A Note on EM Algorithm for Probabilistic Latent Semantic Analysis

Qiaozhu Mei, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

1 Introduction

In many text collections, we encounter the scenario that a document contains multiple topics. Extracting such topics/subtopics/themes from the text collection is important for many text mining tasks, such as search result organization, subtopic retrieval, passage segmentation, document clustering, and contextual text mining.

A well accepted practice is to explain the generation of each document with a probabilistic topic model. In such a model, every topic is represented by a multinomial distribution on the vocabulary (i.e., a unigram language model). Correspondingly, such a probabilistic topic model is usually chosen to be a mixture model of k components, each of which is a topic.

One of the standard probabilistic topic models is the Probabilistic Latent Semantic Analysis (PLSA), which is also known as Probabilistic Latent Semantic Indexing (PLSI) when used in information retrieval [3].

The basic idea of PLSA is to treat the words in each document as observations from a mixture model where the component models are the topic word distributions. The selection of different components are controlled by a set of mixing weights. Words in the same document share the same mixing weights. We may add a component of background and use it to explain the non-topical words (functional words) with a background word distribution.

Specifically, let $\theta_1, ..., \theta_k$ be k topic unigram language models (i.e., word distributions) and θ_B be a background model for the whole collection C. A word w in a document d is regarded as a sample of the following mixture model (based on word generation).

$$p_d(w) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} (\pi_{d,j} p(w|\theta_j)) \tag{1}$$

where w is a word in document d, $\pi_{d,j}$ is the mixing weight for document d for choosing the j-th theme θ_j such that $\sum_{j=1}^{k} \pi_{d,j} = 1$, and λ_B is the mixing weight for θ_B. The purpose of using a background model θ_B is to make the topic models more discriminative; since θ_B gives high probabilities to non-discriminative and non-informative words, we expect such words to be accounted for by θ_B and thus the topic models to be more discriminative. θ_B is estimated using the whole collection C as $p(w|\theta_B) = \frac{\sum_{d \in C} c(w,d)}{\sum_{w \in V} \sum_{d \in C} c(w,d)}$. $p(w|\theta_B)$ doesn't change during the later estimation procedure.

The additional parameters to estimate are $\Lambda = \{\theta_j, \pi_{d,j} | d \in C, 1 \leq j \leq k\}$. The log-likelihood of C is

$$\log p(C|\Lambda) = \sum_{d \in C} \sum_{w \in V} [c(w,d) \times \log(\lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} (\pi_{d,j} p(w|\theta_j)))] \tag{2}$$

where $c(w,d)$ is the count of word w in document d.

2 Expectation-Maximization (EM) Algorithm

The model can be estimated using the Expectation Maximization (EM) algorithm [2] to obtain the topic word distributions and the mixing weights.

The Expectation-Maximization (EM) algorithm is a general algorithm for maximum-likelihood estimation where the data are incomplete or the likelihood function involves latent variables. Note that the notion of incomplete data and latent variables are related: when we have a latent variable, we may regard our data as being incomplete since we do not observe values of the latent variables; similarly, when our data are incomplete, we often can also associate some latent variable with the missing data. For language modeling, the EM algorithm is often used to estimate parameters of a mixture model, in which the exact component model from which a data point is generated is hidden from us. Informally, the EM algorithm starts with randomly assigning values to all the parameters to be estimated. It then iteratively alternates between two steps, called the expectation step (i.e., the E-step) and the maximization step (i.e., the M-step), respectively. In the E-step, it computes the expected likelihood for the complete data (the so-called Q-function) where the expectation is taken w.r.t. the computed conditional distribution of the latent variables (i.e., the hidden variables) given the current settings of parameters and our observed (incomplete) data. In the M-step, it re-estimates all the parameters by maximizing the Q-function. Once we have a new generation of parameter values, we can repeat the E-step and another M-step. This process continues until the likelihood converges, i.e., reaching a local maxima. Intuitively, what EM does is to iteratively augment the data by guessing the values of the hidden variables and to re-estimate the parameters by assuming that the guessed values are the true values. For detailed discussion about EM algorithm, please refer to the note [7].

3 Maximum Likelihood Estimation (MLE)

In this section, we give an intuitive explanation of the EM algorithm for PLSA, which would give you a basic understanding of the model. The basic idea is to estimate the parameters Λ in Section 1 by fitting the data with the model. The straight forward way is to find Λ which maximizes the data likelihood in Equation 2. This means,

$$\begin{aligned}\hat{\Lambda} &= \arg\max_{\Lambda} \log p(C|\Lambda) \\ &= \arg\max_{\Lambda} \sum_{d\in C}\sum_{w\in V}[c(w,d)\times \log(\lambda_B p(w|\theta_B)+(1-\lambda_B)\sum_{j=1}^{k}(\pi_{d,j}p(w|\theta_j)))] \end{aligned} \quad (3)$$

According to the EM algorithm, we can use the following iterative updating formulas to estimate all the parameters. Intuitively, if we know the identity of each word in the collection, i.e., which topic (or background) it is generated with, it is quite easy to estimate Λ. We thus introduce a hidden variable for the identity of each word, $\{z_{d,w}\}$. $\{z_{d,w}\}$ is a hidden variable and $p(z_{d,w}=B)$ is the probability that the word w in document d is generated with the background distribution. $p(z_{d,w}=j)$ in the following formulae indicates that the word w in document d is generated using topic j *given that* w *is not generated from the background model*. Therefore, it is actually better to write $p(z_{d,w}=j)$ in the following formulae as $p(z_{d,w}=j|z_{d,w}\neq B)$. To be consistent with [8] and [5], we still use $p(z_{d,w}=j)$ in the following formulae.

E-Step:

$$p(z_{d,w}=j) = \frac{\pi_{d,j}^{(n)}p^{(n)}(w|\theta_j)}{\sum_{j'=1}^{k}\pi_{d,j'}^{(n)}p^{(n)}(w|\theta_{j'})} \quad (4)$$

$$p(z_{d,w}=B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B)+(1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}^{(n)}p^{(n)}(w|\theta_j)} \quad (5)$$

M-Step:

$$\pi_{d,j}^{(n+1)} = \frac{\sum_{w\in V}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{j'=1}^{k}\sum_{w\in V}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j')} \quad (6)$$

$$p^{(n+1)}(w|\theta_j) = \frac{\sum_{d\in C_i}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{w'\in V}\sum_{d\in C_i}c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)} \quad (7)$$

In E-Step, we are actually estimating the distribution of the hidden variables (or estimating the identity of each word). Please note that this estimated identity is "soft". A word could be splitted into several fractions, each fraction

is generated from one topic, or background. This distribution is simple to compute: just figure out in the likelihood of a word, how much proportion is contributed by the background model, or by Topic j if not contributed by the background.

M-step is essentially aggregating such fractions to estimate a new set of values for Λ. To estimate the new $\pi_{d,j}$ (the mixing weights for a document), we just aggregate all the fractions of words generated by Topic j in document d, and normalize $\{\pi_{d,j}\}_{j=1..k}$ to make $\sum_{j=1...k} \pi_{d,j} = 1$ (Equation 6). To estimate the new $p(w|\theta_j)$, think about how we estimate a language model for a document, where you know the count of every word. Now that we know the identity of every word in the collection, all word fractions generated by Topic j is now a "pseudo-document" for Topic j. Equation 7 is essentially aggregating all the fractions of the word w that are generated by Topic j (over all documents), and normalize $\{p(w|\theta_j)\}_{w\in V}$ to make $\sum_{w\in V} p(w|\theta_j) = 1$.

Before convergence, every iteration of the EM algorithm will yield a larger likelihood value in Equation 2. The algorithm will terminate when it achieves a local maximum of the log likelihood. In our experiments, we use multiple trials to improve the local maximum we obtain, where in each trial we begin with a new starting point of Λ (by assigning random values to Λ). We may use $\hat{j} = \arg\max_j \pi_{d,j}$ to assign a document into the $\hat{j}$th topic, and use $\sum_{d\in C} \pi_{d,j} p(d) = \frac{1}{N} \sum_{d\in C} \pi_{d,j}$ to compute the coverage of Topic j in collection C which contains N documents.

Please refer to [7] if you want to deduct the formulae directly from Equation 3.

4 Maximum A Posterior (MAP) Estimation

Maximum Likelihood Estimation is a reasonable choice if we don't have any prior knowledge about the topic models. But in many scenarios we do. Indeed, given a topic, a user often has some knowledge about what aspects are interesting. For example, when the user is searching for laptops, we know that he is very likely interested in "price" and "configuration". It will be nice if we "guide" the model to enforce two of the topic models to be as close as possible to the predefined facets. We may want to see "retrieval model" among the topics of information retrieval. Rather than directly fitting the data with PLSA model, we use such domain knowledge to define a prior on the topic models and estimate the topic models using the Maximum A Posterior (MAP) estimator. Since the output of topics are language models, it is natural to also input prior knowledge as language models. Specifically, we may want to input $\bar{\theta}_j$ as the prior topic model for Topic j given by the user.

In Bayesian analysis, the prior[1] of a distribution is a *distribution of distribution*, which indicates your belief of the parameters of the target distribution. Since our parameters are Λ, we can denote our prior distribution as $p(\Lambda)$. Conjugate prior is usually adopted as the prior distribution of the target distribution. The basic idea of a conjugate prior is that the prior distribution and the posterior distribution are of the same form. The conjugate prior for multinomial distribution is the Dirichlet distribution [2].

We define the following conjugate Dirichlet prior for the topic model θ_j: $Dir(\{1 + \mu p(w|\bar{\theta}_j)\}_{w\in V})$, where the parameters μ indicate how strong our confidence is on the sentiment model prior.

Since the prior is conjugate, μ can be interpreted as "equivalent sample size", which means that the impact of adding the prior would be equivalent to adding $\mu p(w|\bar{\theta}_j)$ pseudo counts for word w when estimating the topic model $p(w|\theta_j)$. The larger μ is, the closer the estimated topic models would be to the prior language model given by the user.

Therefore, in general, we may assume that the prior on all the parameters in the PLSA model is

$$p(\Lambda) \quad \propto \quad \prod_{j=1}^{k} p(\theta_j) = \prod_{j=1}^{k} \prod_{w\in V} p(w|\theta_j)^{\mu p(w|\bar{\theta}_j)} \tag{8}$$

where $\mu = 0$ if we do not have prior knowledge on θ_j.

With the prior defined above, we may use Bayesian estimation to maximize the posterior probability of parameters, instead of maximizing the likelihood function as in Equation 3. We may use the MAP estimator instead:

$$\hat{\Lambda} = \arg\max_{\Lambda} p(\mathcal{C}|\Lambda) p(\Lambda) \tag{9}$$

The MAP estimation can be conducted by rewriting the M-step in the original EM algorithm in Section 3 to incorporate the pseudo counts given by the prior [4]. Please note that we do not introduce prior for $\pi_{d,j}$. The only thing we need to change in the EM algorithm is thus Equation 7. The new M-step updating formula is now:

[1] http://en.wikipedia.org/wiki/Prior_probability

[2] http://en.wikipedia.org/wiki/Dirichlet_distribution

$$p^{(n+1)}(w|\theta_j) = \frac{\sum_{d\in C_i} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)+\mu p(w|\bar{\theta}_j)}{\sum_{w'\in V}\sum_{d\in C_i} c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)+\mu} \quad (10)$$

Understanding Equation 10 isn't very difficult. Comparing with Equation 7, the difference is that besides the documents in the collection, we also "observe" a pseudo-document for each topic. The size of such a pseudo-document is μ, and the distribution of words in the pseudo-documents follows $p(w|\bar{\theta}_j)$.

The parameters μ is usually empirically set to constants, or be estimated with a regularized estimation [6], which begins with very large μ, and then gradually decay the μ in each iteration until they are equal to the real amount of data generated with the corresponding topic/sentiment models. A detailed discussion can be found in [6].

Please note that we didn't incorporate priors for $\pi_{d,w}$. Can we do this? It is good practice to think about this possibility and then read the paper [1], which leads to a widely used extension of PLSA, LDA.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38, 1977.

[3] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR '99*, pages 50–57, 1999.

[4] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.

[5] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceeding of KDD'05*, pages 198–207, 2005.

[6] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of SIGIR'06*, pages 162–169, 2006.

[7] C. Zhai. A note on the expectation-maximization (em) algorithm. In *Course note of CS410*.

[8] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD '04*, pages 743–748, 2004.