STAT 3503/8109 Lecture 12 Notes

Edoardo Airoldi

*Scribe: Srikar Katta**

Fall 2020: November 16, 2020

1 Introduction

So far, we've described models with observed random variables y, latent random variables x, and unknown constants θ. In Latent Dirichlet Allocation, we had m documents each with n_j words, so that word w_{ij} is the j^{th} word of i^{th} document; we then defined $y = \{w_{11}, \ldots, w_{1n_1}, \ldots, w_{mn_m}\}$, our observed quantities. We also had k topics and V terms in our vocabulary, stored in a V by k matrix, β where each term in β represents the probability of a term appearing in a specific topic. We also had α, an unknown cost underlying the Dirichlet distribution. So, $\theta_i = \{\beta_{V\times k}, \alpha\}$. Lastly, for the j^{th} location in the i^{th} document, there was a variable indicating the topic, labeled z_{ij}. Then, our set of latent random variables could be summarized as $x = \{\theta_1, \ldots, \theta_m, z_{11}, \ldots, z_{1n_1}, \ldots, z_{mn_m}\}$.

In any approach, we first compute two quantities: the complete likelihood and the proper likelihood/log likelihood. If we have no latent variables in our model, then the complete and proper likelihoods are exactly the same. Using these quantities, we perform inference – either for the unknown quantities or latent random variables. If we want to estimate the unknown constants θ, then we computed the likelihood/log-likelihood and use maximum likelihood or the method of moments to utilize the data to find the unknown constants that makes the data most likely.

If we want to estimate the latent random variables x, then we have different approaches. In a two-level model, we have observed random variables $y_1, \ldots, y_n \overset{\text{iid}}{\sim} \mathbb{P}(y|x)$ and a latent

*Please share any comments or suggestions with Srikar Katta at srikar@temple.edu

random variable $x \sim \mathbb{P}(x|\alpha)$, for some unknown constant α. First, we identify a prior distribution for x (whether α is known or unknown is irrelevant for x). Then, we combine the prior and model $(\mathbb{P})y|x))$ to obtain the posterior distribution – the probability of the latent variable *after* seeing the data – $\mathbb{P}(x|\alpha, y_1, \ldots, y_n)$. From the posterior, we estimate $\hat{x}$ using maximum a posteriori or posterior mean.

In the fully general two-layer models, we have latent random variables *and* unknown constants. *If* we can compute the likelihood proper (i.e., the likelihood of the data given the constants, without any latent variables), then we can discuss maximum likelihood estimation or method of moments estimation for θ. The data generating process gives us the complete likelihood. If we can integrate out the latent random variables, then we can estimate the unknown constants with maximum likelihood or method of moments.

However, there are many times in which we cannot integrate out the latent random variables. In these situations, performing maximum likelihood or method of moments is infeasible. To solve this, we will discuss the expectation maximization algorithm.

2 Intuition

Suppose we cannot estimate some unknown constant α using maximum likelihood or method of moments because we cannot integrate out latent random variables from our complete likelihood. Consider the following algorithm – a fairly intuitive approach to estimating unknown constants.

1. Guess α^0. The 0 index represents our guess at the 0^{th} iteration of this algorithm
2. Now, repeat the following until α does not change anymore
   - **Expectation Step 1:** Find the expectation of X, given α^t (α at the t^{th} iteration),

$$\begin{aligned} x^t &= \mathbb{E}_{x|\alpha^t}[x|\alpha^t] \\ &= \int_X x \cdot \mathbb{P}(x|a^t)dx \end{aligned}$$

   - **Expectation Step 2:** Find the joint probability distribution for $y_1, \ldots, y_n$ and "plug in" x^t for x,

$$\mathbb{P}(y_1, \ldots, y_n, x^t|\alpha^t)$$

Here, x^t is no longer an unknown, so we can maximize the *proper* likelihood.

- **Maximization Step:** Now, find α^{t+1} by finding the α that maximizes the joint probability distribution,

$$\alpha^{t+1} = \arg\max_{\alpha} \mathbb{P}(y_1, \ldots, y_n, x^t|\alpha)$$

Throughout this algorithm, we cannot compute the proper likelihood analytically (i.e., using the integral), but this algorithm allows us to find a best guess for x in each iteration based on a guess for α repeatedly until we "converge" on a $\hat{\alpha}_{MLE}$. Note that in the true expectation maximization algorithm will instead compute the *posterior* distribution for α.

3 Expectation-Maximization (E-M) Algorithm

Suppose we observe random variable y and do not observe some constant α and some random variable that depends on α, $x \sim \mathbb{P}(x|\alpha)$. Then, the log of the proper likelihood would be

$$\begin{aligned} logLikelihood &= log\mathbb{P}(y|\alpha) \\ &= log \int_x \mathbb{P}(y, x|\alpha)dx \\ &= log \int_x \frac{\mathbb{P}(y, x|\alpha)}{q(x)} q(x)dx, \end{aligned}$$

for some function of x called q. Suppose we find q such that $\frac{\mathbb{P}(y,x|\alpha)}{q(x)}$ is still a probability distribution function. Then, we can express the log-likelihood as

$$\begin{aligned} logLikelihood &= log \int_x \frac{\mathbb{P}(y, x|\alpha)}{q(x)} q(x)dx \\ &= log\mathbb{E}_{q(x)} \left[\frac{\mathbb{P}(y, x|\alpha)}{q(x)} \right] \end{aligned}$$

Ideally, we want to compute this value, $log\mathbb{E}_{q(x)} \left[\frac{\mathbb{P}(y,x|\alpha)}{q(x)} \right]$, since that is equivalent to the log-likelihood. However, it is difficult to compute this. However, we can utilize Jensen's inequality to find a *lower bound* for this expectation. Namely, the log of a linear function is greater than or equal to the function of the logs. For example, $log(1+2) \geq log(1) + log(2) = 0 + log(2) = log(2)$. Obviously, $log(3) \geq log(2)$. Since the expectation operator is linear, we

can apply this property to find the lower bound for the log-likelihood:

$$\begin{aligned} logLikelihood &= log\mathbb{E}_{q(x)}\left[\frac{\mathbb{P}(y,x|\alpha)}{q(x)}\right] \\ &\geq \mathbb{E}_{q(x)}\left[log\frac{\mathbb{P}(y,x|\alpha)}{q(x)}\right]. \end{aligned}$$

By maximizing the lower bound instead of the log-likelihood, we are still able to estimate the maximum likelihood estimator. Now, we can extend this calculation further:

$$\begin{aligned} logLikelihood &\geq \mathbb{E}_{q(x)}\left[log\frac{\mathbb{P}(y,x|\alpha)}{q(x)}\right] \\ &= \mathbb{E}_{q(x)}\left[log\mathbb{P}(y,x|\alpha) - logq(x)\right] \\ &= \mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)] - \mathbb{E}_{q(x)}[logq(x)]. \end{aligned}$$

If we use the posterior for x, $\mathbb{P}(x|y,\alpha)$, instead of $q(x)$, then

$$\begin{aligned} logLikelihood &\geq \mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)] - \mathbb{E}_{q(x)}[logq(x)] \\ &= \mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)] - \mathbb{E}_{q(x)}[log\mathbb{P}(x|y,\alpha)] \\ &= \mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)] - 0 \\ &= \mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)]. \end{aligned}$$

Note that $\mathbb{E}_{q(x)}[log\mathbb{P}(x|y,\alpha)] = 0$ comes from K.L. divergence, details of which we do not need to be concerned with. Now our calculation is *much* more palatable, as we only need to find $\mathbb{E}_{q(x)}[log\mathbb{P}(y,x|\alpha)]$.

If given an α, we now have the entire prior distribution for x, which has much more information than just a point estimate for the latent variable. To do this, we find the lower bound of the likelihood. And by maximizing the lower bound, we are still able to find the argument that maximizes the likelihood itself.

3.1 Formal E-M Algorithm

Suppose we have observed variables $y_1, \ldots, y_n$, an unknown variable $x \sim \mathbb{P}(x|\alpha)$ where α is an unknown constant. First, we guess α^0. Instead of computing $x^0 = \mathbb{E}[x|\alpha^0]$ as we did in the intuitive set-up, we can compute the posterior distribution for x, $\mathbb{P}(x|y,\alpha^0)$, which contains much more information. Now, we can compute the lower bound for $log\mathbb{P}(y_1,\ldots,y_n|\alpha)$, which is $\mathbb{E}_{\mathbb{P}(x|y,\alpha^0)}[log\mathbb{P}(y_1,\ldots,y_n,x|\alpha)]$. Then, using this, we can find the argument that maximizes

$\mathbb{E}_{\mathbb{P}(x|y,\alpha^0)}[log\mathbb{P}(y_1,\ldots,y_n,x|\alpha)]$ to estimate α^1 in the first iteration of the algorithm. Then we iterate and repeat the steps until there is a very small difference between the estimates of α in each iteration. That is, repeat the algorithm until $|\alpha^{t-1}-\alpha^t|<\varepsilon$ for very small positive number ε.

Algorithm 1: Formal E-M Algorithm

Result: Estimate of the constants

Guess α^0;

Compute $\mathbb{P}(x|y,\alpha^0)$;

Let $\alpha^1 = \arg\max_\alpha \mathbb{E}_{\mathbb{P}(x|y,\alpha^0)}[log\mathbb{P}(y_1,\ldots,y_n,x|\alpha^0)]$;

Let $\varepsilon > 0$;

while $|\alpha^{t-1}-\alpha^t|\geq\varepsilon$ **do**

  Compute $\mathbb{P}(x|y,\alpha^{t-1})$;

  Let $\alpha^t = \arg\max_\alpha \mathbb{E}_{\mathbb{P}(x|y,\alpha^{t-1})}[log\mathbb{P}(y_1,\ldots,y_n,x|\alpha^{t-1})]$;

end

4 Yellowstone National Park Example

At Yellowstone National Park, the famous geyser Old Faithful shoots water at intermittent intervals. However, upon closer inspection, if we have time on the horizontal axis and frequency on the vertical axis, we would see two humps at separate time intervals with some overlap.

So, if we were to fit a normal distribution to the data, it would be over-dispersed and would not be a great fit. If we were to fit two separate normal distributions to the data, it would be a much better fit. So, for each $i = 1,\ldots,n$ time points,

$$x_i = \begin{cases} 0 & \text{with probability } \pi_0 \\ 1 & \text{with probability } \pi_1. \end{cases}$$

Now, suppose our observations of Old Faithful's eruptions, labeled y_i, were drawn from two separate normal distributions:

$$y_i = \begin{cases} Normal(\mu_0,\sigma_1^2) & \text{if } x_i = 0 \\ Normal(\mu_1,\sigma_2^2 & \text{if } x_i = 1. \end{cases}$$

We observe $y_1,\ldots,y_n$ but nothing else. So our 2x2 table would look as follows:

	Observed	**Unobserved**
Variable	$y_1, \ldots, y_n$	$x_1, \ldots, x_n$
Constant	NA	$\pi_0, \pi_1, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2$

So, our proper likelihood would be

$$\begin{aligned} L(\pi_0, \pi_1, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2) &= \mathbb{P}(y_1, \ldots, y_n | \pi_0, \pi_1, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2) \\ &= \int_{x_1} \ldots \int_{x_n} \mathbb{P}(y_1, \ldots, y_n, x_1, \ldots, x_n | \pi_0, \pi_1, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2) dx_n \ldots dx_1. \end{aligned}$$

Now, suppose we could not find the proper likelihood by integrating out the complete likelihood. We could instead proceed using the E-M algorithm.

First, define a vector of our unknowns $\alpha = [\pi_0, \pi_1, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2]$ for ease of notation. Then, the log of the proper likelihood is

$$\begin{aligned} logL(\alpha) &= log\mathbb{P}(y_1, \ldots, y_n, x_1, \ldots, x_n | \alpha) \\ &= log \prod_{i=1}^{n} \mathbb{P}(y_i, x_i | \alpha) \\ &= log \prod_{i=1}^{n} \mathbb{P}(x_i | \pi_0, \pi_1) \mathbb{P}(y_i, | x_i, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2). \end{aligned}$$

Now, recall that we can represent a mixture model using exponents. Utilizing that for our log-likelihood, we have

$$\begin{aligned} logL(\alpha) &= log \prod_{i=1}^{n} \mathbb{P}(x_i | \pi_0, \pi_1) \mathbb{P}(y_i, | x_i, \mu_0, \mu_1, \sigma_1^2, \sigma_2^2) \\ &= log \prod_{i=1}^{n} \left[\pi_0 \cdot Normal(y_i | \mu_0, \sigma_1^2)\right]^{x_i} \left[\pi_1 \cdot Normal(y_i | \mu_1, \sigma_2^2)\right]^{(1-x_i)}. \end{aligned}$$

So, if y_i comes from the second component (i.e., $x_i = 1$), then $[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)]^{x_i} = [\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)]^0 = 1$. So, the likelihood is $\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)$. And if we utilize this same idea when $x_i = 0$, then the likelihood would be $\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)$. Then, the log likelihood is

$$logL(\alpha) = log \prod_{i=1}^{n} \left[\pi_0 \cdot Normal(y_i | \mu_0, \sigma_1^2)\right]^{x_i} \left[\pi_1 \cdot Normal(y_i | \mu_1, \sigma_2^2)\right]^{(1-x_i)}$$

$$= \sum_{i=1}^{n} log\left[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right]^{x_i} + log\left[\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right]^{(1-x_i)}$$
$$= \sum_{i=1}^{n} x_i log\left[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right] + (1-x_i) log\left[\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right].$$

Now, in step 2, we need to compute $\mathbb{P}(x|y,\alpha)$. Then, we find the lower bound,

$$\mathbb{E}_{\mathbb{P}(x|y,\alpha)}[log\mathbb{P}(y,x|\alpha)].$$

In the third step, we iteratively compute

$$\mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}\left[\sum_{i=1}^{n} x_i log\left[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right] + (1-x_i) log\left[\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right]\right].$$

Then, the lower bound is going to simplify to

$$\sum_{i=1}^{n} \mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}\left[x_i log\left[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right]\right] + \mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}\left[(1-x_i) log\left[\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right]\right]$$
$$= \sum_{i=1}^{n} log\left(\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right)\mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}[x_i] + log\left(\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right)\mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}[1-x_i],$$

so all we really need to compute is $\mathbb{E}_{\mathbb{P}(x|y,\alpha^t)}[x_i]$.

We need to randomly guess α. Let us set $\pi_0 = \pi_1 = \frac{1}{2}$, $\mu_0 = \mu_1 = \frac{1}{n}\sum_i y_i$, and $\sigma_1^2 = \sigma_2^2 = \frac{1}{n}\sum_i (y_i - \frac{1}{n}\sum_i y_i)$, which are all reasonable starting points. Then, we compute $\mathbb{E}[x_i|y_1,\ldots,y_n,\pi_0^{(0)},\pi_1^{(0)},\mu_0^{(0)},\mu_1^{(0)},\sigma_1^{2(0)},\sigma_2^{2(0)}]$. We cannot compute the integral, so we instead find the best guess for the latent variable using the posterior distribution. To compute this though, we need to find

$$\begin{aligned}
\mathbb{P}(x_i|y_1,\ldots,y_n,\alpha^{(0)}) &= \frac{\mathbb{P}(x_i,y_1,\ldots,y_n|\alpha^{(0)})}{\mathbb{P}(y_1,\ldots,y_n|\alpha^{(0)})} \\
&= \frac{\mathbb{P}(y_1,\ldots,y_n|x_i,\alpha^{(0)})\mathbb{P}(x_i|\alpha^{(0)})}{\mathbb{P}(y_1,\ldots,y_n|x_i=0,\alpha^{(0)})\mathbb{P}(x_i=0|\alpha^{(0)}) + \mathbb{P}(y_1,\ldots,y_n|x_i=1,\alpha^{(0)})\mathbb{P}(x_i=1|\alpha^{(0)})} \\
&= \frac{\prod_{i=1}^{n}\mathbb{P}(y_i|x_i,\alpha^{(0)})\mathbb{P}(x_i|\alpha^{(0)})}{\prod_{i=1}^{n}\mathbb{P}(y_i|x_i=0,\alpha^{(0)})\mathbb{P}(x_i=0|\alpha^{(0)}) + \prod_{i=1}^{n}\mathbb{P}(y_i|x_i=1,\alpha^{(0)})\mathbb{P}(x_i=1|\alpha^{(0)})} \\
&= \frac{\prod_{i=1}^{n}\left[\pi_0 \cdot Normal(y_i|\mu_0^{(0)},\sigma_1^{2(0)})\right]^{x_i}\left[\pi_1 \cdot Normal(y_i|\mu_1^{(0)},\sigma_2^{2(0)})\right]^{(1-x_i)}}{\prod_{i=1}^{n}\mathbb{P}(y_i|x_i=0,\alpha^{(0)})\pi_0^{(0)} + \prod_{i=1}^{n}\mathbb{P}(y_i|x_i=1,\alpha^{(0)})\pi_1^{(0)}}
\end{aligned}$$

Then, at iteration t,

$$\mathbb{E}_{\mathbb{P}(x_i|y_1,\ldots,y_n,\alpha^{(t)})}[x_i] = \begin{cases} \frac{\prod_{i=1}^n \pi_0 \cdot Normal(y_i|\mu_0^{(0)},\sigma_1^{2(0)})}{\prod_{i=1}^n \mathbb{P}(y_i|x_i=0,\alpha^{(0)})\pi_0^{(0)} + \prod_{i=1}^n \mathbb{P}(y_i|x_i=1,\alpha^{(0)})\pi_1^{(0)}} & \text{if } x_i = 0 \\ \frac{\prod_{i=1}^n \pi_1 \cdot Normal(y_i|\mu_1^{(0)},\sigma_2^{2(0)})}{\prod_{i=1}^n \mathbb{P}(y_i|x_i=0,\alpha^{(0)})\pi_0^{(0)} + \prod_{i=1}^n \mathbb{P}(y_i|x_i=1,\alpha^{(0)})\pi_1^{(0)}} & \text{if } x_i = 1. \end{cases}$$

So, we already guessed $\alpha^{(0)}$. Then, in the E-step, we write down the lower bound. Using this, in the M-step, we find

$$\alpha^{(1)} = \arg\max_{\alpha} \sum_{i=1}^{n} x_i log\left[\pi_0 \cdot Normal(y_i|\mu_0, \sigma_1^2)\right] + (1 - x_i) log\left[\pi_1 \cdot Normal(y_i|\mu_1, \sigma_2^2)\right],$$

and we repeat the E-step and M-step until the difference between our estimates for unknown constants from one iteration to the next is very, very small.

5 Conclusion

There are some nuances in the E-M algorithm. For instance, there are two sets of unknown quantities: one is the "best guess" *entering* this iteration of the E-M algorithm and another is the set of unknown quantities that the log likelihood depends on. The "best guess" is used by coming up with the best guess for the latent random variables. On the other hand, the lower bound is composed of the latent random variables, which is now integrated out because our best guess for those in the expectation set, *and* the other set of constants, which are not specified. We maximize the lower bound with respect to the second set of unknowns to get the next set "best guesses."

This algorithm is quite powerful as it allows us to estimate unknown constants in settings where we cannot find the proper likelihood, which happens quite often. To overcome these challenges, we

- Guess
- Compute the expectation of the lower bound given the guesses to estimate the latent random variables
- Maximize the likelihoods using the latent random variable estimates to find the next set of best guesses.

We simply repeat steps 2 and 3 until the difference between the best guesses between consecutive iterations is quite small.