Stochastic Expectation Maximization for Latent Variable Models

Manzil Zaheer and Satwik Kottur
Carnegie Mellon University
Pittsburgh, PA 15213
{manzil, skottur}@cmu.edu

Abstract

In this project we want to implement and study a type of stochastic optimization. This optimization method based on expectation-maximization will be asynchronous & embarrassingly parallel and thus is useful for inference of latent variable models. The motivation for this stochastic optimization problem comes from an want to directly design a inference procedure from a "comptastical" (computational + statistical) perspective capable of leveraging modern computational resources like GPUs or cloud computing offering massive parallelism. We also find some interesting connection between stochastic expectation-maximization and stochastic gradient descent strengthening validity of proposed method.

1 Introduction

In the past decade, frameworks such as stochastic gradient descent (SGD) [1] and map-reduce [2] have enabled machine learning algorithms to scale to larger and large datasets. However, these frameworks are not always applicable to Bayesian latent variable models (LVM) with rich statistical dependencies and intractable gradients. Markov chain Monte-Carlo (MCMC) [3] is an appealing alternative, but traditional algorithms such as the Gibbs sampler do not match modern computational resources, as they are inherently sequential and the extent to which they can be parallelized depends heavily upon how the structure of the statistical model interacts with the data. Sometimes—due to the concentration of measure phenomenon associated with large sample sizes—computing the full posterior is unnecessary and *maximum a posteriori* (MAP) estimates suffice. EM and/or Variational methods [4] and have thus become the *sine qua non* for inference in these models.

In practice, it is often difficult to determine which inference algorithm will work best since this depends largely on the task, the data, and the model. Therefore, we must often turn to empirical studies to help understand their strengths and weaknesses. Furthermore, for latent variable mixture models, Gibbs and EM possess remarkably different computational properties. For example, collapsed Gibbs sampling is sequential and difficult to parallelize, but has a relatively small memory footprint and is easy to distribute because we only need to communicate imputed values. In contrast, EM is easy to parallelize, but difficult to distribute because the dense conditional puts tremendous pressure on communication bandwidth and memory. However, the two algorithms have complementary strengths.

Table 1: Comparison with other scalable LDA frameworks.

Method	Dataset	Infrastructure	Processing speed
YahooLDA [5]	140K vocab 8.2M docs 797M tokens	10 machines (2011)	12.87M tokens/s
lightLDA [6]	50K vocab 1.2B docs 200B tokens	24 machines (2014)	60M tokens/s
F+LDA [7]	1M vocab 29M docs 1.5B tokens	32 machines (2014)	110M tokens/s
SEM	**140K vocab** **3B docs** 171B tokens	**8 machines** **(2015)**	**503M tokens/s**

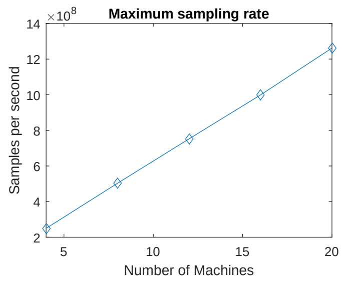


Figure 1: Scalability.

We observe that a relatively esoteric algorithm, stochastic EM (SEM), has the potential to combine the strengths of Gibbs and EM. In particular, SEM replaces the full expectation from the E-step with a sample from it, enabling the subsequent maximization step to operate only on the imputed data and current sufficient statistics. As a result, SEM substantially reduces memory and communication costs (even more than the Gibbs sampler which must also store the latent variables). Furthermore, since SEM is based on EM, it is embarassingly parallel. For example, a simple 300 line C++ implementation of SEM for latent Dirichlet allocation (LDA) [8], easily beats the state-of-the-art and is highly scalable as shown in Table 1 and Figure 1. From a scalable systems perspective SEM has clear computational advantages, but does SEM find good quality solutions to the MAP inference optimization problem in practice?

In this project, we derive Gibbs, EM, and SEM for a somewhat general and representitive class of latent variable models. More importantly, we derive the algorithms for a special instance of the class that highlights the computational advantages of SEM over the other two algorithms. Empirically as well as theoretically, we study the performance of these algorithms and find that SEM performs equally (if not better) than EM and Gibbs across many experimental conditions. Further, we find that SEM may be more robust to poor initialization conditions than either the Gibbs sampler or EM.

2 Proposed Approach

2.1 Latent Variable Exponential Family

Latent variable models are useful when reasoning about partially observed data such as collections of text or images in which each *i.i.d.* data point is a document or image. Since the same local model is applied to each data point, they have the following form

$$p(\mathbf{z}, \mathbf{x}, \eta) = p(\eta) \prod_i p\left(z_i, x_i | \eta\right). \tag{1}$$

Our goal is to obtain a MAP estimate for the parameters η that explain the data $\mathbf{x}$ through the latent variables $\mathbf{z}$. To expose maximum parallelism, we want each cell in the automaton to correspond to a data point and its latent variable. However, this is problematic because in general all latent variables depend on each other via the global parameters η and a naive approach to updating a single cell would then require examining every other cell in the automaton.

Fortunately, if we further suppose that the complete data likelihood is in the exponential family, i.e., $p(z_i, x_i|\eta) = \exp\left(\langle T(z_i, x_i), \eta\rangle - g(\eta)\right)$ then the sufficient statistics are given by $T(\mathbf{z}, \mathbf{x}) = \sum_i T(z_i, x_i)$ and we can thus express any estimator of interest as a function of just $T(\mathbf{z}, \mathbf{x})$ which factorizes over the data. Further, when employing expectation maximization (EM), the M-step is possible in closed form for many members of the exponential family. This allows us to reformulate the cell level updates to depend only upon the sufficient statistics instead of the neighboring cells. The idea is that, unlike MCMC in general which produces a sequence of states that correspond to complete variable assignments $s^0, s^1, \ldots$ via a transition kernel $q(s^{t+1}|s^t)$, we can produce a sequence of sufficient statistics $T^0, T^1, \ldots$ directly via an evolution function $\Phi(T^t) \mapsto T^{t+1}$.

2.2 Stochastic EM

Now we describe stochastic EM (SEM). Suppose we want the MAP estimate for η, $\max_\eta p(\mathbf{x}, \eta) = \max_\eta \int p(\mathbf{z}, \mathbf{x}, \eta)\mu(d\mathbf{z})$ and employ expectation maximization (EM):

E-step Compute in parallel $p(z_i|x_i, \eta^{(t)})$.
M-step Find $\eta^{(t+1)}$ that maximizes the expected log-likelihood with respect to the conditional

$$\eta^{(t+1)} = \arg\max_\eta \mathbb{E}_{\mathbf{z}|\mathbf{x},\eta^{(t)}}[\log p(\mathbf{z}, \mathbf{x}, \eta)] = \xi^{-1}\left(\frac{1}{n+n_0}\sum_i \mathbb{E}_{\mathbf{z}|\mathbf{x},\eta^{(t)}}[T(z_i, x_i)] + T_0\right)$$

where $\xi(\eta) = \nabla g(\eta)$ is invertible as $\nabla^2 g(\theta) \succ 0$ and n_0, T_0 parametrize the conjugate prior. Although EM exposes substantial parallelism, it is difficult to scale, since the dense structure $p(z_i|x_i, \eta^{(t)})$ defines values for all possible outcomes for z and thus puts tremendous pressure on memory bandwidth. To overcome this we introduce sparsity by employing stochastic EM (SEM) [9]. SEM introduces an S-step after the E-step that replaces the full distribution with a single sample:

S-step Sample $z_i^{(t)} \sim p(z_i|x_i; \eta^{(t)})$ in parallel.

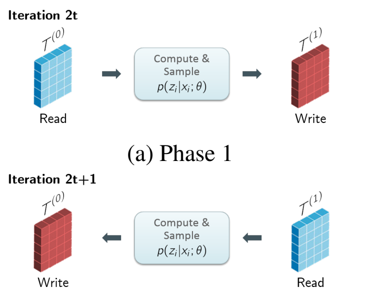
Subsequently, we perform the M-step using the imputed data instead of the expectation. This simple modification overcomes the computational drawbacks of EM for cases in which sampling from $p(z_i|x_i; \eta^{(t)})$ is feasible. (More details about EM and SEM are provided in appendix.) We can now employ fast samplers, such as the alias method, exploit sparsity, reduce CPU-RAM bandwidth while still maintaining massive parallelism. More importantly, the S-step also enables all three steps to now be expressed in terms of the current sufficient statistics. Ths enables distributed and parallel implementations that efficiently execute on modern computational resources offering massive parallelism.

2.3 Implementation

Our implementation has two copies of the data structure containing sufficient statistics $T^{(0)}$ and $T^{(1)}$. We do not compute the values $T(\mathbf{z}, \mathbf{x})$ but maintain their sum as we impute values of the cells/latent variables. During iteration $2t$ of the evolution function, we apply Φ by reading from $T^{(0)}$ and incrementing $T^{(1)}$ as we sample the latent variables (Figure 1). Then in the next iteration $2t+1$ we reverse the roles of data structure, i.e. read from $T^{(1)}$ and increment $T^{(0)}$. See Algorithm 1 and Figure 2.

Algorithm 1 SEM for LVM

```
1: Randomly initialize each cell
2: for $t = 0 \rightarrow$ num iterations do
3:     for all cell $z$ independently in parallel do
4:         Read sufficient statistics from $T^{(t \bmod 2)}$
5:         Compute stochastic updates using $p_z(k|s)$
6:         Write sufficient statistics to $T^{(t+1 \bmod 2)}$
7:     end for
8: end for
```



(a) Phase 1

(b) Phase 2

Figure 2: Efficient (re)use of buffers.

Use of such read/write buffers offer a virtually lock-free (assuming atomic increments) implementation scheme for SEM and is analogous to double-buffering in computer graphics. Although there is a synchronization barrier after each round, its effect is mitigated because each cell's work depends only upon the sufficient statistics and thus does the same amount of work. Therefore, evenly balancing the work load across computation nodes is trivial, even for a heterogeneous cluster.

2.4 Intuitions for working of SEM

In this section we present a pedagogical example that highlights the differences between the three algorithms (Gibbs, EM, SEM) and provides intuition for robust working of SEM. We choose an example that is both simple and representative of the general class. Simplicity is important because it makes it much easier to see the computational differences. In particular, we choose a mixture of Gaussians in which the mixing coefficient π_0 is known, the precision τ of each Gaussian is known, its priors $(\mu_0, \kappa_0\tau)$ are known, and inference must infer the means of the Gaussians μ. That is each datum x_i has an associated latent variable z_i that determines the Gaussian from which it is drawn. The sufficient statistics comprise just a single array of size K (one element per component). More formally, we have the following generative model:

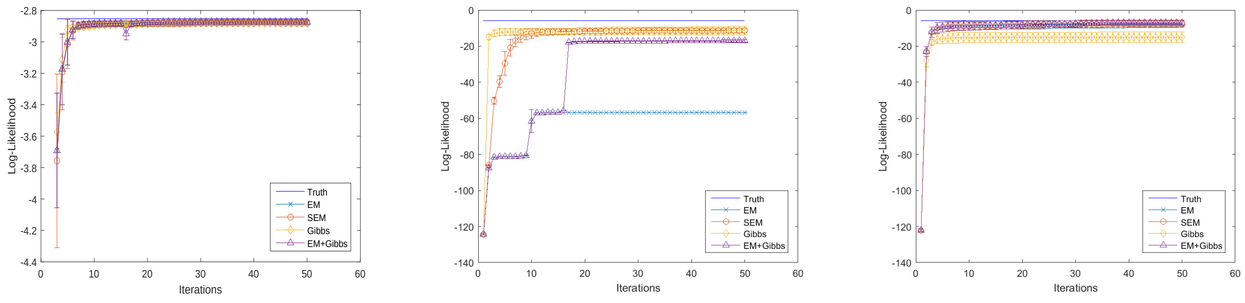
$$\begin{aligned} \mu_k &\sim \mathcal{N}(\mu_0, \kappa_0\tau) \\ z_i &\sim \mathsf{Categorical}(\pi_0) \\ x_i &\sim \mathcal{N}(\mu_{z_i}, \tau) \end{aligned}$$

For this simple example, we take $K = 10$ clusters and $n = 100,000$ training examples generated synthetically to remove any error caused by an incorrect model choice

$$\begin{aligned} \mu_0 = 0 \quad & \kappa_0 = 0.1 \\ \tau = 1 \quad & \pi \sim \mathsf{Dirichlet}(1). \end{aligned}$$

We compare Gibbs, EM and SEM ($T = 50$ iterations) under several experimental conditions. Further, we include an ad-hoc method in which the first 15 iterations are EM and then the rest are collapsed Gibbs sampling (EM+Gibbs). We repeat each condition 10 times using random initialization and report the average log-likelihood on a held-out test set of 10000 instances. First, as a control, in Figure 2a we run all three algorithms on a simple one-dimensional model. As expected, they perform equally well.

Next, we study the robustness of each algorithm to initialization in a two-dimensional model. For this, we choose two extreme initialization conditions under which the algorithms are likely to get stuck in poor local optima. In the first case we initialize to a low entropy configuration (Figure 2b), and the second case a high entropy configuration (Figure 2c). The



(a) Simple 1D case (b) Initialization to single component (c) Uniformly random initialization

Figure 2: Simple Cases

former causes trouble for EM because it is deterministic; in contrast, the stochasticity in Gibbs and SEM allow them to be robust to this condition. The latter causes trouble for Gibbs because the conditional distribution has high entropy forcing the sampler to rely on sheer luck in the early iterations. In summary, we find SEM is more robust than either algorithm since it works well in both extremes.

2.5 Theory

It is desirable to study the theoretical behavior of SEM and understand why it works so well in practice (Sec. 3). Guarantees on the performance of SEM would make it the algorithm of choice for scalable LVM. We first describe some connections between SEM and Stochastic gradient descent (SDG), inspired from the connection between EM and gradient descent. We next briefly discuss related literature helpful in a theoretical understanding of convergence properties for SEM. However, note that we could not concretely obtain guarantees for the performance of SEM and this would be a good future work.

2.5.1 Connections between SEM and SGD

We can view SEM as implicit SGD on MAP. This connection alludes to the convergence rate of SEM. To illustrate this, we consider Latent Dirichlet Allocation (LDA), which is a famous LVM used for topic modelling. For simplicity, we consider only the topic mixture proportions (θ). As pointed out in [10, 11], one EM step is:

$$\theta_m^+ = \theta_m + M\frac{\partial \log p}{\partial \theta_{mk}}$$

which is gradient descent with Frank-Wolfe type update and line search. Similarly, for SEM, one step is

$$\theta_{mk}^+ = \frac{Dn_{mk}}{N_m} = \frac{1}{N_m}\sum_{n=1}^{N_m}\delta(z_{mn}=k)$$

Again vectorizing and re-writing as earlier:

$$\theta_m^+ = \theta_m + Mg$$

where $M = \frac{1}{N_m}\left[\text{diag}(\theta_m) - \theta_m\theta_m^T\right]$ and $g = \frac{1}{\theta_{mk}}\sum_{n=1}^{N_m}\delta(z_{mn}=k)$. The vector g can be shown to be an unbiased noisy estimate of the gradient, i.e.

$$\mathbb{E}[g] = \frac{1}{\theta_{mk}}\sum_{n=1}^{N_i}\mathbb{E}[\delta(z_{ij}=k)] = \frac{\partial \log p}{\partial \theta_{mk}}$$

Thus, a single step of SEM is equivalent to a single step of SGD. Consequently, we could further embrace the connection to SGD and use a subset of the data for the S and M steps, similar to incremental EM [12]. Note that in the limit in which batches comprise just a single token, the algorithm emulates a collapsed Gibbs sampler. This interpretation strengthens the theoretical justification for many existing approximate Gibbs sampling approaches. More details in Appendix E.

2.5.2 Understanding Convergence

We now address the critical question of how the invariant measure of SEM for the model presented in Section 2.1 is related to the true MAP estimates. First, note that SCA is ergodic [13], a result that immediately applies if we ignore the deterministic components of our automata (corresponding to the observations). Now that we have established ergodicity, we next study the properties of the stationary distribution and find that the modes correspond to MAP estimates.

We make a few mild assumptions about the model:

- The observed data Fisher information is non-singular, i.e. $I(\eta) \succ 0$.
- For the Fisher information for $\mathbf{z}|\mathbf{x}$, we need it to be non-singular and central limit theorem, law of large number to hold, i.e. $\mathbb{E}_{\eta_0}[I_Z(\eta_0)] \succ 0$ and

$$\sup_{\eta} \left| \frac{1}{n} \sum_{i=1}^{n} I_{z_i}(\eta) - \mathbb{E}_{\eta_0}[I_X(\eta)] \right| \to 0 \text{ as } n \to \infty$$

- We assume that $\frac{1}{n} \sum_{i=1}^{n} \nabla_\eta \log p(x_i; \eta) = 0$ has at least one solution, let $\hat{\eta}$ be a solution.

These assumptions are reasonable. For example in case of mixture models (or topic models), it just means all component must be exhibited at least once and all components are unique. The details of this case are worked out in Appendix F. Also when the number of parameters grow with the data, e.g., for topic models, the second assumption still holds. In this case, we resort to corresponding result from high dimensional statistics by replacing the law of large numbers with Donsker's theorem and everything else falls into place.

Consequently, we show SEM converges weakly to a distribution with mean equal to some root of the score function ($\nabla_\eta \log p(x_i; \eta)$) and thus a MAP fixed point by borrowing the results known for SEM [14]. In particular, we have:

Theorem 1 *Let the assumptions stated above hold and $\tilde{\eta}$, is the estimate from SEM. Then as the number of* i.i.d. *data point goes to infinity, i.e. $n \to \infty$, we have*

$$\sqrt{n}(\tilde{\eta} - \hat{\eta}) \xrightarrow{\mathfrak{D}} \mathcal{N}\left(0, I(\eta_0)^{-1}[I - F(\eta_0)^{-1}\right) \tag{2}$$

where $F(\eta_0) = I + \mathbb{E}_{\eta_0}[I_X(\eta_0)](I(\eta_0) + \mathbb{E}_{\eta_0}[I_X(\eta_0)])$.

This result implies that SEM flocks around a stationary point under very reasonable assumptions and tremendous computational benefits. Also, for such complicated models, reaching a stationary point is the best that most methods achieve anyway. Now we switch gears to adopt SEM for LDA and perform some simple experimental evaluations.

3 Large Scale Application

To evaluate the strength and weaknesses of SEM for real world applications and not toy examples, we compare against parallel and distributed implementations of a collapsed Gibbs sampler (CGS) and a variational inference (CVB0) for LDA. We also compare our results to performance numbers reported in the literature including those of F+LDA and lightLDA. We choose not to evaluate directly against highly tuned and optimized and proprietary systems such as lightLDA for which no public code is available, as it would not be fair to implement it ourselves. Thus, we only take the best numbers reported by them.

3.1 SEM for LDA

Topic modeling, and latent Dirichlet allocation (LDA) [8] in particular, have become a must-have of analytics platforms and consequently needs to scale to larger and larger datasets. In LDA, we model each document m of a corpus of M documents as a distribution θ_m that represents a mixture of topics. There are K such topics, and we model each topic k as a distribution ϕ_k over the vocabulary of words that appear in our corpus. Each document m contains N_m words w_{mn} from a vocabulary of size V, and we associate a latent variable z_{mn} to each of the words. The latent variables can take one of K values that indicate which topic the word belongs to. Both distributions θ_m and ϕ_k have a Dirichlet prior, parameterized respectively with a constant α and β. See Appendix D for more details.

3.2 Existing systems

Many of the scalable systems for topic modeling are based on one of two core inference methods: the collapsed Gibbs sampler (CGS) [15], and variational inference (VI) [8] and approximations thereof [16]. To scale LDA to large datasets, or for efficiency reasons, we may need to distribute and parallelize them. Both algorithms can be further approximated to meet such implementation requirements.

Collapsed Gibbs Sampling In collapsed Gibbs sampling the full conditional distribution of the latent topic indicators given all the others is

$$p(z_{mn} = k|\boldsymbol{z}^{\neg mn}, \boldsymbol{w}) \propto \boxed{(D_{mk} + \alpha)\frac{W_{kw_{mn}} + \beta}{T_k + \beta V}} \tag{3}$$

where D_{mk} is the number of latent variables in document m that equal k, W_{kv} is the number of latent variables equal to k and whose corresponding word equals v, and T_k is the number of latent variables that equal k, all excluding current z_{mn}.

CGS is a sequential algorithm in which we draw latent variables in turn, and repeat the process for several iterations. The algorithm performs well statistically, and has further benefited from breakthroughs that lead to a reduction of the sampling complexity [17, 18]. This algorithm can be approximated to enable distribution and parallelism, primarily in two ways. One is to partition the data, perform one sampling pass and then assimilate the sampler states, thus yielding an approximate distributed version of CGS (AD-LDA) [19]. Another way is to partition the data and allow each sampler to communicate with a distributed central storage continuously. Here, each sampler sends the differential to the global state-keeper and receives from it the latest global value. A very scalable system built on this principle and leveraging inherent sparsity of LDA is YahooLDA [20]. Further improvement and sampling using alias table was incorporated in lightLDA [6]. Contemporaneously, a nomadic distribution scheme and sampling using Fenwick tree was proposed in F+LDA [7].

Variational Inference In variational inference (VI), we seek to optimize the parameters of an approximate distribution that assumes independence of the latent variables to find a member of the family that is close to the true posterior. Typically, for LDA, document-topic proportions and topic indicators are latent variables and topics are parameter. Then, coordinate ascent alternates between them.

One way to scale VI is stochastic variational inference (SVI) which employs SGD by repeatedly updating the topics via randomly chosen document subsets [21]. Adding a Gibbs step to SVI introduces sparsity for additional efficiency [22]. In some ways this is analogous to our S-step, but in the context of variational inference, the conditional is much more expensive to compute, requiring several rounds of sampling.

Another approach, CVB0, achieves scalability by approximating the collapsed posterior [23]. Here, they minimize the free energy of the approximate distribution for a given parameter γ_{mnk} and then use the zero-order Taylor expansion [16].

$$\gamma_{mnk} \propto \boxed{(D_{mk} + \alpha) \times \frac{W_{kw_{mn}} + \beta}{T_k + \beta\ V}} \tag{4}$$

where D_{mk} is the fractional contribution of latent variables in document m for topic k, W_{kv} is the contribution of latent variables for topic k and whose corresponding word equals v, and T_k is the the contribution of latent variables for topic k. Inference updates the variational parameters until convergence. It is possible to distribute and parallelize CVB0 over tokens [16]. VI and CVB0 are the core algorithms behind several scalable topic modeling systems including Mr.LDA [24] and the Apache Spark machine-learning suite.

Remark It is worth noticing that Gibbs sampling and variational inference, despite being justified very differently, have at their core the very same formulas (shown in a box in formula (3) and (4)). Each of which are literally deciding how important is some topic k to the word v appearing in document m by asking the questions: "How many times does topic k occur in document m?", "How many times is word v associated with topic k?", and "How prominent is topic k overall?". It is reassuring that behind all the beautiful mathematics, something simple and intuitive is happening. As we see next, SEM addresses the same questions via analogous formulas.

3.3 An SEM Algorithm for LDA

To re-iterate, the point of using such a method for LDA is that the parallel update dynamics of the SEM gives us an algorithm that is simple to parallelize, distribute and scale. In the next section, we will evaluate how it works in practice. For now, let us explain how we design our SCA to analyze data.

We begin by writing the stochastic EM steps for LDA (derivation is in Appendix D):

E-step: independently in parallel compute the conditional distribution locally:

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K} \theta_{mk'}\phi_{k'w_{mn}}} \tag{5}$$

S-step: independently in parallel draw z_{ij} from the categorical distribution:

$$z_{mn} \sim \mathsf{Categorical}(q_{mn1}, ..., q_{mnK}) \tag{6}$$

M-step: independently in parallel compute the new parameter estimates:

$$\begin{aligned}\theta_{mk} &= \frac{D_{mk} + \alpha - 1}{N_m + K\alpha - K} \\ \phi_{kv} &= \frac{W_{kv} + \beta - 1}{T_k + V\beta - V}\end{aligned} \tag{7}$$

We simulate these inference steps in SEM, which is a dynamical system with evolution function $\Phi : \mathcal{S} \longrightarrow \mathcal{S}$ over the state space $\mathcal{S}$. For LDA, the state space $\mathcal{S}$ is

$$\mathcal{S} = \mathcal{Z} \longrightarrow \mathcal{K} \times \mathcal{M} \times \mathcal{V} \tag{8}$$

where $\mathcal{Z}$ is the set of cell identifiers (one per token in our corpus), $\mathcal{K}$ is a set of K topics, $\mathcal{M}$ is a set of M document identifiers, and $\mathcal{V}$ is a set of V identifiers for the vocabulary words.

The initial state s_0 is the map defined as follows: for every occurrence of the word v in document m, we associate a cell z to the triple (k_z, m, v) where k_z is chosen uniformly at random from $\mathcal{K}$ and independently from $k_{z'}$ for all $z' \neq z$. This gives us

$$s_0 = z \mapsto (k_z, m, v) \tag{9}$$

We now need to describe the evolution function Φ. First, assuming that we have a state s and a cell z, we define the following distribution:

$$p_z(k|s) \propto \boxed{(D_{mk} + \alpha) \times \frac{W_{kv} + \beta}{T_k + \beta V}} \tag{10}$$

where $D_{mk} = \left|\left\{\, z \mid \exists v.\ s(z) = (k, m, v) \,\right\}\right|$,
$W_{kv} = \left|\left\{\, z \mid \exists m.\ s(z) = (k, m, v) \,\right\}\right|$, and
$T_k = \left|\left\{\, z \mid \exists m.\ \exists v.\ s(z) = (k, m, v) \,\right\}\right|$. Note that we have chosen our local update rule slightly different without an offset of -1 for the counts corresponding to the mode of the Dirichlet distributions and requiring $\alpha, \beta > 1$. Instead, our local update rule allows us to have the relaxed requirement $\alpha, \beta > 0$ which is more common for LDA inference algorithms.

Assuming that $s(z) = (k, m, v)$ and that k' is a sample from p_z (hence the name "stochastic" cellular automaton) we define the local update function as:

$$\begin{aligned} &\phi(s, z) = (k', m, v) \\ \text{where} \quad &s(z) = (k, m, v) \quad \text{and} \quad k' \sim p_z(\,\cdot\,|s)\end{aligned} \tag{11}$$

That is, the document and word of the cell remain unchanged, but we choose a new topic according to the distribution p_z induced by the state. We obtain the evolution function of the stochastic cellular automaton by applying the function ϕ uniformly on every cell.

$$\Phi(s) = z \mapsto \phi(s, z) \tag{12}$$

Finally, the SCA algorithm simulates the evolution function Φ starting with s_0. Of course, since LDA's complete data likelihood is in the exponential family, we never have to represent the states explicitly, and instead employ the sufficient statistics.

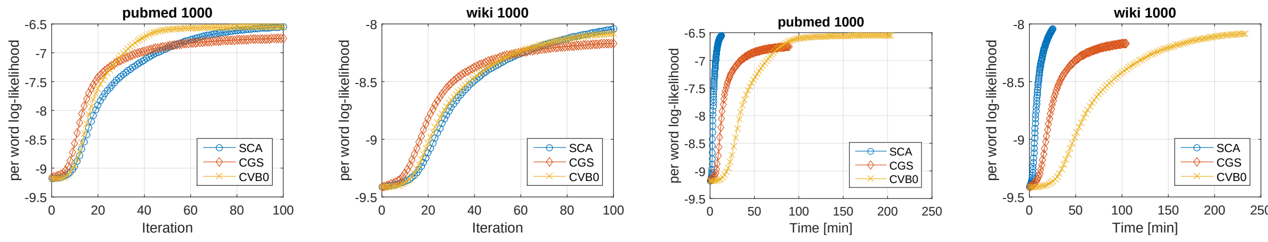
Our implementation has two copies of the count matrices D^i, W^i, and T^i for $i = 0$ or 1 (as in CGS or CVB0, we do not compute the values D_{ik}, W_{kv}, and T_k but keep track of the counts as we assign topics to the cells/latent variables). During iteration i of the evolution function, we apply Φ by reading $D^{i \bmod 2}$, $W^{i \bmod 2}$, and $T^{i \bmod 2}$ and incrementing $D^{i+1 \bmod 2}$, $W^{i+1 \bmod 2}$, and $T^{i+1 \bmod 2)}$ as we assign topics.

3.4 Experimental Results

Software & hardware All three algorithms are implemented in simple C++11. We implement multithreaded parallelization within a node using the work-stealing Fork/Join framework, and the distribution across multiple nodes using the process binding to a socket over MPI. We also implemented a version of SEM with a sparse representation for the array D of counts of topics per documents and Vose's alias method to draw from discrete distributions. We run our experiments on a small cluster of 4 nodes connected through 10Gb/s Ethernet. Each node has two 9-core Intel Xeon E5 processors for a total of 36 hardware threads per node. For random number generation we employ Intel©Digital Random Number Generators through instruction RDRAND, which uses thermal noise within the silicon to output a random stream of bits at 3 Gbit/s, producing true random numbers.

Table 2: Some statistics of the datasets used in experiment

Dataset	**V**	**M**	**Tokens**
PubMed	141,043	8,200,000	737,869,085
Wikipedia	210,233	6,631,176	1,133,050,514
Proprietary	∼140,000	∼3 billion	∼171 billion



(a) PubMed, $K = 1000$, $\alpha = 0.05$, $\beta = 0.1$
(b) Wikipedia, $K = 1000$, $\alpha = 0.05$, $\beta = 0.1$
(c) Pubmed, $K = 1000$, $\alpha = 0.05$, $\beta = 0.1$
(d) Wikipedia, $K = 1000$, $\alpha = 0.05$, $\beta = 0.1$

Figure 3: Evolution of log likelihood on Wikipedia and Pubmed over number of iterations and time.

Datasets We experiment on two public datasets, both of which are cleaned by removing stop words and rare words: PubMed abstracts and English Wikipedia. We also run on a third proprietary dataset. Details are presented in Table 2.

Evaluation To evaluate the proposed method we use predicting power as a metric by calculating the per-word log-likelihood (equivalent to negative log of perplexity) on 10,000 held-out documents conditioned on the trained model. We set $K = 1000$ to demonstrate performance for a large number of topics. The hyper parameters are set as $\alpha = 50/K$ and $\beta = 0.1$ as suggested in [25]; other systems such as YahooLDA and Mallet also use this as the default parameter setting. The results are presented in Figure 3.

Finally, for the large dataset, our implementation (only 300 lines of C++) processes 503 million tokens per second (tps) on our modest 8-node cluster. In comparison, some of the best existing systems achieve 112 million tps (F+LDA, personal communication) and 60 million tps (lightLDA) [6].

4 Conclusion

We have described a novel inference method for latent variable models that is a stochastic version of the Expectation Maximization algorithm. The equilibrium of the dynamics are MAP fixed points and the algorithm has many desirable computational properties: it is embarrassingly parallel, memory efficient, and like HOGWILD!, is virtually lock-free. Further, for many models, it enables the use of approximate counters and the alias method. Thus, we were able to achieve an order of magnitude speed-up over the current state-of-the-art inference algorithms for LDA with accuracy comparable to collapsed Gibbs sampling.

In general, we canno't always guarantee the correct invariant measure [26], and found that parallelizing improperly causes convergence to incorrect MAP fixed points. Even so, SEM is used for simulating Ising models in statistical physics [27].

4.1 Advantages of SEM for LDA

The positive consequences of SEM as a choice for inference on LDA are many:

- Our memory footprint is minimal since we only store the data and sufficient statistics. In contrast to MCMC methods, we do not store the assignments to latent variables **z**. In contrast to variational methods, we do not store the variational parameters γ. Further, variational methods require K memory accesses (one for each topic) per word. In contrast, the S-step ensures we only have a single access (for the sampled topic) per word. Such reduced pressure on the memory bandwidth can improve performance significantly for highly parallel applications.
- We can further reduce the memory footprint by compressing the sufficient statistics with approximate counters [28, 29]. This is possible because updating the sufficient statistics only requires increments as in Mean-for-Mode [30]. In contrast, CGS decrements counts, preventing the use of approximate counters.
- Our implementation is lock-free (in that it does not use locks, but assumes atomic increments) because the double buffering ensures we never read or write to the same data structures. There is less synchronization, which at scale is significant.

References

[1] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

[2] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[3] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1995.

[4] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.

[5] A. Smola and S. Narayanamurthy, “An architecture for parallel topic models,” *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 703–710, Sep. 2010. [Online]. Available: http://dx.doi.org/10.14778/1920841.1920931

[6] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma, “Lightlda: Big topic models on modest computer clusters,” in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1351–1361.

[7] H.-F. Yu, C.-J. Hsieh, H. Yun, S. Vishwanathan, and I. S. Dhillon, “A scalable asynchronous distributed algorithm for topic modeling,” in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1340–1350.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.

[9] G. Celeux and J. Diebolt, “The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem,” *Computational statistics quarterly*, vol. 2, no. 1, pp. 73–82, 1985.

[10] L. Xu and M. I. Jordan, “On convergence properties of the em algorithm for gaussian mixtures,” *Neural computation*, vol. 8, no. 1, pp. 129–151, 1996.

[11] R. Salakhutdinov, S. Roweis, and Z. Ghahramani, “Relationship between gradient and em steps in latent variable models.”

[12] R. M. Neal and G. E. Hinton, “A view of the em algorithm that justifies incremental, sparse, and other variants,” in *Learning in graphical models*. Springer, 1998, pp. 355–368.

[13] P.-Y. Louis, “Automates cellulaires probabilistes : mesures stationnaires, mesures de gibbs associées et ergodicité,” Ph.D. dissertation, Université des Sciences et Technologies de Lille and il Politecnico di Milano, September 2002.

[14] S. F. Nielsen, “The stochastic em algorithm: estimation and asymptotic results,” *Bernoulli*, pp. 457–489, 2000.

[15] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proc. National Academy of Sciences of the United States of America*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[16] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, “On smoothing and inference for topic models,” in *Proc. Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI ’09. Arlington, Virginia, USA: AUAI Press, 2009, pp. 27–34.

[17] L. Yao, D. Mimno, and A. McCallum, “Efficient methods for topic model inference on streaming document collections,” in *Proc. 15th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, ser. KDD ’09. New York: ACM, 2009, pp. 937–946.

[18] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, “Reducing the sampling complexity of topic models,” in *20th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, 2014.

[19] D. Newman, A. Asuncion, P. Smyth, and M. Welling, “Distributed algorithms for topic models,” *J. Machine Learning Research*, vol. 10, pp. 1801–1828, Dec. 2009, http://dl.acm.org/citation.cfm?id=1577069.1755845.

[20] A. Smola and S. Narayanamurthy, “An architecture for parallel topic models,” *Proc. VLDB Endowment*, vol. 3, no. 1-2, pp. 703–710, Sep. 2010.

[21] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *Journal of Machine Learning Research*, vol. 14, pp. 1303–1347, May 2013.

[22] D. Mimno, M. Hoffman, and D. Blei, “Sparse stochastic inference for latent dirichlet allocation,” in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ser. ICML ’12, J. Langford and J. Pineau, Eds. New York, NY, USA: Omnipress, July 2012, pp. 1599–1606.

[23] W. Y. Teh, D. Newman, and M. Welling, “A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation,” in *Advances in Neural Information Processing Systems 19*, ser. NIPS 2006. MIT Press, 2007, pp. 1353–1360.

[24] K. Zhai, J. Boyd-Graber, N. Asadi, and M. L. Alkhouja, “Mr. lda: A flexible large scale topic modeling package using variational inference in mapreduce,” in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 879–888.

[25] T. Griffiths and M. Steyvers, “Finding scientific topics,” *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004.

[26] D. A. Dawson, “Synchronous and asynchronous reversible Markov systems,” *Canadian mathematical bulletin*, vol. 17, pp. 633–649, 1974.

[27] G. Y. Vichniac, “Simulating physics with cellular automata,” *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 96–116, Jan. 1984.

[28] R. Morris, “Counting large numbers of events in small registers,” *Commun. ACM*, vol. 21, no. 10, pp. 840–842, Oct. 1978.

[29] M. Csűrös, “Approximate counting with a floating-point counter,” in *Computing and Combinatorics (COCOON 2010)*, ser. Lecture Notes in Computer Science, M. T. Thai and S. Sahni, Eds. Springer Berlin Heidelberg, 2010, no. 6196, pp. 358–367, see also http://arxiv.org/pdf/0904.3062.pdf.

[30] J.-B. Tristan, J. Tassarotti, and G. L. Steele Jr., “Efficient training of LDA on a GPU by Mean-For-Mode Gibbs sampling,” in *32nd International Conference on Machine Learning*, ser. ICML 2015, vol. 37, 2015, volume 37 of the Journal in Machine Learning Research: Workshop and Conference Proceedings.

A (Stochastic) EM in General

Expectation-Maximization (EM) is an iterative method for finding the maximum likelihood or *maximum a posteriori* (MAP) estimates of the parameters in statistical models when data is only partially, or when model depends on unobserved latent variables. This section is inspired from `http://www.ece.iastate.edu/~namrata/EE527_Spring08/emlecture.pdf`

We derive EM algorithm for a very general class of model. Let us define all the quantities of interest.

Table 3: Notation

Symbol	Meaning
$\mathbf{x}$	Observed data
$\mathbf{z}$	Unobserved data
$(\mathbf{x}, \mathbf{z})$	Complete data
$f_{\mathbf{X};\eta}(\mathbf{x};\eta)$	marginal observed data density
$f_{\mathbf{Z};\eta}(\mathbf{z};\eta)$	marginal unobserved data density
$f_{\mathbf{X},\mathbf{Z};\eta}(\mathbf{x},\mathbf{z};\eta)$	complete data density/likelihood
$f_{\mathbf{Z}\|\mathbf{X};\eta}(\mathbf{z}\|\mathbf{x};\eta)$	conditional unobserved-data (missing-data) density.

Objective: To maximize the marginal log-likelihood or posterior, i.e.

$$L(\eta) = \log f_{\mathbf{X};\eta}(\mathbf{x};\eta). \tag{13}$$

Assumptions:

1. z_i are independent given η. So

$$f_{\mathbf{Z};\eta}(\mathbf{z};\eta) = \prod_{i=1}^{N} f_{Z_i;\eta}(z_i;\eta), \tag{14}$$

2. x_i are independent given missing data z_i and η. So

$$f_{\mathbf{X},\mathbf{Z};\eta}(\mathbf{x},\mathbf{z};\eta) = \prod_{i=1}^{N} f_{X_i,Z_i;\eta}(x_i,z_i;\eta). \tag{15}$$

As a consequence we obtain:

$$f_{\mathbf{Z}|\mathbf{X};\eta}(\mathbf{z}|\mathbf{x};\eta) = \prod_{i=1}^{N} f_{Z_i|X_i;\eta}(z_i|x_i;\eta), \tag{16}$$

Now,

$$L(\eta) = \log f_{\mathbf{X};\eta}(\mathbf{x};\eta) = \log f_{\mathbf{X},\mathbf{Z};\eta}(\mathbf{x},\mathbf{z};\eta) - \log f_{\mathbf{Z}|\mathbf{X};\eta}(\mathbf{z}|\mathbf{x};\eta) \tag{17}$$

or, summing across observations,

$$L(\eta) = \sum_{i=1}^{N} \log f_{X_i;\eta}(x_i;\eta) = \sum_{i=1}^{N} \log f_{X_i,Z_i;\eta}(x_i,z_i;\eta) - \sum_{i=1}^{N} \log f_{Z_i|X_i;\eta}(z_i|x_i;\eta). \tag{18}$$

Let us take the expectation of the above expression with respect to $f_{Z_i|X_i;\eta}(z_i|x_i;\eta_p)$, where we choose $\eta = \eta_p$:

$$\begin{aligned}&\sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i;\eta}(x_i;\eta)|x_i;\eta_p\right] \\ &\quad= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i,z_i;\eta)|x_i;\eta_p\right] - \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{Z_i|X_i;\eta}(z_i|x_i;\eta)|x_i;\eta_p\right]\end{aligned} \tag{19}$$

Since $L(\eta) = \log f_{\mathbf{X};\eta}(\mathbf{x};\eta)$ does not depend on $\mathbf{z}$, it is invariant for this expectation. So we recover:

$$\begin{aligned}L(\eta) &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i,z_i;\eta)|x_i;\eta_p\right] - \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{Z_i|X_i;\eta}(z_i|x_i;\eta)|x_i;\eta_p\right] \\ &= Q(\eta|\eta_p) - H(\eta|\eta_p).\end{aligned} \tag{20}$$

Now, (20) may be written as

$$Q(\eta|\eta_p) = L(\eta) + \underbrace{H(\eta|\eta_p)}_{\leq H(\eta_p|\eta_p)} \tag{21}$$

Here, observe that $H(\eta|\eta_p)$ is maximized (with respect to η) by $\eta = \eta_p$, i.e.

$$H(\eta|\eta_p) \leq H(\eta_p|\eta_p) \tag{22}$$

Simple proof using Jensen's inequality.

As our objective is to maximize $L(\eta)$ with respect to η, if we maximize $Q(\eta|\eta_p)$ with respect to η, it will force $L(\eta)$ to increase. This is what is done repetitively in EM. To summarize, we have:

E-step : Compute $f_{Z_i|X_i;\eta}(z_i|x_i;\eta_p)$ using current estimate of $\eta = \eta_p$.

M-step : Maximize $Q(\eta|\eta_p)$ to obtain next estimate η_{p+1}.

Now assume that the complete data likelihood belongs to the exponential family, i.e.

$$f_{X_i,Z_i;\eta}(x_i, z_i; \eta) = \exp\{T\{z_i \cdot x_i\}\eta - g(\eta)\} \tag{23}$$

then

$$\begin{aligned} Q(\eta|\eta_p) &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i, z_i; \eta)|x_i; \eta_p\right] \\ &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[T\{z_i, \cdot, x_i\}\eta g(\eta)|x_i; \eta_p\right] \end{aligned} \tag{24}$$

To find the maximizer, differentiate and set it to zero:

$$\frac{1}{N}\sum_i \mathbb{E}_{Z_i|X_i;\eta}\left[\{Tz_i, x_i\}\eta|x_i; \eta_p\right] = \frac{dg(\eta)}{d\eta} \tag{25}$$

and one can obtain the maximizer by solving this equation.

Stochastic EM (SEM) introduces an additional simulation after the E-step that replaces the full distribution with a single sample:

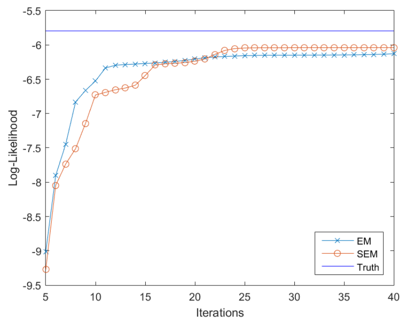
S-step Sample $z_i \sim f_{Z_i|X_i;\eta}(z_i|x_i;\eta_p)$

This essentially means we replace $\mathbb{E}[\cdot]$ with an empirical estimate. Thus, instead of solving (25), we simply have:

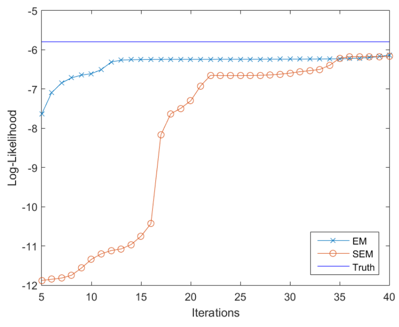
$$\frac{1}{N}\sum_i T(z_i, x_i) = \frac{dg(\eta)}{d\eta}. \tag{26}$$

Computing and solving this system of equations is considerably easier than (25).

Now to demonstrate that SEM is well behaved and works in practice, we run a small experiment. Consider the problem of estimating the parameters of a Gaussian mixture. We choose a 2-dimensional Gaussian with $K = 30$ clusters and 100,000 training points and 1,000 test points. We run EM and SEM with the following initialization:



(a) Same initialization



(b) Bad initialization for SEM

Figure 4: Performance of SEM

- Both SEM and EM are provided the same initialization.
- SEM is deliberately provided a bad initialization, while EM is not.

The log-likelihood on the heldout test set is shown in Figure 4.

B (S)EM Derivation for GMM

Endless flow of equations The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood (ℓ) evaluated using the current estimate for the parameters (initially, random values), and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

We are in a Bayesian world, so parameters are treated as Random Variables:

$$\begin{aligned}
\log p(X,\theta,\pi|m_0,\phi_0,\alpha) &= \log \sum_Z p(X,Z,\theta,\pi|m_0,\phi_0,\alpha) \\
&= \log \sum_Z p(X,Z|\theta,\pi) + \sum_k \log p(\theta_k|m_0,\phi_0) + \log p(\pi|\alpha) \\
&= \sum_i \log \sum_k p(x_i, z_i = k|\theta,\pi) + \sum_k \log p(\theta_k|m_0,\phi_0) + \log p(\pi|\alpha) \\
&= \sum_i \log \sum_k \frac{q(z_i=k|x_i)}{q(z_i=k|x_i)} p(x_i, z_i = k|\theta,\pi) + \sum_k \log p(\theta_k|m_0,\phi_0) + \log p(\pi|\alpha) \\
&= \sum_i \log \sum_k q(z_i=k|x_i) \frac{p(x_i, z_i = k|\theta,\pi)}{q(z_i=k|x_i)} + \sum_k \log p(\theta_k|m_0,\phi_0) + \log p(\pi|\alpha) \\
&\geq \sum_i \sum_k q(z_i=k|x_i) \log \frac{p(x_i, z_i = k|\theta,\pi)}{q(z_i=k|x_i)} + \sum_k \log p(\theta_k|m_0,\phi_0) + \log p(\pi|\alpha)
\end{aligned} \tag{27}$$

Let's denote

$$F(q,\theta,\pi) = \sum_i \sum_k q(z_i=k|x_i) \log \frac{p(x_i, z_i = k|\theta,\pi)}{q(z_i=k|x_i)} + \sum_k \log p(\theta_k|m_0,\phi_0) + p(\pi|\alpha)$$

EM algorithm (in coordinate descent manner) works as follows:

- In E-step, fix θ and π, maximize F over q. On re-arranging one could get:

$$F(q,\theta,\pi) = -\sum_i D_{KL}(q(z_i|x_i)||P(z_i|x_i,\theta,\pi)) + \log P(x_i|\theta,\pi) + \sum_k \log p(\theta_k|m_0,\phi_0) + p(\pi|\alpha)$$

D_{KL} denotes the Kullback Leibler divergence, a measure of the divergence of two distributions, which is defined as $D_{KL}(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$. Since in E-step, θ and π are fixed, $F(q,\theta,\pi)$ is maximized only by q that maximizes the negative KL divergence. Because KL divergence is always non-negative. $D_{KL} = 0$ happens only when p and q are the same.

Summary: In the E-step of t^{th} iteration, we derive $q^{(t)} = \operatorname{argmax}_q F(q,\theta^{(t-1)},\pi^{(t-1)})$, namely

$$\begin{aligned}
n_{ik} = E_{q|x}[z_i = k] = q^{(t)}(z_i=k|x_i) &= p(z_i = k|x_i;\theta^{(t-1)},\pi^{t-1}) \\
&\propto p(x_i|\theta_k^{t-1}, z_i = k)p(z_i = k|\pi^{t-1}) \\
&= \frac{\pi_k^{t-1} p(x_i|\theta_k^{t-1})}{\sum_{k'} \pi_{k'}^{t-1} p(x_i|\theta_{k'}^{t-1})}
\end{aligned} \tag{28}$$

- In M-step, fix q, maximize F over θ and π.

We begin by maximizing over θ, in which case we can drop other terms:

$$\begin{aligned}
F(q,\theta,\pi) &= \sum_i \sum_k q(z_i=k|x_i) \log p(x_i, z_i = k|\theta,\pi) + \sum_k \log p(\theta_k|m_0,\phi_0) + \text{const} \\
&= \sum_i \sum_k q(z_i=k|x_i)(\langle\phi(x_i),\theta_k\rangle - g(\theta_k)) + \sum_k (\langle\phi_0,\theta_k\rangle - m_0 g(\theta_k) + \text{const}
\end{aligned} \tag{29}$$

Taking derivative with respect to θ_k and setting it to 0 yields

$$\begin{aligned}
\nabla g(\hat{\theta}_k) &= \frac{1}{m_0 + \sum_k q(z_i = k|x_i)} \left(\phi_0 + \sum_i q(z_i = k|x_i)\phi(x_i) \right) \\
\hat{\theta}_k &= \eta^{-1} \left(\frac{1}{m_0 + \sum_k q(z_i = k|x_i)} \left(\phi_0 + \sum_i q(z_i = k|x_i)\phi(x_i) \right) \right)
\end{aligned} \tag{30}$$

Similarly solving for π, we first summarize the equation with the terms related to π_k as follwoing:

$$\begin{aligned}
F(q, \theta, \pi) &= \sum_i \sum_k q(z_i = k|x_i) \log p(x_i, z_i = k|\theta, \pi) + \sum_k \log p(\theta_k|m_0, \phi_0) + \log p(\pi|\alpha) \\
F(q, \theta, \pi) &= \sum_i \sum_k q(z_i = k|x_i) \log \pi_k + \sum_k (\alpha_k - 1) \log \pi_k + const
\end{aligned} \tag{31}$$

Now solving over π_k leads to solving the following optimization function,

$$\begin{aligned}
\hat{\pi} = \underset{\pi}{\operatorname{argmax}} &\sum_i \sum_k q(z_i = k|x_i) \log \pi_k + \sum_k (\alpha_k - 1) \log \pi_k \\
\text{s.t.} \qquad &\sum_{k=1}^{K} \pi_k = 1.
\end{aligned} \tag{32}$$

Writing the lagrangian function for the given optimization function,

$$L(\pi, \lambda) = \sum_i \sum_k q(z_i = k|x_i) \log \pi_k + \sum_k (\alpha_k - 1) \log \pi_k + \lambda(1 - \sum_{k=1}^{K} \pi_k) \tag{33}$$

Now, setting the gradient with respect to π_k gives us

$$\begin{aligned}
0 &= \frac{1}{\hat{\pi}_k} \sum_i \left[q(z_i = k|x_i) + (\alpha_k - 1) \right] + \lambda \\
\Leftrightarrow \quad \hat{\pi_k} &= \frac{\sum_i q(z_i = k|x_i) + (\alpha_k - 1)}{\lambda}
\end{aligned} \tag{34}$$

Since $q(z_i = k|x_i) + (\alpha_k - 1) \geq 0$ and π_k have to sum up to 1, solving for λ, and thereby obtaining the solution for π_k as

$$\begin{aligned}
\hat{\pi}_k &= \frac{\sum_i q(z_i = k|x_i) + \alpha_k - 1}{\sum_{i,k} q(z_i = k|x_i) + \sum_k \alpha_k - k} \\
&= \frac{\sum_i q(z_i = k|x_i) + \alpha_k - 1}{N + \sum_k \alpha_k - K}.
\end{aligned} \tag{35}$$

The distribution of Multivariate Normal $\mathcal{N}(\mu, \Sigma)$ is given by

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^{\mathrm{T}} \Sigma^{-1} (x - \mu) \right) \tag{36}$$

where $\mu \in \mathbb{R}^d$ and $\Sigma \succ 0$ is a symmetric positive definite $d \times d$ matrix.

The conjugate prior for Multivariate Normal Distribution can be parametrized as the Normal Inverse Wishart Distribution $\mathcal{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0)$. The distribution is given by:

$$\begin{aligned}
p(\mu, \Sigma;\ \mu_0, \kappa_0, \Sigma_0, \nu_0) &= \mathcal{N}(\mu|\mu_0, \Sigma/\kappa_0)\mathcal{W}^{-1}(\Sigma|\Sigma_0, \nu_0) \\
&= \frac{\kappa_0^{\frac{d}{2}} |\Sigma_0|^{\frac{\nu_0}{2}} |\Sigma|^{-\frac{\nu_0+d+2}{2}}}{2^{\frac{(\nu_0+1)d}{2}} \pi^{\frac{d}{2}} \Gamma_d(\frac{\nu_0}{2})} e^{-\frac{\kappa_0}{2}(\mu-\mu_0)^{\mathrm{T}}\Sigma^{-1}(\mu-\mu_0) - \frac{1}{2}\operatorname{tr}(\Sigma_0\Sigma^{-1})}
\end{aligned} \tag{37}$$

Now, derive the Expectation-Maximiazation rules for the mixture of Mutivariate Normal, $\mathcal{N}(\mu_k, \Sigma_k)$ for $k = 1, \ldots, K$ and with the shared prior $\mathcal{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0)$.

E step:

$$n_{ik}^{(t)} = \frac{\exp\left[-\frac{1}{2}(x_i - \mu_k^{(t)})^\top \Sigma_k^{(t)-1}(x_i - \mu_k^{(t)})\right] \pi_k^{(t)}}{\sum_{j=1}^{k} \exp\left[-\frac{1}{2}(x_i - \mu_j^{(t)})^\top \Sigma_j^{(t)-1}(x_i - \mu_j^{(t)})\right] \pi_j^{(t)}} \tag{38}$$

M step: First,

$$\pi_k^{(t+1)} = \frac{\alpha_k - 1 + \sum_i n_{ik}^{(t)}}{N + \sum_{j=1}^{K} \alpha_j - K}. \tag{39}$$

Now, in natural parameter space, $\theta_1 = \Sigma^{-1}\mu$ and $\theta_2 = -\frac{1}{2}\Sigma^{-1}$. Thus,

$$\begin{aligned} \Sigma &= -\frac{1}{2}\theta_2^{-1} \\ \mu &= -\frac{1}{2}\theta_2^{-1}\theta_1 \end{aligned} \tag{40}$$

and

$$g(\theta) = \begin{bmatrix} -\frac{1}{4}\theta_1^\top \theta_2^{-1}\theta_1 \\ \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|-2\theta_2| \end{bmatrix}. \tag{41}$$

So

$$\frac{\partial g}{\partial \theta_1} = \begin{bmatrix} -\frac{1}{2}\theta_2^{-1}\theta_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mu \\ 0 \end{bmatrix} \qquad \frac{\partial g}{\partial \theta_2} = \begin{bmatrix} \frac{1}{4}\theta_2^{-1}\theta_1\theta_1^\top\theta_2^{-1} \\ -\frac{1}{2}\theta_2^{-1} \end{bmatrix} = \begin{bmatrix} \mu\mu^\top \\ \Sigma \end{bmatrix}. \tag{42}$$

The derivative $\frac{\partial g_1}{\partial \theta_2}$ comes from the identity $\frac{\partial tr(X^{-1}A)}{\partial X} = -(X^{-1})^\top A (X^{-1})^\top$ and the invariance of the trace operator under cyclic permutation; see 'Wikipedia/Matrix calculus'. Also recall that

$$\phi_0 = \begin{pmatrix} \kappa_0\mu_0 \\ \Sigma_0 + \kappa_0\mu_0\mu_0^\top \end{pmatrix} \qquad m_0 = \begin{pmatrix} \kappa_0 \\ \nu_0 + d + 2 \end{pmatrix} \qquad \phi(x) = \begin{pmatrix} x \\ xx^\top \end{pmatrix}. \tag{43}$$

Denote $n_k^{(t)} = \sum_i n_{ik}^{(t)}$ Combining all of this with (**??**) we see that

$$\begin{aligned} \frac{\partial F}{\partial \theta_1} &= \kappa_0\mu_0 + \sum_i n_{ik}^{(t)} x_i - \left(\kappa_0 + n_k^{(t)}\right)\mu_k^{(t+1)} = 0 \\ \frac{\partial F}{\partial \theta_2} &= \Sigma_0 + \kappa_0\mu_0\mu_0^\top + \sum_i n_{ik}^{(t)} x_i x_i^\top - \left(\kappa_0 + n_k^{(t)}\right)\mu_k^{(t+1)}\mu_k^{(t+1)\top} - \left(\nu_0 + d + 2 + n_k^{(t)}\right)\Sigma_k^{(t+1)} = 0 \end{aligned} \tag{44}$$

and hence

$$\begin{aligned} \mu_k^{(t+1)} &= \frac{\kappa_0\mu_0 + \sum_i n_{ik}^{(t)} x_i}{\kappa_0 + n_k^{(t)}} \\ \Sigma_k^{(t+1)} &= \frac{\Sigma_0 + \kappa_0\mu_0\mu_0^\top + \sum_i n_{ik}^{(t)} x_i x_i^\top - \left(\kappa_0 + n_k^{(t)}\right)\mu_k^{(t+1)}\mu_k^{(t+1)\top}}{\nu_0 + d + 2 + n_k^{(t)}}. \end{aligned} \tag{45}$$

B.1 Introducing Stochasticity

After performing the E-step, we add an extra simulation step, i.e. we draw and impute the values for the latent variables from its distribution conditioned on data and current estimate of the parameters. This means basically n_{ik} gets transformed into $\delta(z_i - \tilde{k})$ where $\tilde{k}$ is value drawn from the conditional distribution. Then we proceed to perform the M-step, which is even simpler now. To summarize SEM for GMM will have following steps:

E-step in parallel compute the conditional distribution locally:

$$n_{ik}^{(t)} = \frac{\exp\left[-\frac{1}{2}(x_i - \mu_k^{(t)})^\top \Sigma_k^{(t)-1}(x_i - \mu_k^{(t)})\right] \pi_k^{(t)}}{\sum_{j=1}^{k} \exp\left[-\frac{1}{2}(x_i - \mu_j^{(t)})^\top \Sigma_j^{(t)-1}(x_i - \mu_j^{(t)})\right] \pi_j^{(t)}} \tag{46}$$

S-step in parallel draw z_i from the categorical distribution:

$$z_i^{(t)} \sim \mathsf{Categorical}(n_{i1}^{(t)}, ..., n_{iK}^{(t)}) \tag{47}$$

M-step in parallel compute the new parameter estimates:

$$\begin{aligned}
\pi_k^{(t+1)} &= \frac{\alpha_k - 1 + T_k^{(t)}}{N + \sum_{j=1}^{K} \alpha_j - K} \\
\mu_k^{(t+1)} &= \frac{\kappa_0 \mu_0 + \sum_{i|z_i^{(t)}=k} x_i}{\kappa_0 + T_k^{(t)}} \\
\Sigma_k^{(t+1)} &= \frac{\Sigma_0 + \kappa_0 \mu_0 \mu_0^\top + \sum_{i|z_i^{(t)}=k} x_i x_i^\top - \left(\kappa_0 + T_k^{(t)}\right) \mu_k^{(t+1)} \mu_k^{(t+1)\top}}{\nu_0 + d + 2 + n_k^{(t)}}.
\end{aligned} \tag{48}$$

where $T_k^{(t)} = \left|\left\{ z_i^{(t)} \mid z_i^{(t)} = k \right\}\right|$.

C Gibbs Sampler Derivation for GMM

The Markov blanket for z_i becomes $x_{-i}, z_{-i}, x_i, \phi_0, m_0$, and α in this case. We obtain

$$\begin{aligned}
P(z_i \mid rest) &= \frac{P(x_i \mid z_i, \{x_j : z_j = z_i\}, m_0, \phi_0) P(z_i \mid z_{-i}, \alpha)}{P(x_i \mid x_{-i}, m_0, \phi_0)} \\
&= \frac{\exp[h(m_{z_i}, \phi_{z_i}) - h(m_{z_i} - 1, \phi_{z_i} - \phi(x_i))] \exp[\log \mathbf{B}(\vec{n_k}) - \log \mathbf{B}(\vec{n_k} - \vec{e_{z_i}})]}{\sum_{j=1}^{K} \exp[h(m_j, \phi_j) - h(m_j - 1, \phi_j - \phi(x_i))] \exp[\log \mathbf{B}(\vec{n_k}) - \log \mathbf{B}(\vec{n_k} - \vec{e_j})]} \\
&= \frac{\exp[h(m_{z_i}, \phi_{z_i}) - h(m_{z_i} - 1, \phi_{z_i} - \phi(x_i))] \frac{n_{z_i} - 1}{(\sum_k n_k) - 1}}{\sum_{j=1}^{K} \exp[h(m_j, \phi_j) - h(m_j - 1, \phi_j - \phi(x_i))] \frac{n_j - 1}{(\sum_k n_k) - 1}} \\
&= \frac{\exp[h(m_{z_i}, \phi_{z_i}) - h(m_{z_i} - 1, \phi_{z_i} - \phi(x_i))](n_{z_i} - 1)}{\sum_{j=1}^{K} \exp[h(m_j, \phi_j) - h(m_j - 1, \phi_j - \phi(x_i))](n_j - 1)}
\end{aligned} \tag{49}$$

which yields the conditional needed in Gibbs sampling to sample a latent variable z_i.

As a pedagogical example, we derive Gibbs sampler for the Multivariate Gaussian with a Normal Inverse Wishart.

First, apply the downdate equations in the following order:

$$\begin{aligned}
\kappa_{z_i} \leftarrow \kappa_{z_i} - 1, &\qquad \nu_{z_i} \leftarrow \nu_{z_i} - 1 \\
\mu_{z_i} &\leftarrow \frac{(\kappa_{z_i} - 1)\mu_{z_i}}{\kappa_{z_i}} - x_i \\
\Sigma_{z_i} &\leftarrow \Sigma_{z_i} - \frac{\kappa_{z_i}}{\kappa_{z_i} - 1}(x_i - \mu_{z_i})(x_i - \mu_{z_i})^\top
\end{aligned} \tag{50}$$

Update z_i by smpling from the distribution:

$$\begin{aligned}
P(z_i = k \mid rest) &= \frac{P(x_i \mid z_i = k, x_{-i})(n_k - 1)}{\sum_j P(x_i \mid z_i = j, x_{-i})(n_j - 1)} \\
&= \frac{t_{\nu_k - d + 1}\left(x_i \,\middle|\, \mu_k, \frac{(\kappa_k + 1)\Sigma_k}{\kappa_k(\nu_k - d + 1)}\right)(n_k - 1)}{\sum_{j=1}^{K} t_{\nu_j - d + 1}\left(x_i \,\middle|\, \mu_j, \frac{(\kappa_j + 1)\Sigma_j}{\kappa_j(\nu_j - d + 1)}\right)(n_j - 1)}
\end{aligned} \tag{51}$$

then apply the update equations in the following order:

$$\begin{aligned}
\Sigma_{z_i} &\leftarrow \Sigma_{z_i} + \frac{\kappa_{z_i}}{\kappa_{z_i} - 1}(x_i - \mu_{z_i})(x_i - \mu_{z_i})^\top \\
\mu_{z_i} &\leftarrow \frac{\kappa_{z_i}\mu_{z_i} + x_i}{\kappa_{z_i} + 1} \\
\nu_{z_i} \leftarrow \nu_{z_i} + 1, &\qquad \kappa_{z_i} \leftarrow \kappa_{z_i} + 1
\end{aligned} \tag{52}$$

At the end of the procedure, we obtain

$$\begin{aligned}
\widehat{\mu}_k &= \frac{\kappa_0 \mu_0 + \sum_{i: z_i = k} x_i}{\kappa_0 + \tilde{n}_k} \\
\widehat{\Sigma}_k &= \frac{\Sigma_0 + \sum_{i: z_i = k} x_i x_i^\top + \kappa_0 \mu_0 \mu_0^\top - (\kappa_0 + \tilde{n}_k)\mu_k \mu_k^\top}{\nu_0 + \tilde{n}_k - d - 1}
\end{aligned} \tag{53}$$

where again $\tilde{n}_k = |\{i : z_i = k\}|$.

C.1 Gibbs Derivation

In the last section we used EM for inference and now we turn to Gibbs Sampling, another popular method. Gibbs sampling is a variety of MCMC sampling in which we cycle through all our latent random variables, resampling each conditioned on the currently sampled values of all other random variables.

Gibbs sampling is an MCMC method that traditionally sweeps all the variables in each iteration and one at a time, samples each variable conditioned on the rest (using $p(z_i \mid rest)$, the full conditional). We can often do better (consequence of the Rao-Blackwell theorem) by collapsing and integrating out θ_k and π. See Algorithm **??**, and see Appendix C for more details.

The Markov blanket for z_i becomes $x_{-i}, z_{-i}, x_i, \phi_0, m_0$, and α in this case. We obtain

$$\begin{aligned} P(z_i \mid rest) &= \frac{P(x_i \mid z_i, \{x_j : z_j = z_i\}, m_0, \phi_0) P(z_i \mid z_{-i}, \alpha)}{P(x_i \mid x_{-i}, m_0, \phi_0)} \\ &= \frac{\exp[h(m_{z_i}, \phi_{z_i}) - h(m_{z_i} - 1, \phi_{z_i} - \phi(x_i))](n_{z_i} - 1)}{\sum_{j=1}^{K} \exp[h(m_j, \phi_j) - h(m_j - 1, \phi_j - \phi(x_i))](n_j - 1)} \end{aligned} \tag{54}$$

Note that $m_k - m_0 = n_k - \alpha_k$. Lset $\tilde{n}_k = |\{i : z_i = k\}|$ so that $m_k = m_0 + \tilde{n}_k$ and $n_k = \tilde{n}_k + \alpha$. Thus, we need to maintain only two invariants, n_k and ϕ_k per component in the inference procedure.

Algorithm 2 Collapsed Gibbs sampling for mixture models

```
 1: Initialize $\boldsymbol{z}$ randomly and evaluate initial counts $\tilde{n}_k$ and statistics $\phi_k$.
 2: $t \leftarrow 0$
 3: **while** $t \leq T$ **do**
 4:     **for** $i = 1 \rightarrow N$ **do**
 5:         Remove datum from current component and update statistics: $\underline{\tilde{n}_{z_i} \leftarrow \tilde{n}_{z_i} - 1, \phi_{z_i} \leftarrow \phi_{z_i} - \phi(x_i)}$
 6:         Sample $z_i$ using the PMF stored in
            $p[k] \leftarrow \underline{(\alpha + \tilde{n}_k - 1) \exp\left(h(m_0 + \tilde{n}_k + 1, \phi_k + \phi(x_i)) - h(m_0 + \tilde{n}_k, \phi_k)\right)}$; $\underline{p \leftarrow p/sum(p)}$;
 7:         Add datum to the new component and update statistics: $\underline{\tilde{n}_{z_i} \leftarrow \tilde{n}_{z_i} + 1, \phi_{z_i} \leftarrow \phi_{z_i} + \phi(x_i)}$
 8:     **end for**
 9:     $t \leftarrow t + 1$
10: **end while**
```

D (S)EM Derivation for LDA

We derive an EM procedure for LDA.

D.1 LDA Model

In LDA, we model each document m of a corpus of M documents as a distribution θ_m that represents a mixture of topics. There are K such topics, and we model each topic k as a distribution ϕ_k over the vocabulary of words that appear in our corpus. Each document m contains N_m words w_{mn} from a vocabulary of size V, and we associate a latent variable z_{mn} to each of the words. The latent variables can take one of K values that indicate which topic the word belongs to. We give each of the distributions θ_m and ϕ_k a Dirichlet prior, parameterized respectively with a constant α and β. More concisely, LDA has the following mixed density.

$$p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \left[\prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{Cat}(w_{mn} \mid \phi_{z_{mn}})\, \mathrm{Cat}(z_{mn} \mid \theta_m)\right] \left[\prod_{m=1}^{M} \mathrm{Dir}(\theta_m \mid \alpha)\right] \left[\prod_{k=1}^{K} \mathrm{Dir}(\phi_k \mid \beta)\right] \tag{55}$$

The choice of a Dirichlet prior is not a coincidence: we can integrate all of the variables θ_m and ϕ_k and obtain the following closed form solution.

$$p(\boldsymbol{w}, \boldsymbol{z}) = \left[\prod_{m=1}^{M} \mathrm{Pol}\big(\{z_{m'n} \mid m' = m\}, K, \alpha\big)\right] \left[\prod_{k=1}^{K} \mathrm{Pol}\big(\{w_{mn} \mid z_{mn} = k\}, V, \beta\big)\right] \tag{56}$$

where Pol is the Polya distribution

$$\mathrm{Pol}(S, X, \eta) = \frac{\Gamma(\eta\, K)}{\Gamma(|S| + \eta\, X)} \prod_{x=1}^{X} \frac{\Gamma\big(\big|\{z \mid z \in S, z = x\}\big| + \eta\big)}{\Gamma(\eta)} \tag{57}$$

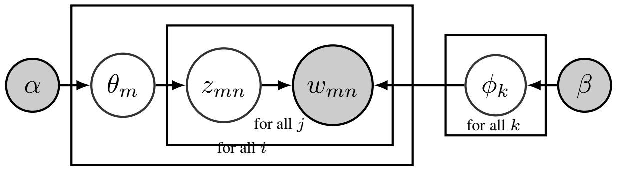


Figure 5: LDA Graphical Model

Algorithm 3 LDA Generative Model

input: $\boldsymbol{\alpha}, \boldsymbol{\beta}$

```
1: for $k = 1 \to K$ do
2:     Choose topic $\boldsymbol{\phi}_k \sim \mathsf{Dir}(\boldsymbol{\beta})$
3: end for
4: for all document $m$ in corpus $D$ do
5:     Choose a topic distribution $\boldsymbol{\theta}_m \sim \mathsf{Dir}(\boldsymbol{\alpha})$
6:     for all word index $n$ from 1 to $N_m$ do
7:         Choose a topic $z_{mn} \sim \mathsf{Categorical}(\boldsymbol{\theta}_m)$
8:         Choose word $w_{mn} \sim \mathsf{Categorical}(\boldsymbol{\phi}_{z_{mn}})$
9:     end for
10: end for
```

The joint probability density can be expressed as:

$$\begin{aligned} p(W, Z, \theta, \phi | \alpha, \beta) &= \left[\prod_{k=1}^{K} p(\phi_k|\beta)\right] \left[\prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N_m} p(z_{mn}|\theta_m) p(w_{mn}|\phi_{z_{mn}})\right] \\ &\propto \left[\prod_{k=1}^{K} \prod_{v=1}^{V} \phi_{kv}^{\beta-1}\right] \left[\prod_{m=1}^{M} \left(\prod_{k=1}^{K} \theta_{mk}^{\alpha-1}\right) \prod_{n=1}^{N_m} \theta_{m z_{mn}} \phi_{z_{mn} w_{mn}}\right] \end{aligned} \tag{58}$$

D.2 Expectation Maximization

We begin by marginalizing the latent variable Z and finding the lower bound for the likelihood/posterior:

$$\begin{aligned}
\log p(W,\theta,\phi|\alpha,\beta) &= \log \sum_{Z} p(W,Z,\theta,\phi|\alpha,\beta) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N_m} \log \sum_{k=1}^{K} p(z_{mn}=k|\theta_m)p(w_{mn}|\phi_k) \\
&\qquad\qquad + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N_m} \log \sum_{k=1}^{K} q(z_{mn}=k|w_{mn})\frac{p(z_{mn}=k|\theta_m)p(w_{mn}|\phi_k)}{q(z_{mn}=k|w_{mn})} \\
&\qquad\qquad + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha) \\
\text{(Jensen Inequality)} \quad &\geq \sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k=1}^{K} q(z_{mn}=k|w_{mn})\log \frac{p(z_{mn}=k|\theta_m)p(w_{mn}|\phi_k)}{q(z_{mn}=k|w_{mn})} \\
&\qquad\qquad + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha)
\end{aligned} \tag{59}$$

Let us define the following functional:

$$\begin{aligned}
F(q,\theta,\phi) := &-\sum_{m=1}^{M}\sum_{n=1}^{N_m} D_{KL}(q(z_{mn}|w_{mn})||p(z_{mn}|w_{mn},\theta_m,\phi)) \\
&+ \sum_{m=1}^{M}\sum_{n=1}^{N_m} p(w_{mn}|\theta_m,\phi) + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha)
\end{aligned} \tag{60}$$

D.2.1 E-Step

In the E-step, we fix θ, ϕ and maximize F for q. As q appears only in the KL-divergence term, it is equivalent to minimizing the KL-divergence between $q(z_{mn}|w_{mn})$ and $p(z_{mn}|w_{mn},\theta_m,\phi)$. We know that for any distributions f and g the KL-divergence is minimized when $f = g$ and is equal to 0. Thus, we have

$$\begin{aligned}
q(z_{mn}=k|w_{mn}) &= p(z_{mn}=k|w_{mn},\theta_m,\phi) \\
&= \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}}
\end{aligned} \tag{61}$$

For simplicity of notation, let us define

$$\boxed{q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}}} \tag{62}$$

D.2.2 M-Step

In the E-step, we fix q and maximize F for θ, ϕ. As this will be a constrained optimization (θ and ϕ must lie on simplex), we use standard constrained optimization procedure of Lagrange multipliers. The Lagrangian can be expressed as:

$$\begin{aligned}
\mathcal{L}(\theta, \phi, \lambda, \mu) = &\sum_{m=1}^{M} \sum_{m=1}^{N_m} \sum_{k=1}^{K} q(z_{mn} = k|w_{mn}) \log \frac{p(z_{mn} = k|\theta_m) p(w_{mn}|\phi_k)}{q(z_{mn} = k|w_{mn})} + \sum_{k=1}^{K} \log p(\phi_k|\beta) \\
&+ \sum_{m=1}^{M} \log p(\theta_m|\alpha) + \sum_{k=1}^{K} \lambda_k \left(1 - \sum_{v=1}^{V} \phi_{kv}\right) + \sum_{m=1}^{M} \mu_i \left(1 - \sum_{k=1}^{K} \theta_{mk}\right) \\
= &\sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{k=1}^{K} q_{mnk} \log \theta_{mk} \phi_{kw_{mn}} + \sum_{k=1}^{K} \sum_{v=1}^{V} (\beta_v - 1) \log \phi_{kv} + \sum_{m=1}^{M} \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_{mk} \\
&+ \sum_{k=1}^{K} \lambda_k \left(1 - \sum_{v=1}^{V} \phi_{kv}\right) + \sum_{m=1}^{M} \mu_m \left(1 - \sum_{k=1}^{K} \theta_{mk}\right) + \text{const.}
\end{aligned} \tag{63}$$

Maximizing θ Taking derivative with respect to θ_{mk} and setting it to 0, we obtain

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \theta_{mk}} = 0 &= \sum_{j=1}^{N_m} \frac{q_{mnk} + \alpha_k - 1}{\theta_{mk}} - \mu_m \\
\mu_m \theta_{mk} &= \sum_{j=1}^{N_i} q_{mnk} + \alpha_k - 1
\end{aligned} \tag{64}$$

After solving for μ_m, we finally obtain

$$\theta_{mk} = \frac{\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1}{\sum_{k'=1}^{K} \sum_{j=1}^{N_m} q_{mnk'} + \alpha_{k'} - 1} \tag{65}$$

Note that $\sum_{k'=1}^{K} q_{mnk'} = 1$, we reach at the optimizer:

$$\boxed{\theta_{mk} = \frac{1}{N_m + \sum(\alpha_{k'} - 1)} \left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right)} \tag{66}$$

Maximizing ϕ Taking derivative with respect to ϕ_{kv} and setting it to 0, we obtain

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \phi_{kv}} = 0 &= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \frac{q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\phi_{kv}} - \lambda_k \\
\lambda_k \phi_{kv} &= \sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1
\end{aligned} \tag{67}$$

After solving for λ_k, we finally obtain

$$\phi_{kv} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\sum_{v'=1}^{V} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \delta(v' - w_{mn}) + \beta_{v'} - 1} \tag{68}$$

Note that $\sum_{v'=1}^{V} \delta(v' - w_{mn}) = 1$, we reach at the optimizer:

$$\boxed{\phi_{kv} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'} - 1)}} \tag{69}$$

D.3 Introducing Stochasticity

After performing the E-step, we add an extra simulation step, i.e. we draw and impute the values for the latent variables from its distribution conditioned on data and current estimate of the parameters. This means basically $q_m nk$ gets transformed into $\delta(z_{mn} - \tilde{k})$ where $\tilde{k}$ is value drawn from the conditional distribution. Then we proceed to perform the M-step, which is even simpler now. To summarize SEM for LDA will have following steps:

E-step in parallel compute the conditional distribution locally:

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{ij}}} \tag{70}$$

S-step in parallel draw z_{mn} from the categorical distribution:

$$z_{mn} \sim \mathsf{Categorical}(q_{mn1}, ..., q_{mnK}) \tag{71}$$

M-step in parallel compute the new parameter estimates:

$$\begin{aligned}\theta_{mk} &= \frac{D_{mk} + \alpha_k - 1}{N_m + \sum(\alpha_{k'} - 1)} \\ \phi_{kv} &= \frac{W_{kv} + \beta_v - 1}{T_k + \sum(\beta_{v'} - 1)}\end{aligned} \tag{72}$$

where $D_{mk} = \left|\left\{\, z_{mn} \mid z_{mn} = k \,\right\}\right|$,
$W_{kv} = \left|\left\{\, z_{mn} \mid w_{mn} = v,\, z_{mn} = k \,\right\}\right|$, and
$T_k = \left|\left\{\, z_{mn} \mid z_{mn} = k \,\right\}\right| = \sum\limits_{v=1}^{V} W_{kv}$.

E Equivalency between (S)EM and (S)GD for LDA

We study the equivalency between (S)EM and (S)GD for LDA.

E.1 EM for LDA

EM for LDA can be summarized by follows:

E-Step

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} \tag{73}$$

M-Step

$$\begin{aligned}
\theta_{mk} &= \frac{1}{N_m + \sum(\alpha_{k'} - 1)}\left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right) \\
\phi_{kv} &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk}\delta(v - w_{mn}) + \beta_v - 1}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'} - 1)}
\end{aligned} \tag{74}$$

E.2 GD for LDA

The joint probability density can be expressed as:

$$\begin{aligned}
p(W, Z, \theta, \phi|\alpha, \beta) &= \left[\prod_{k=1}^{K} p(\phi_k|\beta)\right]\left[\prod_{m=1}^{M} p(\theta_m|\alpha)\prod_{n=1}^{N_m} p(z_{mn}|\theta_m)p(w_{mn}|\phi_{z_{mn}})\right] \\
&\propto \left[\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{kv}^{\beta-1}\right]\left[\prod_{m=1}^{M}\left(\prod_{k=1}^{K}\theta_{mk}^{\alpha-1}\right)\prod_{n=1}^{N_m}\theta_{mz_{mn}}\phi_{z_{mn}w_{mn}}\right]
\end{aligned} \tag{75}$$

The log-probability of joint model with Z marginalized can be written as:

$$\begin{aligned}
\log p(W, \theta, \phi|\alpha, \beta) &= \log\sum_{Z} p(W, Z, \theta, \phi|\alpha, \beta) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\log\sum_{k=1}^{K} p(z_{mn} = k|\theta_m)p(w_{mn}|\phi_k) \\
&\qquad + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\log\sum_{k=1}^{K}\theta_{mk}\phi_{kw_{mn}} \\
&\qquad + \sum_{m=1}^{M}\sum_{k=1}^{K}(\alpha_k - 1)\log\theta_{mk} + \sum_{k=1}^{K}\sum_{v=1}^{V}(\beta_v - 1)\log\phi_{kv}
\end{aligned} \tag{76}$$

Gradient for topic per document Now take derivative with respect to θ_{mk}:

$$\begin{aligned}
\frac{\partial\log p}{\partial\theta_{mk}} &= \sum_{j=1}^{N_m}\frac{\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} + \frac{\alpha_k - 1}{\theta_{mk}} \\
&= \frac{1}{\theta_{mk}}\left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right)
\end{aligned} \tag{77}$$

Gradient for word per topic Now take derivative with respect to ϕ_{kv}:

$$\begin{aligned}\frac{\partial \log p}{\partial \phi_{kv}} &= \sum_{m=1}^{M}\sum_{n=1}^{N_m} \frac{\theta_{mk}\delta(v-w_{mn})}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} + \frac{\beta_v - 1}{\phi_{kv}} \\ &= \frac{1}{\phi_{kv}}\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk}\delta(v-w_{mn}) + \beta_v - 1\right)\end{aligned} \tag{78}$$

E.3 Equivalency

If we look at one step of EM:

For topic per document

$$\begin{aligned}\theta^{+}_{mk} &= \frac{1}{N_m + \sum(\alpha_{k'}-1)}\left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right) \\ &= \frac{\theta_{mk}}{N_m + \sum(\alpha_{k'}-1)}\frac{\partial \log p}{\partial \theta_{mk}}\end{aligned}$$

Vectorize and can be re-written as:

$$\theta^{+}_{m} = \theta_m + \frac{1}{N_m + \sum(\alpha_{k'}-1)}\left[\text{diag}(\theta_m) - \theta_m\theta_m^T\right]\frac{\partial \log p}{\partial \theta_m} \tag{79}$$

For word per topic

$$\begin{aligned}\phi^{+}_{kv} &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk}\delta(v-w_{mn}) + \beta_v - 1}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'}-1)} \\ &= \frac{\phi_{kv}}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'}-1)}\frac{\partial \log p}{\partial \phi_{kv}}\end{aligned}$$

Vectorize and can be re-written as:

$$\theta^{+}_{m} = \theta_m + \frac{1}{N_m + \sum(\alpha_{k'}-1)}\left[\text{diag}(\theta_m) - \theta_m\theta_m^T\right]\frac{\partial \log p}{\partial \theta_m} \tag{80}$$

E.4 SEM for LDA

We summarize our SEM derivation for LDA as follows:

E-Step

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} \tag{81}$$

S-step

$$z_{mn} \sim \mathsf{Categorical}(q_{mn1}, ..., q_{mnK}) \tag{82}$$

M-step

$$\begin{aligned}\theta_{mk} &= \frac{D_{mk} + \alpha_k - 1}{N_m + \sum(\alpha_{k'}-1)} \\ \phi_{kv} &= \frac{W_{kv} + \beta_v - 1}{T_k + \sum(\beta_{v'}-1)}\end{aligned} \tag{83}$$

Here D_{mk} is the total number of tokens that belong to topic k in document m, W_{kv} is the number of times a word v belongs to topic k, i.e.,

$$D_{mk} = \sum_{n=1}^{N_m} z_{mnk} \tag{84}$$

$$W_{kv} = \sum_{n=1}^{N_m}\sum_{m=1}^{N_d} z_{mnk}\delta(w_m = v) \tag{85}$$

However, observe that all our z_{mn} are one-hot categorical random variables and hence, the above sums can be easily computed without going through the entire dataset. This is where the stochastic nature of SEM helps in reducing the training time. We next show the equivalency of SEM to SGD.

E.5 Equivalency

In case of LDA, let us begin with θ for which the update over one step stochastic EM is:

$$\theta_{mk}^{+} = \frac{D_{mk} + \alpha_k - 1}{N_m + \sum_{k'=1}^{K}(\alpha_{k'} - 1)} = \frac{1}{N_m + \sum_{k'=1}^{K}(\alpha_{k'} - 1)} \sum_{n=1}^{N_m} \delta(z_{mnk} = 1) + \alpha_k - 1$$

Again vectorizing and re-writing as earlier:

$$\theta_i^{+} = \theta_i + Mg$$

where $M = \frac{1}{N_m + \sum_{k'=1}^{K}(\alpha_{k'} - 1)} \left[\text{diag}(\theta_m) - \theta_m \theta_m^T\right]$ and $g = \frac{1}{\theta_{mk}} \sum_{n=1}^{N_m} \delta(z_{mnk} = 1) + \alpha_k - 1$. The vector g can be shown to be an unbiased noisy estimate of the gradient, i.e.

$$\begin{aligned}\mathbb{E}[g] &= \frac{1}{\theta_{mk}} \sum_{n=1}^{N_m} \mathbb{E}[\delta(z_{mnk} = 1)] + \alpha_k - 1 \\ &= \frac{1}{\theta_{mk}} \sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1 \qquad = \frac{\partial \log p}{\partial \theta_{mk}}\end{aligned}$$

Thus, it is SGD with constraints. We have a similar result for ϕ_{kv}, where we can see that an unbiased, noisy estimator of the gradient has been used instead of the pure gradient, in the SEM update of parameters. However, note that stochasticity does not arise from sub-sampling data as usually in SGD, rather from the randomness introduced in the S-step. But this immediately hints for developing an online/incremental version where we can subsample data also. This can remove the barrier in current implementation and we can have a revolver like structure, which would be loved by the hardware.

F Non-singularity of Fisher Information for Mixture Models

Let us consider a general mixture model:

$$p(x|\theta,\phi) = \sum_{k=1}^{K} \theta_k f(x|\phi_k) \tag{86}$$

Then the log-likelihood can be written as:

$$\log p(x|\theta,\phi) = \log\left(\sum_{k=1}^{K} \theta_k f(x|\phi_k)\right) \tag{87}$$

The Fisher Information is given by:

$$\begin{aligned} I(\theta,\phi) &= \mathbb{E}\left[(\nabla \log p(x|\theta,\phi))(\nabla \log p(x|\theta,\phi))^T\right] \\ &= \left[\begin{array}{c} \frac{\partial}{\partial\theta}\log p(x|\theta,\phi) \\ \frac{\partial}{\partial\phi}\log p(x|\theta,\phi) \end{array}\right]\left[\begin{array}{c} \frac{\partial}{\partial\theta}\log p(x|\theta,\phi) \\ \frac{\partial}{\partial\phi}\log p(x|\theta,\phi) \end{array}\right]^T \end{aligned}$$

These derivatives can be computed as follows:

$$\begin{aligned} \frac{\partial}{\partial\theta_k}\log p(x|\theta,\phi) &= \frac{\partial}{\partial\theta_k}\log\left((\sum_{k=1}^{K}\theta_k f(x|\phi_k)\right) \\ &= \frac{f(x|\phi_k)}{\sum_{k'=1}^{K}\theta_{k'} f(x|\phi_{k'})} \\ \frac{\partial}{\partial\phi_k}\log p(x|\theta,\phi) &= \frac{\partial}{\partial\phi_k}\log\left((\sum_{k=1}^{K}\theta_k f(x|\phi_k)\right) \\ &= \frac{\theta_k \frac{\partial}{\partial\phi_k} f(x|\phi_k)}{\sum_{k'=1}^{K}\theta_{k'} f(x|\phi_{k'})} \end{aligned} \tag{88}$$

For any $u, v \in \mathbb{R}^K$ (with at least one nonzero), then the Fisher Information is positive definite as:

$$\begin{aligned} (u^T\ v^T) I \left(\begin{array}{c} u \\ v \end{array}\right) &= (u^T\ v^T)\mathbb{E}\left[\left[\begin{array}{c} \frac{\partial}{\partial\theta}\log\left(\sum_{k=1}^{K}\theta_k f(X|\phi_k)\right) \\ \frac{\partial}{\partial\phi}\log\left(\sum_{k=1}^{K}\theta k f(X|\phi_k)\right) \end{array}\right]\left[\begin{array}{c} \frac{\partial}{\partial\theta}\log\left(\sum_{k=1}^{K}\theta_k f(X|\phi_k)\right) \\ \frac{\partial}{\partial\phi}\log\left(\sum_{i=1}^{K}\theta_k f(X|\phi_k)\right) \end{array}\right]^T\right]\left(\begin{array}{c} u \\ v \end{array}\right) \\ &= \mathbb{E}\left[\left(u^T\frac{\partial}{\partial\theta}\log\left(\sum_{k=1}^{K}\theta_k f(X|\phi_k)\right) + v^T\frac{\partial}{\partial\theta}\log\left(\sum_{i=1}^{K}\theta_k f(X|\phi_i)\right)\right)^2\right] \\ &= \mathbb{E}\left[\left(\frac{\sum_{k=1}^{K} u_k f(X|\phi_k) + v_k\theta_k \frac{\partial}{\partial\phi_k} f(X|\phi_k)}{\sum_{k=1}^{K}\theta_k f(X|\phi_k)}\right)^2\right] \end{aligned}$$

This can be 0 if and only if

$$\sum_{k=1}^{K} u_k f(x|\phi_i) + v_k\theta_k\frac{\partial}{\partial\phi_k} f(x;\phi_k) = 0 \quad \forall x. \tag{89}$$

In case of exponential family emission models this cannot hold if all components are unique and all $\theta_k > 0$. Thus, if we assume all components are unique and every component has been observed at least once, the Fisher information matrix becomes non-singular.