10-708: Probabilistic Graphical Models 10-708, Spring 2015

8 : Learning Partially Observed GM: EM Algorithm

Lecturer: Eric P. Xing *Scribes: Ankit Laddha, Anirudh Vemula, Cuong Nguyen*

1 Inference as Subroutine for Learning

Many a times inference is used as subroutine for learning or parameter estimation in graphical models. In the following we describe two such scenarios.

1.1 Partially Observed Directed GM

In case of partially observed directed graphical models the log-likelihood $l(\theta; D)$ involves a marginalization over the unobserved variables. We calculate $l(\theta; D)$ as:

$$l(\theta; D) = \log \sum_z p(x, z|\theta) \tag{1}$$

$$= \log \sum_z p(z|\theta_z) p(x|z, \theta_x) \tag{2}$$

Therefore, to calculate l we need to do inference to find the conditional probability of x given z.

1.2 Fully Observed Undirected GM

In fully observed undirected graphical models the log-likelihood (l)is calculates as

$$l = \sum_x \sum_{x_c} m(x_c) \log \psi_c(x_c) - N \log Z \tag{3}$$

Any method for parameter estimation requires the gradient of l which in tern requires the the gradient of the term $\log Z$, which is calculated as

$$\frac{\partial \log Z}{\partial \psi_c(x_c)} = \frac{p(x_c)}{\psi_c(x_c)} \tag{4}$$

Therefore, to calculate $p(x_c)$ we need to find the $p(x_c)$ for every clique in the model.

2 Examples: Partially Observed Models

2.1 Speech Recognition

As shown in the Figure 1, we can model the speech recognition problem as a HMM where the observed variables X_i are the sounds and the unobserved variables Y_i are the phonetics or words spoken.

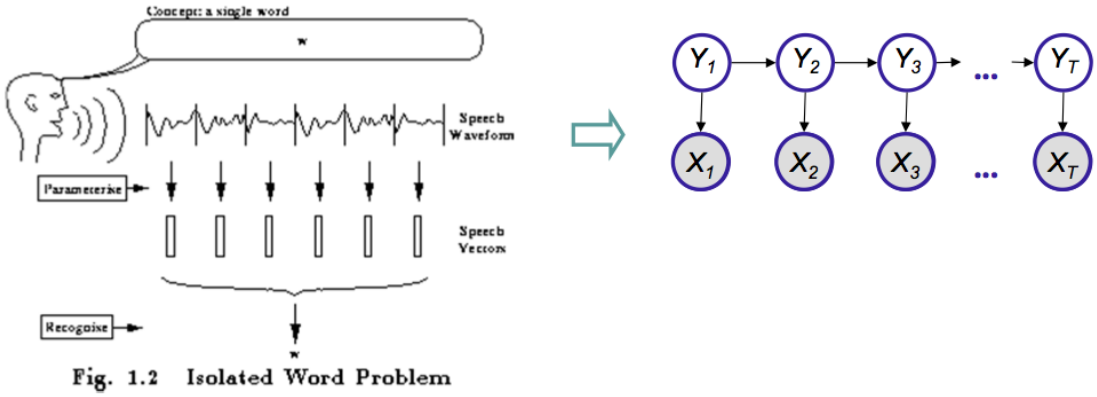

Figure 1: A Graphical Model for Speech Recognition

2.2 Biological Evolution

We could also model the biological evolution as a directed GM as shown in Figure 2. In this, the leaf nodes representing the various organisms are observed. The hidden nodes represent a common ancestor from which the organisms derive a particular trait.

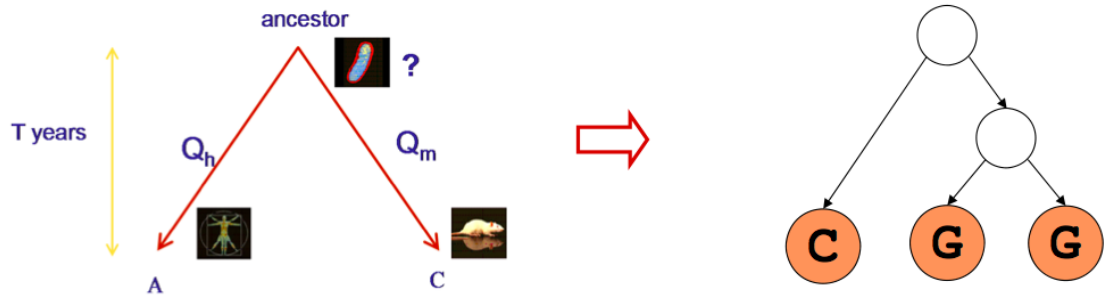

Figure 2: A Graphical Model for Biological Evolution

3 Unobserved Variables

3.1 Why we need them?

Latent variables are extensively used in graphical models as they provide a simplified and abstract view of the data generation process. They can be used to model real-world objects or phenomena which are difficult/impossible to measure or can be only measured with noise (e.g. through faulty sensors)

In case of data where we have clusters (or some kind of grouping), discrete latent variables can be used to model the membership of the data. Continuous latent variables are used in dimensionality reduction

techniques like factor analysis.

3.2 Why is learning hard?

As we have seen before, the log-likelihood for fully-observable directed models decomposes "neatly" into a sum of local terms. In case of undirected models, we observe the same property of log-likelihood in the case of tree-like models and decomposable models.

$$l_c(\theta; D) = \log p(x, z|\theta) = \log p(z|\theta_z) + \log p(x|z, \theta_x) \tag{5}$$

But in the presence of latent variables, the situation becomes "tricky". When some variables are not observed the likelihood is not a joint probability but a marginal probability obtained by summing out all the latent variables. This summation leads to all the parameters getting coupled and the log-likelihood doesn't decompose like before.

$$l_c(\theta; D) = \log \sum_z p(x, z|\theta) = \log \sum_z p(z|\theta_z)p(x|z, \theta_x) \tag{6}$$

This coupling of parameters makes the learning task harder in the presence of latent variables. The usual gradient-based approaches to get maximum-likelihood estimates cannot be used efficiently in this case and we must resort to EM-like approaches.

4 Mixture Models

4.1 GMMs

Mixture models arise naturally from data clustering task, where data may have multi-modal density distribution. A mixture model comprises of a number of components which are uni-modal density functions. In such a setup, each uni-modal density function corresponds to a sub-population of the data.

Gaussian mixture model (GMM) is one of the most mature mixture model. A GMM defines the overall probability density of data as a weighted sum of Gaussian distributions. Figure 3 illustrates the idea of using a GMM for data clustering. More formally, we can define a GMM of k Gaussian components as follows:

$$p(x|\mu, \Sigma) = \sum_k \pi_k N(x|\mu_k, \Sigma_k) \tag{7}$$

Where π_k are the model weights to specify mixture proportion, each $N(x|\mu_k, \Sigma_k)$ is a Gaussian component. Another way to think of GMM is to have a latent discrete variable Z to indicate which Gaussian component is being selected:

$$p(z_n) = multi(z_n : \pi) = \prod_k (\pi_k)^{z_n^k} \tag{8}$$

Given Z, X is a conditional Gaussian variable with specific uni-modal parameters (mean and variance):

$$p(x_n|z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma_k^{1/2}|} \exp(-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^- 1(x_n - \mu_k)) \tag{9}$$

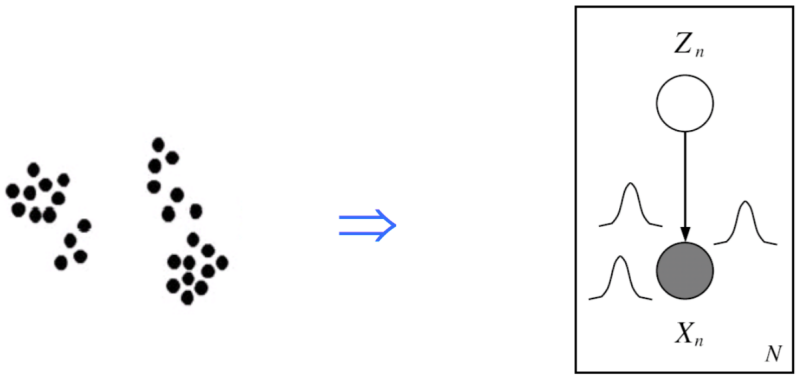

Figure 3: GMM for clustering

The likelihood of a single sample x_n is computed as the product of the probability distribution of Z and the conditional distribution of X given Z:

$$p(x_n|\mu,\Sigma) = \sum_k p(z^k = 1|\pi)p(x|z_k = 1,\mu,\Sigma) \tag{10}$$

$$= \sum_{z_n}\prod_k((\pi_k)^{z_n^k}N(x_n : \mu_k,\Sigma_k)^{z_n^k}) \tag{11}$$

$$= \sum_k \pi_k N(x|\mu_k,\Sigma_k) \tag{12}$$

In a scenario where Z is revealed, i.e. for completely observed data, the MLE solution can be found analytically. The data log-likelihood is given by:

$$l(\theta, D) = \log\prod_n p(z_n, x_n) = \log\prod_n p(z_n|\pi)p(x_n|z_n,\mu,\sigma) \tag{13}$$

$$= \sum_n \log\prod_k \pi_k^{z_n^k} + \sum_n \log\prod_k N(x_n;\mu_k,\sigma)^{z_n^k} \tag{14}$$

$$= \sum_n\sum_k z_n^k \log\pi_k - \sum_n\sum_k z_n^k \frac{1}{2\sigma^2}(x_n - \mu_k)^2 + C \tag{15}$$

The MLE solution for π_k, μ_k, and σ_k is given by:

$$\pi_{k,MLE} = \arg\max_\pi l(\theta, D) \tag{16}$$

$$\mu_{k,MLE} = \arg\max_\mu l(\theta, D) \tag{17}$$

$$\sigma_{k,MLE} = \arg\max_\sigma l(\theta, D) \tag{18}$$

The closed-form solution requires the true value of z_n, for example:

$$\mu_{k,MLE} = \frac{\sum_n z_n^k x_n}{z_n^k} \tag{19}$$

4.2 EM algorithm for GMMs

In cases where Z is unobserved, notice that the expected complete log-likelihood comprises the expected value of z_n:

$$\langle l_c(\theta; x, z)\rangle = \sum_n \langle \log p(z_n|\pi)\rangle_{p(z|x)} + \sum_n \langle \log p(x_n|z_n, \mu, \Sigma)\rangle_{p(z|x)} \tag{20}$$

$$= \sum_n \sum_k \langle z_n^k\rangle \log \pi_k - \frac{1}{2}\sum_n \sum_k \langle z_n^k\rangle ((x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k) + \log|\Sigma_k| + C) \tag{21}$$

Therefore, if $\langle z_n^k\rangle$ can be estimated, the expected log-likelihood can be computed. This task is completed in the E-step of EM algorithm. The overall strategy of EM algorithm is to run E-steps followed by M-steps in an iterative manner to maximize the expected complete log-likelihood $\langle l_c(\theta)\rangle$. In the E-step (expectation step), the expected value of the sufficient statistics of the latent variable is computed. Essentially, we are doing an inference for the sufficient statistics. In this case, the sufficient statistics is Z:

$$\tau_n^{k(t)} = \langle z_n^k\rangle_{q^{(t)}} = p(z_n^k = 1|x_n, \mu^{(t)}, \Sigma^{(t)}) = \frac{\pi_k^{(t)} N(x_n|\mu^{(t)}, \Sigma^{(t)})}{\sum_i \pi_i^{(t)} N(x_n|\mu^{(t)}, \Sigma^{(t)})} \tag{22}$$

In the M-step (maximization step), we maximize the expected log-likelihood by computing the MLE for the parameters π_k, μ_k, Σ_k, using the plug-in value of $\langle z_n^k\rangle$ obtained from E-step:

$$\pi_k^* = \arg\max_\pi \langle l_c(\theta)\rangle \tag{23}$$

$$\mu_k^* = \arg\max_\mu \langle l_c(\theta)\rangle \tag{24}$$

$$\Sigma_k^* = \arg\max_\Sigma \langle l_c(\theta)\rangle \tag{25}$$

4.3 Compare K-means algorithm and EM

Recall that K-means is a clustering algorithm which iteratively runs 2 steps:

1. Assignment step: assign each data sample to its nearest cluster centroid.
2. Update step: recompute the new centroid for each new cluster.

Figure 4 shows a case where K-means converges after 8 iterations, and we are able to identify 2 clusters. In K-means, the assignment step is doing hard assignment, in which a data sample either belong to a cluster (with probability equal to 1) or not (0 probability):

$$z_n^{(t)} = \arg\max (x_n - \mu_k)^T \Sigma_k^{-1(t)} (x_n - \mu_k) \tag{26}$$

In EM, the E-step does soft assignment: for each data sample x_n, it computes the probabilities of x_n belonging to each of the k clusters, as seen in equation (25). In K-means, the update step recomputes the mean as the weighted sum of the data, where all weights are either 1 or 0:

$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)} \tag{27}$$

In EM, the M-step also recomputes the mean, but with soft weights:

$$\mu_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} x_n}{\sum_n \tau_n^{k(t)}} \tag{28}$$

Therefore, K-means can be considered as a hard-assignment version of EM.

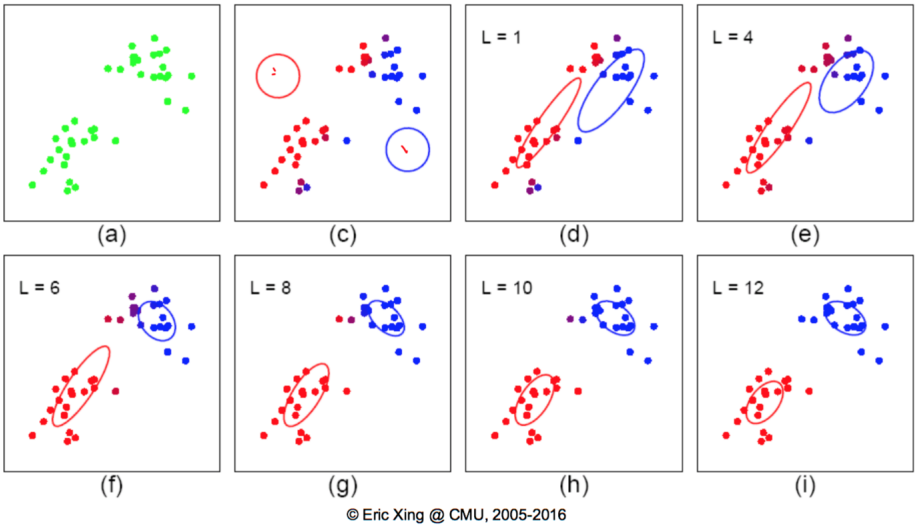

Figure 4: K-means algorithm for clustering

5 EM

5.1 Complete and Incomplete Log-likelihood

Lets denote X as observable variable(s) and Z as hidden variable(s). If we could observe Z, then the complete log-likelihood is defined as:

$$l_c(\theta; x, z) = \log p(x, z|\theta) \tag{29}$$

Usually optimizing $l_c()$ given both x and z is straightforward because it decomposes into a sum of local factors. The parameters for each factor can be estimated separately. However, given that Z is not observed, $l_c()$ is a random quantity which cannot be maximized directly.

With unobserved Z, our objective becomes the log of marginal probability:

$$l_c(\theta; x) = \log \sum_z p(x, z|\theta) = \log \sum_z p(z|\theta_z)p(x|z, \theta_x) \tag{30}$$

This is called incomplete log-likelihood. Now, the objective won't decouple which makes the parameter estimation problem very hard.

5.2 Expected Complete Log-Likelihood

To make the parameter estimation tractable in presence of unobserved variables we define a surrogate function called expected complete log-likelihood. We will also show that it is a lower bound on the incomplete log-likelihood and thus we hope that maximizing this yield a maximizer for the likelihood.

For any distribution $q(z|x, \theta_z)$, we define the expected complete log-likelihood $\langle l_c(\theta; x, z)\rangle$ as:

$$\langle l_c(\theta; x, z)\rangle = \sum_z q(z|x, \theta_z)p(x, z|\theta_x) \tag{31}$$

It is a deterministic function of θ because we are taking an expectation over the unobserved random z. It is also linear in $l_c()$ which implies that it inherits its factorizability.

We could also show that it is a lower bound on the original incomplete log-likelihood using the following arguments:

$$\begin{aligned} l(\theta; x) &= \log p(x|\theta) && (32)\\ &= \log \sum_z p(x, z|\theta) && (33)\\ &= \log \sum_z q(z|x) \frac{p(x, z|\theta)}{q(z|x)} && (34)\\ & && (35) \end{aligned}$$

Note that log is a concave function. Thus, by using Jensen's inequality we get

$$\begin{aligned} l(\theta; x) &\geq \sum_z q(z|x) \log \frac{p(x, z|\theta)}{q(z|x)} && (36)\\ &= \sum_z q(z|x) \log p(x, z|\theta) + H_q && (37)\\ &= \langle l_c(\theta; x, z) \rangle + H_q && (38)\\ &\geq \langle l_c(\theta; x, z) \rangle && (39) \end{aligned}$$

5.3 EM as Coordinate-Ascent on Free Energy

For a fixed data x, we could define a function called the free energy as:

$$F(q, \theta) = \sum_z q(z|x) \log \frac{p(x, z|\theta)}{q(z|x)} \leq l(\theta; x) \tag{40}$$

Now, the EM algorithm can be seen as coordinate ascent on F, where we alternatively minimize q and θ.

5.4 E-step

$$q^{t+1} = \operatorname*{argmax}_q F(q, \theta^t) \tag{41}$$

The solution to the E-step is the posterior distribution over the latent variable given the data and the parameters.

$$q^{t+1} = p(z|x, \theta^t) \tag{42}$$

We could prove this easily by substituting this into $F(q, \theta)$ and showing that it attains the bound $l(\theta; x) \geq$

$F(q, \theta)$.

$$F(p(z|x, \theta^t), \theta^t) = \sum_z p(z|x, \theta^t) \log \frac{p(x, z|\theta^t)}{p(z|x, \theta^t)} \tag{43}$$

$$= \sum_z p(z|x) \log p(x, |\theta^t) \tag{44}$$

$$= \log p(x, |\theta^t) \tag{45}$$

$$= l(\theta^t; x) \tag{46}$$

Before looking at M-step lets define the form of $p(x, z|\theta)$. Without loss of generality we can assume that $p(x, z|\theta)$ is a generalized family distribution:

$$p(x, z|\theta) = \frac{1}{Z(\theta)} h(x, z) \exp\left(\sum_i \theta_i f_i(x, z)\right) \tag{47}$$

Now we can write the $\langle l_c(\theta; x, z)\rangle$ under q^{t+1} as

$$\langle l_c(\theta^t; x, z)\rangle_{q^{t+1}} = \sum_z q(z|x, \theta^t) \log p(x, z|\theta^t) - A(\theta) \tag{48}$$

$$= \sum_i \theta_i^t \langle f_i(x, z)\rangle_{q(z|x,\theta^t)} - A(\theta) \tag{49}$$

Under the special case of that $P(x|z)$ are GLIMs, then

$$f_i(x, z) = \eta^T(z)\xi_i(x) \tag{50}$$

Therefore,

$$\langle l_c(\theta^t; x, z)\rangle_{q^{t+1}} = \sum_i \theta_i^t \langle \eta_i^T\rangle_{q(z|x,\theta^t)} \xi_i(x) - A(\theta) \tag{51}$$

5.5 M-step

$$\theta^{t+1} = \operatorname{argmax}_\theta F(q^{t+1}, \theta) \tag{52}$$

Note that $F(q, \theta)$ breaks into two terms

$$F(q, \theta) = \sum_z q(z|x) \log \frac{p(x, z|\theta)}{q(z|x)} \tag{53}$$

$$= \sum_z q(z|x) \log p(x, z|\theta) - \sum_z q(z|x) \log q(z|x) \tag{54}$$

$$= \langle l_c(\theta; x, z)\rangle_q - H_q \tag{55}$$

The first term is the expected complete log likelihood and the second term is entropy which does not depend on θ. Thus in M-step we only need to consider the first term. So,

$$\theta^{t+1} = \operatorname{argmax}_\theta \sum_z q(z|x) \log p(x, z|\theta) \tag{56}$$

When q is optimal, this is the same as MLE of the fully observed $p(x, z|\theta)$, but instead of the sufficient statistics for z, their expectations with respect to $p(z|x, \theta)$ are used.

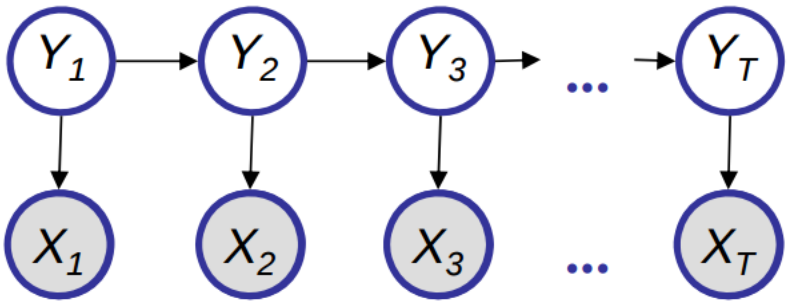

Figure 5: A hidden markov model

6 More examples

6.1 HMMs

We can look at learning Hidden-Markov models in an EM-like framework. Learning in HMMs can be done in both ways: Supervised and unsupervised. In the supervised learning setting, we are given annotated data with the correct labels for the sequences. In such a case, it is a fully-observed model and we can directly apply MLE techniques to learn the parameters of the model. Contrastingly, in the unsupervised learning setting we have unannotated data. We can only observe some of the variables and the remaining variables are unobserved (latent variables). In such a case, it is a partially-observed model and requires us to use a EM-like approach to learn the parameters.

6.1.1 Baum-Welch algorithm

Baum-Welch algorithm is an EM-framework approach to learning the parameters of a HMM. The complete log-likelihood of a HMM can be written as:

$$l_c(\theta; x, Y) = \log p(x, Y) = \log \prod_n (p(Y_{n,1}) \prod_{t=2}^{T} p(Y_{n,t}|Y_{n,t-1}) \prod_{t=1}^{T} p(x_{n,t}|Y_{n,t})) \tag{57}$$

Similarly, the expected complete log-likelihood can be written as:

$$\langle l_c(\theta; x, Y) \rangle = \sum_n (\langle Y^i_{n,1} \rangle_{p(Y_{n,1}|x_n)} \log \pi_i) + \sum_n \sum_{t=2}^{T} (\langle Y^i_{n,t-1} Y^j_{n,t} \rangle_{p(Y_{n,t-1}, Y_{n,t}|x_n)} \log a_{i,j}) \tag{58}$$

$$+ \sum_n \sum_{t=1}^{T} (x^k_{n,t} \langle Y^i_{n,t} \rangle_{p(Y_{n,t}|x_n)} \log b_{i,k}) \tag{59}$$

where $A = a_{i,j}$ is the transition matrix which gives $p(Y_t = j|Y_{t-1} = i)$ and $B = b_{i,k}$ is the emission matrix which gives $p(x_t = k|Y_t = i)$.

The E-step of the EM algorithm is then given by:

$$\gamma^i_{n,t} = \langle Y^i_{n,t} \rangle = p(Y^i_{n,t} = 1|x_n) \tag{60}$$

$$\xi^{i,j}_{n,t} = \langle Y^i_{n,t-1} Y^j_{n,t} \rangle = p(Y^i_{n,t-1} = 1, Y^j_{n,t} = 1|x_n) \tag{61}$$

The M-step is given by:

$$\pi_i^{ML} = \frac{\sum_n \gamma_{n,1}^i}{N} \tag{62}$$

$$a_{i,j}^{ML} = \frac{\sum_n \sum_{t=2}^{T} \xi_{n,t}^{i,j}}{sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i} \tag{63}$$

$$b_{i,k}^{ML} = \frac{\sum_n \sum_{t=1}^{T} \gamma_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i} \tag{64}$$

In the unsupervised setting, Baum-welch algorithm proceeds as follows

1. Start with an initial best guess of parameters θ for the model
2. Estimate $a_{i,j}$ and $b_{i,k}$ from the training data

$$a_{i,j} = \sum_{n,t} \langle Y_{n,t-1}^i Y_{n,t}^j \rangle \tag{65}$$

$$b_{i,k} = \sum_{n,t} \langle Y_{n,t}^i \rangle x_{n,t}^k \tag{66}$$

3. Update θ according to estimated $a_{i,j}$ and $b_{i,k}$. This is the plain old MLE estimation problem, which can be done by any previously explored technique.
4. Repeat steps 2 and 3 until convergence

It can be proven that we get a more likely set of parameters θ at the end of each iteration compared to the previous one.

6.2 EM for BNs

For general bayesian networks, the EM algorithm can be given as:

1. For each node i, reset the expected sufficient statistics $ESS_i = 0$
2. For each data sample n, do inference with $X_{n,H}$ and for each node i, update the expected sufficient statistics correspondingly $ESS_i += \langle SS_i(X_{n,i}, X_{n,\pi_i}) \rangle_{p(X_{n,H}, X_{n,-H})}$
3. For each node i, obtain the MLE parameters $\theta_i = MLE(ESS_i)$
4. Go to step 1 until convergence

6.3 Conditional mixture model

To model $p(y|x)$, we can use a set of different experts each responsible for different regions of the input space. We can use a latent variable Z to choose the expert by a softmax gating function

$$p(z^k = 1|x) = softmax(\xi^T x) \tag{67}$$

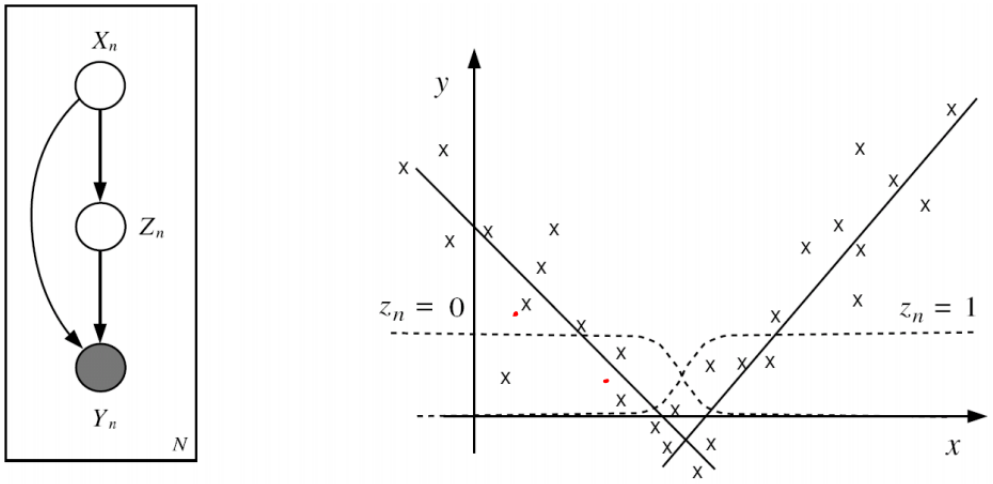

Figure 6: Conditional Mixture Model

The model for a conditional mixture model looks like:

$$P(y|x) = \sum_k p(z^k = 1|x, \xi) p(y|z^k = 1, x, \theta_i, \sigma) \tag{68}$$

The loss function in this case has the following form

$$\langle l_c(\theta; x, y, z) \rangle = \sum_n \langle \log p(z_n|x_n, \xi) \rangle_{p(z|x,y)} + \sum_n \langle \log p(y_n|x_n, z_n, \theta, \sigma) \rangle_{p(z|x,y)} \tag{69}$$

The E-step in the EM algorithm is

$$\tau_n^{k(t)} = p(z^k = 1|x_n, y_n, \theta) = \frac{p(z_n^k = 1|x_n) p_k(y_n|x_n, \theta_k, \sigma_k^2)}{\sum_j p(z_n^k = 1|x_n) p_j(y_n|x_n, \theta_j, \sigma_j^2)} \tag{70}$$

The M-step in the EM algorithm uses the standard normal equation for linear regression $\theta = (X^T X)^{-1} X^T Y$ but with the data reweighted by τ or using the weighted IRLS algorithm to update $\xi_k, \theta_k, \sigma_k$ based on data points (x_n, y_n) with weights $\tau_n^{k(t)}$.

6.4 Other variants of EM

There are several variants of the EM algorithm, few of which are:

- Sparse EM: This algorithm does not recompute the posterior probability on each data point under all data models, because it is almost zero (hence, the sparsity). Instead, it keeps an "active" list which it updates every once in a while
- Generalized (incomplete) EM: In some cases, it is intractable to get the maximum likelihood estimates of the parameters in the M-step, even with complete data. This algorithm still makes progress by doing an M-step that improves the likelihood a bit in a way similar to the IRLS step in the conditional mixture model parameter estimation

7 Summary

In summary, EM is a family of algorithms that help in maximizing the likelihood of latent variable models. It computes the MLE of the parameters in two steps:

1. Estimating "unobserved" or "missing" data from observed data and current parameters. Can also be seen as filling-in the values of the latent variables based on the current best guess
2. Using this "complete" data to get the current MLE parameters. Can also be seen as updating the parameters based on the guesses (or filling-in) that we did in the previous step

Essentially, EM maximizes likelihood by optimizing a lower bound on the log-likelihood in the M-step and closing the gap between the bound and the log-likelihood in the E-step.

EM is currently the most popular method for parameter estimation in partially-observed models. It does not require any learning-rate parameter (unlike many gradient based approaches) and is very fast for low-dimensions. It also ensures convergence as the likelihood increases after each iteration.

The disadvantages of EM are mostly that it can lead to a local optima instead of a global optima. EM can also be slower than conjugate gradient especially near convergence. We can also observe that it needs an expensive inference step (in the E-step) that might be intractable in some situations.