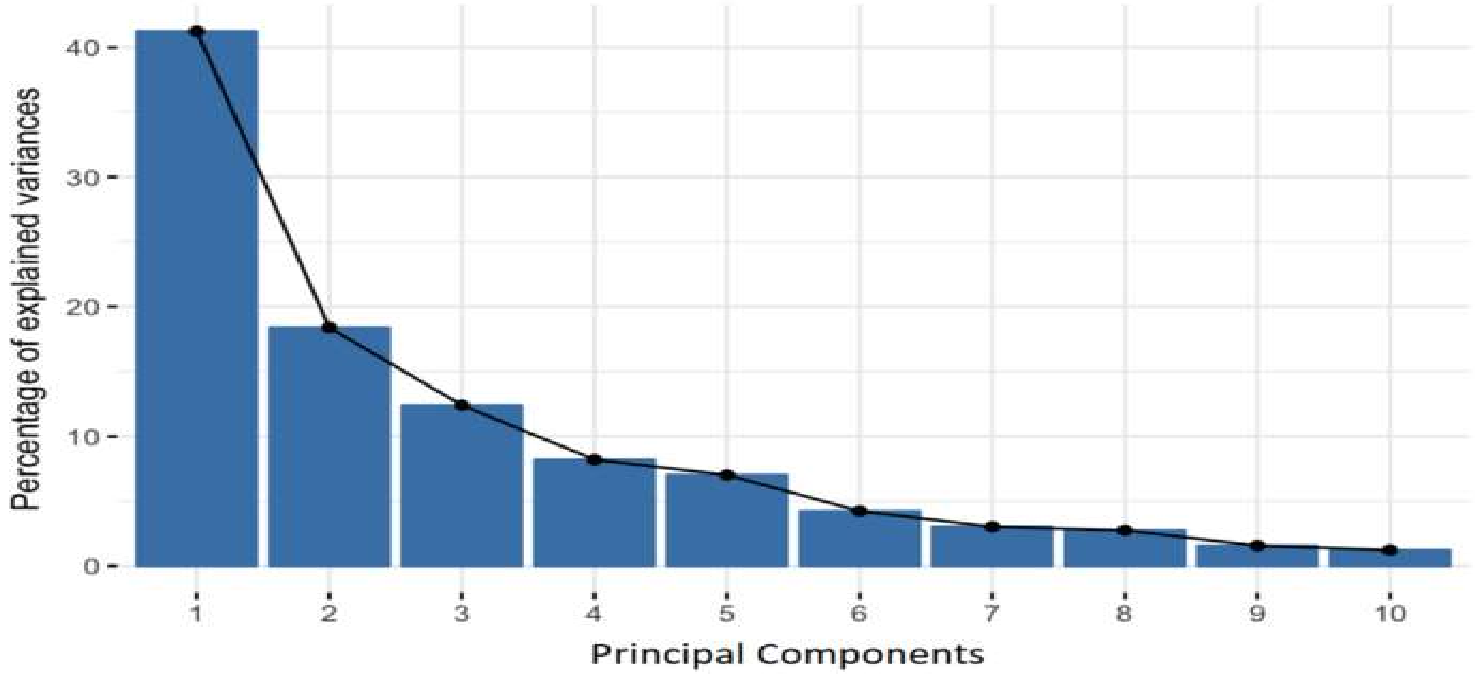
- Principal components are new features that are constructed as a linear combinations or mixtures of the initial feature set.
- These combinations is performed in such a manner that all the newly constructed principal components are uncorrelated.
- Together with reduction task, PCA also preserving as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



PCA- Geometrical Interpretation

- **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.

PCA- Geometrical Interpretation

- Suppose we have the following standardized data (as shown in figure 1).
- If suppose we have to choose 1 feature out of X1 and X2, we will choose feature X1. (The one which explains the maximum variation in the data).
- This is exactly what PCA does. It finds the features which have maximum spread and drop the others with the aim to **minimize information loss**.
- Let's take a slightly complex example where we can not simply drop one feature (as shown in figure 2).
- Here, both the features X1 and X2 have equal spread. So we can't tell whic feature is more important.
- But if we try to find a direction (or axis) which explains the variation in dat we can find a line which fits the data very well. So if we rotate our axis slightly by theta, we get f1 and f2 (perpendicular to f1). We can then drop f2 and say f1 is the most important feature. This is what PCA does.

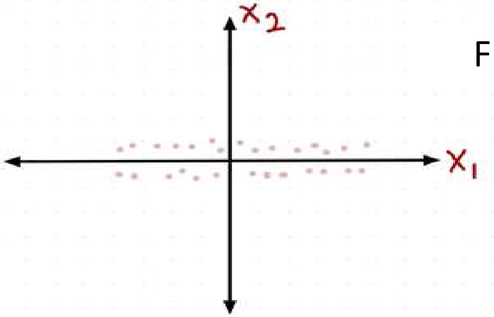


Fig 1

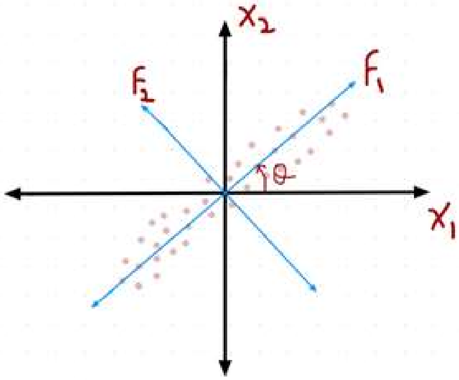
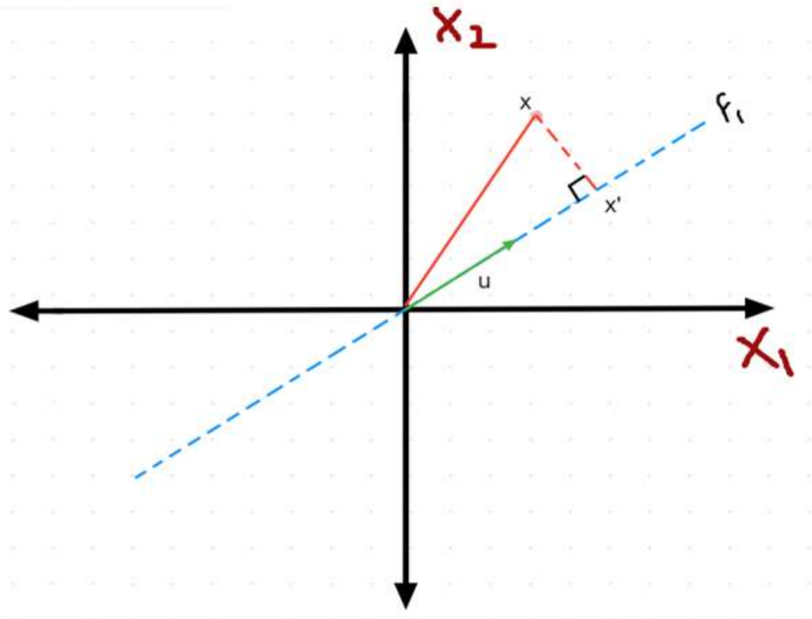


Fig 2

Mathematics behind PCA

Let’s try to find a line that maximizes the distance of projected point to the origin i.e. maximize the variance of the projected distance.



Let f_1 *be the direction of maximum variance and let* u_1 *be a unit vector in the direction of* f_1 *i.e.*

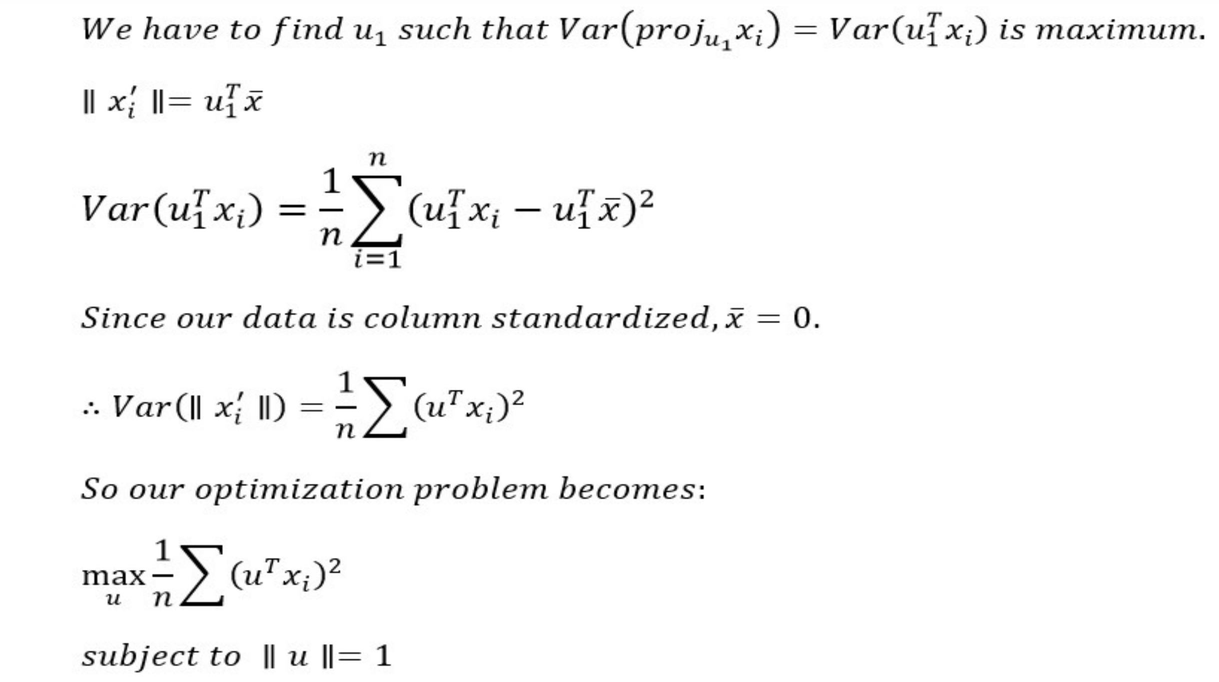
$\| u_1 \| = 1.$

So now our objective is to find this variance maximising vector u_1. *Let* x_i *be any point and* x_i' *be its projection on* u_1, *i.e.*

$$\| x_i' \| = proj_{u_1} x_i = \frac{u_1 \cdot x_i}{\| u_1 \|} = u_1 \cdot x_i = u_1^T x_i \; .$$

$(\cdot \rightarrow Dot\ product)$

Mathematics behind PCA (Contd.....)



$We\ have\ to\ find\ u_1\ such\ that\ Var(proj_{u_1} x_i) = Var(u_1^T x_i)\ is\ maximum.$

$\| x_i' \| = u_1^T \bar{x}$

$$Var(u_1^T x_i) = \frac{1}{n} \sum_{i=1}^{n} (u_1^T x_i - u_1^T \bar{x})^2$$

$Since\ our\ data\ is\ column\ standardized, \bar{x} = 0.$

$$\therefore Var(\| x_i' \|) = \frac{1}{n} \sum (u^T x_i)^2$$

$So\ our\ optimization\ problem\ becomes:$

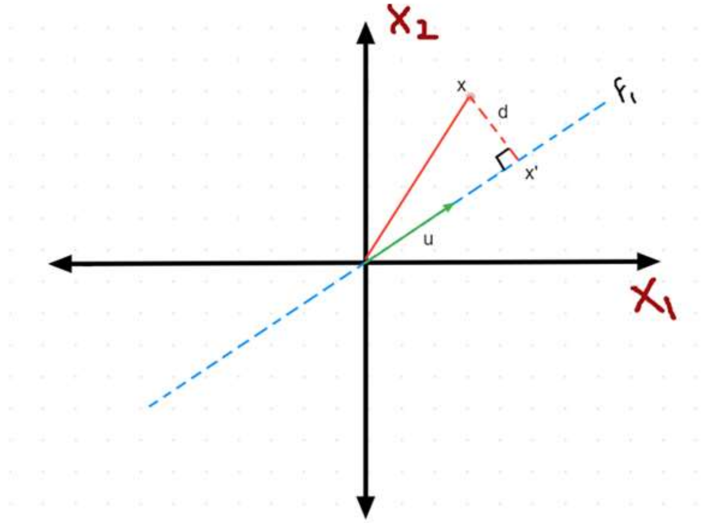
$$\max_u \frac{1}{n} \sum (u^T x_i)^2$$

$subject\ to\ \ \| u \| = 1$

This approach is called *variance maximization approach*

Mathematics behind PCA (Contd.....)

Another way to think of PCA is that it fits the best line that passes through our data with an aim to minimize the projection error 'd' for each point. This approach is called distance minimization approach.



$$d^2 = \| x_i \|^2 - (u^T x_i)^2$$

$$= (x_i^T x_i) - (u^T x_i)^2$$

So our optimization function becomes:

$$\min_u \sum_{i=1}^{n} (x_i^T x_i) - (u^T x_i)^2$$

subject to $\| u \| = 1$

Notice that both the optimization problems, though look different, are same. Since the $x_i^T x_i$ term is independent of u so in order to minimize the function we have to maximize $(u^T x_i)^2$ which is same as our first optimization problem.

Mathematics behind PCA (Contd…..)

Our optimization problem was to find a direction u which

$$\max_u \frac{1}{n}\sum_{i=1}^{n}(u^T x_i)^2\,,\quad subject\ to\ \ \| u \| = 1.$$

On writing our data vectors in matrix notations we can easily prove that

$\frac{1}{n}\sum_{i=1}^{n}(u^T x_i)^2 = u^T \frac{(X^T X)}{n} u = u^T S u.$ (Note: x_i is a row of our data matrix X)

So our optimization problem becomes:

$$\max_u u^T S u\,,\quad subject\ to\ \ \| u \| = u^T u = 1.$$

$$where\ S = \frac{1}{n} X^T X\ \ is\ covariance\ matrix$$

Mathematics behind PCA (Contd…..)

- The given optimization problem is solved using Lagrange Optimization (which is used for constrained optimization)
- The method can be summarized as follows: in order to find the maximum or minimum of a function $f(x)$ subjected to the equality constraint $g(x)=0$, form the Lagrangian function

$$L(x, \lambda)= f(x) - \lambda \, g(x)$$

and find the stationary points of L considered as a function of x and the Lagrange multiplier λ

Mathematics behind PCA (Contd…..)

So, our Lagrange function becomes:

$$L(u,\lambda) = u^T S u - \lambda(u^t u - 1)$$

Partially differentiating wrt u, we get:

$$\frac{\partial\, L(u,\ \lambda)}{\partial u} = 2Su - 2\lambda u$$

Equating the derivative to 0 and solving for u we get:

$$\boldsymbol{Su = \lambda u}$$

Thus u is an eigenvector of the covariance matrix S corresponding to the largest eigenvalue lambda.

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

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

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,Y)	Cov(Y,X)

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

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

$\bar{Y} = 1.91$

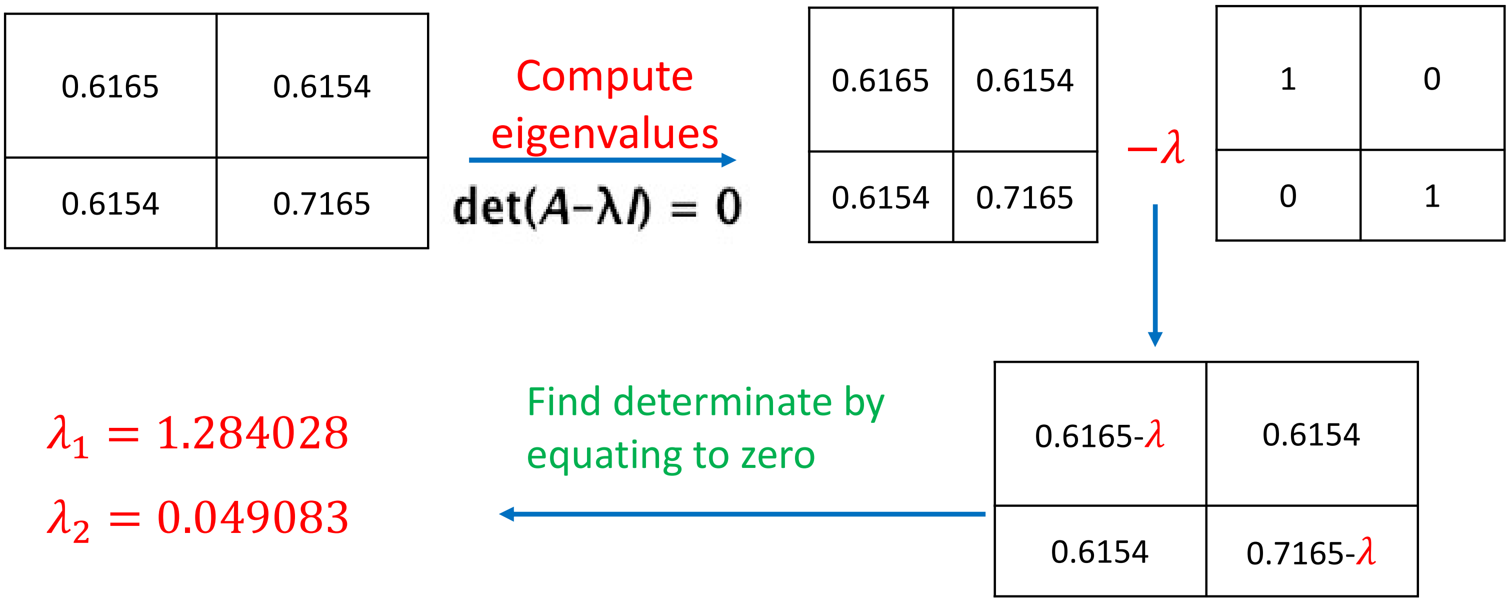
Cov(X,X) = 0.6165

Cov(Y,Y) = 0.7165

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

Principal Component Analysis-Example

Example:



$$\begin{bmatrix}0.6165 & 0.6154\\ 0.6154 & 0.7165\end{bmatrix}$$

Compute eigenvalues

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

$$\begin{bmatrix}0.6165 & 0.6154\\ 0.6154 & 0.7165\end{bmatrix} - \lambda \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}0.6165-\lambda & 0.6154\\ 0.6154 & 0.7165-\lambda\end{bmatrix}$$

Find determinate by equating to zero

$\lambda_1 = 1.284028$

$\lambda_2 = 0.049083$

Principal Component Analysis-Example

Example:

$$\begin{bmatrix} 0.6165-\lambda_1 & 0.6154 \\ 0.6154 & 0.7165-\lambda_1 \end{bmatrix}$$

$$\begin{bmatrix} 0.6165-\lambda_2 & 0.6154 \\ 0.6154 & 0.7165-\lambda_2 \end{bmatrix}$$

$$\xrightarrow{\text{Compute eigenvectors}}$$

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

$$\lambda_1 = 1.284028$$

$$\lambda_2 = 0.049083$$

$$\begin{bmatrix} -0.6675 & 0.6154 \\ 0.6154 & -0.5675 \end{bmatrix} \overset{V_1}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = 0$$

$$\begin{bmatrix} 0.5674 & 0.6154 \\ 0.6154 & 0.6674 \end{bmatrix} \overset{V_2}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = 0$$

Principal Component Analysis-Example

Example:

$$V_1 = \begin{array}{|c|}\hline 0.67787 \\ \hline 0.73517 \\ \hline\end{array} \qquad V_2 = \begin{array}{|c|}\hline -0.73517 \\ \hline 0.67787 \\ \hline\end{array}$$

First Principal Component (PC1)

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-dimentional 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)