Module 4: Dimensionality Reduction and Probabilistic Learning

By
Ms. Sarika Dharangaonkar
Assistant Professor,
KJSCE

Contents: Module1

Linear Discriminant Analysis

- Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique used in machine learning and statistics.
- Its primary purpose is to find a linear combination of features that best characterizes or discriminates between classes or groups in a dataset.
- It's a supervised learning algorithm, meaning it uses labeled data to learn patterns and make predictions.

LDA…

Steps of LDA:

- **Collect Data with Labels:** Obtain a dataset where each sample is labeled with a specific class or category.

  Let's define some terms:

  X: The dataset with *n* samples and *p* features.

  y: The corresponding class labels for each sample.

- The primary objective of LDA is to project the original *p*-dimensional dataset onto a lower-dimensional space (often one dimension) while maximizing the separation between the classes.

LDA…

- **Compute Class Means:** Calculate the mean feature vector for each class. These means represent the centroids of the classes in the feature space.
- Compute the mean of each feature for each class. These are denoted by μi , where i represents the class.

$$\mu_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

- where ni is the number of samples in class i, and Ci is the set of samples in class i.

LDA…

- **Compute Scatter Matrices:** Calculate the within-class scatter matrix (SW) and the between-class scatter matrix (SB).
- Within-Class Scatter Matrix (SW): Measures the spread of data within each class.

$$SW = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$$

- Between-Class Scatter Matrix (SB): Measures the spread of class centroids from each other.

$$SB = \sum_{i=1}^{k} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

- where μ is the overall mean of the dataset and is given by:

$$\mu = \frac{1}{n} \sum_{i=1}^{k} n_i \mu_i$$

LDA…

- **Compute Eigenvectors and Eigenvalues:** Compute the eigenvectors and eigenvalues of the matrix (SW^(-1) * SB), where SW^(-1) is the inverse of the within-class scatter matrix times the between-class scatter matrix.
- Compute the eigenvectors v and corresponding eigenvalues λ *of* $S_W^{-1} S_B$
- **Sort Eigenvectors by Eigenvalues:**
- Sort the eigenvectors based on their corresponding eigenvalues in decreasing order.
- **Select Discriminant Features:**
- Choose the top k eigenvectors corresponding to the k largest eigenvalues to form the transformation matrix W.

LDA…

- **Linear Discriminant Functions:**
- The linear discriminant functions are given by:

$$LD_i(x) = x^T W_i$$

- where *Wi* is the *i*th column of the matrix *W* (formed by stacking the top *k* eigenvectors).
- **Projection onto Lower-Dimensional Space:**
- Finally, the original dataset *X* can be projected onto the lower-dimensional space using the transformation matrix *W*:

$$X_{lda} = X \cdot W$$

LDA…

Advantages	Disadvantages
Reduces data dimensions while retaining information	Assumes data follows a Gaussian distribution
Enhances class separability	Sensitive to outliers
Effective with small data sets	Limited to linear transformations
Facilitates classification	Requires labeled data
Computationally efficient	Prone to overfitting, particularly with small datasets

Application of LDA

Applications	Explanation
Face Recognition	LDA is used to extract discriminant features for face recognition.
Text Classification	In natural language processing, LDA helps classify documents into topics.
Handwriting Recognition	LDA can be used to recognize handwritten characters or symbols.
Bioinformatics	LDA aids in analyzing gene expression and biomarker discovery.
Image Analysis	LDA is utilized to categorize images and features within them.
Financial Risk Assessment	LDA helps analyze financial data to assess and mitigate risks.

Principle Component Analysis (PCA)

- Principal Component Analysis (PCA) is a widely used technique in machine learning and statistics for dimensionality reduction and feature extraction.
- It aims to find a new set of uncorrelated variables called principal components that capture the maximum variance in the original data.

PCA…

Steps of PCA:

- **Standardize the Data:**
- For *n* data points with *m* features each, calculate the mean (μ) and standard deviation (σ) for each feature.

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \quad \text{for } j = 1, 2, \ldots, m$$

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2} \quad \text{for } j = 1, 2, \ldots, m$$

- Standardize the data:

$$\text{Standardized } x_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

PCA…

- **Calculate the Covariance Matrix:**
- Compute the covariance matrix of the standardized data.
- The covariance matrix describes the relationships between different features and is used to understand the dispersion and orientation of the data.
- The covariance between two features *xj* and *xk* is given by:

$$\operatorname{cov}(x_j, x_k) = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \mu_j)(x_{ik} - \mu_k)$$

PCA…

Compute Eigenvectors and Eigenvalues:

- Calculate the eigenvectors and eigenvalues of the covariance matrix.
- Eigenvectors represent the directions of maximum variance, and eigenvalues indicate the magnitude of the variance in those directions.
- Solve the eigenvalue problem for Σ to get the eigenvectors (v) and eigenvalues (λ):

$$\Sigma v = \lambda v$$

Sort Eigenvalues:

- Arrange the eigenvalues in descending order. These eigenvalues represent the amount of variance captured by each principal component.
- Arrange eigenvalues λ in descending order:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m.$$

PCA…

Choose Principal Components:

- Select the top *k* eigenvectors based on the desired number of dimensions you want to reduce the dataset to (usually<total number of features *k*<total number of features).
- These eigenvectors are the principal components.
  - Select the *k* eigenvectors corresponding to the *k* largest eigenvalues to form the feature matrix *Vk*.

Form a Feature Matrix: Create a feature matrix by stacking the selected eigenvectors as columns. Each eigenvector represents a principal component.

  - The feature matrix *Vk* will have the selected *k* eigenvectors as columns.

PCA…

Project Data onto Lower-Dimensional Subspace:

- Multiply the standardized data by the feature matrix to obtain the transformed data in the new lower-dimensional subspace.
- The transformed data contains the same information as the original data but in a reduced number of dimensions based on the selected principal components.
- To transform the original data X of shape $n \times m$ into a new $n \times k$ matrix Y in the lower-dimensional subspace:

$$Y = X \cdot V_k$$

PCA…

Optional Reconstruction:

- If needed, you can reconstruct an approximation of the original data from the transformed data by reversing the transformation.
- However, this reconstruction is an approximation, and some information may be lost during the transformation.
- If needed, you can approximate the original data X from Y using the inverse transformation:

$$X_{\text{approx}} = Y \cdot V_k^T$$

PCA…

Advantages	Disadvantages
Simplifies complex data structures	May result in information loss due to dimension reduction
Reduces overfitting risk	Assumes linear relationships within the data
Speeds up learning algorithms	Interpretability can be challenging in high dimensions
Enhances visualization	Sensitive to feature scaling
Handles multicollinearity	Requires careful handling of missing data

Applications of PCA

Applications	Explanation
Dimensionality Reduction	PCA is widely used to reduce the number of features while maintaining essential information.
Pattern Recognition	PCA helps in recognizing patterns in data by reducing its dimensionality.
Signal Processing	PCA is used to filter noise from signals and compress data.
Image Compression	PCA is applied to reduce the size of images while retaining important features.
Genetics and Genomics Analysis	PCA is used to analyze and visualize complex genetic data, aiding in genetic studies.
Finance and Economics	PCA is used to analyze correlations among financial assets and reduce portfolio risk.

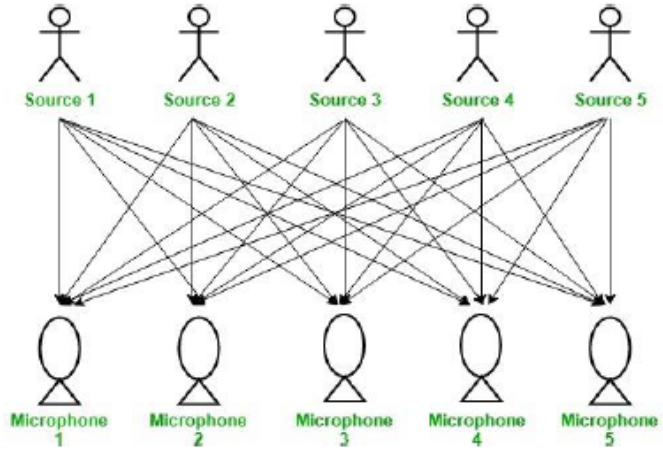
Independent Component Analysis(ICA)

- Independent Component Analysis (ICA) is a statistical technique used to separate a multivariate signal into additive subcomponents that are statistically independent or as independent as possible.
- It's widely used in signal processing, image analysis, and various other domains.

ICA…

- A simple application of ICA is the "cocktail party problem", where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room.
- Usually the problem is simplified by assuming no time delays or echoes. Note that a filtered and delayed signal is a copy of a dependent component, and thus the statistical independence assumption is not violated.

Consider *Cocktail Party Problem* or *Blind Source Separation* problem to understand the problem which is solved by independent component analysis.



Here, There is a party going into a room full of people. There is 'n' number of speakers in that room and they are speaking simultaneously at the party. In the same room, there are also 'n' number of microphones placed at different distances from the speakers which are recording 'n' speakers' voice signals.

ICA…

- Hence, the number of speakers is equal to the number must of microphones in the room. Now, using these microphones' recordings, we want to separate all the 'n' speakers' voice signals in the room given each microphone recorded the voice signals coming from each speaker of different intensity due to the difference in distances between them. Decomposing the mixed signal of each microphone's recording into independent source's speech signal can be done by using the machine learning technique, independent component analysis.

ICA…

Advantages	Disadvantages
Unmixes mixed signals into statistically independent components	Assumes statistical independence, which might not always hold
Useful in separating sources in signal processing	Sensitive to noise in the data
Can discover underlying hidden factors	Non-unique solutions may occur
Widely used in various domains such as image processing	Computational complexity increases with the number of components
Effective in feature extraction	Requires a large amount of data for accurate results

Applications of ICA

Applications	Explanation
Blind Source Separation	ICA helps in separating a mixture of signals into its original sources without prior knowledge.
Image Deblurring and Denoising	ICA can be used to separate image signals from noise and improve image quality.
Speech and Audio Processing	ICA is applied to separate mixed audio sources into individual components like speech or music.
Biomedical Signal Processing	ICA is utilized to separate various physiological signals, aiding in medical diagnosis.
Financial Time Series Analysis	ICA is used to extract meaningful and independent factors from financial time series data.
Neural Network Initialization	ICA can be used to initialize weights in neural networks, enhancing learning performance.

Difference Between PCA and ICA

Principal Component Analysis	Independent Component Analysis
It reduces the dimensions to avoid the problem of overfitting.	It decomposes the mixed signal into its independent sources' signals.
It deals with the Principal Components.	It deals with the Independent Components.
It focuses on maximizing the variance.	It doesn't focus on the issue of variance among the data points.
It focuses on the mutual orthogonality property of the principal components.	It doesn't focus on the mutual orthogonality of the components.
It doesn't focus on the mutual independence of the components.	It focuses on the mutual independence of the components.

Expectation-Maximization Algorithm

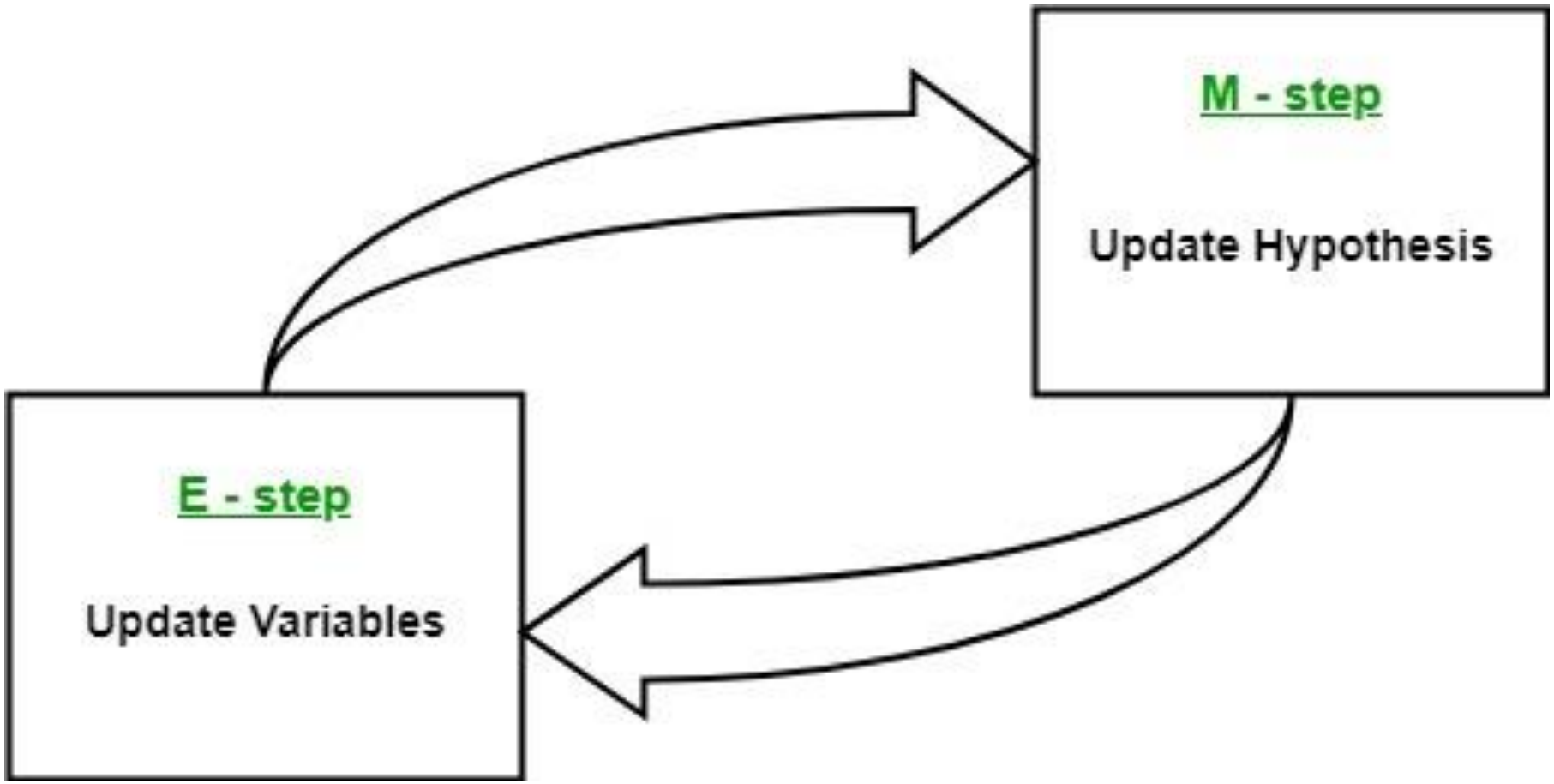
- ***Expectation-Maximization algorithm*** can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of the other observed variables) too in order to predict their values with the condition that the general form of probability distribution governing those latent variables is known to us. This algorithm is actually at the base of many unsupervised clustering algorithms in the field of machine learning.
- The essence of Expectation-Maximization algorithm is to use the available observed data of the dataset to estimate the missing data and then using that data to update the values of the parameters. Let us understand the EM algorithm in detail.

EM Algorithm

Given a set of incomplete data, consider a set of starting parameters.

- **Expectation step (E – step):** Using the observed available data of the dataset, estimate (guess) the values of the missing data.
- **Maximization step (M – step):** Complete data generated after the expectation (E) step is used in order to update the parameters.
- Repeat step 2 and step 3 until convergence.

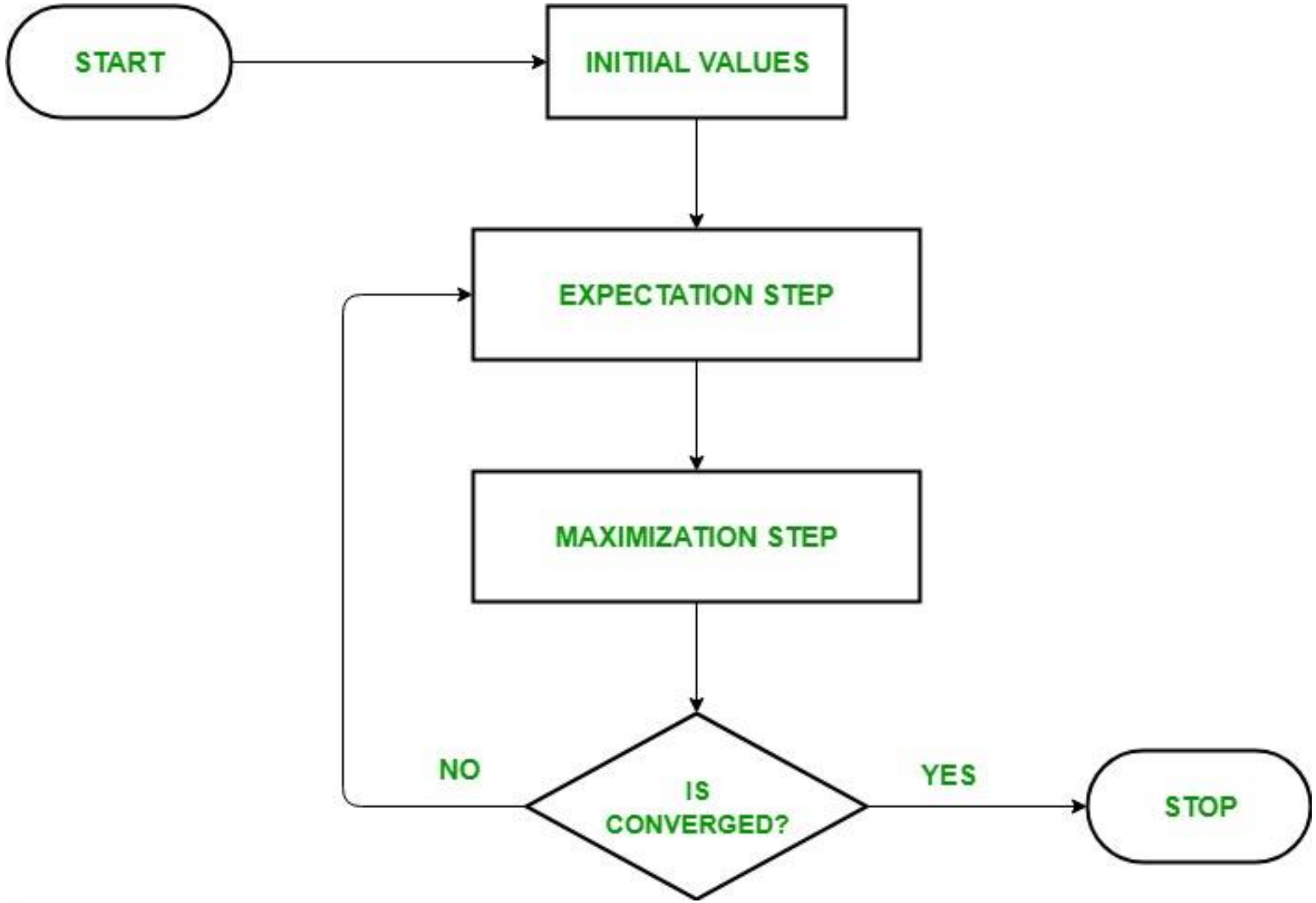
EM Algorithm



EM Algorithm

- Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.
- The next step is known as “Expectation” – step or *E-step*. In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.
- The next step is known as “Maximization”-step or *M-step*. In this step, we use the complete data generated in the preceding “Expectation” – step in order to update the values of the parameters. It is basically used to update the hypothesis.
- Now, in the fourth step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat *step-2* and *step-3* i.e. “Expectation” – step and “Maximization” – step until the convergence occurs.

EM Algorithm



Advantages of EM algorithm –

- It is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- Solutions to the M-steps often exist in the closed form.

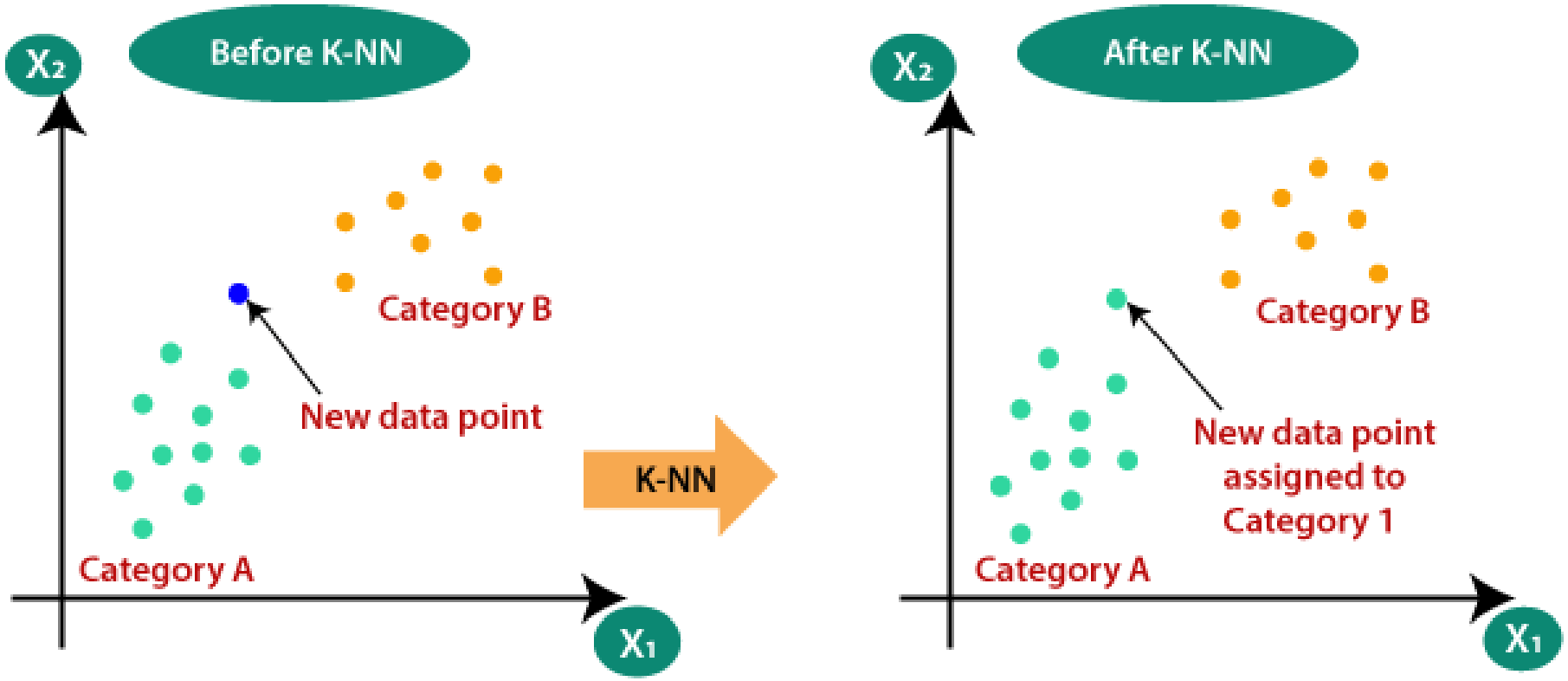
Disadvantages of EM algorithm –

- It has slow convergence.
- It makes convergence to the local optima only.
- It requires both the probabilities, forward and backward (numerical optimization requires only forward probability).

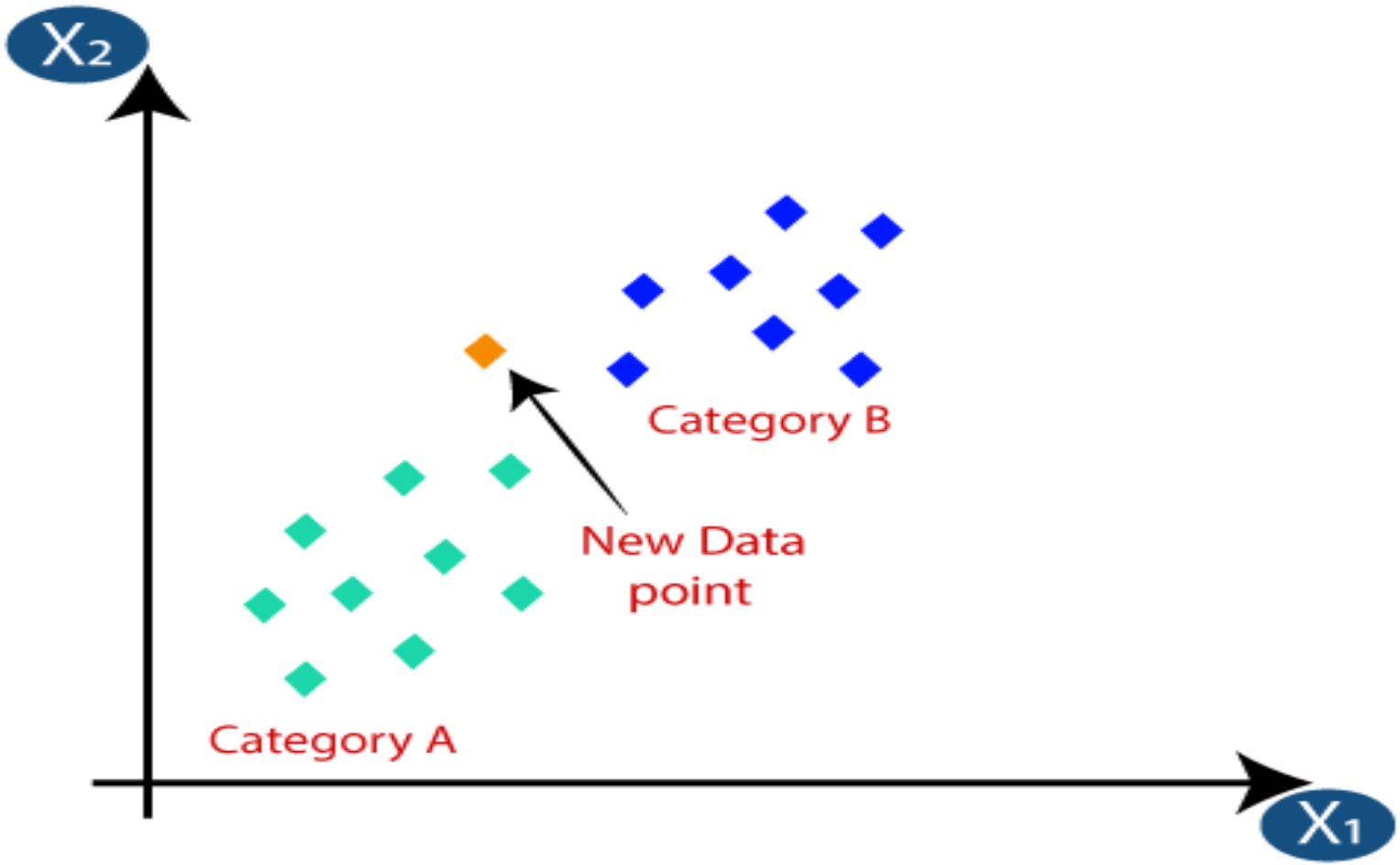
K- Nearest Neighbour

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

KNN…

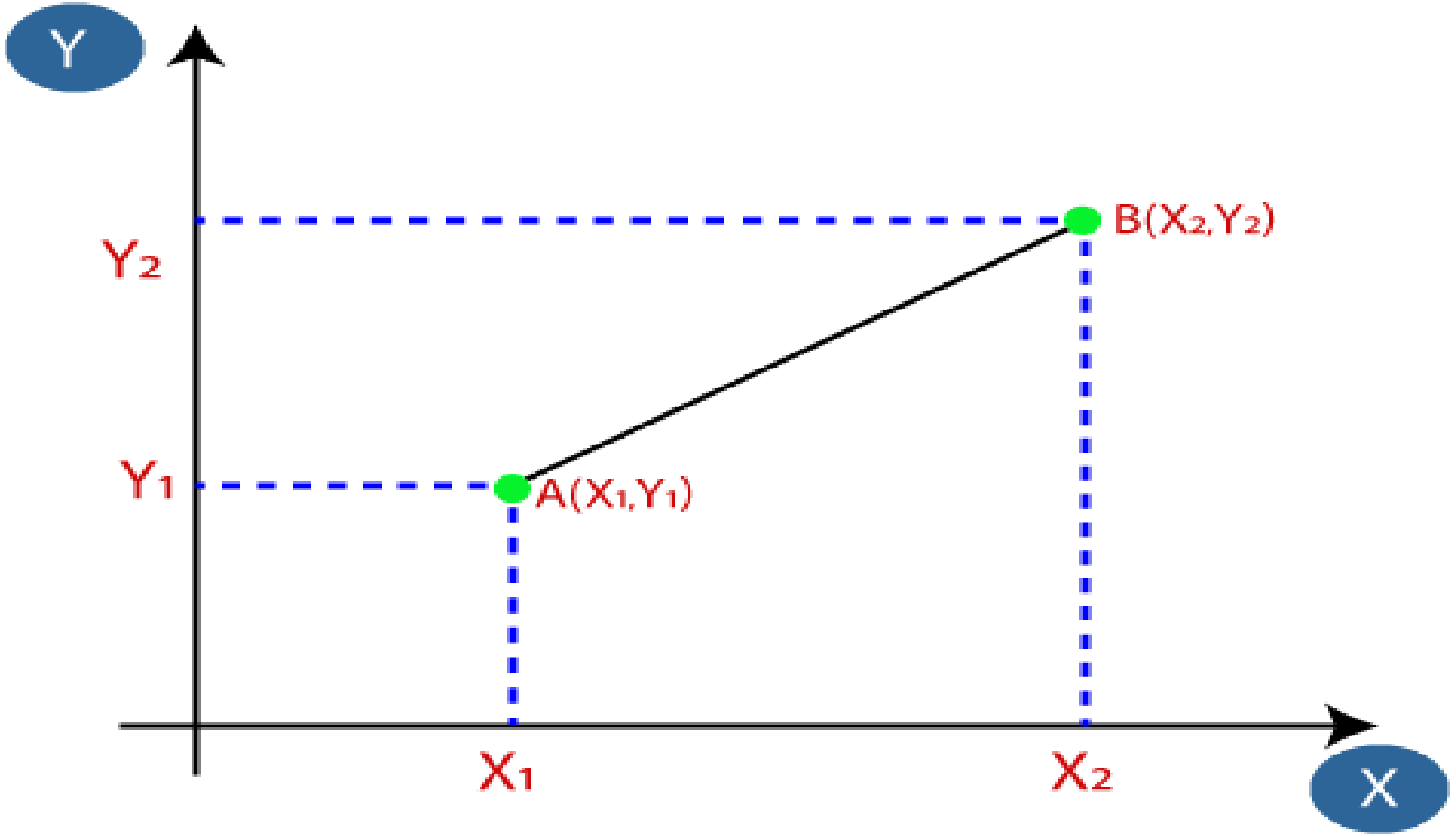

Before K-NN
X2
Category B
New data point
Category A
X1
K-NN
After K-NN
X2
Category B
New data point
assigned to
Category 1
Category A
X1

KNN…



KNN

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points.
- The Euclidean distance is the distance between two points.
- It can be calculated as:



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

How to select the value of K in the K-NN Algorithm?

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

KNN…


Knowledge Alone Liberates
ज्ञानादेव तु कैवल्यम्
Somaiya Vidyavihar

Thank You