e-PGPathshala

Subject : Computer Science

Paper: Machine Learning

Module: Expectation and Maximization

Module No: CS/ML/33

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning.In this module we will discuss another powerful machine learning method – the Expectation Maximization or the EM method.

Learning Objectives:

The learning objectives of this module are as follows:

- To understand the concept of Expectation Maximization (EM)
- To explain EM using a Coin Toss Example
- To understand the use of EM in K-means Algorithm

33.1 Introduction

The Expectation Maximization methodology was first presented in a general way by Dempster, Laird and Rubin in 1977. They define EM algorithm as an iterative estimation algorithm that can derive the maximum likelihood (ML) estimates in the presence of missing/hidden data ("incomplete data"). As an example we have theclassical case of the Gaussian mixture, where we have a set of unknown Gaussian distributions.The goal of the EM algorithm is to facilitate maximum likelihood parameter estimation by introducing so-called hidden random variables which are not observed and therefore define the unobserved data.

There are two main approaches to the applications of the EM algorithm. The first is when the data indeed has missing values, due to problems with or limitations of the observation process. The second approachis the optimization of the likelihood function is complex but however likelihood function can be simplified by assuming the existence of additional but missing (or hidden) parameters. The second type of application is more common in the pattern recognition. Now let us understand the concept of hidden and observed variables. Observed variables are directly measurable from the data, e.g. waveform values of a speech recording, Is it raining today? Did the smoke

alarm go off?. On the other hand hidden variablesinfluence the data, but are not trivial to measure e.g includethe phonemes that produce a given speech recording, the probability P (rain today | rain yesterday) and is the smoke alarm malfunctioning?

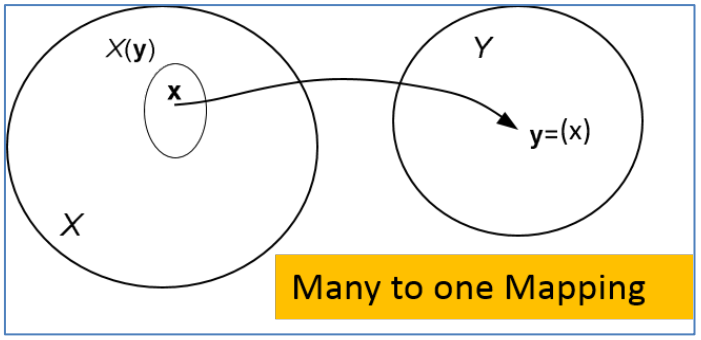

Figure 33.1 Many to One Mapping

In Figure 33.1 X can be considered as the underlying space, x is the complete data (required for ML), *Y* is the observation space, y is the actual observation and x is observed only by means of y(x) and *X*(y) is a subset of X determined by y. The term y=(x) indicates that y is observed as some mapping of x.

The EM approach has found usage in filling in missing data in a sample, discovering the value of latent variables, estimating parameters of HMMs, estimating parameters of finite mixtures, unsupervised learning of clusters and finding parameters of Mixtures of Gaussians (MoG).

33.2 The EM algorithm

The steps of the EM algorithm are as follows:

1. We first consider a set of starting parameters given a set of incomplete (observed) data and we assume that observed data come from a specific model

2. We then use the model to “estimate” the missing data. In other words after formulating some parameters from observed data to build a model, we use this model to guess the missing value/data. This step is called the expectation step.

3. Now we use the “complete” data that we have estimated to update parameters whereusing the missing data and observed data, we find the most likely modified parameters to build the modified model. This is called the maximization step.

4. We repeat steps 2 & 3 until convergence that is there is no change in the parameters of the model and the estimated model fits the observed data.

The general idea of the algorithm is that we start by devising a noisy channel that is we use any model that predicts the corpus observations via some hidden structure and we Initially guess the parameters of the model. It is best to make an educated guess but random assumptions can also work. Then we repeat until convergence the Expectation and Maximization steps. In the**Expectation step** we use current parameters (and observations) to reconstruct hidden structure while in the **Maximization step** weuse that hidden structure (and observations) to re-estimate parameters.

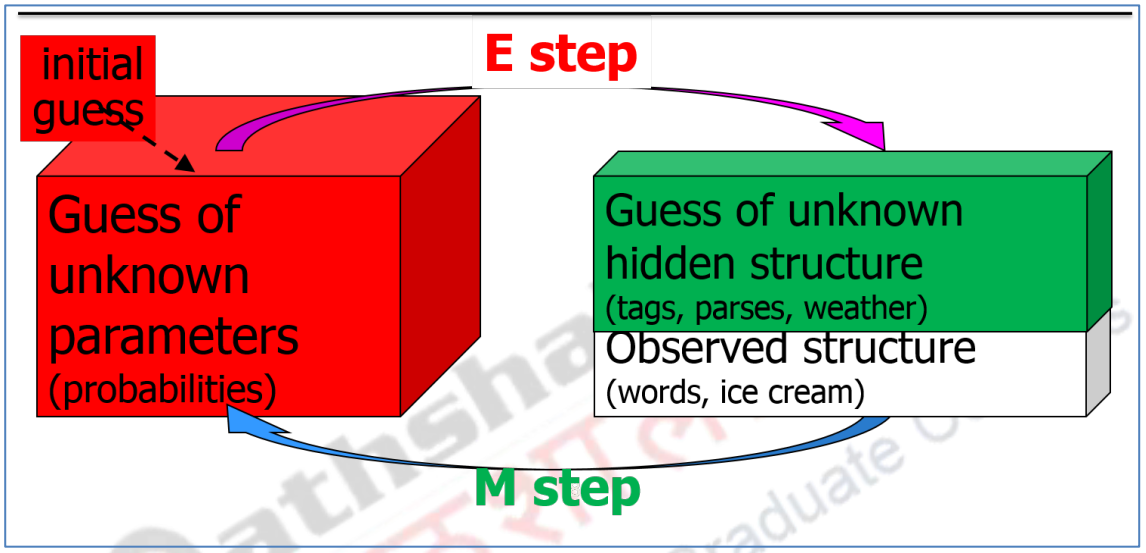

Figure 33.2 General Idea of the EM algorithm

In the example shown in Figure 33.2, we initially guess the probabilities of unknown parameters using which we guess the hidden structure such as tags, parses, weather (E step) and use it along with the observed structure such as words, eating ice cream to again re-estimate the probabilities (M step).

33.3 EM and Maximum Likelihood Estimates

As discussed above parameters describe the characteristics of a population. Their values are estimated from samples collected from that population.A maximum likelihood (ML) estimate is a parameter estimate that is most consistent with the sampled data since it attempts to maximize the likelihood function.

The basic setting of EM which is essentially based on maximum likelihood estimates is outlined below:

Let X be a set of data points considered as**observed** data and Θbethe parameter vector. Now EM is a method to find θ_{ML} where $L(\Theta)$ is likelihood function

$$\theta_{ML} = \arg\max_{\theta \in \Omega} L(\Theta)$$
$$= \arg\max_{\theta \in \Omega} \log P(X \mid \Theta)$$

In essence we need to determine the probability of the observed data given the parameter vector. Now calculating P(X | θ) directly is hard. However calculating P(X,Y|θ) is much simpler, where Y is “hidden” data (or “missing” data) associated with the observed data.

Let us first define Z = (X, Y) where Z is considered as the complete data that is “augmented data”, X is the observed data (“incomplete” data) and Y is the hidden data (“missing” data). EM is an iterative method to perform Maximum Likelihood estimation which starts with an initial estimate for θandrefines the current estimate iteratively to increase the likelihood of the observed data:

$$p(Z/\theta)$$

33.4 EM – Coin Toss Example

The EM strategy can be explained with a coin toss example. This is the example we will be using in subsequent iterations to explain the complete flow of the EM algorithm. In this example we assume that we are tossing a number of coins sequentially to obtain a sequence of Head or Tails. The context of the coin toss example is given in Table 33.1. Here the problem is defined as X, the sequence of Heads and Tails that is observed, Y as the identifier of the coin that is tossed in the sequence, which is hidden and finally θwhich is the parameter vector which is associated with the probabilities ofthe observed and hidden data. Here if we assume three coins are tossed λ is the probability of coin 0 showing H (so 1 − λ is the probability of it showing T), p1 is the probability of coin 1 showing H, and p2 is the probability of coin 2 showing H.

Problem	Coin toss
X(observed)	Head-tail sequences
Y (hidden)	Coin id sequences
Θ	p1, p2, λ

Table 33.1 Parameters of EM

We can also modify the problem to figure out the probability of heads for two coins. Normally the ML estimate can be directly calculated from the results if we know the identity of which of the coins was tossed.

We have two coins indicated as coins**A** and **B**and let us assume that the probabilities for heads are $\mathbf{q}_A$&$\mathbf{q}_B$respectively. We are given 5 measurements sets including 10 coin tosses in each set. Now we know which coin has been tossed in each measurement. The example sets of experiments are given in Table 33.2. In this table A coin has been indicated as red and B coin as blue. The first column in the table indicates the coin type for each of the 5 measurements since here we assume we know the identity of the coin. The second column indicates the sequence of 10 Heads and Tails observed for each measurement. Columns 3 and 4 indicate the number of Heads and Tails obtained in each coin toss for each measurement of each coin type. The final row shows the total number of Heads and Tails obtained for the measurements for each of the A and B coin types.

Coin Type	5sets of 10 tosses each	Coin A	Coin B
B	H T T T H H T H T H		5 H, 5 T
A	H H H H T H H H H H	9 H, 1 T	
A	H T H H H H H T H H	8 H, 2 T	
B	H T H T T T H H T T		4 H, 6 T
A	T H H H T H H H T H	7 H, 3 T	
		24 H, 6 T	9 H, 11T

Table 33.2 Coin Toss Example of 5 measurements of 10 coin Tosses

Now we calculate the ML probabilities$\mathbf{q}_A$&$\mathbf{q}_B$, the probabilities for heads of coins A and B respectively.

Maximum Likelihoodof theprobabilities $\mathbf{q}_A$&$\mathbf{q}_B$arecalculated by dividing the total number of Heads obtained by the total number of Head and Tails observed for each type of coin. Thus we get

$\mathbf{q}_A$ = 24/24+6 = 24/30 = 0.8

$\mathbf{q}_B$ = 9/9+11=9/20=0.45

The above calculation is a basic probability calculation based on observations knowing whether coin A or B has been tossed.

We will now make the problem more interesting and assume that we do not even know which one of the coins is used for the sample set. Now we need to estimate the coin probabilities without knowing which one of the coins is being tossed.

33.5 EM Flow Explained with Coin Toss Example

Note that when we do not know which of the coins is tossed in each set we cannot calculate ML directly and hence we use EM strategy to find the probabilities of which one of the coins is likely to be tossed. Figure 33.3 shows the complete flow of the EM algorithm. Remember we do not know which of the coins is tossed. Hence we start the process by assuming that for each of the coins A (red) and B (blue), the initial probabilities for heads are $\mathbf{q}_A$&$\mathbf{q}_B$respectivelywhich are assumed to have random values. Hence as seen from Figure 33.3 we have randomly fixed $\mathbf{q}_A$to be 0.6 and $\mathbf{q}_B$to be 0.5. Now we observe the number of Heads and Tails for each of the 5 measurements.

E Step:

The first stage is the **Expectation**stage for which we initially use the randomly assumed probabilities of $\mathbf{q}_A$&$\mathbf{q}_B$ and the set of coin tosses observed for each measurement. Now we need to calculate the probabilities for Heads and Tails for both A and B coins for each measurement since we do not know which coin is tossed. We have shown the calculation for the first set .of measurements in Figure 33.3.

Step E-C1

In the first step of the **Expectation stage** we assume that the coin toss sequence follows a binomial distribution, where n is total number of coin tosses, k is number of Heads (Tails) observed and p is the probability of observing heads for each coin.

$$\binom{n}{k} p^k (1-p)_H{}^{n-k}$$

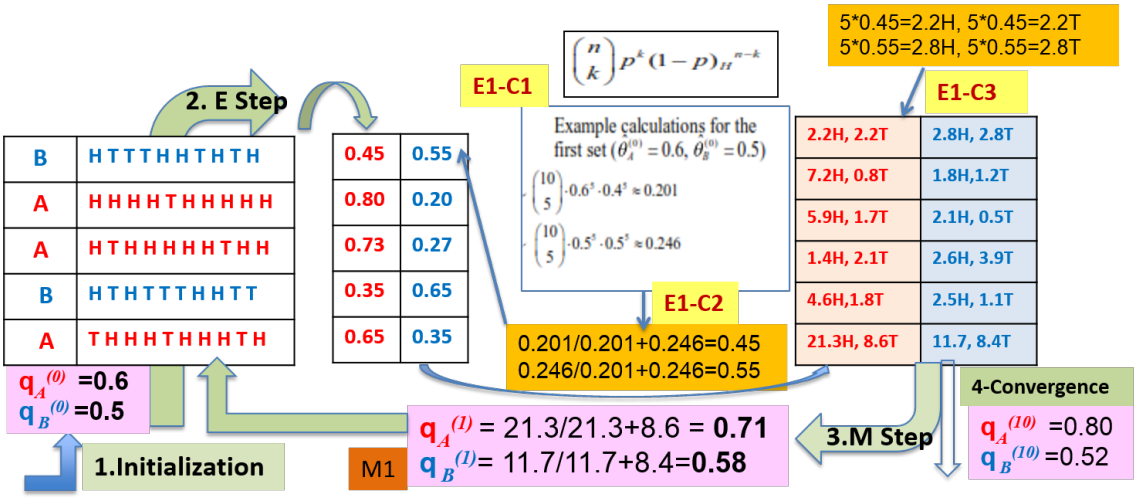

Figure 33.3 Flow of EM - Coin Toss Example

Usingthe distribution we can calculate the probability of observing Heads and Tails for coins A and B for the first set as shown in Figure 33.4. Here

n - the total number of coins tossed in the first measurement = **10,**

k -the total number of heads observed =**5**

Now we need to calculate the probability of observing Heads(Tails) for both coins A and B since we do not know which of the coins was tossed and hence

p – (q_A)- the probability of heads for A (red) coin is initially assumed to be = **0.6**

& (q_b)- the probability of heads for B (blue) coin is initially assumed to be = **0.5**

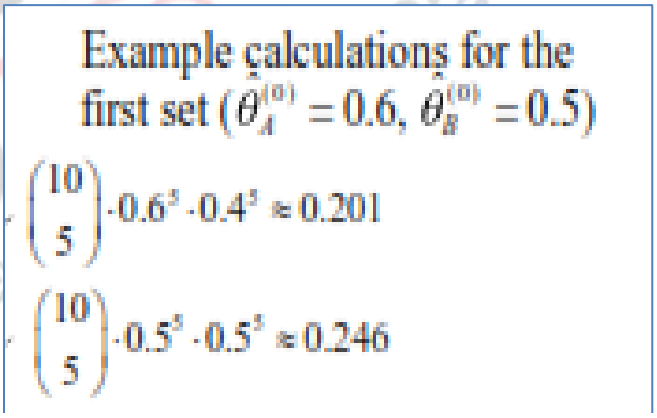

Figure 33.4 Use of Binomial Distribution

Step E-C2

In the second step of the Expectation stage we calculate the probabilities of using coin A or B as follows:

0.201/ (0.201+0.246)=0.45

0.246/ (0.201+0.246)=0.55

We calculate in a similar manner for all the experiments and the values obtained are shown in Figure 33.3.

Step E-C3

In the third step of the Expectation stage using the probability values obtained in step E-C2, we calculate the possible number of Heads and Tails that is likely

to be observed in each experiment if the coin tossed was A and if the coin tossed was B

First Experiment – values corresponding to first row:

(i) 5(number of heads in First experiment)
*0.45(probability of tossing coin A)=2.2H,
(ii) 5(number of Tails in First experiment)
*0.45(probability of tossing coin A)=2.2H,
(iii) 5(number of heads in First experiment)
*0.55(probability of tossing coin B)=2.8H,
(iv)5(number of Tails in First experiment)
*0.55(probability of tossing coin B)=2.8H.

We will also explain the calculations for second experiment.

Second Experiment – values corresponding to second row:

(i) 9(number of heads in Second experiment)
*0.80(probability of tossing coin A)=7.2H,
(ii) 1(number of Tails in Second experiment)
*0.8(probability of tossing coin A)=0.8H,
(iii) 9(number of heads in Second experiment)
*0.20(probability of tossing coin B)=1.8H,
(iv)1(number of Tails in Second experiment)
*0.20(probability of tossing coin B)=0.2H.

Similarly we can do the calculations for all 5 experiments. Using these calculated values for number of Head and Tails for each experiment, we can calculate the total number of Heads and Tails for both Coins A and B.

M Step

Now we have the Maximization stage where we calculate the new values (after 1st iteration) of $\mathbf{q}_A^{(1)}$ and $\mathbf{q}_B^{(1)}$ that is the maximum likelihood estimates of the probability of heads when coin A and probability of heads when coin B are tossed respectively using the values total number of Heads and Tails for both Coins A and B. This calculation is shown below:

$\mathbf{q}_A^{(1)}$= 21.3(total number of Heads when coin A is tossed)/(21.3+8.6)(total number of Heads and Tails when coin A is tossed)= **0.71**

$\mathbf{q}_B^{(1)}$= 11.7(total number of Heads when coin B is tossed)/(11.7+8.4)(total number of Heads and Tails when coin B is tossed)=**0.58**

Continuing and Completing the EM algorithm

Now we have completed one iteration of the EM algorithm. We now continue the second iteration of the algorithm using the above new values of $\mathbf{q}_A$&$\mathbf{q}_B$. We continue the iterations until the values of $\mathbf{q}_A$&$\mathbf{q}_B$ do not change from one iteration to the next. This happens in the 10th iteration for our example (shown in Figure 33.3) when the values $\mathbf{q}_A$&$\mathbf{q}_B$ converge to values **0.80** and **0.52** respectively.

33.6 Kmeans and EM algorithm

We can explain K means as an EM algorithm. First we initialize the k means (μ_k) of the Kmeans algorithm. In the E Step weassign each point to a Cluster and during the M Step given the Clusters we refine mean $\mu_k$of each cluster k. This process is repeated until the change in means is small.

33.6.1 Generating Data from Mixture of Gaussians

Now we can replace the 'hard' clustering of K-means described above with 'soft' probabilistic assignments. Here we assume that each instance x is generated

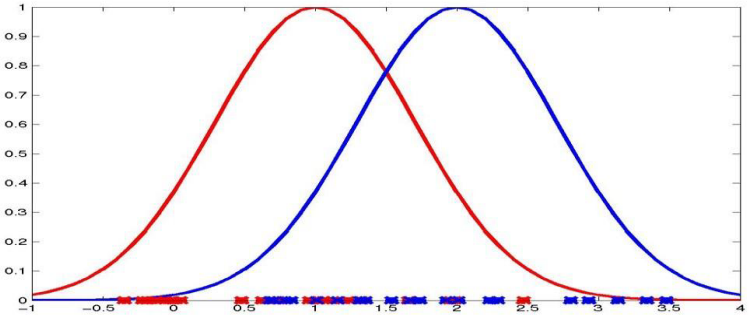

Figure 33.5 The Gaussian Distributions used to Generate Data

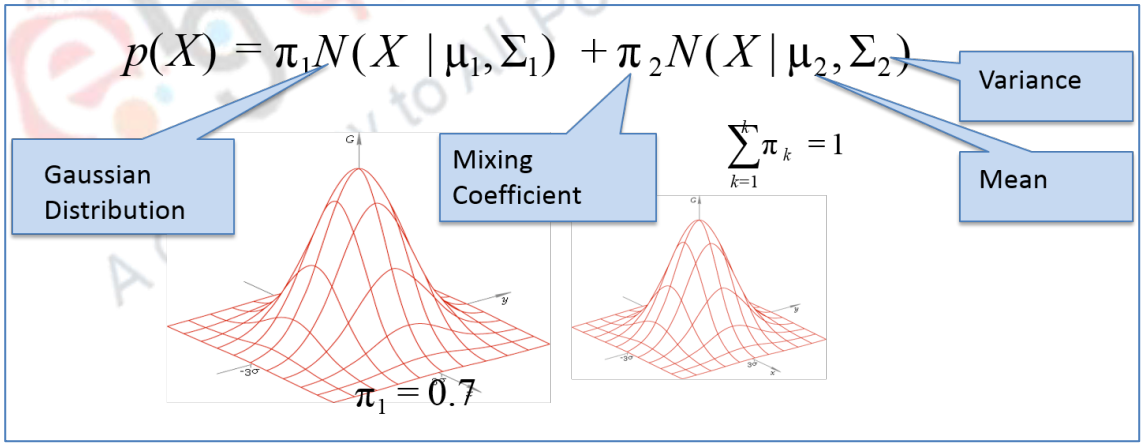

Figure 33.6 The Use of a Mixture of Gaussians for Generating Data

by choosing one of the k Gaussians at random and generating an instance according to that Gaussian(Figure 33.5). This requires more parameters to be determined that would fit the data. Figure 33.6 shows the probability of generating data p(X). This probability is based on the Gaussian distributions used, the mean and the variance of the Gaussian distributions and the mixing coefficients used. The sum of the mixing coefficients of all the Gaussian distributions use must be 1.

33.6.2 K-means and Mixture of Gaussians

Now we know that in a general K-means which is essentially a classifier and we need to find the parameterto fit data – that is we need to find the mean - $^{\mu}{}_{k}$ as already discussed above. However when we use mixture of Gaussians which is a probability model where we are defining a "soft" classifier. Now the parameters that are to be determined to fit to data are the means μ_k*and* covariance $\Sigma_k$which define the Gaussians distributions and the mixing coefficient π_k. Now given the data set, find the mixing coefficients, means and covariance. If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster. However our problem is that the data set is unlabelled or are hidden (Figure 33.7).

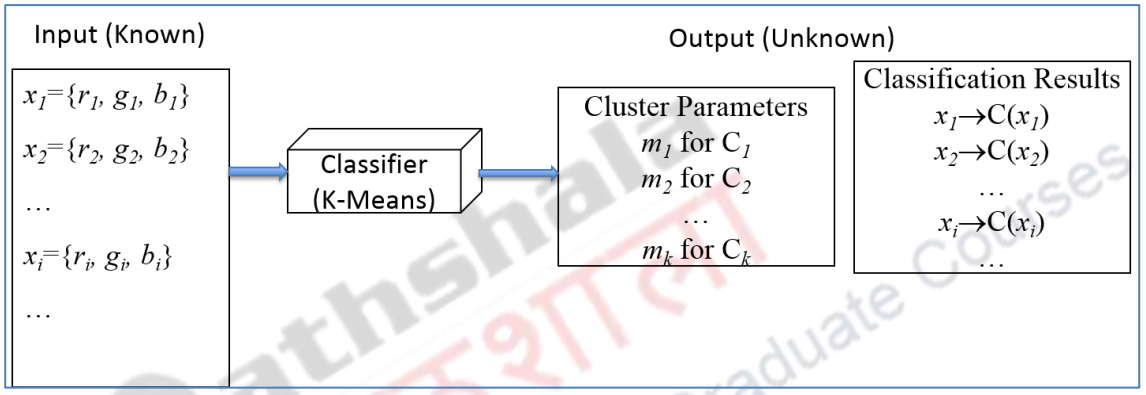

Figure 33.7 K Means Scenario

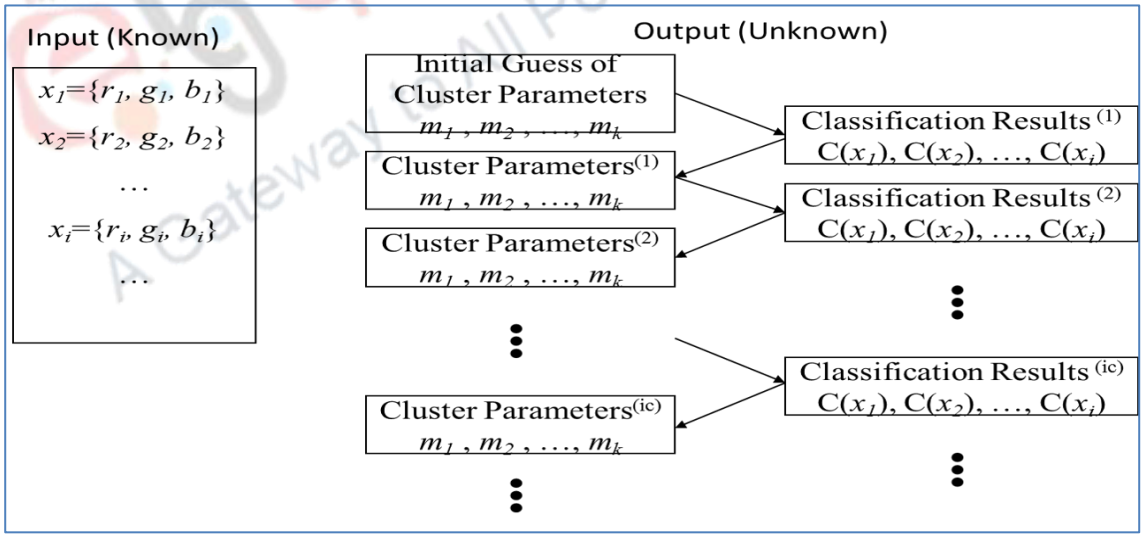

Figure 33.8 K means with Initial Guess of Cluster Points

33.6.3 EM for Estimating k Means

Figure 33.8 shows the scenario of K means the instances from X are generated by a mixture of k Gaussians with unknown means $<m_1,\ldots,m_k>$ of the k Gaussians.Remember we do not know which instance xi was generated by

which Gaussian. Now we need to determine the maximum likelihood estimates of $<m_1,\ldots,m_k>$. Now let us define the full description of each instance as $y_i=<x_i,z_{i1},z_{i2}>$ where z_{ij} is 1 if xi generated by j-th Gaussian. In this case $x_i$is observable but however $z_{ij}$is unobservable.

33.6.4 EM for Gaussian Mixtures

Here we initialize the Gaussian parameters mean μ_k, co-variance $\sum_k$ and mixing coefficient π_k.

E Step:

In the E Step we assign each point x_n an assignment score $\gamma(z_{nk})$ for each cluster or Gaussian kwhich essentially indicates how much this Gaussian k is responsible for point X_n

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Figure 33.9 Calculation of Assignment Score

Here $\gamma(z_{nk})$ is calculated as the mean and covariance of the Gaussian distribution corresponding to the kth mean multiplied by mixture coefficient of kth distribution divided by the summation of mean and covariance of all the Gaussian distributions multiplied by their corresponding mixture coefficient.

M step

During the M Step, given scores, adjust μ_k, $\sum_k$, π_k for each cluster kor Gaussian K. We update parameters using new $\gamma(z_{nk})$ that is find the parameters that fit the new assignment score $\gamma(z_{nk})$ the best. The new values of each of the parameters μ_k^{new}, $\sum_k^{new}$ and π_k^{new} are determined as shown in Figure 33.10.

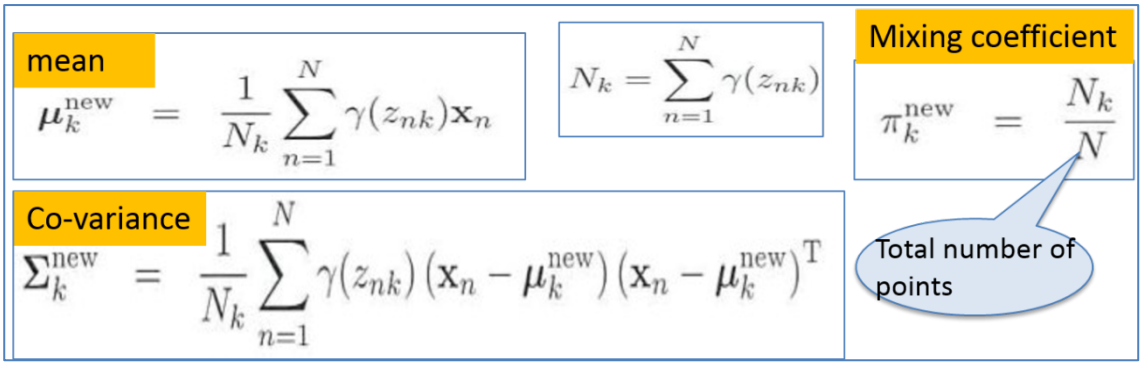

Figure 33.10 Calculating new Values of Parameters during M Step

Now we evaluate log likelihood as shown in Figure 33.11. If likelihood or parameters converge we stop else we iterate with the E and M steps.

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Figure 33.11 Evaluation of Log Likelihood

33.7 Strengths of EM

The major strength of the EM algorithm is its numerical stability where in every iteration of the EM algorithm, the likelihood of the observed data increases that is we are heading towards a solution. In addition, the EM handles parameter constraints gracefully.

33.8 Problems with EM

In the case of EM algorithms can converge very slowly on some problems and this convergence is intimately related to the amount of missing information.It guarantees to improve the probability of the training corpus, which is different from reducing the errors directly. The EM algorithm cannot guarantee to reach global maximum and sometimes could get struck at the local maxima, saddle points, etc. Essentially the guess we make of the initial parameter values is very important and can decide on the time to converge.

Summary

- Explained the concept of Expectation Maximization (EM)
- Discussed EM using a Coin Toss Example
- Outlined the use of EM in K-means Algorithm