UNIT IV ENSEMBLE TECHNIQUES AND UNSUPERVISED LEARNING

Combining multiple learners: Model combination schemes, Voting, Ensemble Learning - bagging, boosting, stacking, Unsupervised learning: K-means, Instance Based Learning: KNN, Gaussian mixture models and Expectation maximization

4.1 Combining Multiple Learners

- When designing a learning machine, we generally make some choices like parameters of machine, training data, representation, etc. This implies some sort of variance in performance. For example, in a classification setting, we can use a parametric classifier or in a multilayer perceptron, we should also decide on the number of hidden units.
- Each learning algorithm dictates a certain model that comes with a set of assumptions. This inductive bias leads to error if the assumptions do not hold for the data.
- Different learning algorithms have different accuracies. The no free lunch theorem asserts that no single learning algorithm always achieves the best performance in any domain. They can be combined to attain higher accuracy.
- Data fusion is the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation. Fusion of data for improving prediction accuracy and reliability is an important problem in machine learning.
- Combining different models is done to improve the performance of deep learning models. Building a new model by combination requires less time, data, and computational resources. The most common method to combine models is by averaging multiple models, where taking a weighted average improves the accuracy.

1. Generating Diverse Learners:

- Different Algorithms: We can use different learning algorithms to train different base-learners. Different algorithms make different assumptions about the data and lead to different classifiers.
- Different Hyper-parameters: We can use the same learning algorithm but use it with different hyper – parameters.
- Different Input Representations: Different representations make different characteristics explicit allowing better identification.
- Different training sets: Another possibility is to train different base – learners by different subsets of the training set.

4.1.1 Model Combination Schemes

- Different methods are used for generating final output for multiple base learners are Multiexpert and multistage combination.

1. Multiexpert combination.

- Multiexpert combination methods have base-learners that work in parallel.
  a) Global approach (Learner fusion) : given an input, all base- learners generate an out put and all these outputs are used, such as voting and stacking
  b) Local approach (learner selection : in mixture of experts, there is a gating model, which looks at the input and chooses one (or every few) of the learners asresponsible for generating the output.

2. **Multistage combination** : Multistage combination methods use a serial approach where the next multistage combination base – learner is trained with or tested on only the instances where the previous base – learners are not accurate enough.

- Let's assume that we want to construct a function that maps inputs to outputs from a set of known N_{train} input -output pairs.
- Let's assume that we want to construct a function that maps inputs to outputs from a set of known N_{train} input -output pairs

  $D_{train} = [(x_i,yi)] N_{train}$

  where$x_i \in \times$is a D dimensional feature input vector, $y_i \in Y$is the out put.

- Classification : When the output takes values in a discrete set of class labels

Y= {c_1; c_2; ... ... $c_x l$, where K is the number of different classes. Regression consists in predicting continuous ordered outputs,Y=R.

4.1.2 Voting

- The simplest way to combine multiple classifiers is by voting, which corresponds to take a linear combination of the learners. Voting is an ensemble machine learning algorithm.
- For regression, a voting ensemble involves make a prediction that the average of multiple other regression models.
- In classification, a hard voting ensembled involves summing the votes for crisp class labels from other models and predicting the class with the most votes, A soft voting ensemble involves summing the predicted probabilities for class labels and predicting the class label with the largest sum probability.

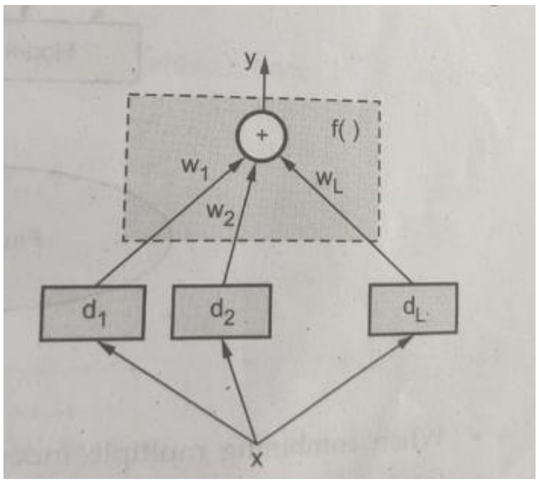


Fig.4.1.1 Base- learners with their outputs.

- In this methods, the first step is to create multiple classification/ regression models using some training dataset. Each base model can be created using different splits of the same training dataset and same algorithm, or using the same dataset with different algorithms, or any other methods.
- Fig. 4.1.2 shows general idea of Base – learners with model combiner.

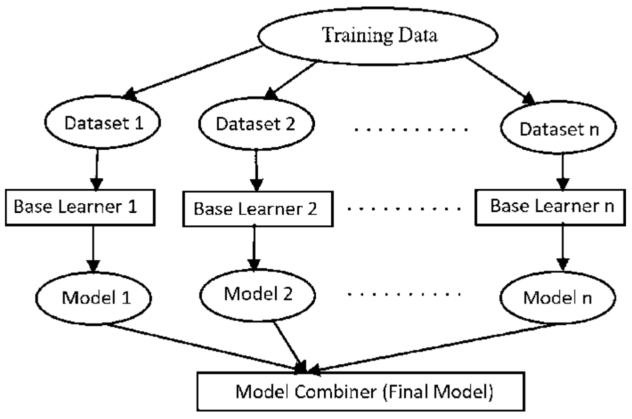


Fig. 4.1.2 Base learners with model combiner

- When combining multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, and correct decisions are reinforced. Human ensembles are demonstrably better.
- Use a single, arbitrary learning algorithm but manipulate training data to make it learn multiple models.

4.1.3 Error – Correcting Output Codes

- In Error – Correcting Output Codes main classification task is defined in terms of a number of subtasks that are implemented by the base – learners. The idea is that the original task of separating one class from all other classes may be a difficult problem.
- So, we want to define a set of simpler classification problems, each specializing in one aspect of the task, and combining these simple classifiers, we get the finalclassifier.
- Base – learners are binary classifiers having output -1/+1, and there is a code matrix W of K × L whose K rows are the binary codes of classes in terms of the L base – learners d_j.
- Code matrix W codes classes in terms of learners
- One per class L= K

$$W = \begin{bmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{bmatrix}_{K \times L}$$

- The problem here is that if there is an error with one of the base-learners, there may be a misclassification because the class code words are so similar. So the approach in error-correcting codes is to have L>K and increase the Hamming distance between the code words.

- One possibilityis pairwise separation of classes where there is a separate base –learner to separate C_i from C_j for i<j.
- Pairwise L= K (K-1) /2

$$W = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ -0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

- Full ode L=2 $^{(k-1)}$ -1

$$w = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & +1 & +1 & +1 & +1 \\ -1 & +1 & +1 & -1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 \end{bmatrix}$$

- With reasonable L, find W such that the Hamming distance between rows and between columns are maximized.
- Voting scheme are

$$y_i = (x + a)^n = \sum_{j=1}^{l} W_{ij}\, D_j$$

and then we choose the class with the highest Y$_i$

- One problem wih ECOC is that because the code matrix W is set a priori, there is no guarantee that the subtasks as defined by the columns of W will be simple.

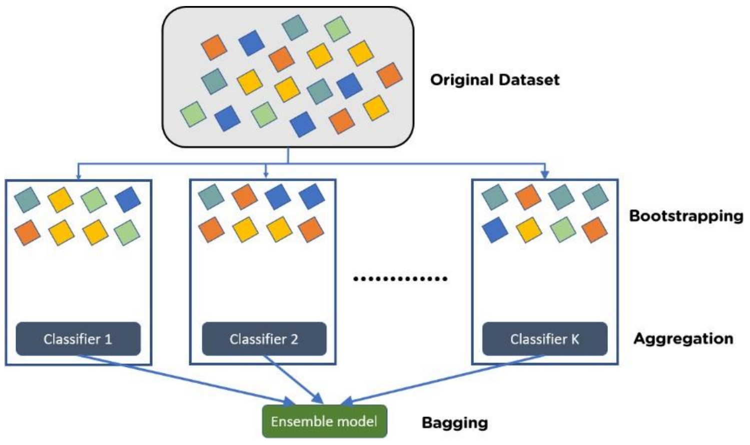
Ensemble Learning

- The idea of ensemble learning is to employ multiple learners and combine their predictions. If we have a committee of M models with uncorrelated errors, simply by averaging them the average error of a model can be reduced by a factor of M.
- Unformtunately, the key assumption that the errors due to the individual models are uncorrelated is unrealistic; in practice, the errors are typically highly correlated, so the reduction in overall error is generally small.

- Ensemble modeling is the process of running two or more related but differentanalytical models and then synthesizing he results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications.
- Ensembles of classifiers is a set of classifiers whose individual decisions combined in some way to classify new examples.
- Ensemble methods combine several decision trees classifiers to produce better predictive performance than a single decision tree classifier. The main principlebehind the ensemble model is that a group of weak learners come together to form a strong learner, thus increasing the accuracy of the model.
- Why do ensemble methods work?
- Based on one of two basic observations:
  1. Variance reduction : If the training sets are completely independent, it will always Helps to average an ensemble because this will reduce variance without affecting bias (e.g. bagging) and reduce sensitivity to individual and points.
  2. Bias reduction: for simple models, average of models has much greater capacity than single model Averaging models can reduce bias substantially by increasing capacity and control variance by Citting one component at a time

Bagging

- Bagging is also called Bootstrap aggregating. Bagging and boosting are meta-algorithms that pool decisions from multiple classifiers. It creates ensembles by repeatedly randomly resampling he training data.
- Bagging was the first effective method of ensemble learning and is one of the simplest methods of arching. The meta – algorithm, which is a special case of the model averaging, was originally designed for classification and is usually applied to decision tree models, but it can be used with any type of model for classification or regression



- Ensemble classifiers such as bagging, boosting and model averaging are known to have improved accuracy and robustness over a single model. Although unsupervised models, such as clustering, do not directly generate label prediction for each individual, they provide useful constraints for the joint prediction of a set of related objects.
- For given a training set of size n, create m samples of size n by drawing n examples from the original data, with replacement. Each bootstrap sample will on average contain 63.2% of the unique training examples, the rest are replicates. It combines the m resulting models using simple majority vote.
- In particular, on each round, the base learner is trained on what is often called a **“bootstrap replicate**” of the original training set. Suppose the training set consists of n examples. Then a bootstrap replicate is a new training set that also consists of n examples, and which is formed by repeatedly selecting uniformly at random and with replacement n examples from the original training set. This means that the same example may appear multiple times in the bootstrap replicate, or it may appear not at all.
- It also decreases error by decreasing the variance in the results due to unstable learners, algorithms (like decision trees) whose out put can change dramatically when the training date is slightly changed.
- Pseudocode:

```
1. Given Training data $(x_1,y_1)$ ........... $(x_m,y_m)$
2. For t = 1,… T :
   a. Form bootstrap replicate dataset $S_t$ by selecting m random examples from the training set with replacement.
   b. Let $h_t$ be the result of training base learn4ng algorithm on $s_t$
```

3. Output combined classifier :

$H(\times) = \text{majority } (h_1(\times) \ldots\ldots\ldots\ldots h_T(\times))$

Bagging Steps :

1. Suppose there are N observations and M features in training data set. A sample from training data set is taken randomly with replacement.
2. A subset of M features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
3. The tree is grown to the largest.
4. Above steps are repeated n times and predictions is given based on the aggregation of predictions from n number of trees.

Advantages of Bagging :

1. Reduces over – fitting of the model.
2. Handles higher dimensionality data very well.
3. Maintains accuracy for missing data.

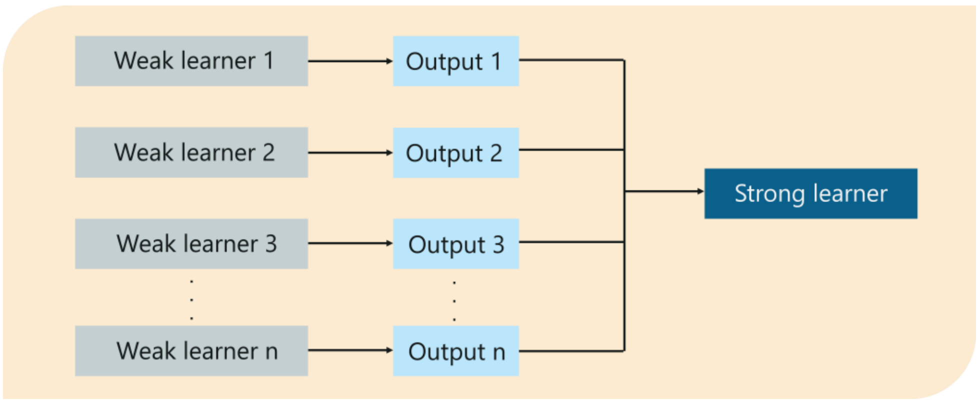
Disadvantages of Bagging:

1. Since final prediction is based on the mean predictions from subset trees, it won't give precise values for the classification and regression model.

4.2.2 Boosting

• Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor..

- Originally developed by computational learning theorists to guarantee performance improvements on fitting training data for a weak learner that only needs to generate a hypothesis with a training accuracy greater than 0.5. Final result is the weighted sum of the results of weak classifiers.



- A learner is weak if it produces a classifier that is only slightly better than random guessing, while a learner is said to be strong if it produces a classifier that achieves a low error with high confidence for a given concept.
- Revised to be a practical algorithm, AdaBoost, for building ensembles that empirically improves generalization performance. Examples are given weights. At each iteration, a new hypothesis is learned and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.
- Boosting is a bias reduction technique. It typically improves the performance of a single tee model. A reason for this is that we often cannot construct trees which are sufficiently large due to thinning out of observations in the terminal nodes.
- Boosting is then a device to come up with a more complex solution by taking linear combination of trees. In presence of high – dimensional predictors, boosting is also very useful as a regularization technique for additive or interaction modeling.
- To begin, we define an algorithm for finding the rules of thumb, which we call a weak learner. The boosting algorithm repeatedly calls this weak learner, each time feeding it a different distribution over the training data. Each call generates a weak classifier and we must combine all of these into a single classifier that, hopefully, is much more accurate than any one of the rules.
- Train a set of weak hypotheses ; h_1 ….,h_T.The combined hypothesis H is a weighted majority vote of the T weak hypotheses. During the training, focus on the examples that are misclassified.

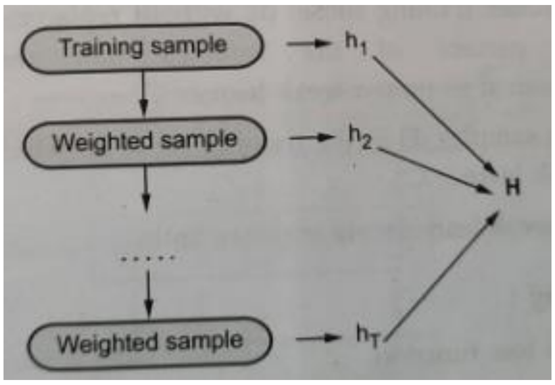


Fig. 4.2.1combined hypothesis

AdaBoost:

- AdaBoost, short for " Adaptive Boosting", is a machine learning meta – algorithm formulated by Yoav Freund and Robert Schapire who won the prestigious "Godel Prize" in 2003 for their work. It can be used in conjunction with ay other types of learning algorithms to improve their performance.
- It can be used to learn weak classifiers and final classification based on weighted vote of weak classifiers.
- It is linear classifier with all is desirable properties. It has good generalization properties.
- To use the weak learner to form a highly accurate prediction rule by calling the weak learner repeatedly on different distributions over the training examples.
- Initially, all weights are set equally, but each round the weights of incorrectly classified examples are increased so that those observations that the previously classifier poorly predicts receive greater weight on the next iteration.

Advantages of AdaBoost:

1. Very simple to implement
2. fairly good generalization
3. The prior error need not be known ahead of time.

Disadvantages of AdaBoost:

1. Suboptimal solution
2. Can over fit in presence of noise.

Boosting Steps:

1. Draw a random subset of training samples d1 without replacement from the training set D to train a weak learner C1
2. Draw second random training subset d2 without replacement from the training set add add 50 percent of the samples that were previously falsely classified / misclassified to train a weak learner C2
3. Find the training samples d3 in the training set D on which C1 and C2 disagree to train a third weak learner C3
4. Combine all the weak learners via majority voting.

Advantages of Boosting:

1. Supports different loos function.
2. Works well with interactions.

Disadvantages of Boosting:

1. Prone to over – fitting.
2. Requires careful tuning of different hyper – parameters.

4.2.3Stacking

- Stacking, sometimes called stacked generalization, is an ensemble machine learning method that combines multiple heterogeneous base or component models via a meta - model.
- The base model is trained on the complete training data, and then he meta – model is trained on the predictions of the base models. The advantage of stacking is the ability to explore the solution space with different models in the same problem.
- The stacking base model can be visualized in levels and has at least two levels of the models. The first level typically trains the two or more base learners (can be heterogeneous) and the second level might be a single meta learner that utilizes the base models predictions as input and gives the final result as output . A stacked modelcan have more than two such levels but increasing the levels doesn’t always guaranteebetter performance.
- In the classification tasks, often logistic regression is used as a meta learner, while linear regression is more suitable as a meta learner for regression – based tasks.
- Stacking is concerned with combining multiple classifiers generated by different learning algorithms $L_1, \ldots L_N$ on a single dataset S, which is composed by a feature vectors $S_i = (x_i, t_i)$

- The stacking process can be broken into two phases:
  1. Generate a set of base – level classifiers $C_1,\ldots..C_N$ where $C_i = B_i$ (S)
  2. Train a meta – level classifier to combine the output of the bae – levelclassifiers.
- Fig. 4.2.2 shows stacking frame.

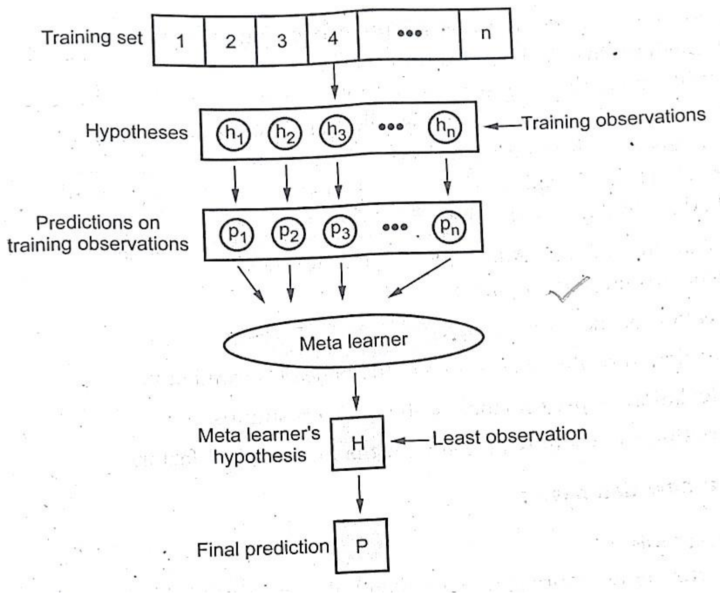


Fig.4.2.2 Stacking frame.

- The training set for he meta – level classifier is generated through a leave – one – out cross validation process.

$$\forall i = 1, \ldots., n and \forall k = 1, \ldots., N : C^i_k$$

$= L_K (S\text{-}s_i)$

- The learned classifiers are then used to generate predictions for s_i: $y^k_i = C^i_k(\times_i)$
- The meta – level dataset consists of examples of the form $\hat{y}^1_k, \ldots \hat{y}^n_i), y_i)$, where the features are the predictions of the base – level classifiers and the class is the correct class of the example in hand.
- Why do ensemble methods work?
- Based on one of two basic observations:
  1. **Variance reduction**: If the training sets are completely independent, it will always helps to average an ensemble because this will reduce variance without affecting bias (e.g. – bagging) and reduce sensitivity to individual data points.
  2. **Bias reduction** : For simple models, average of models has much greater capacity than single model Averaging models can reduce bias substantially by increasing capacity and control variance by Citting one component at a time.

4.2.4Adaboost

- AdaBoost also referred to as adaptive boosting is a method in Machine Learning used as an ensemble method. The maximum not unusual algorithm used with adaBoost is selection trees with one stage meaning with decision trees with most effective I split. These trees also are referred to as decision stumps.

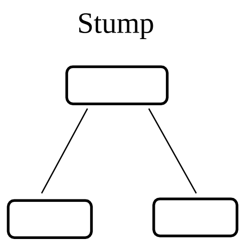


Fig. 4.2.3 Adaboost

- The working of the AdaBoost version follows
  the beneath – referred to path :
  - Creation of the learner.
  - Calculation of the total error via the beneath formulation.
  - Calculation of performance of the decision stumps.
  - Updating the weights in line with the misclassified factors.

Creation of a new database :

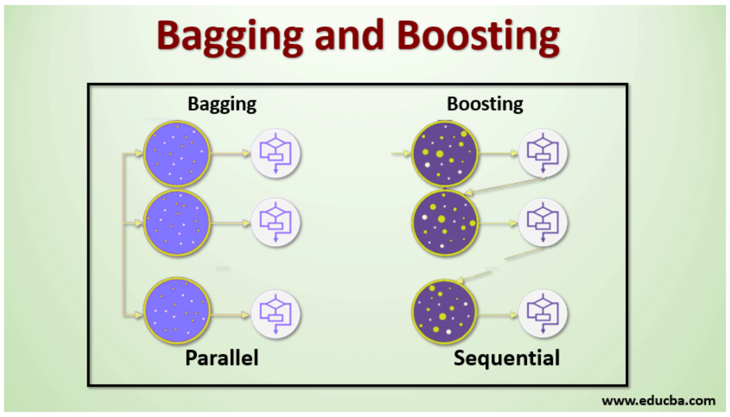
AdaBoost ensemble:

- In the ensemble approach, we upload the susceptible fashions sequentially and then teach them the use of weighted schooling records.
- We hold to iterate the process till we gain the advent of a pre-set range of vulnerable learners or we can not look at further improvement at the dataset. At the end of the algorithm, we are left with some vulnerable learners with a stage fee.

4.2.5 Difference between Bagging and Boosting

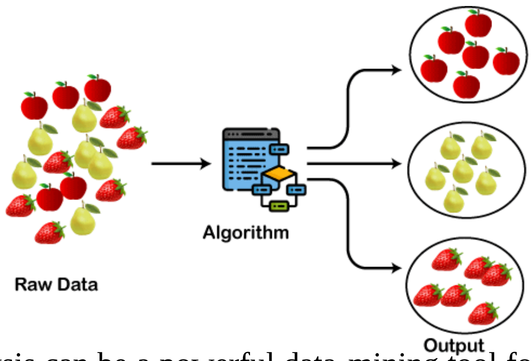
Sr. No.	Bagging	Boosting
1.	Bagging is a technique that builds multiple homogeneous models from Different subsamples of the same training dataset to obtain more accurate predictions than its individual models	Boosting refers to a group of algorithms that utilize weighted averages to make weak learning algorithms stronger learning algorithms.
2	Learns them independently from each other in parallel	Learns them sequentially in very adaptative way
3	It helps in reducing variance.	It helps in reducing bias and

		variance.
4.	Every model receives an equal weight.	Models are weighted by their performance.



Clustering

- **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters).



- Cluster analysis can be a powerful data-mining tool for any organization that needs to identity discrete groups of customers, sales transaction, or other types of behaviors and things. For example, insurance providers use cluster analysis to detect fraudulent claims and banks used it for credit scoring.

- Cluster analysis uses mathematical models to discover groups of similar customers based on the smallest variations among customers within each group.
- Cluster is a group of objects that belongs to the same class. In another words the similar object are grouped in one cluster and dissimilar are grouped in other cluster.
- Clustering is a process of partitioning a set of data in a set of meaningful subclasses. Every data in the sub class shares a common trait. It helps a user understand the natural grouping or structure in a data set.
- Various types of clustering methods are partitioning methods, hierarchical clustering, fuzzy clustering, density based clustering and model based clustering.
- Cluster anlysis is process of grouping a set of data objects into clusters.

Desirable properties of a clustering algorithm are as follows:

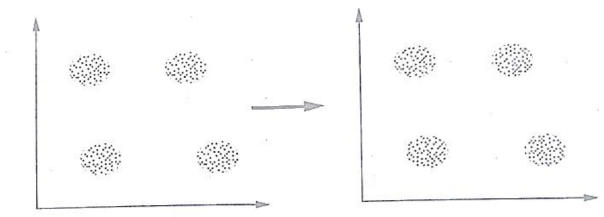
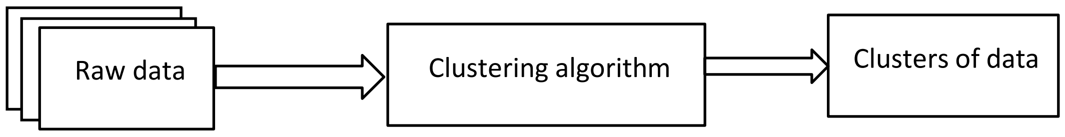


Fig. 4.3.1

1. Scalability (in terms of both time and space)
2. Ability to deal with different data types
3. Minimal requirements for domain knowledge to determine input parameters.
4. Interpretability and usability.

- Clustering of date is a method by which large sets of data are grouped into clusters of smaller sets of similar data. Clustering can be considered the most important unsupervised learning problem.
- A cluster is therefore a collection of objects which are “similar” between them and are dissimilar” to the objects belonging to other clusters. Fig. 4.3.1 shows cluster.
- In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance : two or more objects belong to the same cluster if they are “close” according to a given distance (in this case geometrical distance) This is called **distance – based clustering.**



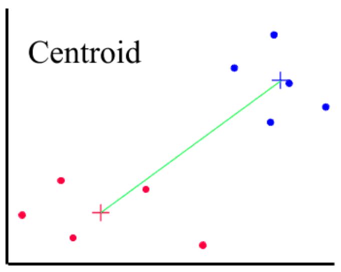


Fig. 4.3.2 cluster centroid

- Clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.
- A clustering algorithm attempts to find natural groups components or data based on some similarity. Also, the clustering algorithm finds the centroid of a group of data sets.

- To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster.
- Cluster centroid : The centroid of a cluster is a point whose parameter values are the means of the parameter values of all the points in the cluster. Each cluster has a well defined centroid.
- Distance : The distance between two points is taken as a common metric to as see the similarity among he components of population. The commonly used distance measure is the Euclidean metric which defines the distance between two points p= (p_1, p_2…) and q=(q_1, q_2,…) is given by,

$$D = \sum_{i=1}^{k} (P_i - q_i)^{2}$$

- The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply criterion, in such a way thatthe result of the clustering will suit their needs.
- Clustering analysis helps construct meaningful partitioning of a large set of objects cluster analysis has been widely used in numerous application, including, pattern recognition, data analysis, image processing etc.
- Clustering algorithms may be classified as listed below:

1. Exclusive clustering
2. Overlapping clustering
3. Hierarchical clustering
4. Probabilistic clustering.

- A good clustering method will produce high quality clusters high intra – class similarity and low inter – class similarity. The quality of a clustering result depends on both the similarity measure used by the method and its implementation. Thequality of a clustering method is also measured by it' s ability to discover some or all of the hidden patterns.

- Clustering techniques types : The major clustering techniques are
  a) Partitioning methods
  b) Hierarchical methods
  c) Density – based methods.

4.3.1 Unsupervised Learning: K-means

K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

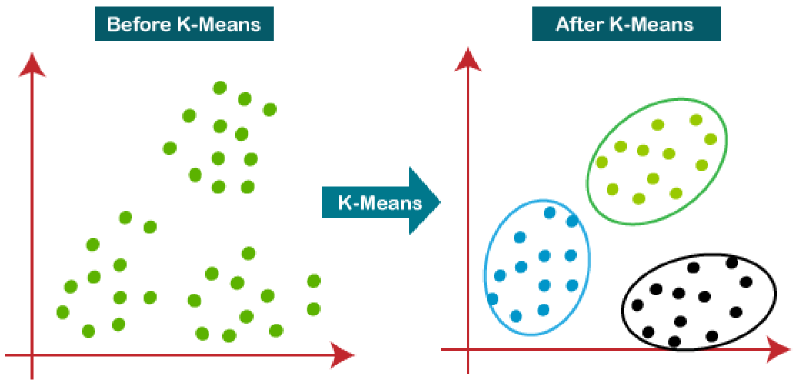
- ❖ K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- ❖ It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.
- ❖ It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
- ❖ It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
- ❖ The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value ofk should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- o Determines the best value for K center points or centroids by an iterative process.
- o Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

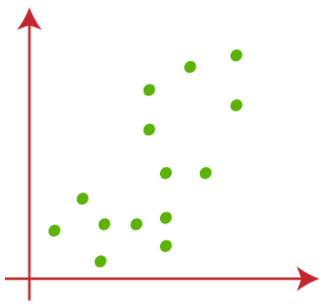
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

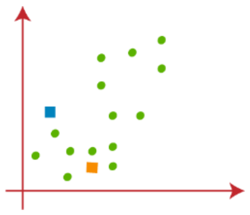
Step-7: The model is ready.

Let's understand the above steps by considering the visual plots:

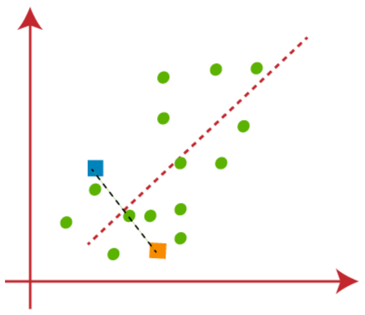
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



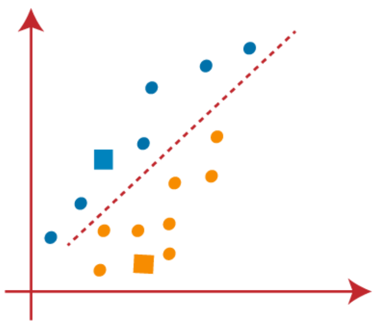
- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider thebelow image:



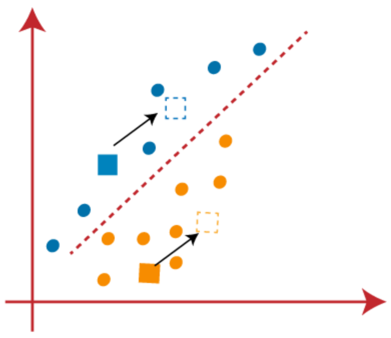
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between boththecentroids. Consider the below image:



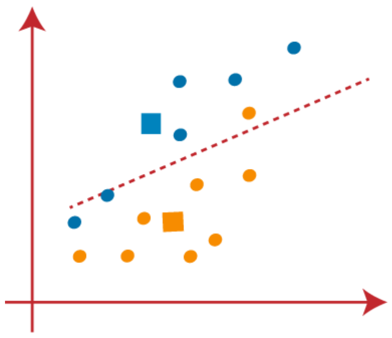
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



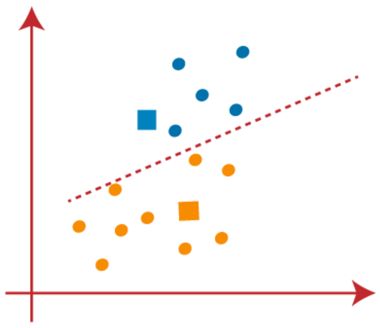
- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



- Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

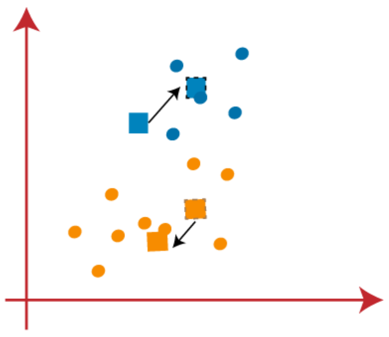


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

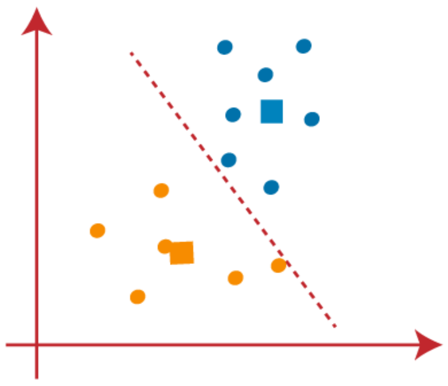


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

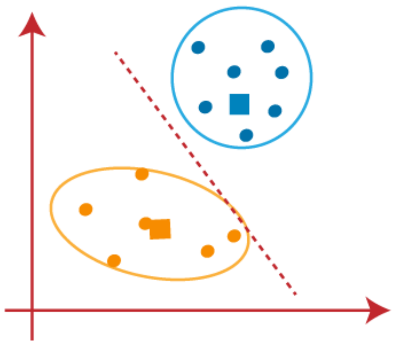
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



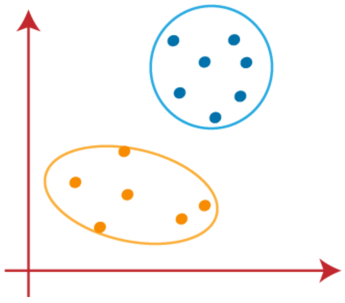
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$\text{WCSS} = \sum_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2 + \sum_{P_i \text{ in Cluster2}} \text{distance}(P_i\ C_2)^2 + \sum_{P_i \text{ in CLuster3}} \text{distance}(P_i\ C_3)^2$$

In the above formula of WCSS,

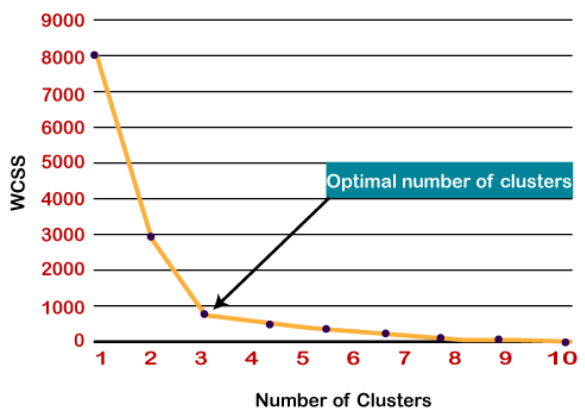
$\sum_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



Instance Based Learning: KNN

- K. Nearest Neighbour is one of only Machine Learning, algorithms based totally on supervised learning approach.
- K-NN algorithm assumes the similarity between the brand new case/facts and available instances and Placed the brand new case into the category that is maximum similar to the to be had classes.
- K-NN set of rules shops all of the to be had facts and classifies a new statistics point based at the similarity. This means when new data seems then it may be effortlessly categorized into a properly suite class by using K-NN algorithm.
- K-NN set of rules can be used for regression as well as for classificationhowever normally it's miles used for the classification troubles.
- K-NN is a non-parametric algorithm, because of this it does no longer makes any assumption on underlying data.
- It is also referred to as a lazy learner set of rules because it does no longer research from the training set immediately as a substitute it shop the dataset andat the time of class, it plays an movement at the dataset.
- The KNN set of rules at the schooling section simply stores the dataset and when it gets new data, then it classifies that statistics into a class that is an awful lot similar to the brand new data.
- Example : Suppose, we've an picture of a creature that looks much like cat and dog, but we want to know both it is a cat or dog. So for this identity, we are able

to use the KNN algorithm, because it works on a similarity degree. Our KNN version will discover the similar features of the new facts set to the cats and dogs snap shots and primarily based on the most similar functions it will place itin both cat or canine class.

Why do We Need KNN ?

- Suppose there are two categories, i.e., category A and category B and we've a brand new statistics point x1, so this fact point will lie within of these classes. To solve this sort of problem, we need a K-NN set of rules. With the help of K-NN, we will without difficulty discover the category or class of a selected dataset. Consider the underneath diagram :

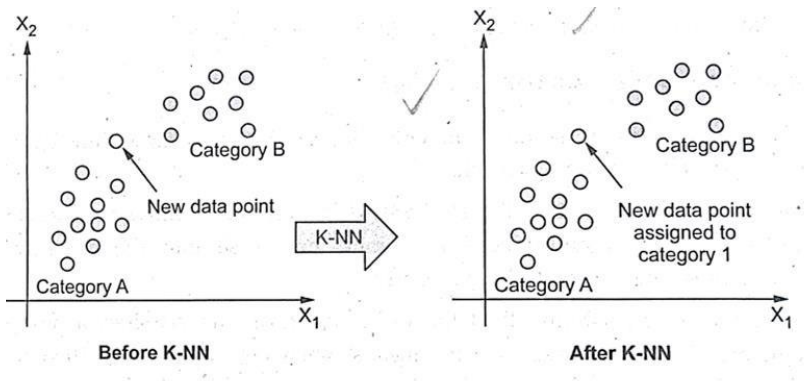


Fig.9.4.1 why do we need KNN

How Does KNN Work?

- The K-NN working can be explained on the basis of the below algorithm :

Step -1 :Select the wide variety K of the acquaintances.

Step -2 :Calculate the Euclidean distance of K variety of friends.

Step -3 :Take the K nearest neighbors as according to the calculated Euclidean distance.

Step -4 :Among these ok pals, count number the number of the data points in each class.

Step -5 :Assign the brand new records points to that category for which the quantity of the neighbor is maximum.

Step -6 :Our model is ready.

- Suppose we've got a brand new information point and we want to place it in the required category, Consider the under image

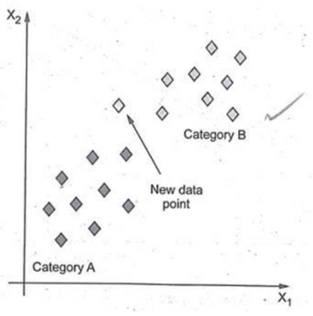
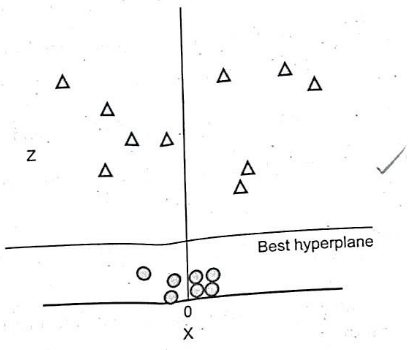


Fig4.4.2 KNN example

- Firstly, we are able to pick the number of friends, so we are able to select the ok=5
- Next, we will calculate the Euclidean distance between the facts points. The Euclidean distance is the gap between points, which we've got already studied in geometry. It may be calculated as :



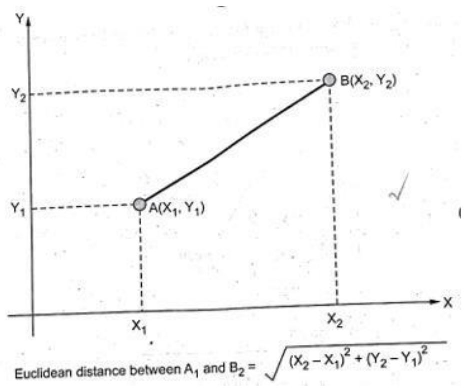

Y
Y_2
$B(X_2, Y_2)$
Y_1
$A(X_1, Y_1)$
X
X_1
X_2
Euclidean distance between A_1 and B_2 = $\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$

Fig4.4.3 KNN example continue

- By calculating the Euclidean distance we got the nearest acquaintances, as 3 nearest neigh bours in category A and two nearest associates in class B. Consider the underneath image.

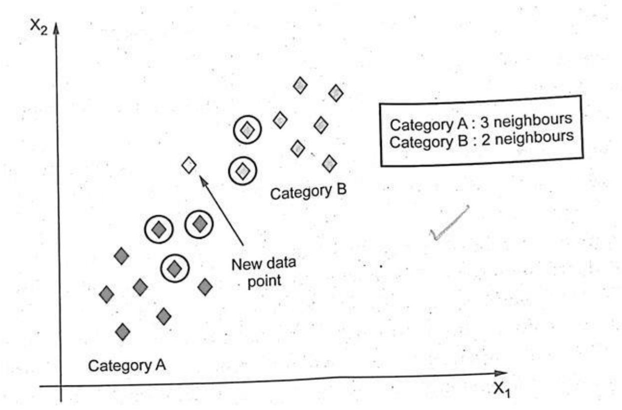


Fig:**KNN example continue**

- As we are able to see the three nearest acquaintances are from category A, subsequently this new fact point must belong to category A.

4.4.3 Difference between K- means and KNN

Sr. No	**K-means**	**KNN**
1	K-Means is an unsupervised machine learning algorithm used for clustering.	KNN is a supervised machine learning algorithm used for classification.
2	K- Means is an eager learner	k-NN is a lazy learner
3	It is used for Clustering	It is used mostly for Classification, and sometimes even for Regression
4	K in K-Means is the number of clusters the algorithm is trying to identify/ learn from the data	K' in KNN is the number of nearest neigh bours used classify or predict a test sample
5	K-means require unlabeled data. It gathers and groups data into k number of clusters.	

Gaussian Mixture models and Expectation Maximization

- Gaussian Mixture Models is a "soft" clustering algorithm, where each point probabilistically "belongs" to all clusters. This is different than k-means where each point belongs to one cluster.

- The Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mix of Gaussian distributions with unknown parameters.
- For example, in modeling human height data, height is typically modeled as a normal distribution for each gender with a mean of approximately 5’10” for males and 5’5” for females. Given only the height data and not the gender assignments for each data point, the distribution of all heights would follow the sum of two scaled (different variance) and shifted (different mean) normal distributions. A model making this assumption is an example of a Gaussian mixture model.
- Gaussian mixture models do not rigidly classify each and every instance into one class or the other. The algorithm attempts to produce K-Gaussian distributions that would take into account the entire training space. Every point can be associated with one or more distributions. Consequently, the deterministic factor would be the probability that each point belongs to a certain Gaussian distribution.
- GMMs have a variety of real - world applications. Some of them are listed below.
  a) Used for signal processing
  b) Used for customer churn analysis
  c) Used for language identification
  d) Used in video game industry
  e) Genre classification of songs

4.5.1 Expectation – maximization

- In Gaussian mixture models, an expectation-maximization method is a powerful tool for estimating he parameters of a Gaussian mixture model. The expectation is termed E and maximization is termed M.

- Expectation is used to find the Gaussian parameters which are used to represent each component of gaussian mixture models. Maximization is termed M and it is involved in determining whether new data points can be added or not.
- The Expectation – Maximization (EM) algorithm is used in maximum likelihood estimation where the problem involves two sets of random variables of which one, X, is observable and the other, Z, is hidden.
- The goal of the algorithm is to find the parameter vector ∅ that maximizes thelikelihood of the observed values of X, L (∅ | X)

- But in cases where this is not feasible, we associated the extra hidden variables Z and express the underlying model using both, to maximize the likelihood of the joint distribution of X and Z the complete likelihood L_c (∅|X,Z)
- Expectation -maximization (EM) is an iterative method used to find maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved, also called latent, variables.
- EM alternates between performing an expectation € step, which computes an expectation of the likelihood by including the latent variables as if they were obserrved, and maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found in the Estep.
- The Parameters found on the M step are then used to start another E step, and the process is repeated until some criterion is satisfied. EM is frequently used for data clustering like for example in Gaussian mixtures.
- In the **Expectation step**, find the expected values of the latent variables (here you need to use the current parameter values)
- In the **Maximization step**, first plug in the expected values of the latent variables in the log-likelihood of the augmented data. The maximize this log-likelihood to reevaluate the parameters.
- Expectation – Maximization (EM) is a technique used in point estimation. Given a set a observable variables X and unknown (latent) variables Z we want to estimate parameters ∅ in a model.
- The expectation maximization (EM) algorithm is a widely used maximum likely- hood estimation procedure for statistical models when the values of some of the variables in the model are not observed
- The EM algorithm is an elegant and powerful method for finding the maximum likelihood of models with hidden variables. The key concept in the EM algorithm is that it iterates between the expectation step (E-setp) and maximization step (M-step) until convergence.
- In the E-step, the algorithm estimates the posterior distribution of the hidden variables Q given the observed data and the current parameter settings; and in the M-step the algorithm calculates the ML parameter settings with Q fixed.

- At the end of each iteration the lower bound on the likelihood is optimized for the given parameter setting (M-step) and the likelihood is set to the bound (E-setp) which guarantees an increase in the likelihood and convergence to a local maximum, or global maximum if the likelihood function is unimodal.
- Generally, EM works best when the fraction of missing information is small and the dimensionality of he data is not too large. EM can require many iterations, and higher dimensionality can dramatically slow down the E-setp.
- EM is useful for several reasons: conceptual simplicity, ease of implementation, and the fact that each iteration improves l (ø). The rate of convergence on the first few steps is typically quite good, but can become excruciatingly slow as you approach local optima.
- Sometimes the M- step is a constrained maximization, which means that there are constraints on valid solutions not encoded in the function itself.
- Expectation maximization is an effective technique that is often used in data analysis to manage missing data. Indeed, expectation maximization overcomes some of the limitations of other techniques, such as mean substitution or regression substitution. These alternative techniques generate biased estimates – and specifically, underestimate the standard errors. Expectation maximization overcomes this problem.