10-708: Probabilistic Graphical Models 10-708, Spring 2014

9 : Learning Partially Observed GM : EM Algorithm

Lecturer: Eric P. Xing *Scribes: Rohan Ramanath, Rahul Goutam*

1 Generalized Iterative Scaling

In this section, we summarize from the previous lecture the use of Generalized Iterative Scaling (GIS) to find the Maximum Likelihood Estimation (MLE) of a feature based Undirected Graphical Model (UGM). Each feature has a weight θ_k which represents the numerical strength of the feature and whether it increases or decreases the probability of the clique k. To represent these features as a part of the prababilistic model, we can multiply in the clique potentials to get:

$$p(x) = \frac{exp\{\sum_i \theta_i f_i(x_{c_i})\}}{Z(\theta)}$$

At the maximum likelihood setting of the parameters, for each clique, the model marginals must be equal to the observed marginals, i.e. $p^*_{MLE}(x) = \frac{m(x)}{N} = \tilde{p}(x)$, where $m(x)$ is the number of times the configuration x is found out of a possible N. The scaled likelihood function can then be represented as:

$$\begin{aligned}
\tilde{l}(\theta; D) &= \frac{l(\theta; D)}{N} = \frac{1}{N}\sum_n \log p(x_n|\theta) \\
&= \sum_x \tilde{p}(x) \log p(x|\theta) \\
&= \sum_x \tilde{p}(x) \sum_i \theta_i f_i(x) - \log Z(\theta)
\end{aligned}$$

Solving the log $Z(\theta)$ term is very complicated. The tangent to the log function forms an upper bound on it ($\log\ Z(\theta) \leq \mu Z(\theta) - \log \mu - 1, \forall \mu$). This fact is used to get a lower bound on the value of $\tilde{l}(\theta; D)$ by using $\mu = Z^{-1}(\theta^{(t)})$. Subsitituting $\Delta\theta_i^{(t)} = \theta_i - \theta_i^{(t)}$, we get:

$$\begin{aligned}
\tilde{l}(\theta; D) &\geq \sum_x \tilde{p}(x) \sum_i \theta_i f_i(x) - \frac{Z(\theta)}{Z(\theta^{(t)})} - \log\ Z(\theta^{(t)}) + 1 \\
&= \sum_x \tilde{p}(x) \sum_i \theta_i f_i(x) - \frac{\sum_x exp\{\sum_i \theta_i f_i(x)\}}{Z(\theta^{(t)})} - \log\ Z(\theta^{(t)}) + 1 \\
&= \sum_i \theta_i \sum_x \tilde{p}(x) f_i(x) - \frac{\sum_x exp\{\sum_i \theta_i^{(t)} f_i(x)\} exp\{\sum_i \Delta\theta_i^{(t)} f_i(x)\}}{Z(\theta^{(t)})} - \log\ Z(\theta^{(t)}) + 1 \\
&= \sum_i \theta_i \sum_x \tilde{p}(x) f_i(x) - \sum_x p(x|\theta^{(t)}) exp\{\sum_i \Delta\theta_i^{(t)} f_i(x)\} - \log\ Z(\theta^{(t)}) + 1
\end{aligned}$$

We can further relax the constrains by assuming $\sum_i f_i(x) = 1$ and $f_i(x) \geq 0$. Since the exponential function is convex, $exp(\sum_i \pi_i x_i) \leq \sum_i \pi_i exp(x_i)$

$$\begin{aligned}
\tilde{l}(\theta; D) &\geq \sum_i \theta_i \sum_x \tilde{p}(x) f_i(x) - \sum_x p(x|\theta^{(t)}) \sum_i f_i(x)\, exp\{\Delta\theta_i^{(t)}\} - \log\, Z(\theta^{(t)}) + 1 = \Lambda(\theta) \\
\frac{\partial \Lambda}{\partial \theta_i} &= \sum_x \tilde{p}(x) f_i(x) - exp(\Delta\theta_i^{(t)}) \sum_x p(x|\theta^{(t)}) f_i(x) = 0 \\
exp(\Delta\theta_i^{(t)}) &= \frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p(x|\theta^{(t)}) f_i(x)} = \frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)} Z(\theta^{(t)})
\end{aligned}$$

where $p^{(t)}(x)$ is the unnormalized version of $p(x|\theta^{(t)})$. Substituting the result in the update equation we get:

$$\begin{aligned}
\theta_i^{(t+1)} &= \theta_i^{(t)} + \Delta\theta_i^{(t)} \\
p^{(t+1)}(x) &= p^{(t)}(x) \prod_i e^{\Delta\theta_i^{(t)} f_i(x)} \\
&= p^{(t)}(x) \prod_i \Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)} \Big)^{f_i(x)} \\
\theta_i^{(t+1)} &= \theta_i^{(t)} + \log \Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)} \Big)
\end{aligned}$$

To summarize we can say that GIS can be used for iterative scaling on general UGM with feature based pottentials. IPF is a special case of GIS in which the clique potential is built on features defined as an indicator function of clique configurations.

2 Mixture Models

Let us consider a dataset X. We wish to model the data by specifying a joint distribution $p(X_i, Z_i) = p(X_i|Z_i)p(Z_i)$. Here, $Z_i \sim Multinomial(\pi)$ (where $\pi_j \geq 0$, $\sum_j^k \pi_j = 1$,and the parameter π_j gives $p(Z_i = j)$,) and $X_i|Z_j = j \sim \mathcal{N}(\mu_j, \Sigma_j)$. We let k denote the number of values that the Z_is can take on. Thus, our model posits that each X_i was generated by randomly choosing Z_i from $\{1, \ldots, k\}$, and then X_i was drawn from one of k Gaussians depending on Z_i. This is called the **mixture of Gaussians** model. Also, note that the $z^{(i)}$s are latent random variables, meaning that theyre hidden/unobserved.

This makes the problem tractable as each conditional distribution $P(X_i|Z_i)$ can now be assumed to be unimodal. In the specific case of a mixture of gaussians, marginal density is factorized as:

$$P(X_i|\mu, \Sigma) = \sum_k \pi_k \, \mathcal{N}(X_i|\mu_k, \Sigma_k)$$

where π_k is called the mixture proportion and $\mathcal{N}(X_i|\mu_k, \Sigma_k)$ is the gaussian distribution for the mixture component.

The task is to learn the values of π_k, μ_k and Σ_k. In the fully observed case, it is fairly straightforward as we would have simply decomposed the log likelihood function and then found the maximum likelihood estimates for all the parameters. The log likelihood can be expressed as:

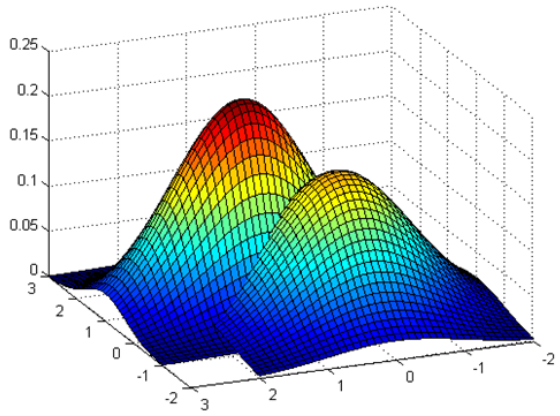

Figure 1: A mixture of 2 Gaussians

$$
\begin{aligned}
l(\theta; D) &= \log \prod_i P(x_i, z_i) \\
&= \log \prod_i P(z_i|\pi) P(x_i|z_i, \mu, \Sigma) \\
&= \sum_i \log \prod_k \pi_k^{z_i^k} + \sum_i \log \prod_k \mathcal{N}(x_i|\mu_k, \Sigma_k)^{z_i^k} \\
&= \sum_i \sum_k z_i^k \log(\pi_k) - \frac{1}{2} \sum_i \sum_k z_i^k (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) + C
\end{aligned}
$$

Since it is fully observed, the log likelihood completely decomposes. However, if we do not observe the values of z_i as in the mixture model setting, all the parameters become coupled together as shown below:

$$
l_c(\theta; D) = \sum_i \log \sum_z P(x_i, z|\theta) = \sum_i \log \sum_z P(z|\theta_z) P(x_i|z, \theta_x)
$$

It is easy to observe that the summation is inside the log and the log likelihood does not decompose nicely anymore. Hence, it is hard to find a closed form solution to the maximum likelihood estimates. This provides motivation for the Expectation Maximization (EM) algorithm.

3 K-Means

Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \leq n$) given by $z = \{z_1, z_2, \ldots, z_n\}$ so as to minimize the within-cluster sum of squares (WCSS). We start with initial random guesses for the class centroids and iterate between the following two steps until convergence.

- **Expectation Step:** Use some distance metric (Euclidean distance, cosine similarity, e.t.c) and the

current guess of the centroids to assign every data point to the nearest centroid.

$$z_i^{(t)} = argmin_k (x_i - \mu_k^{(t)})^T \Sigma_k^{-1^{(t)}} (x_i - \mu_k^{(t)})$$

- **Maximization Step:** Use the new class assignment to recompute centroid values.

$$\mu_k^{(t+1)} = \frac{\sum_i \delta(z_i^{(t)}, k) x_i}{\sum_i \delta(z_i^{(t)}, k)}$$

To understand the EM viewpoint, we can think of each of the clusters being associated with some distribution, i.e. $P(x_i|z_i = k) \sim \mathcal{N}(X_i|\mu_k, \Sigma_k)$ and we would like to learn the parameters (μ_k and Σ_k) for each distribution.

4 EM Algorithm

The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of latent variables. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. As seen in Section 2 the maximum likelihood estimator cannot be used for inference in the partially observed setting. Finding a maximum likelihood solution requires taking derivatives of the likelihood function with respect to both the parameters and the latent variables and simultaneously solving the resulting equations. In statistical models with latent variables, this usually leads to intractable solutions. The EM algorithm works around this by considering the expected complete log likelihood.

Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the latent variables are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology.In the M-step, the likelihood function is maximized under the assumption that the latent variables are known. The estimate of the latent variables from the E-step are used in lieu of the actual latent variables.

The log likelihood from Section 2 can be written in the form of an expected log likelihood as:

$$\sum_i \sum_k < z_i^k >_{P(Z|X;\theta)} \log(\pi_k) - \frac{1}{2} \sum_i \sum_k < z_i^k >_{P(Z|X;\theta} ((x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) + \log|\Sigma_k|) + C$$

where $< z_i^k >_{P(Z|X;\theta)}$ is the expected value of z_i^k with respect to $P(Z|X;\theta)$.

The EM algorithm maximizes $< l_c(\theta; D) >_{P(Z|X;\theta)}$ by iterating between two steps (called E and M steps). In the E step, we compute the sufficient statistics of the hidden variable (i.e., Z) using the current estimate of the parameters (i.e. μ and Σ). The M step, maximizes the value of the parameters using the expected value of the hidden variables computed in the preceding E step. Specifically, in the GMM, we compute $< z_i^k >_{P(Z|X;\theta)}$ in the E step and maximize the expected complete log likelihood with respect to the model parameters μ, Σ and π in the M step to obtain:

$$\begin{aligned}
\pi_k^{(t+1)} &= \frac{< n_k >_{P(Z|X;\theta)}}{N} \\
\mu_k^{(t+1)} &= \frac{\sum_i \tau_i^{k(t)} x_i}{\tau_i^{k(t)}} \\
\Sigma_k^{(t+1)} &= \frac{\sum_i \tau_i^{k(t)} (x_i - \mu_k^{t+1})(x_i - \mu_k^{t+1})^T}{\tau_i^{k(t)}}
\end{aligned}$$

where $\tau_i^{k(t)} =< z_i^k >_{P(Z|X;\theta)}$ and $< n_k >_{P(Z|X;\theta)} = \sum_i \tau_i^{k(t)}$

5 Comparision between K-means and EM

EM algorithm for the mixture of gaussians is a soft version of K means. For K-means, in the E-step, we assign points to clusters and in the M-step, we recompute the cluster centers assuming each point belongs to a single cluster. In the EM version, the E-step we assign points to clusters in a probabilistic manner and in the M-step, we recompute the cluster centroid assuming that the points are assigned to clusters in a probabilistic manner where the weight of each point is given by $\tau^{k(t)}$

6 Theory underlying EM Algorithm

The previous sections tell us how the EM algorithm can be used to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. Let X represent the set of observable variables and Z represent the set of all latent variables. The probability model is $p(X, Z|\theta)$. If Z were observed, we define the complete log likelihood as

$$l_c(\theta; X, Z) = \log p(X, Z|\theta)$$

The ML estimation would then be maximizing the complete log likelihood. If the probability $p(X, Z|\theta)$ factors such that separate components of θ occur in separate factors, then the log separates them into different terms which can be maximized independently (decoupling).

However, since Z is not observed, it has to be marginalized out and the incomplete log likelihood function takes the following form :

$$l_c(\theta; X) = \log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$$

The incomplete log likelihood cannot be decoupled because the summation over $p(X, Z|\theta)$ lies inside the logarithm.

In order to handle this, we average over z to remove the randomness using an averaging distribution $q(Z|X)$. The expected complete log likelihood can then be defined as :

$$< l_c(\theta; x, Z) >_q = \sum_Z q(Z|x, \theta) \log p(x, Z|\theta)$$

The expected complete log likelihood is a deterministic function of θ. It is also linear in the complete log likelihood function and, hence, inherits its factorizability. The expected complete log likelihood also provides a lower bound on the complete log likelihood as shown below:

$$\begin{aligned} l(\theta; x) &= \log p(x|\theta) \\ &= \log \sum_Z p(x, Z|\theta) \\ &= \log \sum_Z q(Z|x) \frac{p(x, Z|\theta)}{q(Z|x)} \end{aligned}$$

$$\begin{aligned} & \geq \sum_Z q(Z|x) \log \frac{p(x,Z|\theta)}{q(Z|x)} \textbf{(Jensen's inequality)} \\ l(\theta;x) & \geq < l_c(\theta;x,Z) >_q + H_q \end{aligned}$$

where H_q is the entropy of the distribution q and does not depend on θ. H_q is a constant for a particular distribution q. Hence, in order to maximize the complete log likelihood with respect to θ, it is sufficient to maximize the exptected complete log likelihood.

The EM algorithm can be seen as a coordinate ascent algorithm. For a fixed data X, let a functional F, called the free energy, be defined as :

$$F(q,\theta) = \sum_Z q(Z|x) \log \frac{p(x,Z|\theta)}{q(Z|x)} \leq l(\theta;x)$$

The EM algorithm is a coordinate ascent algorithm on F. The E-step and the M-step can then be defined as

- E-step : $q^{t+1} = argmax_q F(q,\theta^t)$
- M-step : $\theta^{t+1} = argmax_\theta F(q^{t+1},\theta)$

At the $(t+1)^{th}$ step, we maximize the free energy $F(q,\theta)$ with respect to q. For this optimal choice of q^{t+1}, we maximize $F(q,\theta)$ again with respect to θ to get the optimal updated value θ^{t+1}.

Let us look at the E-step. If we set $q^{t+1}(z|x)$ to the posterior distribution of the latent variables given the data and parameters, $p(z|x,\theta^t)$, we maximize $F(q,\theta^t)$. The proof of this claim is given below :

$$\begin{aligned} F(p(Z|x,\theta^t),\theta^t) &= \sum_Z p(Z|x,\theta^t) \log \frac{p(x,Z|\theta^t)}{p(z|x,\theta^t)} \\ &= \sum_Z q(Z|x) \log p(x|\theta^t) \\ &= \log p(x|\theta^t) \\ &= l(\theta^t;x) \end{aligned}$$

Since $l(\theta^t;x)$ is an upper bound on $F(q,\theta^t)$, the proof shows that $F(q,\theta^t)$ is maximized by setting $q(Z|x)$ to the posterior probability distribution $p(Z|x,\theta^t)$. The above claim can also be proved using variational calculus or the fact that

$$l(\theta;x) - F(q,\theta) = KL(q||p(z|x,\theta))$$

Without loss of generality, we can assume $p(x,Z|\theta)$ to be a generalized exponential family distribution. Then,

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

A special case is when $p(x|Z)$ are GLM. Then,

$$f_i(x,Z) = \eta_i^T(z)\xi_i(x)$$

The expected complete log likelihood under $q^{t+1} = p(Z|x, \theta^t)$ is given below :

$$\begin{aligned} < l_c(\theta^t; x, Z) >_{q^{t+1}} &= \sum_Z q(Z|x, \theta^t) \log p(x, Z|\theta^t) - A(\theta) \\ &= \sum_i \theta_i^t < f_i(x, Z) >_{q(Z|x,\theta^t)} - A(\theta) \\ &=^{p \sim GLIM} \sum_i \theta_i^t < \eta_i(Z) >_{q(Z|x,\theta^t)} \xi_i(x) - A(\theta) \end{aligned}$$

Now, let us analyze the M-step of the EM algorithm. The M-step can be viewed as the maximization of the expected complete log likelihood. This is shown below :

$$\begin{aligned} F(q, \theta) &= \sum_Z q(Z|x) \log \frac{p(x, Z|\theta)}{q(Z|x)} \\ &= \sum_Z q(Z|x) \log p(x, Z|\theta) - \sum_Z q(Z|x) \log q(Z|x) \\ &= < l_c(\theta; x, Z) >_q + H_q \end{aligned}$$

The free energy, hence, breaks into two terms. The first term is the expected complete log likelihood and the second term, which is independent of θ, is the entropy. Thus, maximizing the free energy is equivalent to maximizing the expected complete log likelihood.

$$\theta^{t+1} = argmax_\theta < l_c(\theta; x, Z) >_{q^{t+1}} = argmax_\theta \sum_Z q(Z|x) \log p(x, Z|\theta)$$

Under optimal q^{t+1}, this is equivalent to solving a standard MLE of fully observed model $p(x, Z|\theta)$, with the sufficient statistics involving Z replaced by their expectations w.r.t. $p(Z|x, \theta)$.

7 Example: Hidden Markov Model

Let us look at the learning problem of Hidden Markov Models (HMM) in the framework of the EM algorithm. The learning problem in HMMs could be

- Supervised learning

  In supervised learning, we have annotated data with the correct answer known. Examples include a genomic region $x = x_1 x_2 \dots x_{1000000}$ where we have good annotations of the CpG islands or if a casino player allows us to observe him as he changes dice and produces 10,000 rolls. Since all variables are fully observed, we can learn the parameters θ of the model by applying MLE.

- Unsupervised learning

  In unsupervised learning, we do not have annotated data. We only observe some of the variables and the remaining variables (latent variables) remain unknown. Examples include the porcupine genome where we don't know how frequent the CpG islands are or 10,000 rolls of dice of a casino player without knowing when he changes dice.

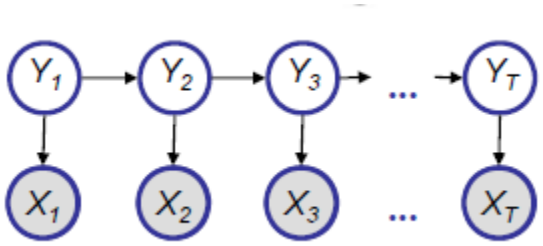

Figure 2: A hidden markov model

7.1 Baum-Welch Algorithm

The Baum-Welch algorithm is used learn the parameters θ of a HMM. The complete log likelihood of a HMM can be written as

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

The expected complete log likelihood can be written as

$$\begin{aligned} < l_c(\theta; x, Y) > \quad = \quad & \sum_n (< Y^i_{n,1} >_{p(Y_{n,1}|x_n)} \log \pi_i) + \sum_n \sum_{t=2}^{T} (< Y^i_{n,t-1} Y^j_{n,t} >_{p(Y_{n,t-1}Y_{n,t}|x_n)} \log a_{i,j}) \\ & + \sum_n \sum_{t=1}^{T} (x^k_{n,t} < Y^i_{n,t} >_{p(Y_{n,t}|x_n)} \log b_{i,k}) \end{aligned}$$

where $A = \{a_{i,j}\} = P(Y_t = j|Y_{t-1} = i)$ is the transition matrix and $B = \{b_{i,j}\} = P(Y_t = i|x_t = j)$ is the emission matrix.
The E-step of the EM algorithm is then

$$\begin{aligned} \gamma^i_{n,t} \quad &= \quad < Y^i_{n,t} >= p(Y^i_{n,t} = 1|x_n) \\ \xi^{i,j}_{n,t} \quad &= \quad < Y^i_{n,t-1} Y^j_{n,t} >= p(Y^i_{n,t-1} = 1, Y^j_{n,t} = 1|x_n) \end{aligned}$$

The M-step of the algorithm is

$$\begin{aligned} \pi^{ML}_i \quad &= \quad \frac{\sum_n \gamma^i_{n,1}}{N} \\ a^{ML}_{ij} \quad &= \quad \frac{\sum_n \sum_{t=2}^{T} \xi^{i,j}_{n,t}}{\sum_n \sum_{t=1}^{T-1} \gamma^i_{n,t}} \\ b^{ML}_{ik} \quad &= \quad \frac{\sum_n \sum_{t=1}^{T} \gamma^i_{n,t} x^k_{n,t}}{\sum_n \sum_{t=1}^{T-1} \gamma^i_{n,t}} \end{aligned}$$

In the unsupervised learning case, the Baum-Welch algorithm proceeds as follows :

1. Start with the best guess of parameters θ for the model

2. Estimate A_{ij}, B_{ik} in the training data.

$$A_{ij} = \sum_{n,t} < Y^i_{n,t-1} Y^j_{n,t} >$$
$$B_{ik} = \sum_{n,t} < Y^i_{n,t} > x^k_{n,t}$$

3. Update θ according to A_{ij}, B_{ik}. Now, problem becomes supervised learning.
4. Repeat steps 2 and 3 until convergence.

It can be proven that we get a more (or equally) likely parameter set θ in each iteration.

8 EM for general BNs

Algorithm 1 shows the algorithm for EM in a general Bayesian Network.

while *not converged* **do**
  for *each node* i **do**
    $ESS_i = 0$
  end
  for *each data sample* n **do**
    do inference with $x_{n,H}$ **for** *each node* i **do**
      $ESS_i += < SS_i(x_{n,i}, x_{n,\pi_i}) >_{p(x_{n,H}|x_{n,-H})}$
    end
  end
  for *each node* i **do**
    $\theta_i = MLE(ESS_i)$
  end
end

Algorithm 1: EM algorithm for general BN

9 Summary of EM algorithm

In summary, the EM algorithm is a way of maximizing the likelihood of the latent variable models. It computes the ML estimate of the parameters by breaking the original problem into two parts:

1. The estimation of unobserved data from observed data and current parameters.
2. Using this model with all models observed to find the ML estimates of parameters.

The algorithm alternates between filling in the latent variables using the best guess and updating the parameters based on this guess.

10 EM for Conditional Mixture Model

The model for the Conditional mixture model is defined as

$$P(Y|x) = \sum_k p(z^k = 1|x, \xi) p(y|z^k = 1, x, \theta_i, \sigma)$$

The objective function for EM is defined as

$$< l_c(\theta; x, y, z) >= \sum_n < \log p(Z_n|x_n, \xi) >_{p(Z|x,Y)} + \sum_n < \log(p(Y_n|x_n, Z_n, \theta, \sigma) >_{p(Z|x,Y)}$$

The E-step in the EM algorithm is

$$\tau_n^{k(t)} = P(Z_n^k = 1|x_n, Y_n, \theta) = \frac{P(Z_n^k = 1|x_n) p_k(Y_n|x_n, \theta_k, \sigma_k^2)}{\sum_j p(Z_n^j = 1|x_n) p_j(y_n|x_n, \theta_j, \sigma_j^2)}$$

The M-step in the EM algorithm uses the normal equation for linear regression $\theta = (x^T x)^{-1} x^T Y$, but with the data re-weighted 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 $\tau_n^{k(t)}$.

11 EM Variants

There are some variants of the EM algorithm.

- Sparse EM: The sparse EM algorithm does not recompute the posterior probability on each data point under all models, which are very close to 0. Instead, it keeps an active list which it updates every once in a while.
- Generalized (Incomplete) EM : Sometimes, it is hard to compute the Maximum likelihood estimate of the parameters in the M-step, even with the complete data. However, we can still make progress by doing a M-step that increases the likelihood a bit (like a gradient step).

12 EM: Report Card

EM is one of the most popular methods to get the Maximum Likelihood Estimate or the Maximum a Posteriori Estimate of parameters in a model where there are some latent unobserved variables. EM does not require any learning rate parameter and is very fast for low-dimensions. It also ensures convergence as each iteration is guaranteed to improve likelihood.

The cons of EM algorithm are that it can give us a local optima instead of global optima. EM can be slower than conjugate gradient, especially near convergence. It requires an expensive inference step and it is a maximum likelihood/MAP method.

Disclaimer: Some portions of the content has been directly taken from the lecture slides[1] and last years scribe notes[2]. The authors do not claim originality of the scribe.

[1] http://www.cs.cmu.edu/ epxing/Class/10708/lectures/lecture9-EM.pdf
[2] http://www.cs.cmu.edu/ epxing/Class/10708-13/lecture/scribe9.pdf