Indian Institute of Technology Kanpur
CS777 Topics in Learning Theory

Scribe: Gurpreet Singh
Instructor: Purushottam Kar
Date: 4 April, 2018

LECTURE

23

Probabilistic Modelling - II

1 Recap

In the last lecture, we introduced probabilistic modeling and discussed the EM Algorithm as an example of the Minorization Maximization technique.

In this lecture, we will add to the discussion of the EM algorithm, and extend our discussion to more Probabilistic Models and techniques, specifically Markov Chain Monte Carlo methods.

2 The EM Algorithm

As was discussed in the last lecture, the EM algorithm is effectively a Minorization Maximization technique. For the sake of completeness, we present the Minorization Maximization technique in Algorithm 1

Algorithm 1: Minorization Maximization

$$g^{t+1} \longleftarrow \text{MIN}\left(f, \boldsymbol{\theta}^{(t)}\right)$$
$$\boldsymbol{\theta}^{(t+1)} \longleftarrow \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ g^{(t+1)}\left(\boldsymbol{\theta}\right)$$

where the function g is a Minorization of a function f with respect to a point $\boldsymbol{\theta}_o$ if the following conditions hold

1. $g(\boldsymbol{\theta}_o) = f(\boldsymbol{\theta}_o)$
2. $\forall \boldsymbol{\theta} \in \Theta$, $g(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta})$
3. $g(.)$ should be easy to minimize

In order to show that the EM algorithm is in fact a Minorization Maximization, we need to derive a well defined function g such that it satisfies the stated conditions.

The EM objective can be stated as finding the values of the parameters $\boldsymbol{\theta}$ such that

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\, \mathbb{P}\left[\mathbf{X} \mid \boldsymbol{\theta}\right]$$

Therefore, our function f is given by $f(\boldsymbol{\theta}) = \mathbb{P}[\mathbf{X} \mid \boldsymbol{\theta}]$. We now move on to find a suitable function g such that the g is a minorization of the function f. Firstly, using rules of probability, we have

$$\begin{aligned}
\log \mathbb{P}[\mathbf{X} \mid \boldsymbol{\theta}] &= \log \left(\sum_{\mathbf{Z} \in \mathcal{Z}} \mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}] \right) \\
&= \log \left(\sum_{\mathbf{Z} \in \mathcal{Z}} \mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right] \frac{\mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}]}{\mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right]} \right) \\
&= \log \left(\underset{\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}}{\mathbb{E}} \left[\frac{\mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}]}{\mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right]} \right] \right)
\end{aligned}$$

Now since log is a concave function, we can use Jensen's Inequality to say

$$\begin{aligned}
\mathbb{P}[\mathbf{X} \mid \boldsymbol{\theta}] &\geq \underset{\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}}{\mathbb{E}} \left[\log \left(\frac{\mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}]}{\mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right]} \right) \right] \\
&= \underset{\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}}{\mathbb{E}} \left[\log \mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}] - \log \mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right] \right]
\end{aligned}$$

Also, one can see that the equality is met if $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$. Hence, this satisfies the first two conditions we stated for the function g. Also, for many classes of models we can compute the Complete Data Likelihood $\mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}]$ as well as the posterior over the latent variables, and it is usually easy to minimize the above function, however, many times, this step itself needs to be approximated. Hence, we have the function g which is stated as below

$$g(\boldsymbol{\theta}) = \underset{\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}}{\mathbb{E}} \left[\log \mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}] - \log \mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right] \right]$$

Using this, we can formulate the EM algorithm in terms of the Minorization Maximization technique. For completeness' sake, we state the final algorithm in Algorithm 2

Algorithm 2: The EM Algorithm

1. Initialize $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^0$
2. For $t = 1, 2 \ldots$

$$\begin{aligned}
g^{t+1}(\boldsymbol{\theta}) &\longleftarrow \underset{\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}}{\mathbb{E}} \left[\log \mathbb{P}[\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}] - \log \mathbb{P}\left[\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}\right] \right] \\
\boldsymbol{\theta}^{t+1} &\longleftarrow \arg\max \boldsymbol{\theta} \in \Theta \; g^{t+1}(\boldsymbol{\theta})
\end{aligned}$$

2.1 Gradient Ascent Expectation Maximization

Many times, the maximization step itself might require gradient ascent methods to maximize the objective function g^{t+1}, therefore, it sometimes makes more sense to take only one step in the direction of the gradient,

and recompute the objective in each iteration. More formally, the maximization step in Algorithm 2 can be replaced by the following computation.

$$\boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^{(t)} + \eta_t \cdot \nabla_{\boldsymbol{\theta}} g^{t+1}(\boldsymbol{\theta}^{(t)})$$

The algorithm thus obtained is known as the Gradient Descent EM Algorithm.

2.2 Stochastic Gradient Ascent Expectation Maximization

Computing the gradient with respect to even one point, $\boldsymbol{\theta}^{(t)}$ can be expensive if the number of data points, represented by $\mathbf{X}$, is very large. Therefore, we can instead perform one iteration of the algorithm using only one data point, or using a small subset of data-points. The algorithm thus obtained is given in Algorithm 3

Algorithm 3: The EM Algorithm

1. Initialize $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^0$
2. For $t = 1, 2 \ldots$

$$I_t \sim n, \text{ uniform and i.i.d.}$$
$$g^{t+1}(\boldsymbol{\theta}) \longleftarrow \text{MIN}\left(\sum_{\mathbf{Z} \in \mathcal{Z}} \mathbb{P}\left[\mathbf{x}^{I_t}, \mathbf{Z} \,|\, \boldsymbol{\theta}\right], \boldsymbol{\theta}^{(t)}\right)$$
$$\boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^{(t)} + \eta_t \cdot \nabla_{\boldsymbol{\theta}} g^{t+1} \boldsymbol{\theta}^t$$

Expectation Minimization suffers from the same problems as MLE or MAP estimates, we are only trusting one model. That is, we are computing only one value of the parameters $\boldsymbol{\theta}$, and therefore there is no uncertainty modeled with respect to the parameters.

The solution to this problem is to model the uncertainty in the parameters using the posterior of the parameters with respect to the data points $\mathbf{X}$, weight or trust the values of the parameters with their posterior values, given by $\mathbb{P}\left[\boldsymbol{\theta} \,|\, \mathbf{X}\right]$

It is possible to compute the Bayesian Predictive Posterior with the posterior, $\mathbb{P}\left[\boldsymbol{\theta} \,|\, \mathbf{X}\right]$ known, instead of the plug-in-posterior. The Bayesian Predictive Posterior is computed as follows

$$\mathbb{P}\left[\mathbf{x}^* \,|\, \mathbf{X}\right] = \int_{\boldsymbol{\theta}} \mathbb{P}\left[\mathbf{x}^* \,|\, \boldsymbol{\theta}\right] \mathbb{P}\left[\boldsymbol{\theta} \,|\, \mathbf{X}\right] \boldsymbol{\theta}$$

Without knowing the posterior, we can only find the plug-in-posterior, which is given as

$$\begin{aligned}\mathbb{P}\left[\mathbf{x}^* \,|\, \mathbf{X}\right] &= \int_{\boldsymbol{\theta}} \mathbb{P}\left[\mathbf{x}^* \,|\, \boldsymbol{\theta}\right] \delta_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}) \boldsymbol{\theta} \\ &= \mathbb{P}\left[\mathbf{x}^* \,|\, \boldsymbol{\theta}^*\right]\end{aligned}$$

where $\boldsymbol{\theta}^*$ is a point estimate (usually MLE or MAP estimate) for the parameters.

The Bayesian Predictive Posterior is a useful quantity, however can be only exactly computed in case there is conjugacy in the model, and the introduction of latent variables again complicate the computations, and usually render the model to be intractable (with the exception of probabilistic PCA for dimensionality reduction). Even without latent variables, if the model does not exhibit conjugacy, computing the posterior becomes intractable.

It should be clear to the reader that we cannot use the EM algorithm to compute the posterior distribution, even approximately, in such cases. Therefore, we need more stronger and general models to achieve our goals. We will state some of the methods in the next section.

3 Approximating Disrtibutions

As previously discussed, our goal is to approximate distributions which are, otherwise, intractable, or hard to compute. There are essentially two classes of techniques to tackle this problem.

1. **Parametric Techniques**

   Parametric techniques includes those techniques which approximate the objective distribution $\mathbb{P}[\boldsymbol{\theta} \mid \mathbf{X}]$ using a "nice" distribution $q(\boldsymbol{\theta})$. The $q \in \mathcal{Q}$ distribution is known as the proposal distribution.

   In order to compute the predictive posterior, we can use Monte Carlo Principle, which is stated in the following theorem.

   Theorem 23.1 (Monte Carlo Principle)**.** If we draw a set of samples $\mathcal{S} = \boldsymbol{\theta}^{(i)}{}_{i=1}^{n}$ from a target density distribution $p(\boldsymbol{\theta})$ defined on Θ, then for a large enough sample set, we can approximate the density as

   $$p(\boldsymbol{\theta}) \approx \frac{1}{n} \sum_{i=1}^{n} \delta_{\boldsymbol{\theta}^{(i)}}(\boldsymbol{\theta})$$

   The predictive posterior can now be approximated using the Monte Carlo method as follows

   $$\mathbb{P}[\mathbf{x}^* \mid \mathbf{X}] \approx \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{\theta} \in \mathcal{S}} \mathbb{P}[\mathbf{x}^* \mid \boldsymbol{\theta}]$$

   where $\mathcal{S} \sim q^{|\mathcal{S}|}$

   Examples of Parametric Techniques incldues Laplace Approximation, which approximates the posterior distribution with a Gaussian Distribution with the mean as the MAP estimate and the Covariance Matrix as the Hessian of the log posterior.

   Some other examples include Variational Bayes where we assume the hypothesis class of the proposal distribution.

   The reader might be wondering if we can directly approximate the distribution $\mathbb{P}[\mathbf{x}^* \mid \mathbf{X}]$. Indeed, we can, however most models have data point dependent predictive posterior. In formal terms, most models are of the form where the data point is composed of two parts, and we wish to model only one part, for example, linear regression, where each data point is the tuple $y, \mathbf{x}$. In this case, we model only $\mathbf{y}$ keeping $\mathbf{X}$ constant, and in such models, the predictive posterior takes the form $\mathbb{P}[y^* \mid \mathbf{x}^*, \mathbf{y}, \mathbf{X}]$. In such cases, we would require to approximate the predictive posterior for every data point separately, which makes the approach infeasible.

   [Non-Parametric Techniques]

   We never require the posterior of a model directly, but only to compute quantities such as the predictive posterior. Therefore, if it is possible to only get samples from the posterior, and not the complete posterior itself, we can approximate the predictive posterior using the Monte Carlo method as seen earlier.

   This is the key idea behind non-parametric techniques, which do not assume anythuing about the objective distribution, but samples from it using various techniques, Some of the common methods that lie in this category are Markov Chain Monte Carlo (MCMC) method, Metropolis Hastings (MH) sampling, Hamiltonian Markov Chain (HMC) and Stochastic Gradient Langevin Dynamics (SGLD).

   As discussed, the goal for non-parametric techniques is to sample from an distribution $p(\boldsymbol{\theta}) = \mathbb{P}[\boldsymbol{\theta} \mid \mathbf{X}]$.

   We assume that we can compute the value of $p(\boldsymbol{\theta})$ up to a proportionality constant, which is otherwise unknown. More formally, we assume

   $$p(\boldsymbol{\theta}) \propto \exp(-f(\boldsymbol{\theta}))$$

and $f(\boldsymbol{\theta})$ is easy to compute or approximate.

This assumption holds in most of the models, where only the denominator of the posterior, which is just a proportionality constant is intractable, but we can easilty compute the numerator.

We will discuss more about the MCMC method in the next section and later look at how these methods can sample from an unknown distribution.

4 Markov Chain Monte Carlo Method

For the sake of simplicity, we assume that the hyopthesis set Θ is finite and discrete for the following discussion, although the results we see hold for the non-finite and continuous cases as well. We represent the possible states as $\boldsymbol{\theta}_j{}_{j=1}^{m}$

The distribution over the parameters $\boldsymbol{\theta} \in \Theta$ will be a probability vector, say $\boldsymbol{\pi}^*$. This is our objective distribution which, in practice, we aim to compute or approximate. The intuitive idea is to start with a base distriubution, say $\boldsymbol{\pi}^0$ and subsequently sample, in a way, from this distribution hoping that the generated samples actually belong to the objective distribution. This is the base idea of the MCMC methods, which is discussed in more detail in this lecture and the subsequent one.

Since we are sampling θ sequentially, we refer to each sample as a state. Also, for a sequence of samples or states, there is a probability distribution which corresponds to the probability for choosing the next state.

MCMC methods rely on a simple assumption, that the probability of choosing a state conditioned on all the previous states is independent of all the states, but only the previous one. More formally, if the sequence of states that we have sampled are given as $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)} \dots \boldsymbol{\theta}^{(t-1)}$, and the current state (to be sampled) is given as $\boldsymbol{\theta}^{(t)}$, then we say

$$\mathbb{P}\left[\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)} \dots \boldsymbol{\theta}^{(t-1)}\right] = \mathbb{P}\left[\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t-1)}\right] \tag{1}$$

Suppose if there are m states, then we can write a transition matrix $P \in \mathbb{R}_{+}^{m \times m}$ such that $P_{i,j} = \mathbb{P}\left[\boldsymbol{\theta}_j \mid \boldsymbol{\theta}_i\right]$.

Therefore, using the Markov Chain rule in equation 1, we can say that the probability of transition to a state $\boldsymbol{\theta}_j$ from a state $\boldsymbol{\theta}_i$, represented as $\mathbb{P}\left[i \to j\right]$ is given by $P_{i,j}$ and the probability distribution over the next state is given by $P_{i,:}$. We can now give the sampling algorithm if we know the transition matrix as given in Algorithm 4.

The sampling strategy for MCMC methods first sample the first state using the probability vector $\boldsymbol{\pi}^{(0)}$. When we say sample from a probability vector, we say that the distribution is a dirac delta distribution and the support points are the states $1 \dots m$, where each state corresponds to a parameter value. Suppose state i corresponds to $\boldsymbol{\theta}_i$, then we sample the parameters from the probability vector $\boldsymbol{\pi}$ as

$$\boldsymbol{\theta}^* \sim \sum_{j=1}^{m} \pi_j \cdot \delta_{\boldsymbol{\theta}_j}\left(\boldsymbol{\theta}\right)$$

Suppose if the current state of the Markov chain is i, then the sampling distribution for the next state can be given by the i^{th} row of the transition matrix P, $P_{i,:}$. Also, the sampling distribution for the first state is given by our initial or proposal distribution $\boldsymbol{\pi}^{(0)}$.

Let us calculate the probability of the t^{th} sample correspond to a particular state or parameter. The probability of the first (initial) state being the state i is equal to $\pi_i^{(0)}$. Speaking more generally, the probability distribution for the first sample is given by $\boldsymbol{\pi}^{(0)}$.

Now suppose if the probability distribution for the t^{th} sample is given by $\boldsymbol{\pi}^{(t)}$, then we are interested in the probality distribution of the next sample. Using rules of probability, we can write this as

$$\begin{aligned}
\mathbb{P}\left[\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}_j\right] &= \sum_{i=1}^{m} \mathbb{P}\left[\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}_i\right] \cdot \mathbb{P}\left[\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}_j \mid \boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}_i\right] \\
&= \boldsymbol{\pi}^{(t)} P_{:,\,j}
\end{aligned}$$

Algorithm 4: MCMC Sampling Strategy

1. Sample the first state as $\boldsymbol{\theta}^{(0)} \sim \sum_{j=1}^{m} \pi_j^0 \cdot \delta_{\boldsymbol{\theta}_j}(\boldsymbol{\theta})$
2. For $t = 1, 2 \dots T$
   Sample state $\boldsymbol{\theta}^{(t+1)}$ using the probability vector given by $P_{s^{(t)},:}$
3. Return the set of samples ${\boldsymbol{\theta}^{(t)}}_{t=1}^{T}$

Therefore the probability distribution $\boldsymbol{\pi}^{(t+1)}$ is given by $\boldsymbol{\pi}^{(t)} \cdot P$. Hence, using induction, we have

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(0)} \cdot P^t$$

Therefore, we can say that if $\boldsymbol{\pi}^{(t)}$ converges to $\boldsymbol{\pi}^*$, the objective distribution, then the sample that is obtained at the end of each chain (when the chain converges) will in fact be a sample from the objective distribution. However that would be make the process extremely slow. Instead, once the chain has converged, we know that the subsequent samples (if we continue sampling) will all be effectively sampled from $\boldsymbol{\pi}^*$. Hence, the sampling strategy given in Algorithm 4 can be used, if we start collecting the samples after the chain converges. That is, the return set can now be written as ${\boldsymbol{\theta}^{(t)}}_{t=T^+}^{T}$ for some large T^+.

We will now state some properties of the transition matrix P which will help us to prove that any choice of our initial distribution (given certain conditions, mentioned later) will converge to the optimal distribution, $\boldsymbol{\pi}^*$. More specifically, we state properties of a right stochastic matrix (defined below).

Definition 23.1 (Right Stochastic Matrix)**.** A Right Stochastic Matrix is a square matrix with real non-negative entries and each row summing to one. a matrix $T \in \mathbb{R}_+^{n \times n}$ is right stochastic iff

$$T\mathbb{1}_n = \mathbb{1}_n$$

where $\mathbb{1}_n$ denotes a n-dimensional vector of ones.

Since P satisfies the contrainsts of a right stochastic matrix, all the properties of a right stochastic matrix are applicable on P. We state some of these properties below.

1. If P is right-stochastic, then $\forall k \in \mathbb{N}$, P^k is right stochastic
2. The largest eigen-value of P is equal to one

   Proof. Since each row of the matrix P denotes a probability vector, and therefore, each element of the vector $P\mathbf{v}$ for some $\mathbf{v} \in \mathbb{R}^n$ is a normalized weighted combination of the dimensions of $\mathbf{v}$. Therefore, each element of $P\mathbf{v}$ is lesser than the maximum dimension value of $\mathbf{v}$, more formally, $\|P\mathbf{v}\|_\infty \leq \|\mathbf{v}\|_\infty$. This suggests that there can be no value of $\lambda > 1$ such that $P\mathbf{v} = \lambda\mathbf{v}$. Also, from the properties of P, we have $P\mathbb{1}_n = \mathbb{1}_n$, therefore there exists an eigen-vector for which the eigen value of P is equal to one. □
3. $\exists \mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{v}P = \mathbf{v}$

In the next lecture, we will see some more properties of the transition matrix, and show the proof of convergence, for any initial distribution, given P, we always converge to $\boldsymbol{\pi}^*$ following the sampling strategy given in Algorithm 4.