LATENT VARIABLE MODELS, PLSM, AND THE EXPECTATION MAXIMIZATION ALGORITHM

GIDEON DRESDNER, HADI DANESHMAND

Abstract. We present the probabilistic perspective on pLSMs because this helps with deriving and getting intuition for the EM algorithm.

1. probabilistic Latent Semantic Models (pLSM)

1.1. **Dataset.** We are given a dataset consisting of some documents $\{d_i\}_{i=1}^N$. We decide to represent each document as a bag of words, $d_i = \{w_j\}$ thereby ignoring the contextual structure between words. Our goal is to learn so-called "topics:" discrete latent variables z which represent the type of document you have. We want to learn the fewest number of topics that best explain the data which means that documents and words will often be a mix of different topics.

1.2. **Model definition.** We begin with the join distribution of the variables.

$$p(w_j, d_i) = \sum_k p(w_j, d_i, z = k) = \sum_k p(w_j, d_i \mid z = k)p(z = k) \tag{1.1}$$

$$\stackrel{*}{=} \sum_k p(w_j \mid z = k)p(d_i \mid z = k)p(z = k) \tag{1.2}$$

The last equality marked $(*)$ follows from a conditional independence assumption: if you know that the topic for a given document is $z = k$, then the document no longer correlates at all with the word frequency w_j. **Exercise:** convince yourself that this assumption is reasonable using an example.

Equation 1.1 is not feasible for learning. The problem is that estimating $p(z = k)$ means making a prior assumption about the distribution of topics and considering that we do not even want to assume what the topics are, this is will be a difficult assumption to make. To avoid this conundrum note that

$$p(w_j \mid z = k)p(z = k) = p(z = k \mid d_i)p(d_i) \tag{1.3}$$

If we substitute this in to Equation 1.1 and then divide by $p(d_i)$, we no longer need to incorporate prior assumptions. Instead we model the conditional distribution

$$p(w_j \mid d_i) = \sum_k p(w_j \mid z = k)p(z = k \mid d_i) \tag{1.4}$$

Intuitively this says that we know the probability of topics given the document, and given the topic, we can define the probability of the words.

Date: March 21-22, 2019.

1.3. **Log-likelihood.** Our goal is to learn topics. We accomplish this by maximizing the log-likelihood of the model we have defined:

$$\sum_{ij} X_{ij} \ln \sum_{k} p(w_j \mid z = k) p(z = k \mid d_i) \tag{1.5}$$

Note, we have added in the X_{ij} term which represents the number of times word j appeared in document i. **Exercise:** understand where this X_{ij} term comes from. (Hint: $c \ln x = \ln x^c$).

1.4. **Challenge in optimizing Equation 1.5.** It is clear that the objective is not concave because of the product terms just as in matrix factorization but it worse than just non-concave.

- Consider the parameters of this model as matrices U $(V \times K)$ and V $(N \times K)$ for the $p(w_j \mid z = k)$ and $p(z = k \mid d_i)$ terms respectively. **Exercise:** You should know the constraints on these matrices. Consider a permutation p of the set $\{1, \ldots, K\}$ and use it to permute the columns of U and V. This does not change the likelihood. To summarize, any given solution to the maximum likelihood problem (including a global optimum), has $K! - 1$ equally good solutions which can be obtained by permuting the columns of the parameter matrices. (For a more in-depth explanation of non-concavity see the solutions to Exercise 5, Problem 3, part (i). Note that the counter example discussed in this part is indeed a permutation).
- Computing gradients scales in k. This hyper-parameter is our control over the complexity of the model. For a larger text corpus, we will want to increase k to allow for more topics to be discovered thus slowing down inference.

If we could somehow flip the ln with the sum, then we would have a concave objective. [1] We will find a way of doing this flip but it will come at a cost of an inequality. We will show that this inequality actually has some good properties. Namely (1) it breaks the symmetry discussed above, (2) it admits closed form updates which are fast, and (3) it is guaranteed to improve the original objective on each iteration.

2. Expectation Maximization (EM)

2.1. **Formulation.** Suppose we have terms q_{ijk} for which $\sum_k q_{ijk} = 1$ and $q_{ijk} \geq 0$. Then if, we incorporate these terms by multiplying by one, we suddenly have a convex combination and we can apply the concavity inequality [2]

$$\ln \sum_k q_{ijk} \frac{p(w_j \mid z = k) p(z = k \mid d_i)}{q_{ijk}} \geq \sum_k q_{ijk} \ln \frac{p(w_j \mid z = k) p(z = k \mid d_i)}{q_{ijk}} \tag{2.1}$$

[1] Given the i.i.d. assumption, the likelihood has many products in it: $p(\mathcal{D}; \theta) \stackrel{iid}{=} \prod_{n=1}^{N} p(x_n; \theta)$. As we have experienced with matrix factorization, products are generally not concave, and so to get a concave likelihood objective to optimize, we often consider the *log-likelihood*. We would like to emphasize that removing the log would thus multiply our problems, not reduce them.

[2] The opposite of the convexity inequality, $f(\lambda x + (1 - \lambda) y) \geq \lambda f(x) + (1 - \lambda) f(y)$ for $\lambda \in [0, 1]$ if f is concave. This is often referred to as Jensen's Inequality.

Note that the ln is now inside the sum term. Note that the objective is concave if you hold the q_{ijk}'s fixed.

$$\sum_{i,j} X_{ij} \sum_{k} q_{ijk} \ln \frac{p(w_j \mid z=k)p(z=k \mid d_i)}{q_{ijk}} \tag{2.2}$$

2.2. **Expectation step.** Note that the constraints on q_{ijk} imply that it is a distribution over the topics z. The lower-bound in Equation 2.1 holds for any such distribution. Observe that q_{ijk} is arbitrary and we introduced it simply to take advantage of the concavity of log. Our goal should be to find the best q_{ijk}'s, i.e. the ones that maximize the lowerbound. Furthermore, if we can match the left hand side of the inequality in Equation 2.1, we also know that we have achieved the optimal value for q_{ijk}.

Out of all the possible distributions on z, suppose we let $q_{ijk} = p(z=k \mid w_j, d_i)$. This is the probability of topic k given the observation (w_j, d_i). Then,[3]

$$\sum_{i,j} \sum_{k} p(z=k \mid w_j, d_i) \ln \frac{p(w_j \mid z=k)p(z=k \mid d_i)}{p(z=k \mid w_j, d_i)} \tag{2.3}$$

$$= \sum_{i,j} \sum_{k} p(z=k \mid w_j, d_i) \ln p(w_j \mid d_i) \tag{2.4}$$

$$= \sum_{i,j} \ln p(w_j \mid d_i) \sum_{k} p(z=k \mid w_j, d_i) \tag{2.5}$$

$$= \sum_{i,j} \ln p(w_j \mid d_i) \tag{2.6}$$

The last line brings us directly back to Equation 1.4.

In summary, we have introduced a surrogate objective which added additional q_{ijk} parameters. We solved in closed-form for the optimal q_{ijk}'s while holding the parameters of our model fixed. Although this is actually a maximization step, it is referred to as the expectation step of the algorithm since Equation 2.2 contains this $\sum_k q_{ijk}(\dots)$ term which is an expectation.

Finally, note that Equation 2.3 shows that we match the upper bound. Thus, when we turn to optimizing the parameters with fixed q_{ijk}, we are guaranteed to find parameters that achieve a better likelihood.

2.3. **Maximization step.** Consider the compact notation $\mathbf{u}_{jk} := p(w_j \mid z=k)$ and $\mathbf{v}_{kd} := p(z=k \mid d_i)$. According to the definition, we know that $\sum_j \mathbf{u}_{jk} = 1$ and $\sum_k v_{ki} = 1$ hold. Let $\mathbf{U}$ and $/\mathbf{V}$ be a corresponding matrices. Using this notation, we rewrite the established lowerbound in Eq. (2.2) as

$$g(\mathcal{Q}, \mathbf{U}, \mathbf{V}) = \sum_{i,j} \sum_{k} q_{ijk} \left(\ln(u_{kj}) + \ln(v_{ki}) - \ln(q_{ijd}) \right). \tag{2.7}$$

Recall the above function lowerbounds the log-likelihood function, namely the following objective function:

$$f(\mathbf{U}, \mathbf{V}) = \sum_{ij} X_{ij} \ln \left(\sum_{k} u_{jk} v_{ki} \right) \tag{2.8}$$

[3] X_{ij} terms omitted for clarity.

In the maximization step, EM optimizes g in $\mathbf{U}$ and $\mathbf{V}$ while keeping variational parameter $\mathcal{Q}$ fixed. It is easy to show that g is concave in $\mathbf{U}$ and $\mathbf{V}$ jointly (**Exercise:** Prove that g is concave in $(\mathbf{U}, \mathbf{V})$). To this end, one has to solve the following constrained concave program:

$$\max_{\mathbf{U},\mathbf{V}} g(\mathcal{Q}, \mathbf{U}, \mathbf{V}), \quad \text{Subject to} \sum_j \mathbf{u}_{jk} = 1 \text{ and} \sum_k v_{ki} = 1. \tag{2.9}$$

(**Exercise:** Why are the constraints $v_{ki} > 0$ and $u_{jk} > 0$ omitted in the program above). The method of Lagrangian multipliers allows us to turn the above constraint program to an unconstrained. This method introduces Lagrangian multipliers associated with each constraint. In the above program, we introduce Lagrangian multiplier α_k for each constraint $\sum_j \mathbf{u}_{jk} = 1$ and β_i for each constraint $\sum_k v_{ki} = 1$. Using these multipliers, we define Lagrangian function as

$$\begin{aligned} L(\mathbf{U}, \mathbf{V}, \alpha, \beta) = &-g(\mathcal{Q}, \mathbf{U}, \mathbf{V}) \\ &+ \sum_k \alpha_k \left(\sum_j u_{jk} - 1 \right) + \sum_i \beta_i \left(\sum_k v_{ki} - 1 \right). \end{aligned} \tag{2.10}$$

The solution of program 2.9 can be obtained by solving

$$\max_{\alpha,\beta} \min_{\mathbf{U},\mathbf{V}} L(\mathbf{U}, \mathbf{V}, \alpha, \beta).$$

(**Exercise:** why solving the above problem recovers the solution of program 2.9?). Setting gradient to zero yields the solution of the above problem (see the solution of problem 3.v of the exercise sheet 5 for more details).

2.4. **EM: an alternative maximization technique.** EM optimizes g through the following recurrence:

$$\text{Expectation step} \qquad \mathcal{Q}_{n+1} = \arg\max_{\mathcal{Q}} g(\mathcal{Q}, \mathbf{U}_n, \mathbf{V}_n) \tag{2.11}$$

$$\text{Maximization step} \qquad \mathbf{U}_{n+1}, \mathbf{V}_{n+1} = \arg\max_{\mathbf{U},\mathbf{V}} g(\mathcal{Q}_{n+1}, \mathbf{U}, \mathbf{V}) \tag{2.12}$$

(**Exercise:** Why the result of section 2.2 concludes the maximization in $\mathcal{Q}$?). The question is how this alternating maximization on g relates to maximization of our target objective f(in Eq. (2.8)). Recall, we have shown in section 2.2 that g touches f after the expectation step, namely

$$f(\mathbf{U}_n, \mathbf{V}_n) = g(\mathcal{Q}_{n+1}, \mathbf{U}_n, \mathbf{V}_n) \tag{2.13}$$

holds. From the other hand, we know that the maximization step increases g, i.e.

$$g(\mathcal{Q}_{n+1}, \mathbf{U}_n, \mathbf{V}_n) \leq g(\mathcal{Q}_{n+1}, \mathbf{U}_{n+1}, \mathbf{V}_{n+1}) \tag{2.14}$$

Putting all together, we have

$$f(\mathbf{U}_n, \mathbf{V}_n) = g(\mathcal{Q}_{n+1}, \mathbf{U}_n, \mathbf{V}_n) \leq g(\mathcal{Q}_{n+1}, \mathbf{U}_{n+1}, \mathbf{V}_{n+1}) \leq g(\mathcal{Q}_{n+2}, \mathbf{U}_{n+1}, \mathbf{V}_{n+1}) = f(\mathbf{U}_{n+1}, \mathbf{V}_{n+1}).$$

The above inequality shows that EM optimises f, hence $f(\mathbf{U}_n, \mathbf{V}_n) \leq f(\mathbf{U}_{n+1}, \mathbf{V}_{n+1})$. Since EM can track solutions of E and M steps in closed-forms, it is often significantly faster than gradient descent method.