MACHINE LEARNING
23DS4PCMLG

UNIT 3

- **Probabilistic Learning(Bayesian Learning):**
- Bayes Theorem and Concept Learning
- Maximum Likelihood
- Minimum Description Length Principle
- Bayes Optimal Classifier
- Gibbs Algorithm
- Naïve Bayes Classifier
- Bayesian Belief Network
- EM Algorithm.

- Bayesian reasoning provides a probabilistic approach to inference.
- It is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data.
- It is important to machine learning because it provides a quantitative approach to weighing the evidence supporting alternative hypotheses.

Relevance of Bayesian Learning

1. Bayesian learning algorithms that calculate explicit probabilities for hypotheses (eg: naive Bayes classifier), are among the most practical approaches to certain types of learning problems.
2. Bayesian methods are important to in study of machine learning as they provide a useful perspective for understanding many learning algorithms that do not explicitly manipulate probabilities.
   a. use a Bayesian analysis to justify a key design choice in neural network learning algorithms:
   b. choosing to minimize the sum of squared errors
   c. derive an alternative error function, cross entropy, that is more appropriate than sum of squared errors when learning target functions that predict probabilities.
   d. use a Bayesian perspective to analyze the bias of decision tree learning algorithms that favour short decision trees

Features of Bayesian learning methods include:

1. Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.
2. Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
   - Prior knowledge is provided by asserting

   i. a prior probability for each candidate hypothesis, and

   ii. a probability distribution over observed data for each possible hypothesis.

Use cases:

1. Bayesian methods can accommodate hypotheses that make probabilistic predictions(e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").
2. New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
3. Bayesian methods can provide a standard of optimal decision making against which other practical methods can be measured.

Challenges while using Bayesian methods

1. They require initial knowledge of many probabilities
   - When probabilities are not known in advance they are estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
2. Significant computational cost is required to determine the Bayes optimal hypothesis.

BAYES THEOREM

- Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.
- Determining the best hypothesis from some space H, given the observed training data D.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- ***P(h)*** denotes the initial probability(***prior probability***) of h
- ***P(D)*** denotes the prior probability that training data D will be observed
- ***P(D|h)*** to denote the probability of observing data ***D*** given some world in which hypothesis ***h*** holds.
- ***P (h|D),*** ***h*** holds the observed training data ***D,*** called the ***posterior probability***

Example

- Consider a medical diagnosis problem in which there are two alternative hypotheses:
- (1) that the patient has a particular form of cancer and
- (2) that the patient does not.
- The available data is from a particular laboratory test with two possible outcomes: + (positive) and -(negative).
- Prior knowledge that over the entire population of people only .008 have this disease.
- The test returns a correct positive result in only 98% of the cases in which the disease is actually present and a correct negative result in only 97% of the cases in which the disease is not present.
- In other cases, the test returns the opposite result.

$$P(cancer) = .008, \quad P(\neg cancer) = .992$$
$$P(\oplus|cancer) = .98, \quad P(\ominus|cancer) = .02$$
$$P(\oplus|\neg cancer) = .03, \quad P(\ominus|\neg cancer) = .97$$

- Suppose we observe a new patient for whom the lab test returns a positive result.
- Should we diagnose the patient as having cancer or not?
- The maximum a posteriori hypothesis can be found

$$P(\oplus|cancer)P(cancer) = (.98).008 = .0078$$
$$P(\oplus|\neg cancer)P(\neg cancer) = (.03).992 = .0298$$

- Therefore, the result of Bayesian inference $h_{MAP} = \neg cancer$ depends strongly on the prior probabilities, which must be available to apply the method directly.

Summary of basic probability formulas

- *Product rule*: probability $P(A \wedge B)$ of a conjunction of two events A and B

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- *Sum rule*: probability of a disjunction of two events A and B

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- *Bayes theorem*: the posterior probability $P(h|D)$ of h given D

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- *Theorem of total probability*: if events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$, then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

BRUTE-FORCE MAP LEARNING algorithm

1. For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h|D)$$

Maximum Likelihood And Least-squared Error Hypotheses

- Consider the problem of learning a continuous-valued target function – faced by many learning approaches such as neural network learning, linear regression, and polynomial curve fitting.
- A Bayesian analysis will show that ***under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a maximum likelihood hypothesis.***

- Consider the following problem setting.
- Learner L considers an instance space X and a hypothesis space H consisting of some class of real-valued functions defined over X.
- The problem faced by L is to learn an unknown target function f: X→ R drawn from H.
- A set of ***m*** training examples is provided, where the target value of each example is corrupted by random noise drawn according to a Normal probability distribution.
- Each training example is a pair of the form (x_i, d_i) where $d_i = f(x_i) + e_i$.
- Here $f(x_i)$ is the noise-free value of the target function and e_i is a random variable representing the noise.
- The task of the learner is to output a maximum likelihood hypothesis, (a MAP hypothesis) assuming all hypotheses are equally probable a priori.

- Normal distribution is a smooth, bell-shaped distribution that can be completely characterized by its mean and its standard deviation.
- Why is it reasonable to choose the Normal distribution to characterize noise?

1. It allows for a mathematically straightforward analysis.
2. The smooth, bell-shaped distribution is a good approximation to many types of noise in physical systems.

Maximum Likelihood Hypotheses For Predicting Probabilities

- When using Maximum Likelihood Estimation (MLE) for predicting probabilities, the goal is to estimate the parameters of a probabilistic model such that the observed data is most probable under the model.
- This is common in various types of probabilistic models, including logistic regression, multinomial logistic regression, and other classification models.

- In neural network learning: learning to predict probabilities.
- Consider the setting to learn a nondeterministic (probabilistic) function $f: X \rightarrow \{0, 1\}$, which has two discrete output values.
- For example

a. the instance space X might represent medical patients in terms of their symptoms, and the target function $f(x)$ might be 1 if the patient survives the disease and 0 if not. OR

b. X might represent loan applicants in terms of their past credit history, and $f(x)$ might be 1 if the applicant successfully repays their next loan and 0 if not.

- In both of these cases expect f to be probabilistic.

- For example, among a collection of patients exhibiting the same set of observable symptoms, we might find that 92% survive, and 8% do not.
- This unpredictability could arise from

a) our inability to observe all the important distinguishing features of the patients or

b) some genuinely probabilistic mechanism in the evolution of the disease.

- Given this problem setting, to learn a neural network (or other real-valued function approximator) whose output is the probability that f(x) = 1.
- In other words, we seek to learn the target function, f': X [0, 1], such that f '(x) = P(f (x) = 1).
- In the medical patient example, if x is one of those indistinguishable patients of which 92% survive, then f'(x) = 0.92
- whereas the probabilistic function f (x) will be equal to 1 in 92% of cases and equal to 0 in the remaining 8%.

How can we learn f' using a neural network?

1. Brute force to first collect the observed frequencies of 1's and 0's for each possible value of x.
2. Then train the neural network to output the target frequency for each x.
3. Obtain an expression for P(D|h).

- Let us assume the training data D is of the form D = {(x_1, d_1) . . . (x_m, d_m)}, where d_i is the observed 0 or 1 value for f (x_i).
- Treating both x_i and d_i as random variables, and assuming that each training example is drawn independently, we can write P(D|h) as

$$P(D|h) = \prod_{i=1}^{m} P(x_i, d_i|h)$$

Gradient Search to Maximize Likelihood in a Neural Net

- In neural networks, parameters (weights and biases) are optimized using gradient-based optimization techniques to maximize the likelihood (or equivalently, to minimize the negative log-likelihood or another loss function).
- The most commonly used method is **gradient descent** and its variants.

STEPS

1. Define the Model: Specify the architecture of the neural network (e.g., number of layers, types of layers, activation functions).
2. Initialize Parameters: Initialize the weights and biases of the network, typically using small random values.
3. Forward Propagation: Pass the input data through the network to obtain the predicted outputs.
4. Compute the Loss: Calculate the loss function, which measures the discrepancy between the predicted outputs and the actual target values.
5. Backward Propagation: Compute the gradients of the loss function for each parameter in the network using the chain rule (backpropagation).
6. Update Parameters: Adjust the parameters in the direction that reduces the loss using gradient descent or one of its variants.

Minimum Description Length(MDL) Principle

- CHOOSE THE SHORTEST EXPLANATION FOR THE OBSERVED DATA.
- The Minimum Description Length principle recommends choosing the hypothesis that minimizes the description length of the hypothesis and the description length of the data given the hypothesis.
- Bayes's theorem and basic results from information theory can be used to provide a rationale for this principle.

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(D|h)P(h)$$

which can be equivalently expressed in terms of maximizing the $\log_2$

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}} \log_2 P(D|h) + \log_2 P(h)$$

or alternatively, minimizing the negative of this quantity

$$h_{MAP} = \underset{h \in H}{\operatorname{argmin}} -\log_2 P(D|h) - \log_2 P(h)$$

Bayes Optimal Classifier

- WHAT IS THE MOST PROBABLE ***CLASSIFICATION*** OF THE NEW INSTANCE GIVEN THE TRAINING DATA?
- Consider a hypothesis space containing three hypotheses, h_1, h_2, and h_3.
- Suppose the posterior probabilities of these hypotheses given the training data are .4, .3, and .3 respectively.
- Thus, h_1 is the MAP hypothesis.
- Suppose a new instance x is encountered, classified positive by h_1, but negative by h_2 and h_3.
- Taking all hypotheses into account, the probability that x is positive is .4, and the probability that it is negative is therefore .6.
- The most probable classification (negative) in this case is different from the classification generated by the MAP hypothesis.

- The most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.
- If the possible classification of the new example can take on any value v_j from some set V, then the probability $P(v_j|D)$ that the correct classification for the new instance is v_j is

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- The optimal classification of the new instance is the value v_j, for which $P(v_j|D)$ is maximum.

$$\underset{v_j \in V}{\operatorname{argmax}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- Any system that classifies new instances according to Equation (prev slide) is called a ***Bayes* optimal *classifier***, or Bayes optimal learner.
- This method maximizes the probability that the new instance is classified correctly, given
  - the available data,
  - hypothesis space, and
  - prior probabilities over the hypotheses.

GIBBS Algorithm

- Bayes optimal classifier obtains the best performance that can be achieved from the given training data, but it is costly to apply.
- The expense is because it computes the posterior probability for every hypothesis in H and then combines the predictions of each hypothesis to classify each new instance.
- An alternative, less optimal method is the Gibbs algorithm
- 1. Choose a hypothesis ***h*** from H at random, according to the posterior probability distribution over H.
- 2. Use ***h*** to predict the classification of the next instance ***x.***

- Under certain conditions, the expected misclassification error for the Gibbs algorithm is at most twice the expected error of the Bayes optimal classifier
- It implies that if the learner assumes a uniform prior over H, and if target concepts are drawn from such a distribution when presented to the learner, ***then***
- ***classifying the next instance according to a hypothesis drawn at random from the current version space (according to a uniform distribution), will have an expected error at most twice that of the Bayes optimal classifier***

Naive Bayes Classifier

- In some domains, ***Naive Bayes classifier*** performance is comparable to that of neural network and decision tree learning.
- The Naive Bayes classifier applies to learning tasks

a. where each instance x is described by a conjunction of attribute values and

b. Where the target function f (x) can take on any value from some finite set V.

- A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values $(a_1, a_2 \ldots a_n)$.
- The learner is asked to predict the target value, or classification, for this new instance.

- The Bayesian approach to classifying the new instance is to assign the most probable target value, V_{MAP}, given the attribute values $(a_1, a_2 \ldots a_n)$ that describe the instance.

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

We can use Bayes theorem to rewrite this expression as

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

- The Naive Bayes classifier assumes that the attribute values are conditionally independent given the target value.
- The assumption is that given the target value of the instance, the probability of observing the conjunction $a_1, a_2 \ldots a_n$, is the product of the probabilities for the individual attributes:

Naive Bayes classifier:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

- where V_{NB} denotes the target value output by the naive Bayes classifier.

Bayesian Belief Networks

- The Naive Bayes classifier makes the assumption that the values of the attributes $a_1, a_2 \ldots\ldots a_n$, are conditionally independent given the target value v.
- This assumption reduces the complexity of learning the target function.
- However, in many cases this conditional independence assumption is restrictive.
- A Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities.

- The naive Bayes classifier assumes that ***all*** the variables are conditionally independent given the value of the target variable.
- Bayesian belief networks allow stating conditional independence assumptions that apply to ***subsets*** of the variables.
- Thus, Bayesian belief networks provide an intermediate approach

 less constraining than the assumption of conditional independence made by the naive Bayes classifier

 And more tractable than avoiding conditional independence assumptions altogether.

- A Bayesian belief network describes the probability distribution over a set of variables.
- Consider an arbitrary set of random variables $Y_1 \ldots Y_n$, where each variable Y_i can take on the set of possible values $V(Y_i)$.
- We define the ***joint space*** of the set of variables Y to be the cross product $V(Y_1)$ x $V(Y_2)$ x . . . $V(Y_n)$.
- Each item in the joint space corresponds to one of the possible assignments of values to the tuple of variables $(Y_1 \ldots Y_n)$.
- The probability distribution over this joint space is called the ***joint probability distribution.***
- A Bayesian belief network describes the joint probability distribution for a set of variables.

Conditional Independence

- Let X, Y, and Z be three discrete-valued random variables.
- X is ***conditionally independent*** of Y given Z if the probability distribution governing X is independent of the value of Y given a value for Z; that is, if

$$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

$$\text{where } x_i \in V(X),\ y_j \in V(Y), \text{ and } z_k \in V(Z)$$

- The above expression can be written as P(X|Y, Z) = P(X|Z).
- The set of variables $X_1 \ldots X_l$ is conditionally independent of the set of variables $Y_1 \ldots Y_m$ given the set of variables $Z_1 \ldots Z_n$, if

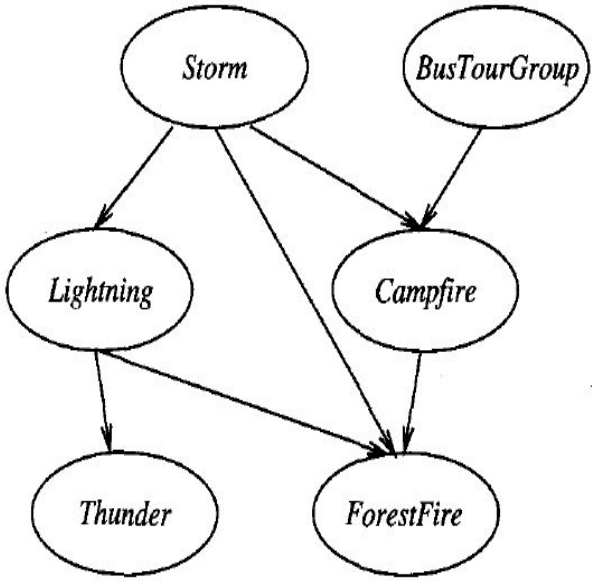
$$P(X_1 \ldots X_l | Y_1 \ldots Y_m, Z_1 \ldots Z_n) = P(X_1 \ldots X_l | Z_1 \ldots Z_n)$$

Representation

- For example, the Bayesian network in Figure represents the joint probability distribution over the boolean variables
- ***Storm, Lightning, Thunder, ForestFire, Camfire,*** and ***BusTourGroup.***
- A Bayesian network

 represents the joint probability distribution by specifying a set of conditional independence assumptions (represented by a DAG)

 with sets of local conditional probabilities.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

Campfire

- In the Bayesian network, each variable is represented by a node, and two types of information are specified.

1. The network arcs represent that the variable is conditionally independent of its nondescendants in the network given its immediate predecessors in the network.

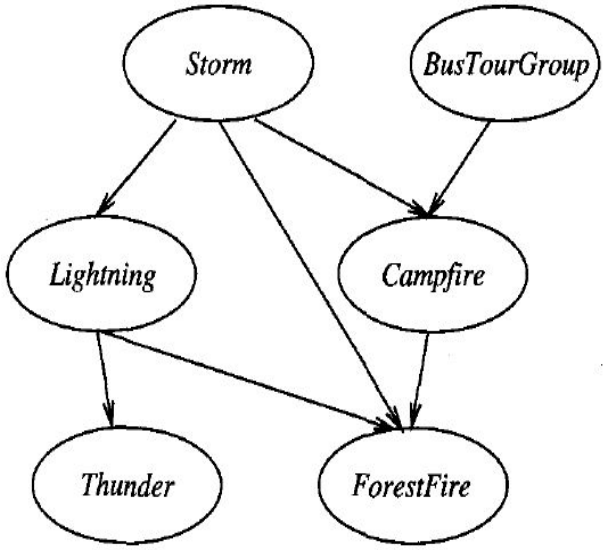
- X is a ***descendant*** of ***Y*** if there is a directed path from ***Y*** to X.

2. A conditional probability table is given for each variable, describing the probability distribution for that variable given the values of its immediate predecessors.

- The joint probability for any desired assignment of values $(y_1, \ldots, y_n)$ to the tuple of network variables $(Y_1, \ldots, Y_n)$ can be computed by the formula

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

- where $Parents(Y_i)$ denotes the set of immediate predecessors of Y_i in the network.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

Campfire

- Consider the node ***Campfire.***
- ***Campfire*** is conditionally independent of ***Lightning*** and ***Thunder,*** given its immediate parents ***Storm*** and ***BusTourGroup.***
- This means that once we know the value of ***Storm*** and ***BusTourGroup,*** ***Lightning*** and ***Thunder*** provide no additional information about ***Campfire.***

- The right side of the fig shows the conditional probability table associated with the variable ***Campfire.***
- The top left entry in this table, for ex, expresses the assertion that

$$P(Campfire = True | Storm = True, BusTourGroup = True) = 0.4$$

- NOTE: This table provides only the conditional probabilities of ***Campfire*** given its parent variables ***Storm*** and ***BusTourGroup.***
- Full Join Probability Distribution For The Network is:
- The set of local conditional probability tables for all the variables, with the set of conditional independence assumptions described by the network.

- Attractive feature of Bayesian belief networks
- A convenient way to represent causal knowledge such as the fact that ***Lightning*** causes ***Thunder.***
- That is:
  - ***Thunder*** is conditionally independent of other variables in the network, given the value of ***Lightning.***

Inference

- Use a Bayesian network to infer the value of some target variable (e.g., **ForestFire)** given the observed values of the other variables.
- This inference step can be straightforward if values for all of the other variables in the network are known.
- Example: We may wish to infer the probability distribution for some variable (e.g., **ForestFire)** given observed values for only a subset of the other variables (e.g., **Thunder** and **BusTourGroup**).
- A Bayesian network can be used to compute the probability distribution for any subset of network variables given the values or distributions for any subset of the remaining variables.

Learning Bayesian Belief Networks

- CAN WE DEVISE EFFECTIVE ALGORITHMS FOR LEARNING BAYESIAN BELIEF NETWORKS FROM TRAINING DATA?
- Several different settings for this learning problem can be considered.

1. First, the network structure might be given in advance, or it might have to be inferred from the training data.
2. Second, all the network variables might be directly observable in each training example, or some might be unobservable.

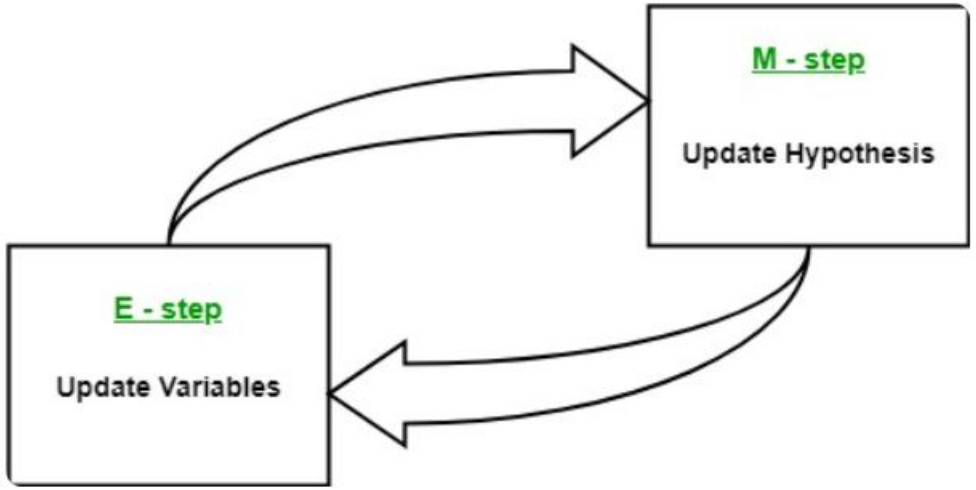
- If 1 & 2 is satisfied, learning the conditional probability tables is straightforward.
- Estimate the conditional probability table entries just as a naive Bayes classifier.
- If only 1 is satisfied, the learning problem is more difficult.
- This problem is same as to learning the weights for the hidden units in an artificial neural network, where the input and output node values are given but the hidden unit values are left unspecified by the training examples.

Expectation-Maximization(EM) algorithm

- The basic idea behind this algorithm is to use the observable samples of latent variables to predict the values of unobservable samples for learning.
- This process is repeated until the convergence of the values occurs.

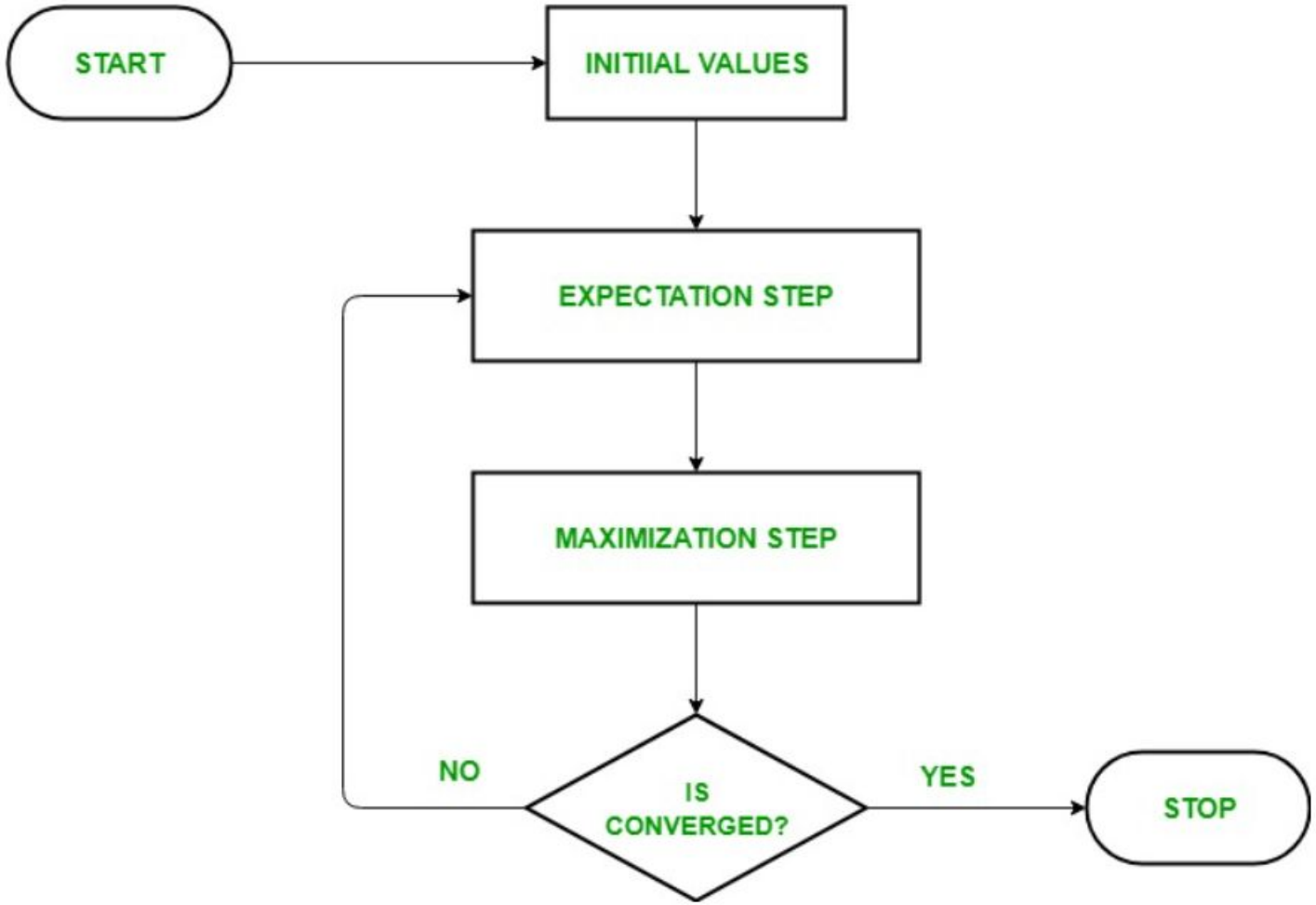
- Given a set of incomplete data, start with a set of initialized parameters.

1. **Expectation step (E-step):** In this expectation step, by using the observed available data of the dataset, we can try to estimate or guess the values of the missing data. Finally, after this step, we get complete data having no missing values.
2. **Maximization step (M-step):** Now, we have to use the complete data, which is prepared in the expectation step, and update the parameters.
3. Repeat step 2 and step 3 until we converge to our solution.



STEPS

1. Initialization Step: Initialize the parameter values with a set of initial values, then give the set of incomplete observed data to the system with the assumption that the observed data comes from a specific model i.e, probability distribution.
2. Expectation Step: This step uses the observed data to estimate or guess the values of the missing or incomplete data. It is used to update the variables.
3. Maximization Step: In this step, use the complete data generated in the "Expectation" step to update the values of the parameters i.e, update the hypothesis.
4. Checking of Convergence Step: Check whether the values are converging or not, if yes, then stop otherwise repeat these two steps i.e., the "Expectation" step and the "Maximization" step until the convergence occurs.


START
INITIIAL VALUES
EXPECTATION STEP
MAXIMIZATION STEP
IS CONVERGED?
NO
YES
STOP

- Advantages

1. The basic two steps of the EM algorithm i.e, E-step and M-step are often pretty easy for many of the machine learning problems in terms of implementation.
2. The solution to the M-steps often exists in the closed-form.
3. It is always guaranteed that the value of likelihood will increase after each iteration.

- Disadvantages

1. It has slow convergence.
2. It converges to the local optimum only.
3. It takes both forward and backward probabilities into account. This thing is in contrast to that of numerical optimization which considers only forward probabilities.

- Applications of EM Algorithm
- Used to calculate the Gaussian density of a function.
- Helpful to fill in the missing data during a sample.
- It finds plenty of use in different domains such as Natural Language Processing (NLP), Computer Vision, etc.
- Used in image reconstruction in the field of Medicine and Structural Engineering.
- Used for estimating the parameters of the Hidden Markov Model (HMM) and also for some other mixed models like Gaussian Mixture Models, etc.
- Used for finding the values of latent variables.

---XXX---