Stochastic Expectation Maximization with Variance Reduction

Jianfei Chen[†], **Jun Zhu**[†,*], **Yee Whye Teh**[‡] **and Tong Zhang**[§]
[†] Dept. of Comp. Sci. & Tech., BNRist Center, State Key Lab for Intell. Tech. & Sys.,
Institute for AI, THBI Lab, Tsinghua University, Beijing, 100084, China
[‡] Department of Statistics, University of Oxford
[§] Tencent AI Lab
{chenjian14@mails,dcszj@}.tsinghua.edu.cn
y.w.teh@stats.ox.ac.uk; tongzhang@tongzhang-ml.org

Abstract

Expectation-Maximization (EM) is a popular tool for learning latent variable models, but the vanilla batch EM does not scale to large data sets because the whole data set is needed at every E-step. Stochastic Expectation Maximization (sEM) reduces the cost of E-step by stochastic approximation. However, sEM has a slower asymptotic convergence rate than batch EM, and requires a decreasing sequence of step sizes, which is difficult to tune. In this paper, we propose a variance reduced stochastic EM (sEM-vr) algorithm inspired by variance reduced stochastic gradient descent algorithms. We show that sEM-vr has the same exponential asymptotic convergence rate as batch EM. Moreover, sEM-vr only requires a constant step size to achieve this rate, which alleviates the burden of parameter tuning. We compare sEM-vr with batch EM, sEM and other algorithms on Gaussian mixture models and probabilistic latent semantic analysis, and sEM-vr converges significantly faster than these baselines.

1 Introduction

Latent variable models are an important class of models due to their wide applicability across machine learning and statistics. Examples include factor analysis in psychology and the understanding of human cognition [32], hidden Markov models for modelling sequences, e.g. speech and language [29], and DNA [15], document and topic models [17, 4] and mixture models for density estimation and clustering [26]. Expectation Maximization (EM) [12] is a basic tool for maximum likelihood estimation for the parameters in latent variable models. It is an iterative algorithm with two steps: an E-step which calculates the expectation of sufficient statistics under the latent variable posteriors given the current parameters, and an M-step which updates the parameters given the expectations.

With the phenomenal growth in big data sets in recent years, the basic batch EM (bEM) algorithm in [12] is quickly becoming infeasible because the whole data set is needed at every E-step. Cappé and Moulines [6] proposed a stochastic EM (sEM) algorithm for exponential family models, which reduces the time complexity for the E-step by approximating the full-batch expectation with an exponential moving average over minibatches of data. sEM has been adopted in many applications including natural language processing [24], topic modeling [16, 14] and hidden Markov models [5]. However, sEM has a slow asymptotic convergence rate due to the high variance of each update. Unlike the original batch EM (bEM), which converges exponentially fast near a local optimum, the distance towards a local optimum only decreases at the rate $O(1/\sqrt{T})$ for sEM, where T is the

[*]corresponding author

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

number of iterations. Moreover, sEM requires a decreasing sequence of step sizes to converge. The decay rate of step sizes is often difficult to tune.

Recently, there has been much progress in accelerating stochastic gradient descent (SGD) by reducing the variance of the stochastic gradients, including SAG, SAGA and SVRG [22, 20, 11]. These algorithms achieve better convergence rates by utilizing infrequently computed batch gradients as control variates. Such ideas have also been brought into gradient-based Bayesian learning algorithms, including stochastic variational inference [25], as well as stochastic gradient Markov-chain Monte-Carlo [13, 8, 7] (SGMCMC).

In this paper, we develop a variance reduced stochastic EM algorithm (sEM-vr). In each epoch, that is, a full pass through the data set, our algorithm computes the full batch expectation as a control variate, and uses this to reduce the variance of minibatch updates in that epoch. Let E be the number of epochs and M be the number of minibatch iterations per epoch. We show that near a local optimum, our algorithm, with a constant step size, enjoys a convergence rate of $O((M^{-1}\log M)^{E/2})$ to the optimum. Like bEM, our convergence rate is exponential with respect to the number of epochs, and is asymptotically faster than sEM. We also show that our algorithm converges globally with a constant step size, under stronger assumptions. Note that leveraging variance reduction ideas in sEM is not straightforward, since sEM is not a stochastic gradient descent algorithm but rather a stochastic approximation [21] algorithm. In particular, the proof techniques we utilize are different than those in stochastic gradient descent algorithms. We demonstrate our algorithm on Gaussian mixture models and probabilistic latent semantic analysis [18]. sEM-vr achieves significantly faster convergence comparing with sEM, bEM, and other gradient-based and Bayesian algorithms.

2 Background

We review batch and stochastic EM algorithms in this section. Throughout the paper we focus on exponential family models with tractable E- and M-steps, which stochastic EM [6] is designed for.

2.1 EM Algorithm

The EM algorithm is designed for models with some observed variable x and hidden variable h. We assume an exponential family joint distribution $p(x,h;\theta) = b(x,h)\exp\{\eta(\theta)^\top\phi(x,h) - A(\theta)\}$ parameterized by θ. Given a data set of N ($\gg 1$) observations $X = \{x_i\}_{i=1}^N$, we want to obtain a maximum likelihood estimation (MLE) of the parameter θ, by maximizing the log marginal likelihood $\mathcal{L}(\theta) := \sum_{i=1}^N \log p(x_i;\theta) = \sum_{i=1}^N \log\int_{h_i} p(x_i,h_i;\theta)d\theta$, where the variables (x_i,h_i) are i.i.d. given θ. Denote $H = \{h_i\}_{i=1}^N$. Batch expectation-maximization (bEM) [12] optimizes the log marginal likelihood $\mathcal{L}(\theta)$ by constructing a lower bound of it:

$$\mathcal{L}(\theta) \geq Q(\theta;\hat{\theta}) - E_{p(H|X;\hat{\theta})}\left[\log p(H|X,\hat{\theta})\right], \tag{1}$$

$$Q(\theta;\hat{\theta}) := \mathbb{E}_{p(H|X;\hat{\theta})}\left[\log p(X,H;\theta)\right] = N\left(\eta(\theta)^\top F(\hat{\theta}) - A(\theta)\right) + \text{constant}, \tag{2}$$

where we define $F(\hat{\theta}) := \frac{1}{N}\sum_{i=1}^N f_i(\hat{\theta})$ as the full-batch expected sufficient statistics, and where $f_i(\hat{\theta}) := \mathbb{E}_{p(h_i|x_i;\hat{\theta})}\left[\phi(x_i,h_i)\right]$ is the expected sufficient statistics conditioned on observed datum x_i.

Let $\hat{\theta}_e$ be the estimated parameter at iteration or epoch e, where each epoch is a complete pass through the data set. In the E-step, bEM tightens the bound in Eq. (1) by setting $\hat{\theta} = \hat{\theta}_e$, and computes the expected sufficient statistics $F(\hat{\theta}_e)$. In the M-step, bEM finds a maximizer $\hat{\theta}_{e+1}$ of the lower bound with respect to θ, by solving the optimization problem $\operatorname{argmax}_\theta\{\eta(\theta)^\top F(\hat{\theta}) - A(\theta)\}$. The solution is denoted as $R(F(\hat{\theta}))$, and is assumed to be tractable. In summary, the bEM updates can be written simply as

$$\text{E-step: compute } F(\hat{\theta}_e), \qquad \text{M-step: let } \hat{\theta}_{e+1} = R(F(\hat{\theta}_e)). \tag{3}$$

The algorithm is also applicable to maximum *a posteriori* (MAP) estimation of parameters, with a conjugate prior $p(\theta;\alpha) = \exp\{\eta(\theta)^\top\alpha - A(\theta)\}$ with the hyperparameter α. Instead of $\mathcal{L}(\theta)$, MAP maximizes $\mathcal{L}(\theta) + \log p(\theta;\alpha) \geq N\eta(\theta)^\top\left(\alpha/N + F(\hat{\theta})\right) - NA(\theta) + \text{constant}$, and we still apply Eq. (3), but with $f_i(\hat{\theta}) := \alpha/N + \mathbb{E}_{p(h_i|x_i;\hat{\theta})}\left[\phi(x_i,h_i)\right]$ instead.

2.2 Stochastic EM Algorithm

When the data set is large, that is, N is large, computing $F(\hat{\theta}_t)$ in the E-step is too expensive because it needs a full pass though the entire data set. Stochastic EM (sEM) [6] avoids this by maintaining an exponentially moving average $\hat{s}_t$ as an approximation of the full average $F(\hat{\theta}_t)$. At iteration t, sEM picks a single random datum i, and updates:

$$\text{E step: } \hat{s}_{t+1} = (1-\rho_t)\hat{s}_t + \rho_t f_i(\hat{\theta}_t), \quad \text{M step: } \hat{\theta}_{t+1} = R(\hat{s}_{t+1}),$$

where (ρ_t) is a sequence of step sizes that satisfy $\sum_t \rho_t = \infty$ and $\sum_t \rho_t^2 < \infty$. We deliberately choose different iteration indices e and t for bEM and sEM to emphasize their different time complexity per iteration. In practice, sEM can take a minibatch of data instead of a single datum per iteration, but we stick to a single datum for cleaner presentation. The two sEM updates can be rolled into a single update

$$\hat{s}_{t+1} = (1-\rho_t)\hat{s}_t + \rho_t f_i(\hat{s}_t). \tag{4}$$

where for simplicity we have overloaded the notation with $f_i(s) := f_i(R(s))$. This first maps s, which can be interpreted as the estimated mean parameter of the model, into the parameters $\theta = R(s)$, before computing the required expected sufficient statistics $f_i(\theta)$ under the posterior given observation x_i. Which of the two definitions should be clear from the type of its argument and we feel this helps reduce notational burden on the reader. We similarly overload $F(s) := F(R(s))$ and $\mathcal{L}(s) := \mathcal{L}(R(s))$ accordingly, so we can also write bEM updates (Eq. 3) as simply $\hat{s}_{e+1} = F(\hat{s}_e)$. Intuitively, we want to find a stationary point s_* under bEM iterations, i.e., $s_* = F(s_*)$. We can view bEM as a fixed-point algorithm, and sEM as a Robbins-Monro [30] algorithm to solve the equation $s_* = F(s_*)$.

Because of the cheap updates, sEM can converge faster than bEM on large data sets in the beginning. However, due to the variance of the estimator $\hat{s}_t$, sEM has a slower asymptotic convergence rate than bEM for finite data sets. Specifically, let $s_* = F(s_*)$ be a stationary point, Cappe and Monlines [6] showed that $\mathbb{E}\left\|\hat{s}_T - s_*\right\|^2 = O(\rho_T)$ for sEM, which is at best $O(T^{-1})$ since $\sum_t \rho_t = \infty$. In contrast, Dempster et al. [12] showed that bEM converges as $\left\|\hat{s}_E - s_*\right\|^2 \le (1-\lambda)^{-2E}\left\|\hat{s}_0 - s_*\right\|$, where $1-\lambda \in [0,1)$ is a constant that is defined in Sec. 3.3. As long as the data set is finite, the exponential rate of bEM is faster than sEM. [2] Moreover, sEM needs a decreasing sequence of step sizes to converge, whose decay rate is difficult to tune.

3 Variance Reduced Stochastic Expectation Maximization

In this section, we describe a variance reduced stochastic EM algorithm (sEM-vr), and develop the theory for its convergence. sEM-vr enjoys an exponential convergence rate with a constant step size.

3.1 Algorithm Description

We run the algorithm for E epochs and M minibatch iterations per epoch, so that there are $T := ME$ iterations in total. For simplicity we choose $M = N$ and use minibatches of size 1, though our analysis is not limited to this case. Each epoch has the same time complexity as bEM. We index iteration t in epoch e as e, t. Let $\hat{s}_{e,t}$ be the estimated sufficient statistics at iteration e, t. Starting from the initial estimate $\hat{s}_{0,0}$, sEM-vr performs the following updates in epoch e,

Stochastic EM with Variance Reduction

1. Compute $F(\hat{s}_{e,0})$, and save $F(\hat{s}_{e,0})$ as well as $\hat{s}_{e,0}$
2. For each iteration $t = 1, \ldots, M$, randomly sample a datum i, and update

$$\hat{s}_{e,t+1} = (1-\rho)\hat{s}_{e,t} + \rho\left[f_i(\hat{s}_{e,t}) - f_i(\hat{s}_{e,0}) + F(\hat{s}_{e,0})\right]. \tag{5}$$

3. Let $\hat{s}_{e+1,0} = \hat{s}_{e,M}$.

Let $\mathbb{E}_{e,t}$ and $\mathrm{Var}_{e,t}$ be the expectation and variance over the random index i in iteration e, t. Comparing Eq. (5) with Eq. (4), we observe that the sEM and sEM-vr updates have the same expectation

[2] Without affecting the convergence rates, we slightly adjust the convergence theorems in [6, 12] to view them in a uniformed way, see Appendix A for details.

$\mathbb{E}_t[\hat{s}_{t+1}] = (1-\rho)\hat{s}_t + \rho F(\hat{s}_t)$. However their variances are different: sEM has $\text{Var}_t[\hat{s}_{t+1}] = \rho_t^2 \text{Var}_t[f_i(\hat{s}_t)]$, while sEM-vr has $\text{Var}_{e,t}[\hat{s}_{e,t+1}] = \rho^2 \text{Var}_{e,t}[f_i(\hat{s}_{e,t}) - f_i(\hat{s}_{e,0})]$. If the algorithm converges, i.e., the sequence $(\hat{s}_{e,t})$ converges to a point s_*, and $f_i(\cdot)$ is continuous, the variance of sEM-vr will converge to zero, while that of sEM will remain positive. Therefore, sEM-vr has asymptotically smaller variance than sEM, and we will see that this leads to better asymptotic convergence rates.

The time complexity of sEM-vr per epoch is the same as bEM and sEM, with a constant factor up to 3, for computing $f_i(\hat{s}_{e,t})$, $f_i(\hat{s}_{e,0})$ and $F(\hat{s}_{e,0})$. The space complexity also has a constant factor up to 3, for storing $\hat{s}_{e,0}$ and $F(\hat{s}_{e,0})$ along with $\hat{s}_{e,t}$. In practice, the difference is less than 3 times because the time and space costs for other aspects of the methods are the same, e.g. data storage.

3.2 Related Works

A possible alternative to sEM is Titterington's online algorithm [33], which replaces the exact M-step with a gradient ascent step to optimize $Q(\theta; \hat{\theta})$, where the gradient is multiplied with the inverse Fisher information of $p(x, h; \theta)$. Titterington's algorithm is locally equivalent to sEM [6]. However, as argued by Cappé and Moulines [6], Titterington's algorithm has several issues, including the Fisher information being expensive to compute in high dimensions, the need for explicit matrix inversion, and that the updated parameters are not guaranteed to be valid. Moreover, leveraging variance reduced stochastic gradient algorithms [20, 22, 11] for Titterington's algorithm is not straightforward as the Fisher information matrix changes with θ. Zhu et al. has proposed a variance reduced stochastic gradient EM algorithm [39]. There are also some theoretical analysis of EM algorithm for high dimensional data [3, 35].

Instead of performing point estimation of parameters, Bayesian inference algorithms, including variational inference (VI) and Markov-chain Monte-Carlo (MCMC), can also be adopted, to infer the posterior distribution of parameters. Variance reducing techniques have also been applied to these settings, including smoothed stochastic variational inference (SSVI) [25] and variance reduced stochastic gradient MCMC (VRSGMCMC) algorithms [13, 8, 7]. However, convergence guarantees for SSVI have not been developed, while VRSGMCMC algorithms are typically much slower than sEM-vr due to the intrinsic randomness of MCMC. For example, the time complexity to converge to an ϵ-precision in terms of the 2-Wasserstein distance of the true posterior and the MCMC distribution is $O(N + \kappa^{3/2}\sqrt{d}/\epsilon)$, where κ is a condition number and d is the dimensionality of the parameters [7].

3.3 Local Convergence Rate

We analyze the local convergence rate of a sequence $\{\hat{s}_{e,t}\}$ of sEM-vr iterates to a stationary point s_* with $s_* = F(s_*)$. Let $\theta_* := R(s_*)$ be the natural parameter corresponding to the mean parameter s_*.

Theorem 1. *If*

(a) The Hessian $\nabla^2 \mathcal{L}(\theta_*)$ is negative definite, i.e., θ_* is a strict local maximum of $\mathcal{L}(\theta_*)$.
(b) $\forall i$, $f_i(s)$ is L_f-Lipschitz continuous, and $F(s)$ is β_f-smooth.
(c) $\forall e, t, \|\hat{s}_{e,t} - s_*\| < \lambda/\beta_f$, where $1-\lambda$ is the maximum eigenvalue of $J_* := \partial F(s_*)/\partial s_*$.

Then, for any step size $\rho \leq \lambda/(32L_f^2)$, we have

$$\mathbb{E}\left\|\hat{s}_{E,0} - s_*\right\|^2 \leq \left[\exp\left(-M\lambda\rho/4\right) + 32L_f^2\rho/\lambda\right]^E \left\|\hat{s}_{0,0} - s_*\right\|^2. \tag{6}$$

In particular, if $\rho = \rho_* := 4\log(M/\kappa^2)/(\lambda M)$, where $\kappa^2 := 128L_f^2/\lambda^2$, then we have

$$\mathbb{E}\left\|\hat{s}_{E,0} - s_*\right\|^2 \leq \left[\left(1 + \log(M/\kappa^2)\right)\kappa^2/M\right]^E \left\|\hat{s}_{0,0} - s_*\right\|^2. \tag{7}$$

Remarks. Assumption (a) follows directly from the original EM paper (Theorem 4) [12]. [12] analyzed the convergence only in an infinitesimal neighbourhood of s_*, while Assumption (c) gives an explicit radius of convergence. Assumption (b) is new and required to control the variance and radius of convergence. Note also that we analyse the convergence of the mean parameters, while [12] analysed that for parameters. However they are equivalent if $R(s)$ is Lipschitz continuous. In Appendix A.1 we show that negative definite $\nabla^2 \mathcal{L}(\theta_*)$ in Assumption (a) implies that $\lambda > 0$ in Assumption (c).

Proof. We first analyze the convergence behavior at a specific epoch e, and omit the epoch index $_e$ for concise notations. We further denote $\Delta_t := \hat{s}_t - s_*$ for any t. By Eq. (5),

$$\begin{aligned}\mathbb{E}_t \left\|\Delta_{t+1}\right\|^2 &= \mathbb{E}_t \left\|(1-\rho)\hat{s}_t + \rho F(\hat{s}_t) - s_* + \rho\left[f_i(\hat{s}_t) - f_i(\hat{s}_0) - F(\hat{s}_t) + F(\hat{s}_0)\right]\right\|^2 \\ &= \left\|(1-\rho)\hat{s}_t + \rho F(\hat{s}_t) - s_*\right\|^2 + \rho^2 \mathbb{E}_t \left\|f_i(\hat{s}_t) - f_i(\hat{s}_0) - F(\hat{s}_t) + F(\hat{s}_0)\right\|^2, \end{aligned} \tag{8}$$

where the second equality is due to $\mathbb{E}_t\left[f_i(\hat{s}_{e,t}) - f_i(\hat{s}_{e,0}) + F(\hat{s}_{e,0})\right] = F(\hat{s}_{e,t})$. We have

$$\begin{aligned}&\left\|(1-\rho)\hat{s}_t + \rho F(\hat{s}_t) - s_*\right\|^2 = \left\|(1-\rho)\Delta_t + \rho(F(\hat{s}_t) - s_*) + \rho J_*\Delta_t - \rho J_*\Delta_t\right\|^2 \\ \le& \left[\, \left\|(1-\rho)\Delta_t + \rho J_*\Delta_t\right\| + \rho\left\|F(\hat{s}_t) - s_* - J_*\Delta_t\right\| \,\right]^2 \\ \le& \left[(1-\rho\lambda)\left\|\Delta_t\right\| + (\rho/2)\beta_f\left\|\Delta_t\right\|^2\right]^2 = \left[1 - \rho\left(\lambda - \beta_f\left\|\Delta_t\right\|/2\right)\right]^2\left\|\Delta_t\right\|^2 \\ \le& \left(1-\rho\lambda/2\right)^2\left\|\Delta_t\right\|^2 \le \left(1-\rho\lambda/2\right)\left\|\Delta_t\right\|^2, \end{aligned} \tag{9}$$

where the second line utilizes triangular inequality, the third line utilizes $\|(1-\rho)I + \rho J_*\| \le 1 - \rho + \rho(1-\lambda) = 1 - \rho\lambda$,where $\|\cdot\|$ is the ℓ_2 operator norm, and the smoothness in (b), which implies $\|F(\hat{s}_t) - s_* - J_*(\hat{s}_t - s_*)\| \le (\beta_f/2)\left\|\hat{s}_t - s_*\right\|^2$. The last line utilizes (c).

By (b), F is L_f-Lipschitz and $\forall i$, $f_i - F$ is $2L_f$-Lipschitz continuous. Therefore

$$\mathbb{E}_t \left\|f_i(\hat{s}_t) - f_i(\hat{s}_0) - F(\hat{s}_t) + F(\hat{s}_0)\right\|^2 \le 4L_f^2\left\|\hat{s}_t - \hat{s}_0\right\|^2 \le 8L_f^2(\left\|\Delta_t\right\|^2 + \left\|\Delta_0\right\|^2). \tag{10}$$

Combining Eq. (8, 9, 10), and utilizing our assumption $\rho \le \lambda/(32L_f^2)$, we have

$$\mathbb{E}\left\|\Delta_{t+1}\right\|^2 \le \left(1 - \rho\lambda/2 + 8\rho^2L_f^2\right)\left\|\Delta_t\right\|^2 + 8\rho^2L_f^2\left\|\Delta_0\right\|^2 \le \left(1-\rho\lambda/4\right)\left\|\Delta_t\right\|^2 + 8\rho^2L_f^2\left\|\Delta_0\right\|^2.$$

We get Eq. (6, 7) by analyzing the sequence $a_{t+1} \le (1-\epsilon\rho)a_t + c\rho^2 a_0$, where $a_t = \mathbb{E}\left\|\Delta_t\right\|^2$, $\epsilon = \lambda/4$ and $c = 8L_f^2$. The analysis is in Appendix B. □

Comparison with bEM: As mentioned in Sec. 2.2, bEM has $\mathbb{E}\left\|\hat{s}_E - s_*\right\|^2 \le (1-\lambda)^{2E}\left\|\hat{s}_0 - s_*\right\|^2$. The distance decreases exponentially for both bEM and sEM-vr, but at different speeds. If M is large, sEM-vr (Eq. 7) converges much faster than bEM because $\left(1 + \log(M/\kappa^2)\right)\kappa^2/M \ll (1-\lambda)^2$, thanks to its cheap stochastic updates.

Comparison with sEM: As mentioned in Sec. 2.2, sEM has $\mathbb{E}\left\|\hat{s}_T - s_*\right\|^2 = O(T^{-1})$, which is not exponential, and is asymptotically slower than sEM-vr. The key difference is we can bound the variance term for sEM-vr by $\left\|\hat{s}_t - \hat{s}_0\right\|^2$ in Eq. (10), so the variance goes to zero as $(\hat{s}_{e,t})$ converges. The advantage of sEM-vr over sEM is especially significant when E is large. Moreover, sEM requires a decreasing sequence of step sizes to converge [6], which is more difficult to tune comparing with the constant step size of sEM-vr.

3.4 Global Convergence

Theorem 1 only considers the case near a local maximum of the log marginal likelihood. We now show that under stronger assumptions, there exists a constant step size, such that sEM-vr can globally converge to a stationary point $s_* = F(s_*)$, one with $\nabla\mathcal{L}(s_*) = 0$ [12].

Theorem 2. *Suppose*

(a) The natural parameter function $\eta(\theta)$ is L_η-Lipschitz, and $f_i(s)$ is L_f-Lipschitz for all i,
(b) for any x and h, $\log p(x, h; \theta)$ is γ-strongly-concave w.r.t. θ.

Then for any constant step size $\rho < \gamma/\left(M(M-1)L_\eta L_f\right)$, sEM-vr converges to a stationary point, starting from any valid sufficient statistics vector $\hat{s}_{0,0}$.

A sufficient condition for (b) is the exponential family is canonical, i.e., $\eta(\theta) = \theta$, and we want the MAP estimation instead of MLE, where the prior $\log p(\theta)$ is γ-strongly-concave. We leave the proof in Appendix C. The idea is first show that sEM-vr is a generalized EM (GEM) algorithm [36], which improves $\mathbb{E}[Q(\theta; \hat{\theta})]$ after each epoch, and then apply Wu's convergence theorem for GEM [36].

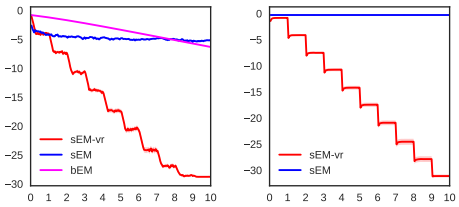


Figure 1: Toy Gaussian Mixture. Left: $\log_{10} \mathbb{E} \left\| \hat{\mu}_t - \mu_* \right\|^2$, Right: $\log_{10} \mathrm{Var}_t[\hat{\mu}_t]/\rho_t^2$, X-axis: number of epochs.

Data set	D	V	$\lvert\mathcal{I}\rvert$
NIPS [1]	1.5k	12k	1.93m
NYTimes [1]	0.3m	102k	99m
Wiki [38]	3.6m	8k	524m
PubMed [1]	8.1m	141k	731m

Table 1: Statistics of datasets for pLSA. k=thousands, m=millions.

4 Applications and Experiments

We demonstrate the application of sEM-vr on a toy Gaussian mixture model and probabilistic latent semantic analysis.

4.1 Toy Gaussian Mixture

We fit a mixture of two Gaussians, $p(x|\mu) = 0.2\mathcal{N}(\mu, 1) + 0.8\mathcal{N}(-\mu, 1)$, with a single unknown parameter μ. Let $X = \{x_i\}_{i=1}^N$ be the data set, and $h_i \in \{1, 2\}$ be the cluster assignment of x_i. We write $h_{ik} := \mathbb{I}(h_i = k)$ as a shortcut, where $\mathbb{I}(\cdot)$ is the indicator function. The joint likelihood is $p(X, H|\mu) \propto \exp\{\sum_i \sum_k h_{ik} \log \mathcal{N}(x_i; \mu_k, 1)\} \propto \exp\{\sum_i \eta(\mu)^\top \phi(x_i, h_i)\}$, where the natural parameter $\eta(\mu) = (\mu, -\mu, -\mu^2/2, \mu^2/2)$ and the sufficient statistics $\phi(x_i, h_i) = (x_i h_{i1}, x_i h_{i2}, h_{i1}, h_{i2})$. Let $\gamma_{ik}(\mu) = p(h_i = k|x_i, \mu) \propto \pi_i \mathcal{N}(x_i; \mu_k, 1)$ for $k \in \{1, 2\}$ be the posterior probabilities. The expected sufficient statistics $f_i(\mu) = \mathbb{E}_{p(h_i, x_i|\mu)} \phi(x_i, h_i) = (x_i \gamma_{i1}(\mu), x_i \gamma_{i2}(\mu), \gamma_{i1}(\mu), \gamma_{i2}(\mu))$, and $F(\mu) = 1/N \sum_i f_i(\mu)$. The mapping from sufficient statistics to parameters is $R(s) = (s_1 - s_2)/(s_3 - s_4)$. bEM, sEM, and sEM-vr updates are then defined respectively as Eq. (3), Eq. (4), and Eq. (5).

We construct a dataset of $N = 10,000$ samples drawn from the model with $\mu = 0.5$, and run bEM until convergence (to double precision) to obtain the MLE μ_*. We then measure the convergence of $\mathbb{E} \left\| \hat{\mu}_t - \mu_* \right\|^2$ as well as the variance term $\mathrm{Var}_t[\hat{\mu}_t]/\rho_t^2$ for bEM, sEM, and sEM-vr with respective to the number of epochs. $\mathrm{Var}_t[\hat{\mu}_t]$ is always quadratic with respect to the step size ρ_t, so we divide it by ρ_t^2 to cancel the effect of the step size, and just study the intrinsic variance. We tune the step size manually, and set $\rho_t = 3/(t + 10)$ for sEM and $\rho = 0.003$ for sEM-vr.

The result is shown as Fig. 1. sEM converges faster than bEM in the first 8 epochs, and then it is outperformed by bEM, because sEM is asymptotically slower, as mentioned in Sec. 2.2. The convergence curve of sEM-vr exhibits a staircase pattern. In the beginning of each epoch it converges very fast because $\|\hat{s}_{e,t} - \hat{s}_{e,0}\|$ is small, so the variance is small. The variance then becomes larger and the convergence slows down. Then we start a new epoch and compute a new $F(\hat{s}_{e,0})$, so that the convergence is fast again. On the other hand, the variance of sEM remains constant.

4.2 Probabilistic Latent Semantic Analysis

4.2.1 Model and Algorithm

Probabilistic Latent Semantic Analysis (pLSA) [18] represents text documents as mixtures of topics. pLSA takes a list $\mathcal{I}$ of tokens, where each token i is represented by a pair of document and word IDs (d_i, v_i), that indicates for the presence of a word v_i in document d_i. Denote $[n] = \{1, \ldots, n\}$, we have $d_i \in [D]$ and $v_i \in [V]$. pLSA assigns a latent topic $z_i \in [K]$ for each token, and defines the joint likelihood as $p(\mathcal{I}, Z|\boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{i \in \mathcal{I}} \mathrm{Cat}(z_i; \theta_{d_i}) \mathrm{Cat}(v_i; \phi_{z_i})$, with the parameters $\boldsymbol{\theta} = \{\theta_d\}_{d=1}^D$ and $\boldsymbol{\phi} = \{\phi_k\}_{k=1}^K$. We have priors $p(\theta_d) = \mathrm{Dir}(\theta_d; K, \alpha')$ and $p(\phi_k) = \mathrm{Dir}(\phi_k; V, \beta')$, where $\mathrm{Dir}(K, \alpha)$ is a K-dimensional symmetric Dirichlet distribution with the concentration parameter α, and find an MAP estimation $\mathrm{argmax}_{\boldsymbol{\theta}, \boldsymbol{\phi}} \log \sum_Z p(W, Z|\boldsymbol{\theta}, \boldsymbol{\phi}) + \log p(\boldsymbol{\theta}) + \log p(\boldsymbol{\phi})$. Only the updates are presented here and the derivation is in Appendix D. Let $\gamma_{ik}(\boldsymbol{\theta}, \boldsymbol{\phi}) := p(z_i = k|v_i, \boldsymbol{\theta}, \boldsymbol{\phi}) \propto \theta_{d_i,k} \phi_{k,v_i}$ be the posterior topic assignment of the token v_i, bEM updates $\gamma_{dk}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{i \in \mathcal{I}_d} \gamma_{ik}(\boldsymbol{\theta}, \boldsymbol{\phi})$, and

$\gamma_{kv}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{i \in \mathcal{I}_v} \gamma_{ik}(\boldsymbol{\theta}, \boldsymbol{\phi})$ in E-step, where $\mathcal{I}_d = \{(d_i, v_i) | d_i = d\}$ and $\mathcal{I}_v = \{(d_i, v_i) | v_i = v\}$. M-step is $\theta_{dk} = (\gamma_{dk} + \alpha)/(\sum_k \gamma_{dk} + K\alpha)$, and $\phi_{kv} = (\gamma_{kv} + \beta)/(\sum_v \gamma_{kv} + V\beta)$, where $\alpha = \alpha' - 1$ and $\beta = \beta' - 1$. We distinguish $(\gamma_{ik}, \gamma_{dk}, \gamma_{vk})$ and $(\mathcal{I}, \mathcal{I}_d, \mathcal{I}_v)$ by indices for simplicity.

sEM approximates the full batch expected sufficient statistics γ_{dk} and γ_{kv} with exponential moving averages $\hat{s}_{t,d,k}$ and $\hat{s}_{t,k,v}$ at iteration t, and updates $\hat{s}_{t+1,d,k} = (1-\rho_t)\hat{s}_{t,d,k} + \rho_t \frac{|\mathcal{I}|}{|\hat{\mathcal{I}}|} \sum_{i \in \hat{\mathcal{I}}_d} \gamma_{ik}(\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{\phi}}_t)$, and $\hat{s}_{t+1,k,v} = (1-\rho_t)\hat{s}_{t,k,v} + \rho_t \frac{|\mathcal{I}|}{|\hat{\mathcal{I}}|} \sum_{i \in \hat{\mathcal{I}}_v} \gamma_{ik}(\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{\phi}}_t)$, where we sample a minibatch $\hat{\mathcal{I}} \subset \mathcal{I}$ of tokens per iteration, $\hat{\mathcal{I}}_d, \hat{\mathcal{I}}_v$ are defined in the same way as $\mathcal{I}_d, \mathcal{I}_v$. $\hat{\boldsymbol{\theta}}_t$ and $\hat{\boldsymbol{\phi}}_t$ are computed in the M-step with $\hat{s}_{t,d,k}$ and $\hat{s}_{t,k,v}$. This sEM algorithm is known as SCVB0 [16].

sEM-vr updates as $\hat{s}_{e,t+1,d,k} = (1-\rho)\hat{s}_{e,t,d,k} + \rho \frac{|\mathcal{I}|}{|\hat{\mathcal{I}}|} \sum_{i \in \hat{\mathcal{I}}_d} (\gamma_{ik}(\hat{\boldsymbol{\theta}}_{e,t}, \hat{\boldsymbol{\phi}}_{e,t}) - \gamma_{ik}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})) + \rho\gamma_{dk}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})$, and $\hat{s}_{e,t+1,k,v} = (1-\rho)\hat{s}_{e,t,k,v} + \rho \frac{|\mathcal{I}|}{|\hat{\mathcal{I}}|} \sum_{i \in \hat{\mathcal{I}}_v} (\gamma_{ik}(\hat{\boldsymbol{\theta}}_{e,t}, \hat{\boldsymbol{\phi}}_{e,t}) - \gamma_{ik}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})) + \rho\gamma_{kv}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})$, where $\gamma_{dk}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})$ and $\gamma_{kv}(\hat{\boldsymbol{\theta}}_{e,0}, \hat{\boldsymbol{\phi}}_{e,0})$ is computed by bEM per epoch. We have pseudocode for sEM and sEM-vr in Appendix D.

If $\boldsymbol{\theta}$ is integrated out instead of maximized, we recover an MAP estimation [14] of latent Dirichlet allocation (LDA) [4]. Many existing algorithms for LDA actually optimize the pLSA objective as an approximation of the LDA objective, including CVB0 [2, 31, 19], SCVB0 [16], BP-LDA [10], ESCA [37], and WarpLDA [9]. This approximation works well in practice when the number of topics is small [2]. We have more discussions in Appendix D.1.

4.2.2 Experimental Settings

We compare sEM-vr with bEM and sEM (SCVB0), which is the start-of-the-art algorithm for pLSA, on four datasets listed in Table 1. We also compare with two gradient based algorithms, stochastic mirror descent (SMD) [10] and reparameterized stochastic gradient descent (RSGD) as well as their variants with SVRG-style [20] variance reduction, denoted as SMD-vr and RSGD-vr, despite their convergence properties are unknown. Both SMD and RSGD replace the M-step with a stochastic gradient step. SMD updates as $\theta_{dk} \propto \theta_{dk} \exp\{\rho \nabla_{\theta_{dk}} Q\}$ and $\phi_{kv} \propto \phi_{kv} \exp\{\rho \nabla_{\phi_{kv}} Q\}$, where Q is defined as Eq. (1). RSGD adopts the reparameterization $\theta_{dk} = \frac{\exp \lambda_{dk}}{\sum_k \exp \lambda_{dk}}$ and $\phi_{kv} = \frac{\exp \tau_{kv}}{\sum_v \exp \tau_{kv}}$, and directly optimize Q w.r.t. λ and τ by stochastic gradient descent. Derivations of SMD and RSGD are in Appendix D.6. All the algorithms are implemented in C++, and are highly-optimized and parallelized. The testing machine has two 12-core Xeon E5-2692v2 CPUs and 64GB main memory.

We assess the convergence of algorithms by the training objective $\log p(W|\theta, \phi) + \log p(\theta|\alpha') + \log p(\phi|\beta')$, i.e., logarithm of unnormalized posterior distribution $p(\theta, \phi|W, \alpha', \beta')$. For each dataset and the number of topics $K \in \{50, 100\}$, we first select the hyperparameters by a grid search $K\alpha \in \{0.1, 1, 10, 100\}$ and $\beta \in \{0.01, 0.1, 1\}$.[3] Then, we do another grid search to choose the step size. For sEM-vr, we choose $\rho \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$, and for all other stochastic algorithms, we set $\rho_t = a/(t + t_0)^\kappa$, and choose $a \in \{10^{-7}, \dots, 10^0\}, t_0 \in \{10, 100, 1000\}$ and $\kappa \in \{0.5, 0.75, 1\}$.[4] Finally, we repeat 5 runs with difference random seeds for each algorithm with its best step size. E is 20 for NIPS and NYTimes, and 5 for Wiki and PubMed. M is 50 for NIPS and 500 for all the other datasets.

4.2.3 Results for pLSA

We plot the training objective against running time as first and second row of Fig. 2. We find that gradient-based algorithms and bEM are not competitive with sEM and sEM-vr, so we only report their results on NIPS, to make the distinction sEM and sEM-vr more clear. Full results and more explanations of the slow convergence of gradient-based algorithms are available in Appendix D.6. Due to the reduced variance, sEM-vr consistently converges faster to better training objective than sEM and bEM on all the datasets, while the constant step size of sEM-vr is easier to tune than the decreasing sequence of step sizes for sEM.

[3] We find that all the algorithms have the same best hyperparameter configuration.

[4] We have tried constant step sizes for SMD-vr and RSGD-vr but found it worse than decreasing step sizes.

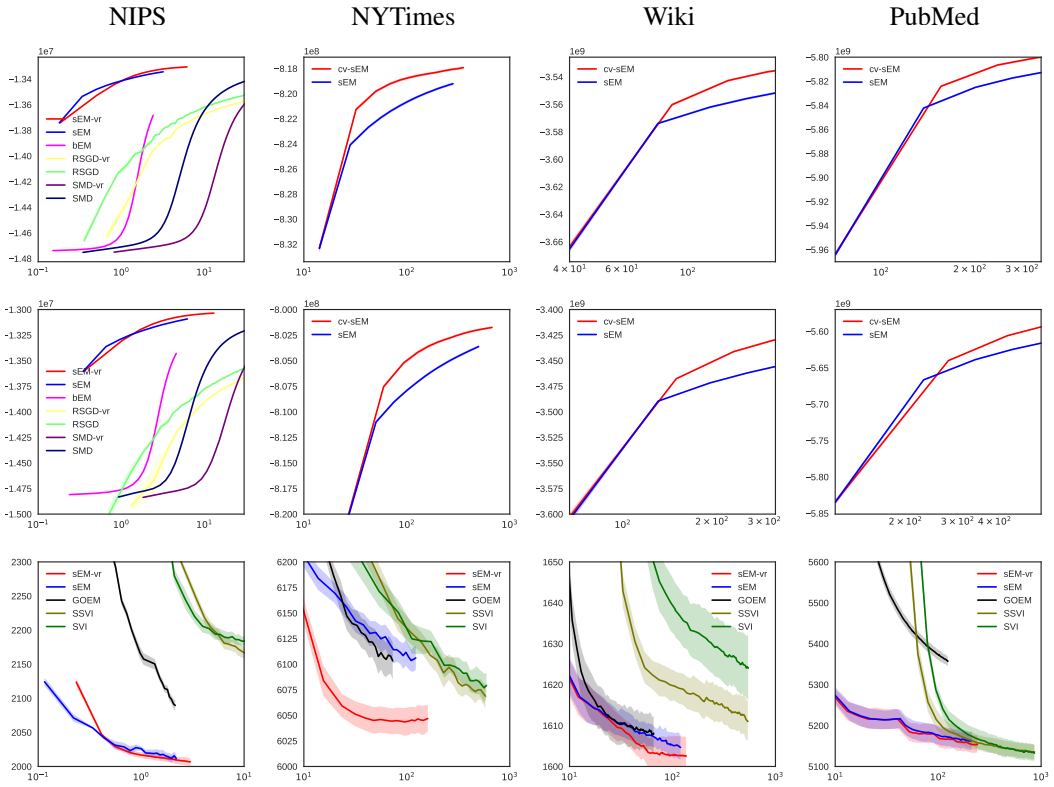


Figure 2: pLSA and LDA convergence results. X-axis is running time in seconds. First and second row: pLSA with $K = 50$ and $K = 100$, y-axis is the training objective. Third row: LDA with $K = 10$, y-axis is the testing perplexity.

4.3 Results for LDA

As discussed in Sec. 4.2.1, algorithms for pLSA also work well as approximate training algorithms for LDA, if the number of topics is small. Therefore, we also evaluate our sEM-vr algorithm for LDA, with a small number of $K = 10$ topics. The training algorithm is exactly the same, but the evaluation metric is different. We hold out a small testing set, and report the testing perplexity, computed by the left-to-right algorithm [34] on the testing set. We compare with a state-of-the-art algorithm, Gibbs online expectation maximization (GOEM) [14], which outperforms a wide range of algorithms including SVI [17], hybrid variational-Gibbs [27], and SGRLD [28]. We also compare with stochastic variational inference (SVI) [17] and its variance reduced variant SSVI [25].

The third row of Fig. 2 shows the results. We observed that sEM-vr converges the fastest on all the datasets except NIPS, where sEM converges faster due to its cheaper iterations. sEM-vr always gets better results than sEM in the end. GOEM converges slower due to its high Monte-Carlo variance. SVI and SSVI converge slower due to their inexact mean field assumption and expensive iterations, including an inner loop for inferring the local latent variables and frequent evaluation of the expensive digamma function. For a larger number of topics, such as 100, we find that GOEM performs the best since it does not approximate LDA as pLSA, and does not make mean field assumptions as SVI and SSVI. Extending our algorithm to variational EM and Monte-Carlo EM, when the E-step is not tractable, is an interesting future direction.

5 Conclusions and Discussions

We propose a variance reduced stochastic EM (sEM-vr) algorithm. sEM-vr achieves a $\big(1 + \log(M/\kappa^2)\big)^{-E}$ local convergence rate, which is faster than both the $(1 - \lambda)^{-2E}$ rate of batch EM and $O(T^{-1})$ rate of plain stochastic EM (sEM). Unlike sEM, which requires a decreasing sequence of step sizes to converge, sEM-vr only requires a constant step size to achieve this local

convergence rate as well as global convergence, under stronger assumptions. We compare sEM-vr against bEM, sEM and other gradient and Bayesian algorithms, on GMM and pLSA tasks, and find that sEM-vr converges significantly faster than these alternatives.

An interesting future direction is leveraging recent progress on variance reduced stochastic gradient descent for non-convex optimization [23] to relax our assumptions on strongly-log-concavity, and extend sEM-vr to stochastic control variates, which works better on very large data sets. Extending our work to variational EM and Monte-Carlo EM is also interesting.

Acknowledgments

We thank Chris Maddison, Adam Foster, and Jin Xu for proofreading. J.C. and J.Z. were supported by the National Key Research and Development Program of China (No.2017YFA0700904), NSFC projects (Nos. 61620106010, 61621136008, 61332007), the MIIT Grant of Int. Man. Comp. Stan (No. 2016ZXFB00001), Tsinghua Tiangong Institute for Intelligent Computing, the NVIDIA NVAIL Program and a Project from Siemens. YWT was supported by funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071, and from Tencent AI Lab through the Oxford-Tencent Collaboration on Large Scale Machine Learning.

References

[1] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[2] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press, 2009.

[3] Sivaraman Balakrishnan, Martin J Wainwright, Bin Yu, et al. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Olivier Cappé. Online em algorithm for hidden markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749, 2011.

[6] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[7] Niladri S Chatterji, Nicolas Flammarion, Yi-An Ma, Peter L Bartlett, and Michael I Jordan. On the theory of variance reduction for stochastic gradient monte carlo. *arXiv preprint arXiv:1802.05431*, 2018.

[8] Changyou Chen, Wenlin Wang, Yizhe Zhang, Qinliang Su, and Lawrence Carin. A convergence analysis for a class of practical variance-reduction stochastic gradient mcmc. *arXiv preprint arXiv:1709.01180*, 2017.

[9] Jianfei Chen, Kaiwei Li, Jun Zhu, and Wenguang Chen. Warplda: a cache efficient o (1) algorithm for latent dirichlet allocation. *Proceedings of the VLDB Endowment*, 9(10):744–755, 2016.

[10] Jianshu Chen, Ji He, Yelong Shen, Lin Xiao, Xiaodong He, Jianfeng Gao, Xinying Song, and Li Deng. End-to-end learning of lda by mirror-descent back propagation over a deep architecture. In *Advances in Neural Information Processing Systems*, pages 1765–1773, 2015.

[11] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[12] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[13] Kumar Avinava Dubey, Sashank J Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient langevin dynamics. In *Advances in neural information processing systems*, pages 1154–1162, 2016.

[14] Christophe Dupuy and Francis Bach. Online but accurate inference for latent variable models with local gibbs sampling. *The Journal of Machine Learning Research*, 18(1):4581–4625, 2017.

[15] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[16] James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. Stochastic collapsed variational bayesian inference for latent dirichlet allocation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 446–454. ACM, 2013.

[17] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[18] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[19] Katsuhiko Ishiguro, Issei Sato, and Naonori Ueda. Averaged collapsed variational bayes inference. *Journal of Machine Learning Research*, 18(1):1–29, 2017.

[20] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[21] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[22] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

[23] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156, 2017.

[24] Percy Liang and Dan Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619. Association for Computational Linguistics, 2009.

[25] Stephan Mandt and David Blei. Smoothed gradients for stochastic variational inference. In *Advances in Neural Information Processing Systems*, pages 2438–2446, 2014.

[26] Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[27] David Mimno, Matt Hoffman, and David Blei. Sparse stochastic inference for latent dirichlet allocation. *arXiv preprint arXiv:1206.6425*, 2012.

[28] 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.

[29] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[30] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[31] Issei Sato and Hiroshi Nakagawa. Rethinking collapsed variational bayes inference for lda. In *ICML*, 2012.

[32] Charles Spearman and L. W. Jones. *Human Ability*. Macmillan, 1950.

[33] D Michael Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 257–267, 1984.

[34] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112. ACM, 2009.

[35] Zhaoran Wang, Quanquan Gu, Yang Ning, and Han Liu. High dimensional expectation-maximization algorithm: Statistical optimization and asymptotic normality. *arXiv preprint arXiv:1412.8729*, 2014.

[36] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[37] Manzil Zaheer, Michael Wick, Jean-Baptiste Tristan, Alex Smola, and Guy Steele. Exponential stochastic cellular automata for massively parallel inference. In *Artificial Intelligence and Statistics*, pages 966–975, 2016.

[38] Aonan Zhang, Jun Zhu, and Bo Zhang. Sparse online topic models. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1489–1500. ACM, 2013.

[39] Rongda Zhu, Lingxiao Wang, Chengxiang Zhai, and Quanquan Gu. High-dimensional variance-reduced stochastic gradient expectation-maximization algorithm. In *International Conference on Machine Learning*, pages 4180–4188, 2017.

A Some Clarifications

A.1 Convergence rate of bEM

Dempster et al. [12] showed in their Theorem 4 that the convergence rate of bEM is

$$\left\|\hat{\theta}_E - \theta_*\right\|^2 \leq (1-\lambda)^{-2E} \left\|\hat{\theta}_0 - \theta_*\right\|^2,$$

where $1-\lambda$ is the maximum eigenvalue of $\partial R(F(\theta_*))/\partial\theta_*$. We define $1-\lambda$ in Sec. 3.3 as the maximum eigenvalue of $\partial F(s_*)/\partial s_*$. The two definitions are equivalent because at the stationary point (θ_*, s_*), we have $\theta_* = R(s_*)$ and $s_* = F(\theta_*)$. Note that for two matrices A and B, AB and BA have the same spectrum. Therefore, $\partial F(s_*)/\partial s_* = \partial F(R(s_*))/\partial s_* = (\partial F/\partial\theta_*)(\partial R/\partial s_*)$ has the same spectrum with $\partial R(F(\theta_*))/\partial\theta_* = (\partial R/\partial s_*)(\partial F/\partial\theta_*)$, so $1-\lambda$ is the maximum eigenvalue of both $\partial F(s_*)/\partial s_*$ and $\partial R(F(\theta_*))/\partial\theta_*$.

Dempster et. al [12] also showed that $\partial R(F(\theta_*))/\partial\theta_* = (I_* - \nabla^2\mathcal{L}(\theta_*))^{-1} I_*$, where $I_* = -\mathbb{E}_{p(H|X,\theta_*)}\nabla^2 \log p(H|X,\theta_*) \succeq 0$ is the Fisher information of $p(H|X,\theta_*)$, $A \succeq B$ means $A-B$ is positive semidefinite, and $A \succ B$ means $A-B$ is positive definite. If $0 \succ \nabla^2\mathcal{L}(\theta_*)$, as we assumed in Theorem 1, then $I_* - \nabla^2\mathcal{L}(\theta_*) \succ I_* \succeq 0$, and the eigenvalues of $\partial R(F(\theta_*))/\partial\theta_*$ are between $[0,1)$, so $\lambda > 0$.

Finally, the convergence of the sequence of parameters $(\hat{\theta}_t)$ and sufficient statistics $(\hat{s}_t)$ are equivalent as long as the mappings between them, $R(s)$ and $F(\theta)$, are Lipschitz continuous.

A.2 Convergence rate of sEM

Cappe and Moulines [6] showed in their Theorem 2 that the sequence $\rho_T^{-1/2}(\hat{\theta}_T - \theta_*)$ converge in distribution to $\mathcal{N}(0, \Sigma(\theta_*))$, where $\Sigma(\theta_*)$ is irrelevant with ρ_T. This implies $\rho_T^{-1}\left\|\hat{\theta}_T - \theta_*\right\|^2 \to \Sigma(\theta_*)$, that is $\left\|\hat{\theta}_T - \theta_*\right\|^2 = O(\rho_T)$. Finally, we convert the convergence of $(\hat{\theta}_t)$ to the convergence of $(\hat{s}_t)$ as mentioned in Sec. A.1.

B Remaining Proof of Theorem 1

Proof. We continue the analysis in Sec. 3.3 of the sequence $a_{t+1} \leq (1-\epsilon\rho)a_t + c\rho^2 a_0$, where $a_t = \mathbb{E}\left\|\Delta_t\right\|^2$, $\epsilon = \lambda/4$ and $c = 8L_f^2$. We have

$$\begin{aligned}
a_M &\leq (1-\epsilon\rho)a_{M-1} + c\rho^2 a_0 \\
&\leq (1-\epsilon\rho)^M a_0 + c\rho^2\left[1 + (1-\epsilon\rho) + \cdots + (1-\epsilon\rho)^{M-1}\right] a_0 \\
&\leq \exp(-M\epsilon\rho)a_0 + c\rho^2 \frac{1-(1-\epsilon\rho)^M}{\epsilon\rho} a_0 \\
&\leq \left[\exp(-M\epsilon\rho) + \frac{c\rho}{\epsilon}\right] a_0 := A_M, \qquad (11)
\end{aligned}$$

where the third line utilizes the inequality $1 + x \leq \exp(x), \forall x \in \mathbb{R}$. Taking derivative of the upper bound A_M w.r.t. ρ, we have

$$(A_M)'_\rho = \left[-M\epsilon\exp(-M\epsilon\rho) + \frac{c}{\epsilon}\right] a_0.$$

Let the derivative be zero, we obtain the optimal upper bound and its corresponding ρ, denoted as ρ_*

$$\rho_* = \log\left(\frac{\epsilon^2 M}{c}\right)/(\epsilon M), \qquad (12)$$

$$a_M \leq \frac{c}{\epsilon^2 M}\left(1 + \log\frac{\epsilon^2 M}{c}\right) a_0. \qquad (13)$$

Plugging $a_t = \mathbb{E}\left\|\Delta_t\right\|^2$, $\epsilon = \lambda/4$ and $c = 8L_f^2$ into Eq. (11, 12, 13), we have

$$\begin{aligned}
\mathbb{E}\left\|\Delta_M\right\|^2 &\leq \left[\exp\left(-M\lambda\rho/4\right) + 32L_f^2\rho/\lambda\right]\left\|\Delta_0\right\|^2 \\
\rho_* &= 4\log(M/\kappa^2)/(\lambda M) = 4\log\left(\lambda^2 M/(128L_f^2)\right)/(\lambda M) \\
\mathbb{E}\left\|\Delta_M\right\|^2 &\leq \left[\left(1+\log(M/\kappa^2)\right)\kappa^2/M\right]\left\|\Delta_0\right\|^2,
\end{aligned}$$

where $\kappa^2 = \frac{c}{\epsilon^2} = \frac{128L_f^2}{\lambda^2}$. We can verify that $\rho_* = 4\log\left(\lambda^2 M/(128L_f^2)\right)/(\lambda M)$ is less equal than $\lambda/(32L_f^2)$, assumed by Theorem 1 because $\log x < x$ for all $x > 0$, where $x = \lambda^2 M/(128L_f^2)$.

Finally, because we take $\hat{s}_{E+1,0} = \hat{s}_{E,M}$, we get Eq. (6, 7). □

C Proof of Theorem 2

We construct an auxiliary function

$$\hat{Q}_{e,t}(\theta) = N\left(\eta(\theta)^\top \hat{s}_{e,t} - A(\theta)\right),$$

and its equivalent recursive definition

$$\begin{aligned}
\hat{Q}_{e,t+1}(\theta) &= (1-\rho)\hat{Q}_{e,t}(\theta) + \rho(Q_i(\theta;\hat{\theta}_{e,t}) - Q_i(\theta;\hat{\theta}_{e,0}) + Q(\theta;\hat{\theta}_{e,0})), \\
\hat{Q}_{0,0}(\theta) &= Q(\theta;\hat{\theta}_{0,0}),
\end{aligned}$$

where $Q(\theta;\hat{\theta}_{e,0})$ is defined in Eq. (1), $Q_i(\theta;\hat{\theta}_{e,0}) = \mathbb{E}_{p(h_i|x_i,\hat{\theta}_{e,0})}[\log p(x_i,h_i;\theta)] = \eta(\theta)^\top f_i(\hat{\theta}_{e,0}) - A(\theta)$, and $\hat{\theta}_{e,t} := \operatorname{argmax}_\theta \hat{Q}_{e,t}(\theta) = R(\hat{s}_{e,t})$. This is similar to the original from of sEM [6] rather than its exponential family form we present in the main text.

According to Assumption (b), $\log p(x_i,h_i;\theta)$ is γ-strongly-concave, so $Q_i(\theta;\hat{\theta}) = \mathbb{E}_{p(h_i|x_i,\hat{\theta})}[\log p(x_i,h_i;\theta)]$ is also γ-strongly-concave with respect to θ. By induction, $\mathbb{E}_{e,t}[\hat{Q}_{e,t+1}(\theta)] = (1-\rho)\hat{Q}_{e,t}(\theta) + \rho Q(\theta;\hat{\theta}_{e,t})$ is also γ-strongly-concave for all e and t.

By the recursive formulation, we have

$$\begin{aligned}
Q(\hat{\theta}_{e,t+1};\hat{\theta}_{e,0}) - Q(\hat{\theta}_{e,t};\hat{\theta}_{e,0}) =& \frac{1}{\rho}\left(\hat{Q}_{e,t+1}(\hat{\theta}_{e,t+1}) - \hat{Q}_{e,t+1}(\hat{\theta}_{e,t})\right) + \frac{1-\rho}{\rho}\left(\hat{Q}_{e,t}(\hat{\theta}_{e,t}) - \hat{Q}_{e,t}(\hat{\theta}_{e,t+1})\right) \\
&+ Q_i(\theta_{e,t};\hat{\theta}_{e,t}) - Q_i(\theta_{e,t+1};\hat{\theta}_{e,t}) + Q_i(\theta_{e,t+1};\hat{\theta}_{e,0}) - Q_i(\theta_{e,t};\hat{\theta}_{e,0}).
\end{aligned}$$

According to the definition of $\hat{\theta}_{e,t}$, and assuming that the algorithm has not converged, we have

$$\begin{aligned}
\hat{Q}_{e,t+1}(\hat{\theta}_{e,t+1}) - \hat{Q}_{e,t+1}(\hat{\theta}_{e,t}) > 0, \\
\hat{Q}_{e,t}(\hat{\theta}_{e,t}) - \hat{Q}_{e,t}(\hat{\theta}_{e,t+1}) > 0,
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\mathbb{E}_{e,t}[Q_i(\hat{\theta}_{e,t};\hat{\theta}_{e,t}) - Q_i(\hat{\theta}_{e,t+1};\hat{\theta}_{e,t}) + Q_i(\hat{\theta}_{e,t+1};\hat{\theta}_{e,0}) - Q_i(\hat{\theta}_{e,t};\hat{\theta}_{e,0})]. \\
=&\mathbb{E}_{e,t}[\eta(\hat{\theta}_{e,t})^\top f_i(\hat{\theta}_{e,t}) - \eta(\hat{\theta}_{e,t+1})^\top f_i(\hat{\theta}_{e,t}) + \eta(\hat{\theta}_{e,t+1})^\top f_i(\hat{\theta}_{e,0}) - \eta(\hat{\theta}_{e,t})^\top f_i(\hat{\theta}_{e,0})] \\
=&\left(\eta(\hat{\theta}_{e,t}) - \eta(\hat{\theta}_{e,t+1})\right)^\top \left(F(\hat{\theta}_{e,t}) - F(\hat{\theta}_{e,0})\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{e,t}[Q(\hat{\theta}_{e,t+1};\hat{\theta}_{e,0}) - Q(\hat{\theta}_{e,t};\hat{\theta}_{e,0})] \\
>&\frac{1}{\rho}\left(\hat{Q}_{e,t+1}(\hat{\theta}_{e,t+1}) - \hat{Q}_{e,t+1}(\hat{\theta}_{e,t})\right) + \left(\eta(\hat{\theta}_{e,t}) - \eta(\hat{\theta}_{e,t+1})\right)^\top \left(F(\hat{\theta}_{e,t}) - F(\hat{\theta}_{e,0})\right) \\
\geq&\frac{\gamma}{2\rho}\left\|\hat{\theta}_{e,t+1} - \hat{\theta}_{e,t}\right\|^2 - L_\eta L_f \left\|\hat{\theta}_{e,t} - \hat{\theta}_{e,t+1}\right\| \left\|\hat{\theta}_{e,t} - \hat{\theta}_{e,0}\right\|, \qquad (14)
\end{aligned}$$

where the last line utilizes the γ-strong-concavity of $\hat{Q}_{e,t+1}$ (recall that $\nabla \hat{Q}_{e,t+1}(\hat{\theta}_{e,t+1}) = 0$, according to the definition of $\hat{\theta}_{e,t+1}$) as well as Lipschitz continuity of η and f_i. Summing up Eq. (14), we have

$$\begin{aligned}
&\mathbb{E}[Q(\hat{\theta}_{e,M}; \hat{\theta}_{e,0}) - Q(\hat{\theta}_{e,0}; \hat{\theta}_{e,0})] \\
>&\frac{\gamma}{2\rho} \sum_{t=0}^{M-1} \left\| \hat{\theta}_{e,t+1} - \hat{\theta}_{e,t} \right\|^2 - L_\eta L_f \sum_{t=0}^{M-1} \left\| \hat{\theta}_{e,t} - \hat{\theta}_{e,t+1} \right\| \left\| \hat{\theta}_{e,t} - \hat{\theta}_{e,0} \right\| \\
\geq&\frac{\gamma}{2\rho} \Delta_e^2 - M(M-1) L_\eta L_f \Delta_e^2 / 2,
\end{aligned}$$

where $\Delta_e := \max_t \left\| \hat{\theta}_{e,t+1} - \hat{\theta}_{e,t} \right\|$. Therefore, when $\rho < \frac{\gamma}{M(M-1)L_\eta L_f}$, we have $\mathbb{E}[Q(\hat{\theta}_{e,M}; \hat{\theta}_{e,0}) - Q(\hat{\theta}_{e,0}; \hat{\theta}_{e,0})] > 0$ for any $\hat{\theta}_{e,0}$ and $\hat{\theta}_{e,M}$. That is, sEM-vr improves the lower bound of the log marginal likelihood $\mathcal{L}$ in each epoch. Hence sEM-vr can be considered as a generalized EM (GEM) algorithm [36], which improves the ELBO in every epoch. Applying Wu's Theorem 1 [36], we conclude that sEM-vr converges globally to a stationary point.

D Details of Probabilistic Latent Semantic Analysis

In pLSA, we want to model a collection of D documents $W = \{w_d\}_{d=1}^D$, where each document $w_d = \{w_{dn}\}_{n=1}^{N_d}$ is a list of tokens, and each token $w_{dn} \in \{1, \dots, V\}$ is represented by its ID in a vocabulary of V words. The notations here is different with the main text, but rather similar with the SCVB0 paper [16].

We define the following generative procedure of the documents:

1. for each topic $k \in [K]$, generate $\phi_k \sim \text{Dir}(V, \beta')$;
2. for each document $d \in [D]$, generate $\theta_d \sim \text{Dir}(K, \alpha')$;
3. for each position $d \in [D], n \in [N_d]$, generate $z_{dn} \sim \text{Cat}(\theta_d)$, generate $w_{dn} \sim \text{Cat}(\phi_{z_{dn}})$,

where $[K] := \{1, \dots, K\}$, $\text{Dir}(K, \alpha)$ is a K-dimensional symmetric Dirichlet distribution with the concentration parameter α, and $\text{Cat}(\cdot)$ is a categorical distribution. This is exactly the same generative procedure with latent Dirichlet allocation (LDA) [4].

Denote Z, θ and ϕ to be the collection of z_{dn}, θ_d and ϕ_k, the generative procedure defines a joint distribution $p(W, Z, \theta, \phi | \alpha', \beta')$. Our goal is a maximum *a posteriori* (MAP) estimate of the parameters (θ, ϕ).

$$\begin{aligned}
&\operatorname*{argmax}_{\theta,\phi} \log p(\theta, \phi | W, \alpha', \beta') \\
=&\operatorname*{argmax}_{\theta,\phi} \log \sum_Z \left\{ p(W, Z | \theta, \phi) p(\theta | \alpha') p(\phi | \beta') \right\}. \qquad (15)
\end{aligned}$$

Let $\alpha = \alpha' - 1$ and $\beta = \beta' - 1$, we have

$$\begin{aligned}
p(W, Z | \theta, \phi) p(\theta | \alpha') p(\phi | \beta') \propto& \prod_{dn} \theta_{d, z_{dn}} \phi_{z_{dn}, w_{dn}} \prod_{dk} \theta_{dk}^{\alpha} \prod_{kv} \phi_{kv}^{\beta}. \\
=& \prod_{dk} \theta_{dk}^{C_{dk} + \alpha} \prod_{kv} \phi_{kv}^{C_{kv} + \beta} \\
=& \exp \left\{ \sum_{dk} (C_{dk} + \alpha) \log \theta_{dk} + \sum_{kv} (C_{kv} + \beta) \log \phi_{kv} \right\}, \qquad (16)
\end{aligned}$$

where $C_{dk} = \sum_n \mathbb{I}(z_{dn} = k)$ and $C_{kv} = \sum_{dn} \mathbb{I}(z_{dn} = k) \mathbb{I}(w_{dn} = v)$, and $\mathbb{I}(\cdot)$ is the indicator function. Eq. (16) is in an exponential family form where (C_{dk}, C_{kv}) are the sufficient statistics, and $(\log \theta_{dk}, \log \phi_{kv})$ are the natural parameters.

Then,

$$\begin{aligned}
Q(\theta, \phi; \theta', \phi') =& \mathbb{E}_{p(Z | W, \theta', \phi')} \left[\log p(W, Z | \theta, \phi) \right] + \log p(\theta | \alpha') + \log p(\phi | \beta') + \text{const.} \\
=& \sum_{dk} \left(\gamma_{dk}(\theta', \phi') + \alpha \right) \log \theta_{dk} + \sum_{kv} \left(\gamma_{kv}(\theta', \phi') + \beta \right) \log \phi_{kv}, \qquad (17)
\end{aligned}$$

where const. is a constant term w.r.t. θ and ϕ,

$$\gamma_{dnk}(\theta, \phi) = \mathbb{E}_{p(z_{dn}|w_{dn},\theta,\phi)}[\mathbb{I}(z_{dn} = k)] = p(z_{dn} = k|w_{dn}, \theta, \phi) = \frac{\theta_{dk}\phi_{k,w_{dn}}}{\sum_k \theta_{dk}\phi_{k,w_{dn}}}.$$

and

$$\gamma_{dk}(\theta, \phi) := \mathbb{E}_{p(Z|W,\theta,\phi)}[C_{dk}] = \sum_n \gamma_{dnk}(\theta, \phi), \tag{18}$$

$$\gamma_{kv}(\theta, \phi) := \mathbb{E}_{p(Z|W,\theta,\phi)}[C_{kv}] = \sum_{dn} \mathbb{I}(w_{dn} = v)\gamma_{dnk}(\theta, \phi). \tag{19}$$

D.1 Connection with LDA

The only difference of our pLSA objective Eq. (15) with LDA [4] is whether treating θ as latent variable or parameter. If θ is marginalized out, we recover the LDA objective

$$\begin{aligned} &\operatorname*{argmax}_{\phi} \log p(\phi|W, \alpha', \beta') \\ =&\operatorname*{argmax}_{\phi} \log \sum_Z \int_\theta \{p(W, Z|\theta, \phi)p(\theta|\alpha')p(\phi|\beta')\}\, d\theta. \end{aligned} \tag{20}$$

Due to their resemblance, a number of LDA training algorithms optimizes the pLSA training objective Eq. (15) instead of the LDA training objective Eq. (20) for faster convergence, including CVB0 [2, 31, 19], SCVB0 [16], BP-LDA [10], ESCA [37], and WarpLDA [9]. This approximation works well in practice [2] when the number of topics is small.

D.2 E-step

In the E-step, we compute the expected sufficient statistics $\gamma_{dk}(\theta, \phi)$ and $\gamma_{kv}(\theta, \phi)$ as Eq. (18, 19).

D.3 M-step

In the M-step, we solve the maximization problem

$$\begin{aligned} &\operatorname*{argmax}_{\theta,\phi} \sum_{dk} (\gamma_{dk} + \alpha) \log \theta_{dk} + \sum_{kv} (\gamma_{kv} + \beta) \log \phi_{kv}. \\ \text{s.t. } &\sum_k \theta_{dk} = 1, \forall d \in [D], \\ &\sum_v \phi_{kv} = 1, \forall k \in [K]. \end{aligned} \tag{21}$$

The solution is

$$\theta_{dk} = \frac{\gamma_{dk} + \alpha}{\sum_k \gamma_{dk} + K\alpha}, \quad \phi_{kv} = \frac{\gamma_{kv} + \beta}{\sum_v \gamma_{kv} + V\beta}.$$

D.4 Stochastic EM Updates

According to Sec. 2.2, we can derive an sEM algorithm by replacing the E-step with stochastic approximation. sEM algorithm for pLSA is known as SCVB0 [16]. SCVB0 optimizes θ and ϕ alternatively. To optimize θ_d for a document d given ϕ, SCVB0 replaces γ_{dk}, the sum over all the tokens $n \in [N_d]$ (Eq. 18), with a stochastic approximation

$$\begin{aligned} \text{E step: } &\hat{s}_{t+1,d,k} = (1 - \rho_t)\hat{s}_{t,d,k} + \rho_t N_d \gamma_{dnk}(\theta_t, \phi), \quad n \sim \text{Uniform}(N_d), \\ \text{M step: } &\theta_{t+1,d,k} = (\hat{s}_{t+1,d,k} + \alpha)/(\sum_k \hat{s}_{t+1,d,k} + K\alpha), \end{aligned}$$

where $\hat{s}_{t,d,k}$ is an approximation of the batch sufficient statistics γ_{dk}, and $\theta_{t,d,k}$ is the estimated parameter at iteration t.

Algorithm 1 Batch E-step for PLSA.

Require: θ, ϕ, W
  $\forall d, k, \gamma_{dk} \leftarrow 0$
  $\forall k, v, \gamma_{kv} \leftarrow 0$
  for each document d **do**
    for each token w_{dn} **do**
      $\forall k, \gamma_{dnk} = \theta_{dk}\phi_{k,w_{dn}} / (\sum_k \theta_{dk}\phi_{k,w_{dn}})$
      $\forall k, \gamma_{dk} \leftarrow \gamma_{dk} + \gamma_{dnk}, \gamma_{k,w_{dn}} \leftarrow \gamma_{k,w_{dn}} + \gamma_{dnk}$
    end for
  end for
  Return γ_{dk}, γ_{kv}.

Algorithm 2 SCVB0 algorithm for PLSA.

Require: Initial θ, ϕ
  $\hat{s}_{d,k}, \hat{s}_{k,v} \leftarrow$ BatchEStep(θ, ϕ, W) (Alg. 1)
  for each minibatch of M documents **do**
    Compute the step size ρ
    (Update θ)
    for each document d **do**
      for each token w_{dn} **do**
        Compute $\forall k, \gamma_{dnk} = \theta_{dk}\phi_{k,w_{dn}} / (\sum_k \theta_{dk}\phi_{k,w_{dn}})$,
        E-step: $\forall k, \hat{s}_{d,k} \leftarrow (1-\rho)\hat{s}_{d,k} + \rho N_d \gamma_{dnk}$
        M-step: $\forall k, \theta_{dk} \leftarrow (\hat{s}_{d,k} + \alpha)/(N_d + K\alpha)$.
      end for
    end for
    (Update ϕ)
    $\forall k, v, \hat{s}_{kv} \leftarrow (1-\rho)\hat{s}_{kv}$
    for each document d **do**
      for each token w_{dn} **do**
        Compute $\forall k, \gamma_{dnk} = \theta_{dk}\phi_{k,w_{dn}} / (\sum_k \theta_{dk}\phi_{k,w_{dn}})$,
        E-step: $\forall k, \hat{s}_{k,w_{dn}} \leftarrow \hat{s}_{k,w_{dn}} + \rho \frac{D}{M} \gamma_{dnk}$
      end for
    end for
    M-step: $\forall k, v, \phi_{kv} \leftarrow (\hat{s}_{kv} + \beta)/(\sum_v \hat{s}_{kv} + V\beta)$.
  end for

To optimize ϕ given θ, SCVB0 randomly sample a minibatch $\mathcal{D} = \{d_1, \ldots, d_M\}$ of M documents, and approximate the sum over the entire corpus, γ_{kv}, with $\hat{s}_{t,k,v}$

$$\text{E step: } \hat{s}_{t+1,k,v} = (1-\rho_t)\hat{s}_{t,k,v} + \rho_t \frac{D}{M} \sum_{d \in \mathcal{D}} \sum_n \mathbb{I}(w_{dn} = v)\gamma_{dnk}(\theta, \phi_t),$$

$$\text{M step: } \phi_{t+1,k,v} = (\hat{s}_{t+1,k,v} + \beta)/(\sum_v \hat{s}_{t+1,k,v} + V\beta),$$

where ϕ_t is the estimated ϕ at iteration t. See Alg. 2 for a pseudocode.

D.5 Stochastic EM with Variance Reduction

At each epoch e, sEM-vr computes the full-batch sufficient statistics $\gamma_{dk}(\theta_{e,0}, \phi_{e,0})$ and $\gamma_{kv}(\theta_{e,0}, \phi_{e,0})$ according to Eq. (18, 19), and performs the following E-step updates:

$$\hat{s}_{e,t+1,d,k} = (1-\rho)\hat{s}_{e,t,d,k} + \rho \left(N_d \gamma_{dnk}(\theta_{e,t}, \phi_{e,t}) - N_d \gamma_{dnk}(\theta_{e,0}, \phi_{e,0}) + \gamma_{dk}(\theta_{e,0}, \phi_{e,0}) \right),$$

$$\hat{s}_{e,t+1,k,v} = (1-\rho)\hat{s}_{e,t,k,v} + \rho \left(\frac{D}{M} \sum_{d \in \mathcal{D}} \sum_n \mathbb{I}(w_{dn} = v) \left(\gamma_{dnk}(\theta_{e,t}, \phi_{e,t}) - \gamma_{dnk}(\theta_{e,0}, \phi_{e,0}) \right) + \gamma_{kv}(\theta_{e,0}, \phi_{e,0}) \right),$$

see Alg. 3 for the pseudocode.

Algorithm 3 sEM-vr for PLSA.

Require: Initial θ, ϕ
  $\hat{s}_{d,k}, \hat{s}_{k,v} \leftarrow$ BatchEStep(θ, ϕ, W) (Alg. 1)
  for each epoch e **do**
    Store $\forall d, k, \tilde{\theta}_{d,k} \leftarrow \hat{\theta}_{d,k}, \forall k, v, \tilde{\phi}_{k,v} \leftarrow \hat{\phi}_{k,v}$
    $\tilde{s}_{d,k}, \tilde{s}_{k,v} \leftarrow$ BatchEStep(θ, ϕ, W) (Alg. 1)
    for each minibatch of M documents **do**
      (Update θ)
      for each document d **do**
        for each token w_{dn} **do**
          Compute $\forall k, \gamma_{dnk} = \theta_{dk}\phi_{k,w_{dn}}/(\sum_k \theta_{dk}\phi_{k,w_{dn}})$,
          Compute $\forall k, \tilde{\gamma}_{dnk} = \tilde{\theta}_{dk}\tilde{\phi}_{k,w_{dn}}/(\sum_k \tilde{\theta}_{dk}\tilde{\phi}_{k,w_{dn}})$,
          E-step: $\forall k, \hat{s}_{d,k} \leftarrow (1-\rho)\hat{s}_{d,k} + \rho(N_d\gamma_{dnk} - N_d\tilde{\gamma}_{dnk} + \tilde{\gamma}_{dk})$
          M-step: $\theta_d \leftarrow$ Proj$(\hat{s}_d, \alpha, K)$.
        end for
      end for
      (Update ϕ)
      $\forall k, v, \hat{s}_{kv} \leftarrow (1-\rho)\hat{s}_{kv} + \rho\tilde{\gamma}_{kv}$
      for each document d **do**
        for each token w_{dn} **do**
          Compute $\forall k, \gamma_{dnk} = \theta_{dk}\phi_{k,w_{dn}}/(\sum_k \theta_{dk}\phi_{k,w_{dn}})$,
          Compute $\forall k, \tilde{\gamma}_{dnk} = \tilde{\theta}_{dk}\tilde{\phi}_{k,w_{dn}}/(\sum_k \tilde{\theta}_{dk}\tilde{\phi}_{k,w_{dn}})$,
          E-step: $\forall k, \hat{s}_{k,w_{dn}} \leftarrow \hat{s}_{k,w_{dn}} + \rho(\frac{D}{M}\gamma_{dnk} - \frac{D}{M}\tilde{\gamma}_{dnk})$
        end for
      end for
      M-step: $\phi_k \leftarrow$ Proj$(\hat{s}_k, \beta, V)$.
    end for
  end for

A subtlety here is $\hat{s}_{e,t,d,k}$ and $\hat{s}_{e,t,k,v}$ can be negative, so we need additional constraints to ensure that θ_{dk} and ϕ_{kv} are non-negative. We solve the following problem in M-step instead of Eq. (21).

$$\begin{aligned}
&\operatorname*{argmax}_{\theta,\phi} \sum_{dk}(\gamma_{dk}+\alpha)\log\theta_{dk} + \sum_{kv}(\gamma_{kv}+\beta)\log\phi_{kv}.\\
\text{s.t. } &\sum_k \theta_{dk} = 1, \forall d \in [D],\\
&\sum_v \phi_{kv} = 1, \forall k \in [K].\\
&\theta_{dk} > \epsilon, \forall d \in [D], k \in [K]\\
&\phi_{kv} > \epsilon, \forall k \in [K], v \in [V],
\end{aligned}$$

where $\epsilon > 0$ is a threshold to avoid numerical problems. We adopt $\epsilon = 10^{-10}$ in all our experiments. The solution is

$$\theta_d = \text{Proj}(\gamma_d, \alpha, K), \quad \phi_k = \text{Proj}(\gamma_k, \beta, V),$$

where

$$\text{Proj}(\gamma_d, \alpha, K)_k = \epsilon + (1 - K\epsilon)[\gamma_{dk} + \alpha]_+ / \sum_k [\gamma_{dk} + \alpha]_+, \quad [a]_+ := \max\{a, 0\}.$$

D.6 Gradient-based Updates

Instead of performing exact maximization of the ELBO in the M-step, we can also do a stochastic gradient step. However, as the parameters θ_d and ϕ_k are on probabilistic simplex, i.e., $\theta_{dk} > 0$, $\sum_k \theta_{dk} = 1$, $\phi_{kv} > 0$ and $\sum_v \phi_{kv} = 0$, standard stochastic gradient descent (SGD) for unconstrained minimization is not applicable. We implement two algorithms, stochastic mirror descent (SMD) [10] and reparameterized SGD (RSGD), for minimizing on the simplex.

SMD update parameters by $\theta_{dk} \propto \theta_{dk} \exp(\rho \nabla_{\theta_{dk}} Q)$ and $\phi_{kv} \propto \phi_{kv} \exp(\rho \nabla_{\phi_{kv}} Q)$, where Q is the ELBO defined as Eq. (17). The updates are

$$\theta_{dk} \propto \theta_{dk} \exp\left(\rho(N_d \gamma_{dnk} + \alpha)/\theta_{dk}\right),$$
$$\phi_{kv} \propto \phi_{kv} \exp\left[\rho\left(\beta + \frac{D}{M}\sum_{d\in\mathcal{D}}\sum_{n}\mathbb{I}(w_{dn}=v)\gamma_{dnk}\right)/\phi_{kv}\right].$$

RSGD applies the reparametrization

$$\theta_{dk} = \frac{\exp\lambda_{dk}}{\sum_k \exp\lambda_{dk}}, \quad \phi_{kv} = \frac{\exp\tau_{kv}}{\sum_v \exp\tau_{kv}},$$

and optimizes the reparameterized ELBO

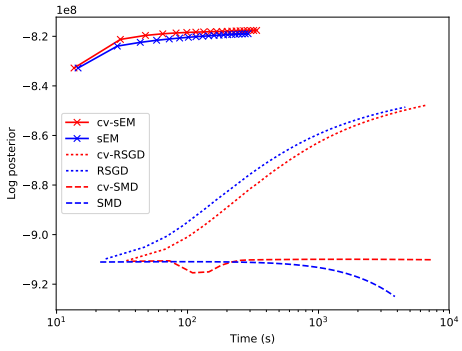
$$Q(\lambda,\tau;\theta,\phi) = \sum_{dk}\left(\gamma_{dk}(\theta,\phi)+\alpha\right)\lambda_{dk} - \sum_d (N_d + K\alpha)\log\left(\sum_k \exp\lambda_{dk}\right)$$
$$+ \sum_{kv}\left(\gamma_{kv}(\theta,\phi)+\beta\right)\tau_{kv} - \sum_k\left(\sum_v \gamma_{kv}(\theta,\phi) + V\beta\right)\log\left(\sum_v \exp\tau_{kv}\right).$$

The updates are

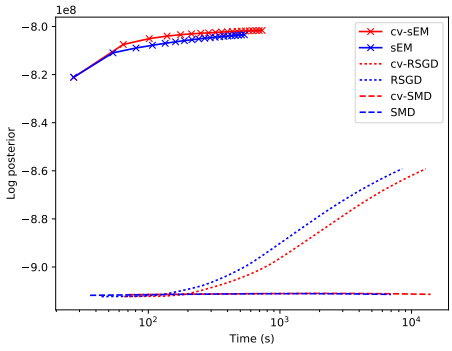
$$\lambda_{dk} \leftarrow \lambda_{dk} + \rho\left[N_d\gamma_{dnk} + \alpha - (N_d + K\alpha)\theta_{dk}\right], \tag{22}$$
$$\tau_{kv} \leftarrow \tau_{kv} + \rho\left[\hat{\gamma}_{kv} + \beta - (\sum_v \hat{\gamma}_{kv} + V\beta)\phi_{kv}\right],$$

where $\hat{\gamma}_{kv} = \frac{D}{M}\sum_{d\in D}\sum_n \mathbb{I}(w_{dn}=v)\gamma_{dnk}$.

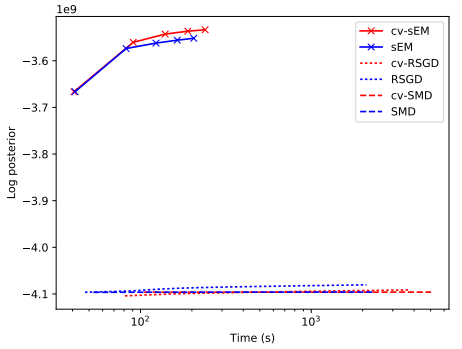
We also implement SVRG-based [20] variance reduction for SMD and RSGD, denoting their variance-reduced version as SMD-vr and RSGD-vr. We compare SMD and RSGD with sEM in Fig. 3 for PLSA. The gradient based algorithms converges slower than SEM, which has an exact M-step. Moreover, SMD and RSGD almost make no progress on the large Wiki and PubMed datasets, because of the bad scaling of the gradient. Take RSGD (Eq. 22) as an example, the gradient is proportional with the document length N_d. The document length can vary greatly, from less than ten to thousands. Therefore, the parameters for long documents changes faster than short documents due to the larger gradient. If the learning rate is large, the gradients of long documents can be so large that the update is not stable. Therefore, the learning rate is limited by the length of the longest document, and all the other shorter documents will converge slowly. In contrast, sEM updates do not have this problem because all the documents forgets the past sufficient statistics at the same rate. SMD and RSGD can be improved with better tuning of the learning rate, such as line search and adaptive learning rates [10]. However this significantly complicates the implementation, and how to apply variance reduction to these algorithms are unclear.



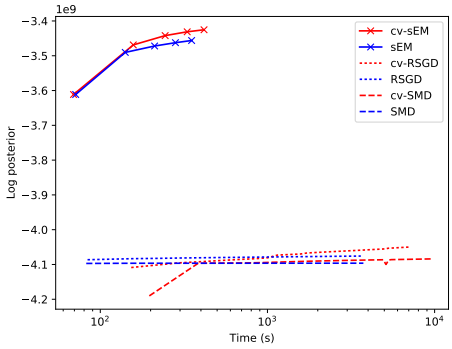
(a) NYTimes $K = 50$



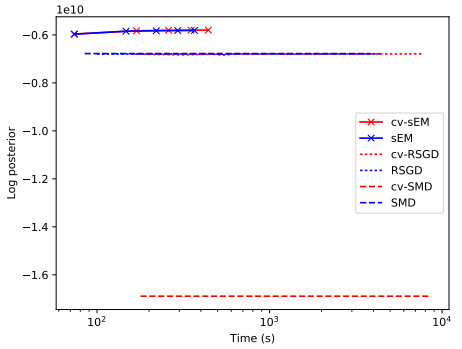
(b) NYTimes $K = 100$



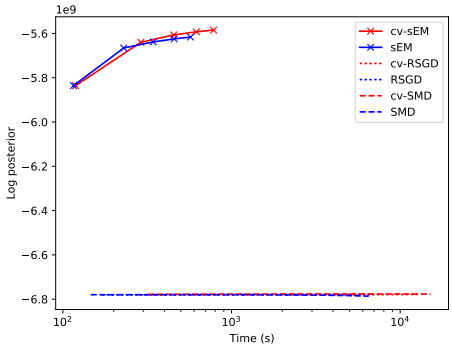
(c) Wiki $K = 50$



(d) Wiki $K = 100$



(e) PubMed $K = 50$



(f) PubMed $K = 100$

Figure 3: PLSA convergence experiments.