20-Nov-2023

EM - Algorithm

EM: Expectation - Maximization

- In the real-world applications of machine learning, it is very common that there are many relevant features available for learning but only a small subset of them are observable.
- The 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).
- This algorithm is actually the base for many unsupervised clustering algorithms in the field of machine learning.

Steps:

Step-1: 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.

Step-2: 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.

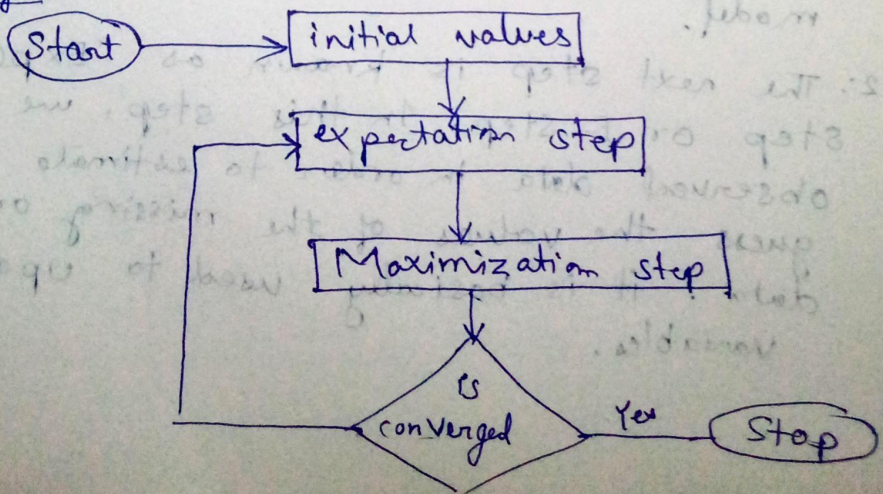
Step-3: The next step is known as "Maximization" step or M-step. In this step, we use the complete data generated in the preceding "Expectation"-step in order to update the values of the parameters. it is basically used to update the hypothesis.

Step-4: 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.

Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. Expectation step (E-step): using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. Maximization step (M-step): Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step-3 until convergence.

Flow diagram



Usage of EM Algorithm:

- it can be used to fill the missing data in sample.
- it can be used as the basis of unsupervised learning of clusters.
- it can be used for the purpose of estimating the parameters of Hidden markov model (HMM)
- it can be used for discovering the values of latent variables.

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:

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

- expectation - Maximization (EM) - a very popular technique for estimating parameters of probabilistic models.
- Many popular algorithms like HMM, Gaussian Mixtures, Kalman filters, and other uses EM technique.
- It is beneficial when working with data that is incomplete or has missing data points or has unobserved latent variables.

Example:

- Assume that we have two coins, C_1 and C_2
- Assume the bias of C_1 is θ_1 (i.e., probability of getting heads with C_1)
- Assume the bias of C_2 is θ_2 (i.e., probability of getting heads with C_2)
- We want to find θ_1 and θ_2 by performing a number of trials (i.e., coin tosses)

First experiment

- we choose 5 times one of the coins
- We toss the chosen coin 10 times

(B) H T T T H H T H T H

(A) H H H H T H H H H H

(A) H T H H H H H T H H

(B) H T H T T T H H T T

(A) T H H H T H H H T H

$$\theta_1 = \frac{\text{number of heads using } C_1}{\text{total number of flips using } C_1}$$

$\theta_2 = \frac{\text{number of heads using } c_2}{\text{total number of flips using } c_2}$

coin A	coin B
	5H, 5T
9H, 1T	
8H, 2T	
	4H, 6T
7H, 3T	
Total = 24 H 6T	9H 11T

$\theta_1 = \frac{24}{24+6} = 0.8 \qquad \theta_2 = \frac{9}{9+11} = 0.45$

Assume a more challenging problem.
- we do not know the identities of the coins used for each set of tosses (we treat them as hidden variables)

H T T T H H T H T H
H H H H T H H H H H
H T H H H H H T H H
H T H T T T H H T T
T H H H T H H H T H

assume $\theta_A^{(0)} = 0.60 \qquad \theta_B^{(0)} = 0.50$

likelihood $L(c) = x^k (1-x)^{n-k}$

$P(E|z_A) = P(H\,H\,H\,H\,H\,H\,H\,H\,T \mid A \text{ chosen}) = \binom{n}{x} \theta_A^x (1-\theta_A)^n$

$= \binom{9}{1} (0.6)^9 (1-0.6)^{10-9} = 0.036$

$P(E|z_B) = P(H\,H\,H\,H\,H\,H\,H\,H\,T \mid B \text{ chosen}) = \binom{n}{x} \theta_B^x (1-\theta_B)^n$

$= \binom{9}{1} (0.5)^9 (1-0.5)^{10-9} = 0.009$

$$P(Z_A|E) = \frac{0.036}{0.036 + 0.009} = 0.80$$

$$P(Z_B|E) = \frac{0.009}{0.036 + 0.009} = 0.20$$

$P(Z_A|E) \times$ number of head $= 0.80 \times 9 = 7.2$

$P(Z_A|E) \times$ number of tail $= 0.80 \times 1 = 0.8$

$P(Z_B|E) \times$ number of head $= 0.20 \times 9 = 1.8$

$P(Z_B|E) \times$ number of tail $= 0.20 \times 1 = 0.20$

the above step is ~~over~~ on first ~~item~~ iteration.

on 2nd raw data.

H H H H T H H H H

then apply the above step in all the data raw then our output will be:

coin A	coin B
2.2H, 2.2T	2.8H, 2.8T
7.2H, 0.8T	1.8H, 0.2T
5.9H, 1.5T	2.1H, 0.5T
1.4H, 2.1T	2.6H, 3.9T
4.5H, 1.9T	2.5H, 1.1T
≈ 21.3H, 8.6T	≈ 11.7H, 8.4T

step-3. maximization step

current $\begin{cases} \theta_A^{(1)} = \frac{21.3}{21.3 + 8.6} = 0.71 \\ \theta_B^{(1)} = \frac{11.7}{11.7 + 8.4} = 0.58 \end{cases}$

previous $\begin{cases} \theta_A^{(0)} = 0.60 \\ \theta_B^{(0)} = 0.50 \end{cases}$

if previous and current are same then.

$\theta_A^{(10)} = 0.60$ $\theta_B^{(10)} = 0.50$

else our new θ_A and θ_B will be current example. Here

$\theta_A^{(1)} = 0.71$ $\theta_B^{(1)} = 0.58$

After the 10th iteration we get the answer it means current == previous.

$\theta_A^{(20)} = 0.80$ $\theta_B^{(10)} = 0.52$