Lecture 3 — April 7th

Lecturer: Lester Mackey *Scribe: Jordan Bryan, Dangna Li*

3.1 Recap: Gaussian Mixture Modeling

In the last lecture, we discussed the Gaussian mixture model (GMM), a model for clustered data with real-valued components. We represented a GMM as a generative model for our data:

$$\begin{aligned} z_i &\overset{\text{iid}}{\sim} \text{Mult}(\pi, 1) \\ x_i|z_i &\overset{\text{ind}}{\sim} \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}). \end{aligned}$$

where each z_i represented a latent class / cluster for data point x_i. Our clustering goal in this setting was to infer the z_i's given the x_i's, and we identified a potential solution based on the EM algorithm and probabilistic inference.

Our last lecture ended with a number of questions concerning the behavior of the EM algorithm and its relevance to the maximum likelihood objective. In this lecture, we will answer these questions by considering a more general version of the EM for arbitrary latent variable models, but, first, we will consider a second "teaser" from the GMM setting that hints at the relationship between EM and likelihood.

3.1.1 EM and the expected complete log likelihood

Let $\theta = (\pi, \mu_{1:k}, \Sigma_{1:k})$ be our shorthand for all unknown parameters in the GMM, and let $\theta^{(t)}$ be our estimate for θ after t steps in our EM Algorithm. Recall our definition of the **complete log likelihood** over $(x, z) = (x_{1:n}, z_{1:n})$:

$$\begin{aligned} l_c(\theta; z) &= \sum_{i=1}^{n} \log p(x_i, z_i; \theta) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}[z_i = j] \log(\pi_j \phi(x_i; \mu_j, \Sigma_j)). \end{aligned}$$

If we view Z as a random variable and take the conditional expectation of l_c given x and $\theta^{(t)}$, we obtain

$$\begin{aligned} \mathbb{E}[l_c(\theta; Z)|x, \theta^{(t)}] &= \sum_{i=1}^{n} \sum_{j=1}^{k} p(z_i = j|x_i, \theta^{(t)}) \log(\pi_j \phi(x_i; \mu_j, \Sigma_j)) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \tau_{ij}^{(t)} \log(\pi_j \phi(x_i; \mu_j, \Sigma_j)), \end{aligned} \tag{3.1}$$

where the quantities $\tau_{ij}^{(t)}$ are the very same conditional expectations defined in the E-step of the EM algorithm. Essentially, by forming the conditional expectation of the complete log likelihood, we are able to replace the unknown indicator $\mathbb{I}[z_i = j]$ by the observable conditional expectations $\tau_{ij}^{(t)}$. In a sense, the E-step of EM is computing this **expected complete log likelihood** (ECLL) (3.1). Moreover, if we attempt to maximize the ECLL over θ, we exactly recover the parameter updates of the M-step in the EM algorithm. It appears then that the ECLL – an observable approximation to the unknown complete log likelihood – is central to the EM algorithm, but why, and how does this relate to our goal of maximizing the likelihood of x? To obtain a fully satisfying answer, we turn to a more general latent variable setting.

3.2 EM for General Latent Variable Models

Suppose that our collection of datapoints $X = (X_1, ..., X_n)$ has joint density $p(x; \theta)$ with

$$p(x; \theta) = \sum_{z \in \mathcal{Z}} p(x, z; \theta)$$

for θ a vector of unknown parameters and z a vector of latent variables in a set $\mathcal{Z}$.

Remark. We are implicitly assuming that z is discrete to simplify the exposition. The discussion to follow applies equally in the setting in which z has continuous components.

Our ultimate goal is to estimate the parameters θ by (at least, approximately) maximizing the likelihood of our data. To that end, we will take introduce an important inequality from convex analysis that will allow us to relate the EM algorithm to the maximum likelihood objective.

3.2.1 Jensen's inequality

Definition 1. A twice differentiable function f is **concave** on $\mathcal{X}$ if $f''(x) \leq 0$, $\forall x \in \mathcal{X}$.

Intuitively, if f is concave, then the line segment connecting $(x, f(x))$ and $(y, f(y))$ always lies below the function f.

Example 1. The function $f(x) = \log x$ is concave on $\mathbb{R}_{>0}$ since $f''(x) = -1/x^2 < 0, \forall x \in \mathbb{R}_{>0}$.

Theorem 1 (Jensen's Inequality)**.** If a function f is concave on $\mathcal{X}$, then $\mathbb{E}(f(X)) \leq f(\mathbb{E}(X))$ for any random variable $X \in \mathcal{X}$.

3.2.2 Lower bounding the likelihood

With Jensen's inequality in mind, let us return to our discussion of the general EM algorithm. Since maximizing the likelihood of x directly can be difficult in a latent variable model, we will introduce a tractable lower bound for likelihood that we can maximize in its stead.

First note that we have the following equalities

$$\log p(x;\theta) = \log \sum_{z\in\mathcal{Z}} p(x,z;\theta) = \log \sum_{z\in\mathcal{Z}} q(z)p(x,z;\theta)/q(z).$$

for any distribution q over the latent variable space $\mathcal{Z}$ that contains the support of $p(z|x;\theta)$. We call q the **auxiliary distribution** and we allow this distribution to depend on the observed data x. The right-hand side in the above equation is in fact an expectation under q:

$$\log \sum_{z\in\mathcal{Z}} q(z)p(x,z;\theta)/q(z) \quad = \quad \log \mathbb{E}_q[p(x,Z;\theta)/q(Z)].$$

Since $x \mapsto \log(x)$ is a concave function, we can apply Jensen's inequality to pull the expectation out of the log and thereby obtain a lower bound on the log likelihood:

$$\log p(x;\theta) = \log \mathbb{E}_q[p(x,Z;\theta)/q(Z)] \geq \mathbb{E}_q[\log(p(x,Z;\theta)/q(Z))] = \mathcal{L}(q,\theta).$$

We refer to this final lower-bounding term, $\mathcal{L}(q,\theta)$, as the **auxiliary function**.

3.2.3 The EM algorithm

It turns out that the EM algorithm in its general form is precisely coordinate ascent on the two arguments q and θ of the log likelihood lower bound $\mathcal{L}(q,\theta)$:

The EM algorithm

Step (0): Initialize $\theta^{(1)}$ arbitrarily.

Step (1): Alternate the following two steps until convergence:

(i) **E-Step**: $q^{(t+1)} = \operatorname{argmax}_q \mathcal{L}(q,\theta^{(t)})$

(ii) **M-Step**: $\theta^{(t+1)} = \operatorname{argmax}_\theta \mathcal{L}(q^{(t+1)},\theta)$

However, this formulation is a bit abstract, so we will unpack each step to better understand what the algorithm is doing.

Unpacking the M-Step

In the M-Step, the EM algorithm maximizes the auxiliary function with respect to parameters θ. A close examination of the auxiliary function reveals that

$$\begin{aligned}\mathcal{L}(q,\theta) &= \sum_{z\in\mathcal{Z}} q(z)\log(p(X,z;\theta)/q(z)) \\ &= \sum_{z\in\mathcal{Z}} q(z)\log p(X,z;\theta) - \sum_{z\in\mathcal{Z}} q(z)\log q(z),\end{aligned}$$

where the first term on the right-hand side is the expectation of the complete log likelihood with respect to q and the second has no dependence on θ. We can conclude that the M-step of maximizing the auxiliary function with respect to θ is equivalent to maximizing the ECLL under q. To determine which q to use, we shall next examine the E-step.

Unpacking the E-step

In the E-step, we find the auxiliary density that maximizes the auxiliary function over q when the parameters θ are held fixed.

Claim: The solution is $q^{(t+1)}(z) = p(z|x;\theta^{(t)})$.

Claim Check:

$$\begin{aligned}\mathcal{L}(p(z|x;\theta^{(t)}),\theta^{(t)}) &= \sum_{z\in\mathcal{Z}} p(z|x,\theta^{(t)})\log(p(x,z;\theta^{(t)})/p(z|x,\theta^{(t)})) \\ &= \sum_{z\in\mathcal{Z}} p(z|x,\theta^{(t)})\log p(x;\theta^{(t)}) \\ &= \log p(x;\theta^{(t)})\end{aligned}$$

Here we see that the final term in this string of equalities is the log likelihood of the data. Since we know that the log likelihood provides an upper bound for the auxiliary function for all q, the fact that we have equality means that our choice of q must be maximal!

3.2.4 Consequences

Our coordinate ascent formulation and the explicit expressions of the E and M-steps have a number of important consequences for our understanding of the EM algorithm.

1. We have a **second, operational view of EM**:

   - The E-step computes the expected complete likelihood, i.e. $\mathbf{E}_{p(\cdot|x,\theta^{(t)})}[l_c(\theta;Z)]$.
   - Then, the M-step maximizes this quantity over θ.

   In practice, the E-step often amounts to computing a few expected statistics of Z. For example in the Gaussian mixture model case, it suffices to compute $\mathbb{P}(z_i = j \mid x_i)$, for $j = 1, 2, \cdots, k$.

   Typically, maximizing the ECLL in the M-step is no more difficult than maximizing the complete log likelihood with observed z. Hence, EM is often practical and valuable when maximizing the complete log likelihood is straightforward (as it was in the GMM case).

2. **Claim:** The log likelihood $\log p(x;\theta^{(t)})$ improves monotonically on each EM iteration.

   Proof. Each M-step maximizes $\mathcal{L}(q^{(t+1)},\theta)$ with respect to θ, so

   $$\mathcal{L}(q^{(t+1)},\theta^{(t)}) \le \mathcal{L}(q^{(t+1)},\theta^{(t+1)}).$$

   We also know that $\mathcal{L}(q^{(t+1)},\theta^{(t+1)})$ provides a lower bound for $\log p(x;\theta^{(t+1)})$, and hence

   $$\mathcal{L}(q^{(t+1)},\theta^{(t+1)}) \le \log p(x;\theta^{(t+1)}).$$

Now the E-step ensures that $\mathcal{L}(q^{(t+1)}, \theta^{(t)}) = \log p(x; \theta^{(t)})$, the log likelihood at step t. Combining these inequalities, we obtain

$$\log p(x; \theta^{(t)}) \leq \log p(x; \theta^{(t+1)}).$$

Thus, the log likelihood at step t is no worse than the log likelihood at step $t+1$, which proves our claim. □

Our proof shows that *any* improvement in $\mathcal{L}(q^{(t+1)}, \theta)$ over $\mathcal{L}(q^{(t+1)}, \theta^{(t)})$ would suffice to improve the likelihood on each step. If full maximization of $\mathcal{L}(q^{(t+1)}, \theta)$ is expensive, one might imagine simply selecting some $\theta^{(t+1)}$ with an improved auxiliary function value. This idea is often called **generalized EM**.

3. The **objective** $\log p(x, \theta^{(t)})$ **converges** as $t \to \infty$. This fact warrants a few remarks.
   - The objective will typically not converge to a global optimum. In practice, random restarts are common, just like in the k-means setting.
   - Under some technical conditions, $\theta^{(t)}$ will converge as well.
   - Any convergence rates will heavily depend on the specific properties of the density $p(x; \theta)$.

The EM algorithm enables unsupervised inference in latent variable models by providing a general-purpose tool for parameter estimation. We will make use of this tool in many contexts throughout this course, and we will begin by demonstrating its applicability to a new class of mixture models. Recall that the GMM was primarily suitable for clustering continuous data. In the next section we consider the multinomial mixture model (MMM) which is often used for clustering discrete data.

3.3 Multinomial Mixture Models

3.3.1 Motivation

Consider the task of document review for a law firm, in which you need to scan tens of thousands of emails to determine their relevance to a legal matter. It could be helpful to pre-group these emails according to their topics, e.g., business, lunch, spam, etc. A **multinomial mixture model (MMM)** would model each topic as a distribution over English words (the lunch topic, for example, might place higher weight on the word "pizza" than the business topic) and view words in an email as independent draws from a single topic distribution.

3.3.2 The model

Let $X_1, X_2, \cdots, X_n$ be per-document word count vectors, where n is the total number of documents. In particular, $X_{iv} = m$, if word v appears m times in document i. To simplify

notation, we will assume that each document contains M words in total (i.e., $\sum_v X_{iv} = M$); this assumption can easily be relaxed.

The MMM models all documents as independent draws from the probability mass function $p(x) = \sum_{j=1}^k \pi_j f(x_j; \theta_j)$. As in the GMM case, we have k different topic components but the nature of the components has changed. Now each component

$$f(x_j; \theta_j) \propto \prod_{v=1}^{p} \theta_{jv}^{x_v}$$

is the pmf of a multinomial distribution, $\text{Mult}(\theta_j, M)$, where $\theta_j \in \mathbb{R}^p$, p is the total number of possible words, and θ_{jv} is the unknown probability of selecting word v for topic j. As usual, π_j is the unknown probability of selecting topic j, and $\sum_{j=1}^k \pi_j = 1$.

As in the GMM case, the MMM can also be viewed as an equivalent generative model with explicit latent variables:

- First we draw latent topic indicators $z_i \overset{\text{iid}}{\sim} \text{Mult}(\pi, 1)$.
- Given each topic z_i, x_i is drawn independently from a topic-specific multinomial distribution:
$$x_i \mid z_i \overset{\text{ind}}{\sim} \text{Mult}(\theta_{z_i}, M).$$
Notice that the only difference from the GMM is that we have replaced the Gaussian distributions with multinomial distributions.

Our goal is to infer the latent topics z_i by first estimating the parameters $\theta_{1:k}$. We use our general formulation to derive an EM algorithm for the MMM and see that the result will be almost identical to the EM algorithm for GMMs.

3.3.3 EM for MMMs

The complete log likelihood

A central quantity in any new EM algorithm derivation is the complete log likelihood. For an MMM, this takes the form

$$\begin{aligned}
&\log p(x_{1:n}, z_{1:n}; \pi, \theta_{1:k}) \\
&= \sum_{i=1}^{n} \log p(x_i, z_i; \pi, \theta_{1:k}) \\
&= \sum_{i=1}^{n} \log p(z_i; \pi) + \sum_{i=1}^{n} \log p(x_i | z_i; \theta_{1:k}) \\
&= \sum_{i=1}^{n} \log \pi_{z_i} + \sum_{i=1}^{n} \sum_{v=1}^{P} x_{iv} \log \theta_{z_i v} + \text{parameter-free terms} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}\{z_i = j\} \log \pi_j + \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{v=1}^{P} x_{iv} \mathbb{I}\{z_i = j\} \log \theta_{jv} + \text{parameter-free terms.}
\end{aligned}$$

The E-step

The E-step requires that we compute the expected complete log likelihood under the conditional distribution $p(z_{1:n}|x_{1:n};\pi,\theta_{1:k})$:

$$\mathbf{E}_{p(\cdot|x_{1:n},\pi,\theta)}[\log p(x_{1:n},z_{1:n};\pi_j,\theta_{1:k})] = \sum_{i=1}^{n}\sum_{i=1}^{k}\tau_{ij}\log\pi_j + \sum_{i=1}^{n}\sum_{j=1}^{k}\sum_{v=1}^{P}x_{iv}\tau_{ij}\log\theta_{jv}$$

where $\tau_{ij} = p(z_i = j|x_i;\pi,\theta_{1:k})$. As in the GMM case, it suffices to compute τ_{ij}'s to complete this step. By Bayes' Rule,

$$p(z_i = j|x_i;\pi,\theta_{1:k}) \propto p(z_i = j;\pi)p(x_i|z_i;\theta_{1:k}).$$

If we plug in the MMM-specific densities, we obtain the expression

$$p(z_i = j|x_i;\pi,\theta_{1:k}) = \frac{\pi_j f(x_i;\theta_j)}{\sum_{l=1}^{k}\pi_l f(x_i;\theta_j)}.$$

The M-step

The M-step now maximizes the expected complete log likelihood. In this case, we have a closed-form solution for the parameter updates:

$$\pi_j = \frac{1}{n}\sum_{i=1}^{n}\tau_{ij},$$

which is the relative frequency of topic j in the dataset, and

$$\theta_{jv} = \frac{\sum_{i=1}^{n}x_{iv}\tau_{ij}}{M\sum_{i=1}^{n}\tau_{ij}},$$

which is the relative frequency of word v in topic j.

3.4 Mixture Modeling + EM: Practical Considerations

Many of the practical considerations that arose in our discussion of k-means are also relevant in the probabilistic mixture modeling setting:

- While GMMs are not appropriate for all data types (e.g., discrete data), we can handle most data types by employing appropriate generative models.
- The solutions of the EM algorithm are usually suboptimal. Might clever initialization strategies (like that advocated in k-means++ for Lloyd's algorithm) lead to suboptimality bounds and improved practical performance? Are there other ways to counter the suboptimality of EM? These might be a worthy questions to tackle as a final project.

- There is no single satisfying approach to choosing k. Fortunately, many of the methods we have discussed for k-means also apply to the mixture modeling setting.
- Our models have assumed that all data points are *independent* draws from a common mixture, but many datasets have known structural dependencies among datapoints. Next time, we will learn to leverage sequential structure in unsupervised learning, using hidden Markov models.