Note for pLSA and LDA-Version 1.1

Wayne Xin Zhao

March 2, 2011

1 Disclaimer

In this part of PLSA, I refer to [4, 5, 1]. In LDA part, I refer to [3, 2]. Due to the limit of my English ability, in some place, I just copy the words from original papers or notes. I'm sorry I don't have time to get the approvement from these authors. I would remove or rewrite all such parts later. This note is strictly for studying instead of any other activies. Please just keep by yourself and don't distribute it. This note is done in two days, I didn't have time for reviewing but straightforward releasing it for possible materials of studying topic models. I would keep updating this note and try to make the notations consistent and correct all the errors. If you find any problem in it, feel free to contact me via batmanfly@gmail.com.

2 pLSA

2.1 Introduction

pLSA is one kind of mixture model. We assume that there are a collection of D documents, denoted by $\mathcal{C}$, and there are totally K latent topics, which are represented by a multinomial distribution over vocabulary and denoted by variable $z_i \in \{z_1, .., z_K\}$. In this model, all the words $\mathbf{w}$ are assumed observed variables while $\{z_i\}_{i=1}^{K}$ are unobserved.

Let us introduce the following probabilities:

- $p(d)$ is used to denote the probability that a word occurrence will be observed in a particular document d,
- $p(w|z)$ denotes the class-conditional probability of a specific word conditioned on the unobserved variable z,
- and finally $p(z|d)$ denotes a document specific probability distribution over the latent variable space.

Using these definitions, one may define a generative model for word/document co-occurrences by the following scheme:

- select a document d with probability $p(d)$,
- pick a latent class z with probability $p(z|d)$,
- generate a word w with probability $p(w|z)$.

We model the joint probability of one observation pair (d, w) as follows: $p(d, w) = p(d)(w|d)$ and $p(w|d) = \sum_{k=1}^{K} p(w|z_k)p(z_k|d)$.

For topic models, topics refer to the K multinomial word distribution, i.e. $\{p(\cdot|z)\}_z$. Additionally, another important kind of parameters is document-topic distribution, i.e. $\{p(\cdot|d)\}_d$. So the key problem for application of pLSA is the estimation of these two groups of parameters.

2.2 parameter Estimation

As usual, one may use a maximum likelihood formulation of the learning problem, i.e., one has to maximize

$$\begin{aligned} \mathcal{L} &= \sum_d \sum_w n(d, w) \log p(d, w) \\ &\propto \sum_d \sum_w n(d, w) \log \left[\sum_{k=1}^{K} p(w|z_k)p(z_k|d) \right]. \end{aligned} \tag{1}$$

However, due to the log-sum, it is infeasible to solve this optimization problem. Here we describle how to use EM algorithm for parameter estimation.

E-step

For the E-step one simply applies Bayes formula, we estimate the posterior probability of latent variable based on current parameters as follows:

$$\begin{aligned} p(z|w, d) &= \frac{p(z, w, d)}{\sum_z p(z, w, d)}, \\ &= \frac{p(w|d, z)p(z|d)p(d)}{\sum_z \left(p(w|d, z)p(z|d)p(d) \right)} \\ &= \frac{p(w|z)p(z|d)}{\sum_z p(w|z)p(z|d)} \end{aligned} \tag{2}$$

In this step, we assumet that all the parameters are known.

M-step

Because the optimazation of Equation 1 is intractable, we study to maximize the expected complete data log-likelihood $\mathbf{E}(\mathcal{L})$. One complete data instance refers to one triple (d, w, z), however, z is not observed in the dataset, we seek to study the expectation of complete data log-likelihood

$$\mathbb{E}(\mathcal{L}) \quad = \quad \sum_d \sum_w n(d,w) \sum_z p(z|w,d) \log \left[p(w|z_k) p(z_k|d) \right] \tag{3}$$

Here we assume all $p(z|w,d)$ are estimated in the previous E step. To find out the optimal solutions of Equation 3, we use Lagrange multiplier method[1], which provides a strategy for finding the maxima and minima of a function subject to constraints. For Lagrange multiplier method, the optimization problem should include two parts: objective function and contraints. For objective function, it's the expected complete data log-likelihood in Equation 3. For contraints, the parameters should satisfy the following normalization:

- $\forall d, \sum_z p(z|d) = 1$.
- $\forall z, \sum_w p(w|z) = 1$.

Accoring to Lagrange multiplier method, we put objective function and contraints together as follows:

$$\mathcal{H} = \mathbb{E}(\mathcal{L}) + \sum_d \lambda_d \sum_z (1 - p(z|d)) + \sum_z \delta_z \sum_w (1 - p(w|z)). \tag{4}$$

Maximization of $\mathcal{H}$ with respect to parameters(i.e. $p(z|d)$ and $p(w|z)$) leads to the following set of stationary equations

$$\sum_w n(d,w) p(z|w,d) - \lambda_d p(z|d) = 0, \forall d, z, \tag{5}$$

$$\sum_d n(d,w) p(z|w,d) - \delta_z p(w|z) = 0, \forall z, w. \tag{6}$$

After eliminating the Lagrange multipliers[2], we obtain the M-step re-estimation equations

$$p(w|z) = \frac{\sum_d n(d,w) p(z|w,d)}{\sum_{w'} \sum_d n(d,w') p(z|w',d)}, \tag{7}$$

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

[2] Sum the left side of Equation 5 by z and sum the left side of Equation 6 by w.

$$p(z|d) = \frac{\sum_w n(d,w)p(z|w,d)}{n_d}. \tag{8}$$

Sometimes, it's useful to incorporate one background model $p(w|\mathcal{C})$ in Equation 1. It's very easy to modify EM algorithm for that.

"Maximum Likelihood Estimation is a reasonable choice if we dont 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 inter- esting. 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.

In Bayesian analysis, the prior of a distribution is a distribution of distribution, which indicates your belief of the parameters of the target distribution. Since our parameters are Θ ($\{p(\cdot|z)\}_z$ and $\{p(\cdot|d)\}_d$), we can denote our prior distribution as $p(\Theta)$. 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[3]. "[4]

For our problem setting, we may just consider putting Dirichlet prior on $p(w|z)$. The formula of Dirichlet distribution is very similar to multinomial distribution except that they have different coefficients. Since we consider studying log-likelihood, the coefficients could be discarded.

We define the following conjugate Dirichlet prior for the topic model $\theta_z (= p(\cdot|z))$ as Dir(1 + $\mu p(w|\phi)_w$), where the parameters μ indicate how strong our confidence is on the topic model prior and $\sum_w p(w|\phi)$.

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

$$p(\Theta) \propto \prod_z p(\theta_z) = \prod_z \prod_w p(w|\theta_z)^{\mu p(w|\phi_z)} \tag{9}$$

With the prior defined above, we may use Bayesian estimation to maximize the posterior probability of parameters, instead of maximizing the likelihood function

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

[4] Copied from Qiaozhu Mei's note.

as in Equation1. We may use the MAp estimator instead:

$$\hat{\Theta} = \mathbf{arg}\max_{\Theta} p(\mathcal{C}|\Theta)p(\Theta). \tag{10}$$

The MAp estimation can be conducted by rewriting the M-step in the original EM algorithm in Equation 7 to incorporate the pseudo counts given by the prior, i.e. adding $\mu p(w|\phi)$ to $p(w|z), \forall z$. please note that we do not introduce prior for $p(z|d)$. The only thing we need to change in the EM algorithm is thus Equation 7. The new M-step updating formula is now:

$$p(w|z) = \frac{\sum_d n(d,w)p(z|w,d) + \mu p(w|\phi_z)}{\sum_{w'}\sum_d n(d,w')p(z|w',d) + \mu}. \tag{11}$$

Here let's see how to understand pseudo counts. Generally, we can estimate $p(w|z) = \frac{\#(w,z)}{\#(\cdot,z)}$. In the maximum likelihood estimatin

$$p(w|z) = \frac{\sum_d n(d,w)p(z|w,d)}{\sum_{w'}\sum_d n(d,w')p(z|w',d)}. \tag{12}$$

, we can assume that $\#(w,z) = \sum_d n(d,w)p(z|w,d)$ and $\#(\cdot,z) = \sum_{w'}\sum_d n(d,w')p(z|w',d)$ if $p(z|w,d)$ is fixed.

Compared Equation 12 with Equation 11, there seemed to be more $\mu p(w|\phi_z)$ counts of words to be added to $\#(w,z)$ and μ to be added to $\#(\cdot,z)$.

2.3 Implementation

To implement a pLSA tool on a median size(10,000) of corpus is very straightforward. We first malloc two arrays: $V \times K$ for $p(w|z)$ and $K \times D$ for $p(z|d)$. To make memory avaiable for our algorithm, we don't implement E-step explicitly. Instead, we go to M-step first. For Equation 7, we have two loops, namely by w and z; for Equation 8, we just use one loop by d. Here we may use index or map(hashtable) to enumerate all terms which appear in d instead of going through the whole vocabulary. For each query of $p(z|d,w)$, we then come to Equation 7 to compute it. In a general setting of desktop, for 10,000 docs and 100 topics, it may take such implementation 1h to iterate 10 times. There should be more efficient algorithms to implement pLSA, this is just the simplest one.

2.4 Understanding EM

EM is often used in incomplete data. In text mining, we design models with hidden(latent) variables, these variables are not seen in the dataset. In such setting, we may consider using EM algorithm to solve the optimization problem.

Supposing that all the observed variables denoted by $\mathbf{X}$ and all the hidden variables denoted by $\mathbf{Z}$, the joint distribution of all the variables are governed by a set of parameters denoted by Θ. Our goal is to maximize the likelihood function that is given by

$$p(\mathbf{X}|\Theta) = \sum_{\mathbf{z}} p(\mathbf{X}, \mathbf{Z}|\Theta). \tag{13}$$

Further we assume suppose the direct optimization of $p(\mathbf{X}|\Theta)$ but that optimizatio of complete-data likelihood function $p(\mathbf{X}, \mathbf{Z}|\Theta)$ is significantly easier. We further introduct a distribution $q(\mathbf{Z})$ defined over latent variables. No matter how we choose $q(\mathbf{Z})$, we can have the following decomposition

$$\log p(\mathbf{X}|\Theta) = \mathcal{L}(q, \Theta) + KL(q||p), \tag{14}$$

where we have defined

$$\mathcal{L}(q, \Theta) = \sum_{\mathbf{z}} q(\mathbf{Z}) \log \frac{p(\mathbf{X}, \mathbf{Z}|\Theta)}{q(\mathbf{Z})}. \tag{15}$$

$$KL(q||p) = -\sum_{\mathbf{z}} q(\mathbf{Z}) \log \frac{p(\mathbf{Z}|\mathbf{X}, \Theta)}{q(\mathbf{Z})}. \tag{16}$$

After these prelimenaries, we start to study why EM works, i.e. why $\log p(\mathbf{X}|\Theta)$ increase. Noting that $\log p(\mathbf{X}|\Theta) \geqslant \mathcal{L}(q, \Theta)$ due to the non-negativity of KL function, . $\mathcal{L}(q, \theta)$ equals to the expected complete-data likelihood plus the entropy of variable $\mathbf{Z}$. It's one lower bound of $\log p(\mathbf{X}|\Theta)$.

In EM algorithm, our objective function is expectation of complete-data likelihood, i.e. $\mathbb{E}_{q(\mathbf{Z})}(\log p(\mathbf{X}, \mathbf{Z}|\Theta))$. In the E step, we try to find out an estimate of $q(\mathbf{Z})$ while in M step, we try to maximize $\mathbb{E}_{q(\mathbf{Z})}(\log p(\mathbf{X}, \mathbf{Z}|\Theta))$ to find out the local(global) optimal parameters.

E-step

In E-step, we assume that Θ is fixed, so $\log p(\mathbf{X}|\Theta)$ should be a constant, i.e. $\log p(\mathbf{X}|\Theta) = C$. Then we estimate $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \Theta)$. Let's see what would happen after this operation: $KL(q||p)$ goes down to zero. Since Θ is fixed, $\log p(\mathbf{X}|\Theta)$ is fixed. So this operation leads to the increase of $\mathcal{L}(q, \Theta)$, in other words, the lower bound of $\log p(\mathbf{X}|\Theta)$ increases to C. Actually, though this step named as "Expectation", it did the maximization of $\mathcal{L}(q, \Theta)$.

M-step

In M-step, we maximize $\mathcal{L}(q, \Theta)$. Noting that in our previous EM algorithm for pLSA, $\mathcal{L}'(q, \Theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}|\Theta)$ instead of $\mathcal{L}(q, \Theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \frac{p(\mathbf{X},\mathbf{Z}|\Theta)}{q(\mathbf{Z})}$. Since we assume that $q(\mathbf{Z})$ is fixed, then $\mathcal{L}(q, \Theta) = \mathcal{L}'(q, \Theta) +$ constant, arg $\max_\Theta \mathcal{L}(q, \Theta) =$ arg $\max_\Theta \mathcal{L}'(q, \Theta)$.

After maximizing $\mathcal{L}(q, \Theta)$, we get new Θ^{new} and $\mathcal{L}(q, \Theta) > C$(unless $\mathcal{L}(q, \Theta)$ is alread in at a (local or global) maximum.).

We still used Θ^{old} to evaluate $q(\mathbf{Z})$, so $KL(q(\mathbf{Z}|\mathbf{X}, \Theta^{old})||p(\mathbf{Z}|\mathbf{X}, \Theta^{new})) \geqslant 0$.

After these two steps, the lower bound of $\log p(\mathbf{X}|\Theta)$ increases, so $\log p(\mathbf{X}|\Theta)$ will increase(or converge to a local(global) maximum), i.e. $\log p(\mathbf{X}|\Theta) \geqslant \mathcal{L}(q, \theta) > C$.

3 LDA

3.1 Introduction

The essence of LDA is very similar to that of PLSA in terms of how one text is generated. The difference is that LDA is a complete generative model, also called Hierarchy Bayesian model.

As we can see that, the original PLSA model didn't model the prior on the parameters. In LDA, it views theses parameters as variables and put prior on them, which are called hyper-parameters.

Now let me introduce some formulas for presenting Gibbs Sampling method.

Let $\alpha = (\alpha_1, .., \alpha_K)$ be the hyper-parameters for document-topic distribution(i.e. $\theta_d = \{p(z|d)\}_z$). Let $\beta = (\beta_1, .., \beta_V)$ be the hyper-parameters for topic-word distribution(i.e. $\phi_z = \{p(w|z)\}_w$). The prior probabilities on parameters are defined in a Dirichlet distribution

$$\begin{aligned} p(\theta|\alpha) &= \prod_{m=1}^{D} \left(\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{m,k}^{\alpha_k - 1} \right), \\ p(\phi|\beta) &= \prod_{k=1}^{K} \left(\frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \right). \end{aligned} \tag{17}$$

The probabilities of all hidden variables given parameters $\{\theta_d\}_d$ is defined in a multinomial distribution

$$p(\mathbf{z}|\theta) = \prod_{m=1}^{D} \prod_{k=1}^{K} \theta_{m,k}^{n_{m,k}}, \tag{18}$$

where $n_{m,k}$ denotes the count of words assigned to topic k in document m.

The probabilities of all observed variables(words) given parameters $\{\phi z\}_z$ and all the hidden variables is defined in a multinomial distribution

$$p(\mathbf{w}|\mathbf{z},\phi) = \prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}}, \tag{19}$$

where $n_{k,v}$ denotes the count of word v assigned to topic k in corpus.

Noting that, in these two formulas, we assume that given parameters all the variables are indepedent of hyper-parameters.

Given the hyper-parameters, the joint probability of all the variables (including parameters) is defined

$$p(\mathbf{w},\mathbf{z},\theta,\phi|\alpha,\beta) = p(\mathbf{w}|\mathbf{z},\theta)p(\mathbf{z}|\phi)p(\phi|\beta)p(\theta|\alpha). \tag{20}$$

So after deriving the joint probability, we integrate all the parameters

$$\begin{aligned}
p(\mathbf{w},\mathbf{z}|\alpha,\beta) &= \int_\theta\int_\phi p(\mathbf{w}|\mathbf{z},\theta)p(\mathbf{z}|\phi)p(\phi|\beta)p(\theta|\alpha)d\phi d\theta, \qquad (21)\\
&= \int_\theta\int_\phi \left(\prod_{m=1}^{D}\prod_{k=1}^{K}\theta_{m,k}^{n_{m,k}}\right)\left(\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}}\right)\\
&\quad \left(\prod_{k=1}^{K}\Big(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\prod_{v=1}^{V}\phi_{k,v}^{\beta_v-1}\Big)\right)\left(\prod_{m=1}^{D}\Big(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{m,k}^{\alpha_k-1}\Big)\right)d\phi d\theta,\\
&= \int_\theta\left(\Big(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\Big)^D\prod_{m=1}^{D}\prod_{k=1}^{K}\theta_{m,k}^{n_{m,k}+\alpha_k-1}\right)d\theta\\
&\quad \int_\phi\left(\Big(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\Big)^K\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}+\phi_v-1}\right)d\phi\\
&= \Big(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\Big)^D\prod_{m=1}^{D}\int_\theta\left(\prod_{k=1}^{K}\theta_{m,k}^{n_{m,k}+\alpha_k-1}\right)d\theta\\
&\quad \Big(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\Big)^K\prod_{k=1}^{K}\int_\phi\left(\prod_{v=1}^{V}\phi_{k,v}^{n_{k,v}+\phi_v-1}\right)d\phi\\
&= \Big(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\Big)^D\prod_{m=1}^{D}\frac{\prod_{k=1}^{K}\Gamma(\alpha_k+n_{m,k})}{\Gamma(\sum_{k=1}^{K}\alpha_k+n_{m,k})}\\
&\quad \Big(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\Big)^K\prod_{k=1}^{K}\frac{\prod_{v=1}^{V}\Gamma(\beta_v+n_{k,v})}{\Gamma(\sum_{v=1}^{V}\beta_v+n_{k,v})}
\end{aligned}$$

3.2 Gibbs Sampling

Introduction

Gibbs sampling, in its basic incarnation, is a special case of the MetropolisHastings algorithm. The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Suppose we want to obtain k. samples of $\mathbf{X} = \{x_1, \ldots, x_n\}$ from a joint distribution $p(x_1, \ldots, x_n)$. Denote the ith sample by $\mathbf{X}^{(i)} = \{x_1^{(i)}, \ldots, x_n^{(i)}\}$. We proceed as follows:

- We begin with some initial value $\mathbf{X}^{(0)}$ for each variable.
- For each sample $i = \{1 \ldots k\}$, sample each variable $x_j^{(i)}$ from the conditional distribution $p(x_j^{(i)}|x_1^{(i)}, \ldots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \ldots, x_n^{(i-1)})$. That is, sample each variable from the distribution of that variable conditioned on all other variables, making use of the most recent values and updating the variable with its new value as soon as it has been sampled.

Collapsed Gibbs Sampling for LDA

The idea of Collapsed Gibbs Sampling is that we don't explicitly solve the parameters first but infer the values of all the hidden variables. The general sampling rule for LDA is defined as follows:

$$\text{sample each hidden variable } z_i \text{ according to : } p(z_i|\mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta) \tag{22}$$

In each iteration, we update the values for all the variables $\mathbf{z}$ according to $p(z_i|\mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta)$. One important question is how to define $p(z_i|\mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta)$.

$$
\begin{aligned}
p(z_i = t|\mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta) &= \frac{p(z_i, \mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta)}{p(\mathbf{w}, \mathbf{z}_{\neg i}, \alpha, \beta)} \\
&= \frac{p(\mathbf{w}, \mathbf{z}, \alpha, \beta)}{p(\mathbf{w}_{\neg i}, w_i, \mathbf{z}_{\neg i}, \alpha, \beta)} \\
&= \frac{p(\mathbf{w}, \mathbf{z}, \alpha, \beta)}{p(\mathbf{w}_{\neg i}, \mathbf{z}_{\neg i}, \alpha, \beta)p(w_i|\mathbf{w}_{\neg i}, \mathbf{z}_{\neg i}, \alpha, \beta)} \\
&\propto \frac{p(\mathbf{w}, \mathbf{z}|\alpha, \beta)}{p(\mathbf{w}_{\neg i}, \mathbf{z}_{\neg i}|\alpha, \beta)} \\
&= \frac{\left(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\right)^D \prod_{m=1}^{D}\frac{\prod_{k=1}^{K}\Gamma(\alpha_k+n_{m,k})}{\Gamma(\sum_{k=1}^{K}\alpha_k+n_{m,k})} \left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\right)^K \prod_{k=1}^{K}\frac{\prod_{v=1}^{V}\Gamma(\beta_v+n_{k,v})}{\Gamma(\sum_{v=1}^{V}\beta_v+n_{k,v})}}{\left(\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\right)^D \prod_{m=1}^{D}\frac{\prod_{k=1}^{K}\Gamma(\alpha_k+n_{m,k}^{-i})}{\Gamma(\sum_{k=1}^{K}\alpha_k+n_{m,k}^{-i})} \left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\right)^K \prod_{k=1}^{K}\frac{\prod_{v=1}^{V}\Gamma(\beta_v+n_{k,v}^{-i})}{\Gamma(\sum_{v=1}^{V}\beta_v+n_{k,v}^{-i})}} \\
&= \frac{\alpha_t + n_{m,t}^{-i}}{\sum_{k=1}^{K}\alpha_k + n_{m,k}^{-i}} \frac{\beta_v + n_{t,v}^{-i}}{\sum_{v=1}^{V}\beta_v + n_{t,v}^{-i}}.
\end{aligned} \tag{23}
$$

The derivations above used the results in Equation 21 and also one important property of Gamma function $\Gamma(x+1) = x\Gamma(x)$.

The key idea is that we try to represent the joint probabilities(integrating all the paraters) using counts.

Implementation

The implementation of Collapsed Gibbs Sampling is very easy. You can just malloc arrays to store all the counts with one exception is that for storing the original content we have to use dynamically malloc.

4 Useful notes or papers

Here I list all the useful notes for PLSA and LDA:

- [1] has one very insightful section for EM algorithms. If you would like to understand EM, please read the corresponding section in that book.
- Qiaozhu Mei's note [5]. For incorporating background models and priors, definitely refer to this note.
- Freddy Chong Tat Chua's note [2]: This is a good note with lots technical details. You can also learn knowledge about LSA and PLSA.

- [3] is a must read note for understanding the inference of LDA.
- [4] is really an excellent paper which make me learn how to use EM for solving PLSA. In addition, there are lots of insightful points to understand PLSA.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] Freddy Chong Tat Chua. Dimensionality reduction and clustering of text documents. Technical report, 2009.

[3] Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[4] Thomas Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 2001.

[5] Qiaozhu Mei. A note on em algorithm for probabilistic latent semantic analysis. Technical report.