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 lecture of Dr Namrata Vaswani available at `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 2: 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
$f_{\mathbf{Z}\vert\mathbf{X};\eta}(\mathbf{z}\vert\mathbf{x};\eta)$	conditional unobserved-data (missing-data) density.

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

$$L(\eta) = \log f_{\mathbf{X};\eta}(\mathbf{x};\eta). \tag{24}$$

Assumptions:

1. z_i are independent given η. So
$$f_{\mathbf{Z};\eta}(\mathbf{z};\eta) = \prod_{i=1}^{N} f_{Z_i;\eta}(z_i;\eta), \tag{25}$$

2. x_i are independent given missing data z_i and η. So
$$f_{\mathbf{X},\mathbf{Z};\eta}(\mathbf{x},\mathbf{z};\eta) = \prod_{i=1}^{N} f_{X_i,Z_i;\eta}(x_i,z_i;\eta). \tag{26}$$

As a consequence we obtain:

$$f_{\mathbf{Z}|\mathbf{X};\eta}(\mathbf{z}|\mathbf{x};\eta) = \prod_{i=1}^{N} f_{Z_i|X_i;\eta}(z_i|x_i;\eta), \tag{27}$$

Now,

$$L(\eta) = \log f_{\mathbf{X};\eta}(\mathbf{x};\eta) = \log f_{\mathbf{X},\mathbf{Z};\eta}(\mathbf{x},\mathbf{z};\eta) - \log f_{\mathbf{Z}|\mathbf{X};\eta}(\mathbf{z}|\mathbf{x};\eta) \tag{28}$$

or, summing across observations,

$$L(\eta) = \sum_{i=1}^{N} \log f_{X_i;\eta}(x_i;\eta) = \sum_{i=1}^{N} \log f_{X_i,Z_i;\eta}(x_i,z_i;\eta) - \sum_{i=1}^{N} \log f_{Z_i|X_i;\eta}(z_i|x_i;\eta). \tag{29}$$

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 $\eta = \eta_p$:

$$\begin{aligned} &\sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i;\eta}(x_i;\eta)|x_i;\eta_p\right] \\ &\quad = \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i,z_i;\eta)|x_i;\eta_p\right] - \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{Z_i|X_i;\eta}(z_i|x_i;\eta)|x_i;\eta_p\right] \end{aligned} \tag{30}$$

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:

$$\begin{aligned} L(\eta) &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i,z_i;\eta)|x_i;\eta_p\right] - \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{Z_i|X_i;\eta}(z_i|x_i;\eta)|x_i;\eta_p\right] \\ &= Q(\eta|\eta_p) - H(\eta|\eta_p). \end{aligned} \tag{31}$$

Now, (31) may be written as

$$Q(\eta|\eta_p) = L(\eta) + \underbrace{H(\eta|\eta_p)}_{\leq H(\eta_p|\eta_p)} \tag{32}$$

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

$$H(\eta|\eta_p) \leq H(\eta_p|\eta_p) \tag{33}$$

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.

$$f_{X_i,Z_i;\eta}(x_i,z_i;\eta) = \exp\left(\langle T\left(z_i,x_i\right),\eta\rangle - g(\eta)\right) \tag{34}$$

then

$$\begin{aligned} Q(\eta|\eta_p) &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\log f_{X_i,Z_i;\eta}(x_i,z_i;\eta)|x_i;\eta_p\right] \\ &= \sum_{i=1}^{N} \mathbb{E}_{Z_i|X_i;\eta}\left[\langle T\left(z_i,x_i\right),\eta\rangle - g(\eta)|x_i;\eta_p\right] \end{aligned} \tag{35}$$

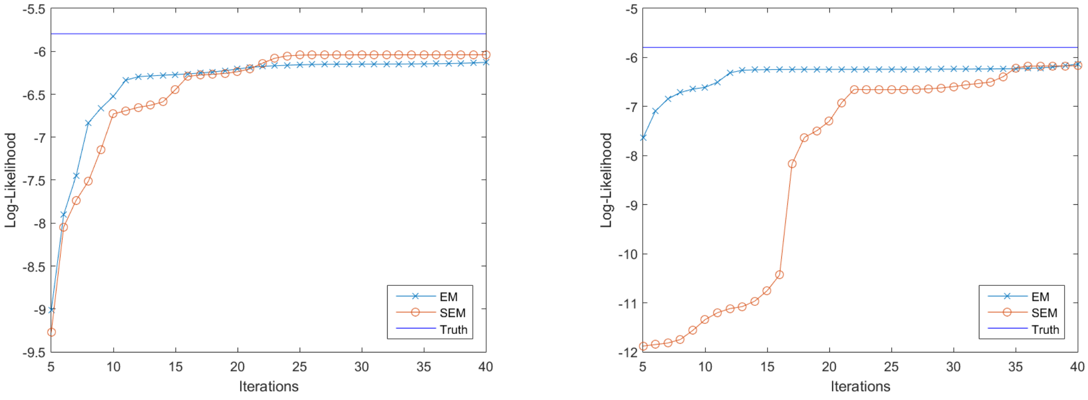
To find the maximizer, differentiate and set it to zero:

$$\frac{1}{N}\sum_i \mathbb{E}_{Z_i|X_i;\eta}\left[\langle T\left(z_i,x_i\right),\eta\rangle |x_i;\eta_p\right] = \frac{dg(\eta)}{d\eta} \tag{36}$$

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



(a) Same initialization

(b) Bad initialization for SEM

Figure 3: Performance of SEM

This essentially means we replace $\mathbb{E}[\cdot]$ with an empirical estimate. Thus, instead of solving (36), we simply have:

$$\frac{1}{N}\sum_i T(z_i, x_i) = \frac{dg(\eta)}{d\eta}. \tag{37}$$

Computing and solving this system of equations is considerably easier than (36).

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:

- 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 3.

B (S)EM Derivation for LDA

We derive an EM procedure for LDA.

B.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.

$$p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\phi}) = \left[\prod_{m=1}^{M}\prod_{n=1}^{N_m}\mathrm{Cat}(w_{mn}\mid\phi_{z_{mn}})\ \mathrm{Cat}(z_{mn}\mid\theta_m)\right]\left[\prod_{m=1}^{M}\mathrm{Dir}(\theta_m\mid\alpha)\right]\left[\prod_{k=1}^{K}\mathrm{Dir}(\phi_k\mid\beta)\right] \tag{38}$$

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.

$$p(\boldsymbol{w},\boldsymbol{z}) = \left[\prod_{m=1}^{M}\mathrm{Pol}\big(\{z_{m'n}\mid m'=m\},K,\alpha\big)\right]\left[\prod_{k=1}^{K}\mathrm{Pol}\big(\{w_{mn}\mid z_{mn}=k\},V,\beta\big)\right] \tag{39}$$

where Pol is the Polya distribution

$$\mathrm{Pol}(S,X,\eta) = \frac{\Gamma(\eta K)}{\Gamma(|S|+\eta X)}\prod_{x=1}^{X}\frac{\Gamma\big(\big|\{z\mid z\in S, z=x\}\big|+\eta\big)}{\Gamma(\eta)} \tag{40}$$

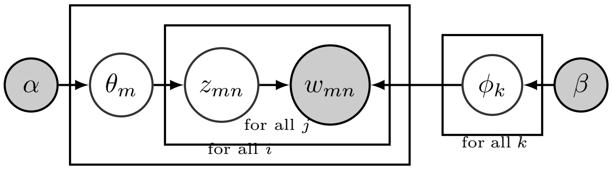


Figure 4: LDA Graphical Model

Algorithm 2 LDA Generative Model

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

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

The joint probability density can be expressed as:

$$\begin{aligned} p(W,Z,\theta,\phi|\alpha,\beta) &= \left[\prod_{k=1}^{K}p(\phi_k|\beta)\right]\left[\prod_{m=1}^{M}p(\theta_m|\alpha)\prod_{n=1}^{N_m}p(z_{mn}|\theta_m)p(w_{mn}|\phi_{z_{mn}})\right] \\ &\propto \left[\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{kv}^{\beta-1}\right]\left[\prod_{m=1}^{M}\left(\prod_{k=1}^{K}\theta_{mk}^{\alpha-1}\right)\prod_{n=1}^{N_m}\theta_{mz_{mn}}\phi_{z_{mn}w_{mn}}\right] \end{aligned} \tag{41}$$

B.2 Expectation Maximization

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

$$\begin{aligned}
\log p(W, \theta, \phi|\alpha, \beta) &= \log \sum_Z p(W, Z, \theta, \phi|\alpha, \beta) \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log \sum_{k=1}^{K} p(z_{mn} = k|\theta_m) p(w_{mn}|\phi_k) \\
&\qquad\qquad + \sum_{k=1}^{K} \log p(\phi_k|\beta) + \sum_{m=1}^{M} \log p(\theta_m|\alpha) \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log \sum_{k=1}^{K} q(z_{mn} = k|w_{mn}) \frac{p(z_{mn} = k|\theta_m) p(w_{mn}|\phi_k)}{q(z_{mn} = k|w_{mn})} \\
&\qquad\qquad + \sum_{k=1}^{K} \log p(\phi_k|\beta) + \sum_{m=1}^{M} \log p(\theta_m|\alpha) \\
\text{(Jensen Inequality)} \qquad &\geq \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{k=1}^{K} q(z_{mn} = k|w_{mn}) \log \frac{p(z_{mn} = k|\theta_m) p(w_{mn}|\phi_k)}{q(z_{mn} = k|w_{mn})} \\
&\qquad\qquad + \sum_{k=1}^{K} \log p(\phi_k|\beta) + \sum_{m=1}^{M} \log p(\theta_m|\alpha)
\end{aligned} \tag{42}$$

Let us define the following functional:

$$\begin{aligned}
F(q, \theta, \phi) := &- \sum_{m=1}^{M} \sum_{n=1}^{N_m} D_{KL}(q(z_{mn}|w_{mn})||p(z_{mn}|w_{mn}, \theta_m, \phi)) \\
&+ \sum_{m=1}^{M} \sum_{n=1}^{N_m} p(w_{mn}|\theta_m, \phi) + \sum_{k=1}^{K} \log p(\phi_k|\beta) + \sum_{m=1}^{M} \log p(\theta_m|\alpha)
\end{aligned} \tag{43}$$

B.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

$$\begin{aligned}
q(z_{mn} = k|w_{mn}) &= p(z_{mn} = k|w_{mn}, \theta_m, \phi) \\
&= \frac{\theta_{mk} \phi_{k w_{mn}}}{\sum_{k'=1}^{K} \theta_{mk'} \phi_{k' w_{mn}}}
\end{aligned} \tag{44}$$

For simplicity of notation, let us define

$$\boxed{q_{mnk} = \frac{\theta_{mk} \phi_{k w_{mn}}}{\sum_{k'=1}^{K} \theta_{mk'} \phi_{k' w_{mn}}}} \tag{45}$$

B.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:

$$\begin{aligned}\mathcal{L}(\theta, \phi, \lambda, \mu) &= \sum_{m=1}^{M} \sum_{m=1}^{N_m} \sum_{k=1}^{K} q(z_{mn} = k | w_{mn}) \log \frac{p(z_{mn} = k|\theta_m) p(w_{mn}|\phi_k)}{q(z_{mn} = k|w_{mn})} + \sum_{k=1}^{K} \log p(\phi_k|\beta) \\ &\quad + \sum_{m=1}^{M} \log p(\theta_m|\alpha) + \sum_{k=1}^{K} \lambda_k \left(1 - \sum_{v=1}^{V} \phi_{kv}\right) + \sum_{m=1}^{M} \mu_i \left(1 - \sum_{k=1}^{K} \theta_{mk}\right) \\ &= \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{k=1}^{K} q_{mnk} \log \theta_{mk} \phi_{k w_{mn}} + \sum_{k=1}^{K} \sum_{v=1}^{V} (\beta_v - 1) \log \phi_{kv} + \sum_{m=1}^{M} \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_{mk} \\ &\quad + \sum_{k=1}^{K} \lambda_k \left(1 - \sum_{v=1}^{V} \phi_{kv}\right) + \sum_{m=1}^{M} \mu_m \left(1 - \sum_{k=1}^{K} \theta_{mk}\right) + \text{const.}\end{aligned} \tag{46}$$

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

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta_{mk}} &= 0 = \sum_{j=1}^{N_m} \frac{q_{mnk} + \alpha_k - 1}{\theta_{mk}} - \mu_m \\ \mu_m \theta_{mk} &= \sum_{j=1}^{N_i} q_{mnk} + \alpha_k - 1\end{aligned} \tag{47}$$

After solving for μ_m, we finally obtain

$$\theta_{mk} = \frac{\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1}{\sum_{k'=1}^{K} \sum_{j=1}^{N_m} q_{mnk'} + \alpha_{k'} - 1} \tag{48}$$

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

$$\boxed{\theta_{mk} = \frac{1}{N_m + \sum(\alpha_{k'} - 1)} \left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right)} \tag{49}$$

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

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi_{kv}} &= 0 = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \frac{q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\phi_{kv}} - \lambda_k \\ \lambda_k \phi_{kv} &= \sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1\end{aligned} \tag{50}$$

After solving for λ_k, we finally obtain

$$\phi_{kv} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\sum_{v'=1}^{V} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \delta(v' - w_{mn}) + \beta_{v'} - 1} \tag{51}$$

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

$$\boxed{\phi_{kv} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} \delta(v - w_{mn}) + \beta_v - 1}{\sum_{m=1}^{M} \sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'} - 1)}} \tag{52}$$

B.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_mnk 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:

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{ij}}} \tag{53}$$

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

$$z_{mn} \sim \mathsf{Categorical}(q_{mn1}, ..., q_{mnK}) \tag{54}$$

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

$$\begin{aligned} \theta_{mk} &= \frac{D_{mk} + \alpha_k - 1}{N_m + \sum(\alpha_{k'} - 1)} \\ \phi_{kv} &= \frac{W_{kv} + \beta_v - 1}{T_k + \sum(\beta_{v'} - 1)} \end{aligned} \tag{55}$$

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
$T_k = \left|\left\{\, z_{mn} \mid z_{mn} = k \,\right\}\right| = \sum_{v=1}^{V} W_{kv}$.

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

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

C.1 EM for LDA

EM for LDA can be summarized by follows:

E-Step

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} \tag{56}$$

M-Step

$$\begin{aligned}\theta_{mk} &= \frac{1}{N_m + \sum(\alpha_{k'} - 1)}\left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right)\\ \phi_{kv} &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk}\delta(v - w_{mn}) + \beta_v - 1}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'} - 1)}\end{aligned} \tag{57}$$

C.2 GD for LDA

The joint probability density can be expressed as:

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

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

$$\begin{aligned}\log p(W, \theta, \phi|\alpha, \beta) &= \log\sum_{Z} p(W, Z, \theta, \phi|\alpha, \beta)\\ &= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\log\sum_{k=1}^{K} p(z_{mn} = k|\theta_m)p(w_{mn}|\phi_k)\\ &\qquad + \sum_{k=1}^{K}\log p(\phi_k|\beta) + \sum_{m=1}^{M}\log p(\theta_m|\alpha)\\ &= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\log\sum_{k=1}^{K}\theta_{mk}\phi_{kw_{mn}}\\ &\qquad + \sum_{m=1}^{M}\sum_{k=1}^{K}(\alpha_k - 1)\log\theta_{mk} + \sum_{k=1}^{K}\sum_{v=1}^{V}(\beta_v - 1)\log\phi_{kv}\end{aligned} \tag{59}$$

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

$$\begin{aligned}\frac{\partial\log p}{\partial\theta_{mk}} &= \sum_{j=1}^{N_m}\frac{\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} + \frac{\alpha_k - 1}{\theta_{mk}}\\ &= \frac{1}{\theta_{mk}}\left(\sum_{n=1}^{N_m} q_{mnk} + \alpha_k - 1\right)\end{aligned} \tag{60}$$

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

$$\begin{aligned}\frac{\partial \log p}{\partial \phi_{kv}} &= \sum_{m=1}^{M}\sum_{n=1}^{N_m} \frac{\theta_{mk}\delta(v-w_{mn})}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} + \frac{\beta_v - 1}{\phi_{kv}} \\ &= \frac{1}{\phi_{kv}}\left(\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk}\delta(v-w_{mn}) + \beta_v - 1\right)\end{aligned} \tag{61}$$

C.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:

$$\theta^+_m = \theta_m + \frac{1}{N_m + \sum(\alpha_{k'} - 1)}\left[\mathrm{diag}(\theta_m) - \theta_m\theta_m^T\right]\frac{\partial \log p}{\partial \theta_m} \tag{62}$$

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:

$$\phi^+_k = \phi_k + \frac{1}{\sum_{m=1}^{M}\sum_{n=1}^{N_m} q_{mnk} + \sum(\beta_{v'} - 1)}\left[\mathrm{diag}(\phi_k) - \phi_k\phi_k^T\right]\frac{\partial \log p}{\partial \phi_k} \tag{63}$$

C.4 SEM for LDA

We summarize our SEM derivation for LDA as follows:

E-Step

$$q_{mnk} = \frac{\theta_{mk}\phi_{kw_{mn}}}{\sum_{k'=1}^{K}\theta_{mk'}\phi_{k'w_{mn}}} \tag{64}$$

S-step

$$z_{mn} \sim \mathsf{Categorical}(q_{mn1}, ..., q_{mnK}) \tag{65}$$

M-step

$$\begin{aligned}\theta_{mk} &= \frac{D_{mk} + \alpha_k - 1}{N_m + \sum(\alpha_{k'} - 1)} \\ \phi_{kv} &= \frac{W_{kv} + \beta_v - 1}{T_k + \sum(\beta_{v'} - 1)}\end{aligned} \tag{66}$$

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.,

$$D_{mk} = \sum_{n=1}^{N_m} z_{mnk} \tag{67}$$

$$W_{kv} = \sum_{n=1}^{N_m} \sum_{m=1}^{N_d} z_{mnk}\delta(w_m = v) \tag{68}$$

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.

C.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.

D Non-singularity of Fisher Information for Mixture Models

Let us consider a general mixture model:

$$p(x|\theta,\phi) = \sum_{k=1}^{K} \theta_k f(x|\phi_k) \tag{69}$$

Then the log-likelihood can be written as:

$$\log p(x|\theta,\phi) = \log\left(\sum_{k=1}^{K} \theta_k f(x|\phi_k)\right) \tag{70}$$

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:

$$\begin{aligned} \frac{\partial}{\partial\theta_k}\log p(x|\theta,\phi) &= \frac{\partial}{\partial\theta_k}\log\left((\sum_{k=1}^{K}\theta_k f(x|\phi_k)\right) \\ &= \frac{f(x|\phi_k)}{\sum_{k'=1}^{K}\theta_{k'} f(x|\phi_{k'})} \\ \frac{\partial}{\partial\phi_k}\log p(x|\theta,\phi) &= \frac{\partial}{\partial\phi_k}\log\left((\sum_{k=1}^{K}\theta_k f(x|\phi_k)\right) \\ &= \frac{\theta_k \frac{\partial}{\partial\phi_k} f(x|\phi_k)}{\sum_{k'=1}^{K}\theta_{k'} f(x|\phi_{k'})} \end{aligned} \tag{71}$$

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

$$\sum_{k=1}^{K} u_k f(x|\phi_i) + v_k\theta_k\frac{\partial}{\partial\phi_k}f(x;\phi_k) = 0 \quad \forall x. \tag{72}$$

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.

E Alias Sampling Method

The alias sampling method is an efficient method for drawing samples from a K outcome discrete distribution in $O(1)$ amortized time and we describe it here for completeness. Denote by p_i for $i \in \{1 \dots K\}$ the probabilities of a distribution over K outcomes from which we would like to sample. If p were the uniform distribution, i.e. $p_i = K^{-1}$, then sampling would be trivial. For the general case, we must pre-process the distribution p into a table of K triples of the form (i, j, π_i) as follows:

- Partition the indices $\{1 \dots K\}$ into sets U and L where $p_i > K^{-1}$ for $i \in U$ and $p_i \leq K^{-1}$ for $i \in L$.
- Remove any i from L and j from U and add (i, j, p_i) to the table.
- Update $p_j = p_i + p_j - K^{-1}$ and if $p_j > K^{-1}$ then add j to U, else to L.

By construction the algorithm terminates after K steps; moreover, all probability mass is preserved either in the form of π_i associated with i or in the form of $K^{-1} - \pi_i$ associated with j. Hence, sampling from p can now be accomplished in constant time:

- Draw (i, j, π_i) uniformly from the set of k triples in K.
- With probability $K\pi_i$ emit i, else emit j.

Hence, if we need to draw from p at least K times, sampling can be accomplished in amortized $O(1)$ time.

F Applicability of ESCA

We begin with a simple Gaussian mixture model (GMM) with K components. Let $x_1, ..., x_n$ be *i.i.d.* observations, $z_1, ..., z_n$ be hidden component assignment variable and $\eta = \eta(\theta_1, ..., \theta_K, \mu_1, \Sigma_1, \mu_2, \Sigma_2, ..., \mu_K, \Sigma_K)$ be the parameters. Then the GMM fits into ESCA with sufficient statistics given by:

$$\begin{aligned} T(x_i, z_i) = [&\mathbb{1}\{z_i = 1\}, ..., \mathbb{1}\{z_i = K\}, \\ &x_i\mathbb{1}\{z_i = 1\}, ..., x_i\mathbb{1}\{z_i = K\}, \\ &x_i x_i^T\mathbb{1}\{z_i = 1\}, ..., x_i x_i^T\mathbb{1}\{z_i = K\}]. \end{aligned} \tag{73}$$

The conditional distribution for the E-step is:

$$p(z_i = k|x_i; \eta) \propto \theta_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \tag{74}$$

In the S-step we draw from this conditional distribution and the M-step, through inversion of link function, is:

$$\begin{aligned} \tilde{\theta}_k &= \frac{1}{n + K\alpha - K}\sum_{i=1}^n(\mathbb{1}\{z_i = k\} + \alpha - 1) \\ \tilde{\mu}_k &= \frac{\kappa_0\mu_0 + \sum_{i=1}^n x_i\mathbb{1}\{z_i = k\}}{\kappa_0 + \sum_{i=1}^n \mathbb{1}\{z_i = k\}} \\ \tilde{\Sigma}_k &= \frac{\Psi_0 + \kappa_0\mu_0\mu_0^T + \sum_{i=1}^n x_i x_i^T\mathbb{1}\{z_i = k\} - (\kappa_0 + \sum_{i=1}^n \mathbb{1}\{z_i = k\})\tilde{\mu}_k\tilde{\mu}_k^T}{\nu_0 + d + 2 + \sum_{i=1}^n \mathbb{1}\{z_i = k\}} \end{aligned} \tag{75}$$

and is only function of the sufficient statistics.

Next, we provide more details on how to employ ESCA for any conditional exponential family mixture model; i.e., in which n random variables x_i, $i = 1, \ldots, n$ correspond to observations, each distributed according to a mixture of K components, with each component belonging to the same exponential family of distributions (e.g., all normal, all multinomial, etc.), but with different parameters:

$$p(x_i|\phi) = \exp(\langle\psi(x_i), \phi\rangle - g(\phi)). \tag{76}$$

The model also has n latent variables z_i that specify the identity of the mixture component of each observation x_i, each distributed according to a K-dimensional categorical distribution. A set of K mixture weights θ_k, $k = 1, \ldots, K$, each of which is a probability (a real number between 0 and 1 inclusive) and collectively sum to one. A Dirichlet prior on the mixture weights with hyper-parameters α. A set of K parameters ϕ_k, $k = 1, \ldots, K$, each specifying the parameter of the corresponding mixture component. For example, observations distributed according to a mixture of one-dimensional Gaussian distributions will have a mean and variance for each component. Observations distributed according to a mixture of V-dimensional categorical distributions (e.g., when each observation is a word from a vocabulary of size V) will have a vector of V probabilities, collectively summing to 1. Moreover, we put a shared conjugate prior on these parameters:

$$p(\phi; n_0, \psi_0) = \exp\left(\langle\psi_0, \phi\rangle - n_0 g(\phi) - h(m_0, \psi_0)\right). \tag{77}$$

Then joint sufficient statistics would be given by:

$$\begin{aligned} T(z_i, x_i) = [&\mathbb{1}\{z_i = 1\}, ..., \mathbb{1}\{z_i = K\}, \\ &\psi(x_i)\mathbb{1}\{z_i = 1\}, ..., \psi(x_i)\mathbb{1}\{z_i = K\}] \end{aligned} \tag{78}$$

In the E-step of t^{th} iteration, we derive the conditional distribution $p(z_i|x_i, \eta)$, namely

$$\begin{aligned} p(z_i = k|x_i, \eta) &\propto p(x_i|\phi_k^{t-1}, z_i = k)p(z_i = k|\theta^{t-1}) \\ &= \frac{\theta_k^{t-1}p(x_i|\phi_k^{t-1})}{\sum_{k'}\theta_{k'}^{t-1}p(x_i|\phi_{k'}^{t-1})} \end{aligned} \tag{79}$$

In the S-step we draw z_i^t from this conditional distribution and the M-step through inversion of the link function yields:

$$\begin{aligned} \nabla g(\tilde{\phi}_k) &= \frac{\phi_0 + \sum_i \psi(x_i)\mathbb{1}\{z_i = k\})}{n_0 + \sum_i \mathbb{1}\{z_i = k\}} \\ \text{or} \qquad \tilde{\phi}_k &= \xi^{-1}\left(\frac{\psi_0 + \sum_i \psi(x_i)\mathbb{1}\{z_i = k\}}{n_0 + \sum_i \mathbb{1}\{z_i = k\}}\right) \\ \tilde{\theta}_k &= \frac{\sum_i \mathbb{1}\{z_i = k\} + \alpha_k - 1}{n + \sum_k \alpha_k - k} \ . \end{aligned} \tag{80}$$

This encompasses most of the popular mixture models (and with slight more work all the mixed membership or admixture models) with Binomial, multinomial, or Gaussian emission model, e.g. beta-binomials for identification, Dirichlet-multinomial for text or Gauss-Wishart for images as listed in Table 1.

Note further, ESCA is applicable to models such as restricted Boltzmann machines (RBMs) as well which are also in the exponential family. For example, if the data were a collection of images, each cell could independently compute the S-step for its respective image. For RBMs the cell would flip a biased coin for each latent variable, and for deep Boltzmann machines, the cells could perform Gibbs sampling.

To elabortate, consider 2-layer RBM (1 observed, 1 latent), then ESCA should work as it is. That is, we sample latent variables conditioned on data and weights. Then optimize weights, given latent variables and observed data. Now if we have deep RBM, i.e. one with many hidden layers. Then ESCA will have similar problem as Ising model. But there is a quick fix borrowing ideas from chromatic samplers.

```
for each iteration
    1. Sample all odd layers of the RBM
    2. Optimize for weights
    3. Sample all even layers of the RBM
    4. Optimize for weights
end for
```

We save a precise derivation and empirical evaluation for future work.

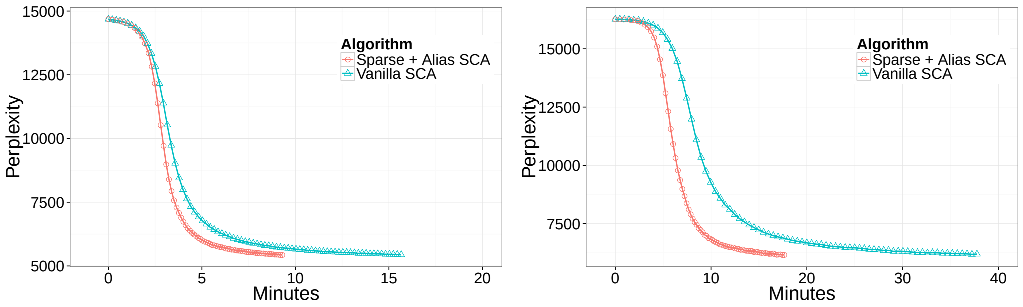
G More experimental results

In addition to the experiments reported in main paper, we perform another set of experiments. As before, to evaluate the strength and weaknesses of our algorithm, we compare against parallel and distributed implementations of CGS and CVB0.

Software & hardware All three algorithms were first implemented in the **Java** programming language. (We later switched to C++ for achieving better performance and those results are reported in the main paper.) To achieve good performance in the Java programming language, we use only arrays of primitive types and pre-allocate all of the necessary structures before the learning starts. We implement multithreaded parallelization within a node using the work-stealing Fork/Join framework, and the distribution across multiple nodes using the Java binding to OpenMPI. We also implemented a version of SCA 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 16 nodes connected through 10Gb/s Ethernet. Each node has two 8-core Intel Xeon E5 processors (some nodes have Ivy Bridge processors while others have Sandy Bridge processors) for a total of 32 hardware threads per node and 256GB of memory.

Datasets We experiment on two datasets, both of which are cleaned by removing stop words and rare words: Reuters RCV1 and English Wikipedia. Our Reuters dataset is composed of 806,791 documents comprising 105,989,213 tokens with a vocabulary of 43,962 vocabulary words. Our Wikipedia dataset is composed of 6,749,797 documents comprising 6,749,797 tokens with a vocabulary of 291,561 words. (Note this Wikipedia dump was collected at a different time than the main paper, hence different numbers.) We also apply the SCA algorithm to a third larger dataset composed of more than 3 billion documents comprising more than 171 billion tokens with a vocabulary of about 140,000 words.

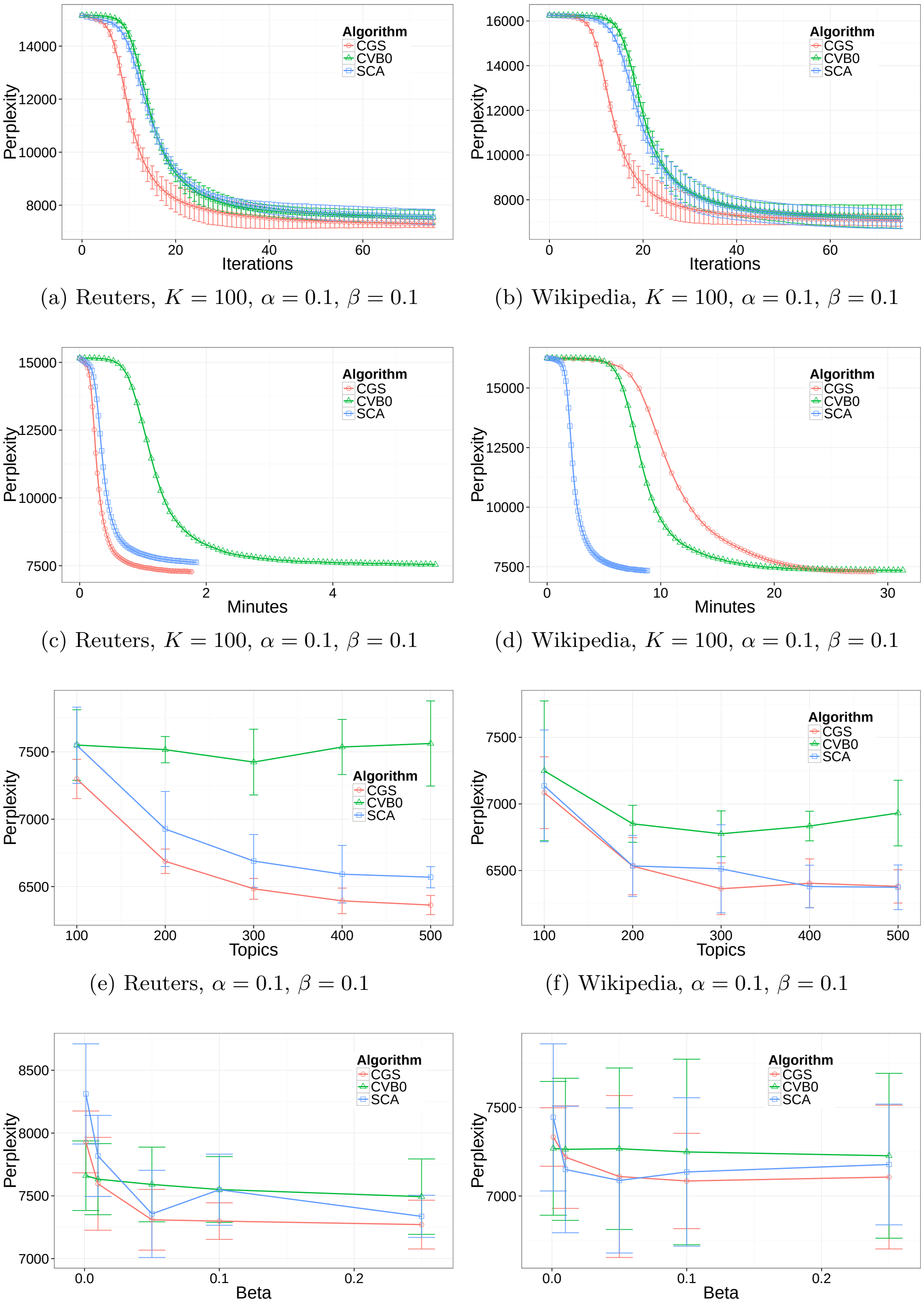
Protocol We use perplexity on held-out documents to compare the algorithms. When comparing algorithms trained on Wikipedia, we compute the perplexity of 10,000 Reuters documents. Vice versa, when comparing algorithms trained on Reuters, we compute the perplexity of 10,000 Wikipedia documents. We run four sets of experiment on each dataset: (1) how perplexity evolves for some numbers of training iterations (100 topics); (2) how perplexity evolves over time (100 topics); (3) perplexity as a function of the number of topics (75 iterations); and (4) perplexity as a function of the value of β (100 topics, 75 iterations). With the exception of the second experiment, we ran all experiments five times with five different seeds, and report the mean and standard deviation of these runs. The results are presented in Figure 6. We also ran an experiment to compare vanilla SCA and its improved version that uses a sparse representation and Vose's alias method for discrete sampling. The results are presented in Figure 5.



(a) Wikipedia, $K = 200$, $\alpha = 0.1$, $\beta = 0.1$

(b) Wikipedia, $K = 500$, $\alpha = 0.1$, $\beta = 0.1$

Figure 5: Evolution of perplexity over time for plain SCA and a sparse one using the alias method.



(a) Reuters, $K = 100$, $\alpha = 0.1$, $\beta = 0.1$

(b) Wikipedia, $K = 100$, $\alpha = 0.1$, $\beta = 0.1$

(c) Reuters, $K = 100$, $\alpha = 0.1$, $\beta = 0.1$

(d) Wikipedia, $K = 100$, $\alpha = 0.1$, $\beta = 0.1$

(e) Reuters, $\alpha = 0.1$, $\beta = 0.1$

(f) Wikipedia, $\alpha = 0.1$, $\beta = 0.1$

(g) Reuters, $K = 100$, $\alpha = 0.1$

(h) Wikipedia, $K = 100$, $\alpha = 0.1$

Figure 6: Evolution of perplexity on Wikipedia and Reuters over number of iterations, time, number of topics, value of β. Here SCA does not use alias method or sparsity and hence slower.

Topics

Here are the first five topics inferred via ESCA on LDA from both PubMed and Wikipedia:

PubMed				
Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
seizures	data	local	gene	state
epilepsy	information	block	transcript	change
seizure	available	lidocaine	exon	transition
epileptic	provide	anethesia	genes	states
temporal_lobe	regarding	anethetic	expression	occur
anticonvulsant	sources	acupuncture	region	process
convulsion	literature	bupivacaine	mrna	shift
kindling	concerning	anaesthesia	mouse	condition
partial	limited	under	expressed	changed
generalized	provided	anaesthetic	human	dynamic

Wikipedia				
Topic 0	Topic 1	Topic 2	Topic 3	Topic 4
hockey	medical	von	boy	music
ice	medicine	german	youth	music
league	hospital	karl	boys	pop
played	physician	carl	camp	music
junior	doctor	friedrich	girl	artists
nhl	clinical	wilhelm	scout	electronic
professional	md	johann	girls	duo
games	physicians	ludwig	guide	genre
playing	doctors	prussian	scouts	genres
national	surgeon	heinrich	scouting	musicians

Comparison

Table 3: Comparison with existing scalable LDA frameworks.

Method	Dataset	Number of Topics	Size of Vocabulary	Number of Documents	Number of Tokens	Infrastructure	Year	Processing Speed
YahooLDA [30]	PubMed	1K	140K	8.2M	797M	10 machines on hadoop	2010	12.87M tokens/s
lightLDA [39]	Bing "web chunk"	1000K	50K	1.2B	200B	24 machines (480 cores)	2014	60M tokens/s
F+LDA [38]	Amazon reviews	1K	1680K	29M	1.5B	32 machines (640 cores)	2014	110M tokens/s
ESCA	**100 copies of Wikipedia**	**1K**	**210K**	**667M**	**128B**	**8 Amazon c4.8x large (288 virtual cores)**	2015	**503M tokens/s**
ESCA	**100 copies of Wikipedia**	**1K**	**210K**	**667M**	**128B**	**20 Amazon c4.8x large (288 virtual cores)**	2015	**1200M tokens/s**

References

[1] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proc. Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, USA, 2009. AUAI Press.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[3] J. Canny. Gap: a factor model for discrete data. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129. ACM, 2004.

[4] Gilles Celeux and Jean Diebolt. The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, 2(1):73–82, 1985.

[5] Miklós Csűrös. Approximate counting with a floating-point counter. In M. T. Thai and Sartaj Sahni, editors, *Computing and Combinatorics (COCOON 2010)*, number 6196 in Lecture Notes in Computer Science, pages 358–367. Springer Berlin Heidelberg, 2010. See also http://arxiv.org/pdf/0904.3062.pdf.

[6] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, Beijing, China, July 2015. Association for Computational Linguistics.

[7] Donald A. Dawson. Synchronous and asynchronous reversible Markov systems. *Canadian mathematical bulletin*, 17:633–649, 1974.

[8] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[9] Anton K Formann and Thomas Kohlmann. Latent class analysis in medical research. *Statistical methods in medical research*, 5(2):179–211, 1996.

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

[11] Joseph Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel gibbs sampling: from colored fields to thin junction trees. In *International Conference on Artificial Intelligence and Statistics*, pages 324–332, 2011.

[12] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[13] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, May 2013.

[14] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999.

[15] Joel L. Lebowitz, Christian Maes, and Eugene R. Speer. Statistical mechanics of probabilistic cellular automata. *Journal of statistical physics*, 59:117–170, April 1990.

[16] Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. Reducing the sampling complexity of topic models. In *20th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, 2014.

[17] Pierre-Yves Louis. *Automates Cellulaires Probabilistes : mesures stationnaires, mesures de Gibbs associées et ergodicité*. PhD thesis, Université des Sciences et Technologies de Lille and il Politecnico di Milano, September 2002.

[18] Jean Mairesse and Irène Marcovici. Around probabilistic cellular automata. *Theoretical Computer Science*, 559:42–72, November 2014.

[19] David Mimno, Matt Hoffman, and David Blei. Sparse stochastic inference for latent dirichlet allocation. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 1599–1606, New York, NY, USA, July 2012. Omnipress.

[20] Robert Morris. Counting large numbers of events in small registers. *Commun. ACM*, 21(10):840–842, October 1978.

[21] R. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical Report 9815, University of Toronto, 1998.

[22] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[23] A. U. Neumann and B. Derrida. Finite size scaling study of dynamical phase transitions in two dimensional models: Ferromagnet, symmetric and non symmetric spin glasses. *J. Phys. France*, 49:1647–1656, 08 1988.

[24] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *J. Machine Learning Research*, 10:1801–1828, December 2009. http://dl.acm.org/citation.cfm?id=1577069.1755845.

[25] Søren Feodor Nielsen. The stochastic em algorithm: estimation and asymptotic results. *Bernoulli*, pages 457–489, 2000.

[26] Sam Patterson and Yee Whye Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

[27] B. Recht, C. Re, S.J. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In Peter Bartlett, Fernando Pereira, Richard Zemel, John Shawe-Taylor, and Kilian Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 693–701, 2011.

[28] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

[29] Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. Relationship between gradient and em steps in latent variable models.

[30] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endowment*, 3(1-2):703–710, September 2010.

[31] Whye Yee Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 19*, NIPS 2006, pages 1353–1360. MIT Press, 2007.

[32] Jean-Baptiste Tristan, Joseph Tassarotti, and Guy L. Steele Jr. Efficient training of LDA on a GPU by Mean-For-Mode Gibbs sampling. In *32nd International Conference on Machine Learning*, volume 37 of *ICML 2015*, 2015. Volume 37 of the Journal in Machine Learning Research: Workshop and Conference Proceedings.

[33] Gérard Y. Vichniac. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):96–116, January 1984.

[34] Michael D Vose. A linear algorithm for generating random numbers with a given distribution. *Software Engineering, IEEE Transactions on*, 17(9):972–975, 1991.

[35] Max A Woodbury, Jonathan Clive, and Arthur Garson. Mathematical typology: a grade of membership technique for obtaining disease definition. *Computers and biomedical research*, 11(3):277–298, 1978.

[36] Lei Xu and Michael I Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[37] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proc. 15th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, 2009. ACM.

[38] Hsiang-Fu Yu, Cho-Jui Hsieh, Hyokun Yun, SVN Vishwanathan, and Inderjit S Dhillon. A scalable asynchronous distributed algorithm for topic modeling. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1340–1350. International World Wide Web Conferences Steering Committee, 2015.

[39] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1351–1361. International World Wide Web Conferences Steering Committee, 2015.

[40] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad 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*, pages 879–888. ACM, 2012.