High-Dimensional Variance-Reduced Stochastic Gradient Expectation-Maximization Algorithm

Rongda Zhu [1] **Lingxiao Wang** [2] **Chengxiang Zhai** [3] **Quanquan Gu** [2]

Abstract

We propose a generic stochastic expectation-maximization (EM) algorithm for the estimation of high-dimensional latent variable models. At the core of our algorithm is a novel semi-stochastic variance-reduced gradient designed for the Q-function in the EM algorithm. Under a mild condition on the initialization, our algorithm is guaranteed to attain a linear convergence rate to the unknown parameter of the latent variable model, and achieve an optimal statistical rate up to a logarithmic factor for parameter estimation. Compared with existing high-dimensional EM algorithms, our algorithm enjoys a better computational complexity and is therefore more efficient. We apply our generic algorithm to two illustrative latent variable models: Gaussian mixture model and mixture of linear regression, and demonstrate the advantages of our algorithm by both theoretical analysis and numerical experiments. We believe that the proposed semi-stochastic gradient is of independent interest for general nonconvex optimization problems with bivariate structures.

1. Introduction

As a popular algorithm for the estimation of latent variable models, the expectation-maximization (EM) algorithm (Dempster et al., 1977; Wu, 1983) has been widely used in machine learning and statistics (Jain et al., 1999; Tseng, 2004; Han et al., 2011; Little & Rubin, 2014). Although EM is well-known to often converge to an empirically good local estimator (Wu, 1983), finite sample theoretical guarantees for its performance have not been established until recent studies (Balakrishnan et al., 2014; Wang et al., 2014; Yi & Caramanis, 2015). Specifically, the first local convergence theory and finite sample statistical rates of convergence for the conventional EM algorithm and its gradient ascent variant were established in Balakrishnan et al. (2014). Later on, Wang et al. (2014) extended the conventional EM algorithm as well as gradient ascent EM algorithm to the high-dimensional setting, where the number of parameters is comparable to or even larger than the sample size. A key idea used in their algorithms is an additional truncation step after the maximization step (M-step), which is able to exploit the intrinsic sparse structure of the high-dimensional latent variable models. Yi & Caramanis (2015) also proposed a high-dimensional EM algorithm, which, instead of using truncation, uses a regularized M-estimator in the M-step. In the high-dimensional setting, the gradient EM algorithm is especially appealing, because exact maximization based M-step can be very time consuming, or even ill-posed. Nonetheless, gradient EM algorithms can still be computationally prohibitive when the number of observations is also large, since they need to calculate the full gradient at each iteration, whose time complexity is linear in the sample size.

In this paper, we address the aforementioned computational challenge in the large-scale high-dimensional setting, by proposing a novel variance-reduced stochastic gradient EM algorithm with theoretical guarantees. Our algorithm is along the line of gradient EM algorithms (Balakrishnan et al., 2014; Wang et al., 2014), where the M-step is achieved by one-step gradient ascent rather than (regularized) exact maximization (Yi & Caramanis, 2015). Instead of using a full gradient at each iteration as in existing gradient EM algorithms, we significantly reduce the computational cost by utilizing stochastic variance-reduced gradient, which is inspired by recent advances in stochastic optimization (Roux et al., 2012; Johnson & Zhang, 2013; Shalev Shwartz & Zhang, 2013; Defazio et al., 2014; Zhang & Gu, 2016). To accommodate the special bivariate structure of the Q-function (i.e., the expected value of the log likelihood function, with respect to the conditional distribution of the latent variable given the observed variable under the current estimate of the model parameter) in EM algorithm, we design a novel semi-stochastic variance-reduced gradient which sets

[1]Facebook, Inc., Menlo Park, CA 94025 [2]Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA [3]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. Correspondence to: Quanquan Gu <qg5w@virginia.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

our work apart from all existing methods and greatly helps reduce the intrinsic variance of the stochastic gradient of the Q-function in the EM algorithm. We apply our algorithm to two popular latent variable models and thorough numerical experiments are provided to backup our theory. In particular, we summarize our major contributions as follows:

- We propose a novel high-dimensional EM algorithm by incorporating variance reduction into the stochastic gradient method for EM. Specifically, we design a novel semi-stochastic gradient tailored to the bivariate structure of the Q-function in the EM algorithm. To the best of our knowledge, this is the first work ever that brings variance reduction into the stochastic gradient EM algorithm in the high-dimensional scenario.
- We prove that our proposed algorithm converges at a linear rate to the unknown model parameter and achieves the best-known statistical rate of convergence with a mild condition on the initialization.
- We show that the proposed algorithm has an improved overall computational complexity over the state-of-the-art algorithm. Specifically, to achieve an optimization error of ϵ, our algorithm needs $O\big((N + b\kappa^2) \cdot \log(1/\epsilon)\big)$ gradient evaluations[1], where N is the sample size, b is the mini batch size that will be discussed later, and κ is the restricted condition number. In contrast, the gradient complexity of the state-of-the-art high-dimensional EM algorithm (Wang et al., 2014) is $O\big(\kappa N \log(1/\epsilon)\big)$. As long as $\kappa \leq N/b$, which holds in most real cases, the overall gradient complexity of our algorithm is less than Wang et al. (2014).
- Different from the proof technique used in existing work (Balakrishnan et al., 2014; Wang et al., 2014; Yi & Caramanis, 2015), which analyzes both the population and sample versions of the Q-function, we directly analyze the sample version of the Q-function. Our proof is much simpler and provides a good interface to analyze the semi-stochastic gradient.

The rest of the paper is organized as follows. We introduce the related work in Section 2, and then present our algorithm and its applications to two representative latent variable models in Section 3. We demonstrate the main theoretical result as well as its implication to specific latent variable models in Section 4, followed by experimental results in Section 5. Finally, we conclude our paper and point out some future work in Section 6.

Notation: Let $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{d\times d}$ be a matrix and $\mathbf{v} = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$ be a vector. We define the ℓ_q-norm ($q \geq 1$) of $\mathbf{v}$ as $\|\mathbf{v}\|_q = \big(\sum_{j=1}^d |v_j|^q\big)^{1/q}$. Specifically, $\|\mathbf{v}\|_0$ denotes the number of nonzero entries of $\mathbf{v}$, $\|\mathbf{v}\|_2 = \sqrt{\sum_{j=1}^d v_j^2}$ and $\|\mathbf{v}\|_\infty = \max_j |v_j|$. For $q \geq 1$, we define $\|\mathbf{A}\|_q$ as the operator norm of $\mathbf{A}$. Specifically, $\|\mathbf{A}\|_2$ is the spectral norm. We let $\|\mathbf{A}\|_{\infty,\infty} = \max_{i,j} |A_{ij}|$. For an integer $d > 1$, we define $[d] = \{1, \ldots, d\}$. For an index set $\mathcal{I} \in [d]$ and vector $\mathbf{v} \in \mathbb{R}^d$, we use $\mathbf{v}_\mathcal{I} \in \mathbb{R}^d$ to denote the vector where $[\mathbf{v}_\mathcal{I}]_j = v_j$ if $j \in \mathcal{I}$, and $[\mathbf{v}_\mathcal{I}]_j = 0$ otherwise. We use $\mathrm{supp}(\mathbf{v})$ to denote the index set of its nonzero entries, and $\mathrm{supp}(\mathbf{v}, s)$ to denote the index set of top s largest $|v_j|$'s. C is used to denote some absolute constants. The values of these constants may be different from case to case. $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ are used to denote the largest and smallest eigenvalues of matrix $\mathbf{A}$. We use $\mathcal{B}(r; \boldsymbol{\beta})$ to denote the ball centered at $\boldsymbol{\beta}$ with radius r.

2. Related Work

In this section, we discuss some related work in detail. Even with its long history in theory and practice of the EM algorithm (Dempster et al., 1977; Wu, 1983; Tseng, 2004), the finite sample statistical guarantees on EM algorithm have not been pursued until recent research (Balakrishnan et al., 2014; Wang et al., 2014; Yi & Caramanis, 2015). In a pioneering work by Balakrishnan et al. (2014), both statistical and computational analysis of EM algorithm was conducted in the classical regime. Specifically, the authors treated EM algorithms as a special perturbed form of standard gradient methods, and they showed that with an appropriate initialization, their algorithm achieves a locally linear convergence rate to the unknown model parameter. However, their work is limited to the classical regime. While in the high-dimensional regime, when data dimension is much larger than the number of samples, the M-step is often intractable or even not well defined. In order to extend this work to the high-dimensional scenario, Wang et al. (2014) addressed this challenge by inserting a truncation step to enforce the sparsity of the parameter. They proved that their algorithm also enjoys locally linear convergence to the model parameter up to certain statistical error. Yi & Caramanis (2015) proposed a high-dimensional extension of EM algorithms via a regularized M-estimator, and provided similar theoretical guarantees. In addition, both Balakrishnan et al. (2014) and Wang et al. (2014) proposed gradient variants of the EM algorithm, which can be computationally faster than exact maximization based EM.

Although the gradient based EM algorithms (Balakrishnan et al., 2014; Wang et al., 2014) have been proved to achieve guaranteed performance, these deterministic approaches can incur huge computational cost in big data and high-dimensional scenario since they need costly calculation of full gradient at each iteration. Stochastic gradient methods are a common workaround to large scale optimization (Bot-

[1]Throughout this paper, we consider the calculation of the gradient of the Q-function over a data point as a unit gradient evaluation cost. And we use the gradient complexity, i.e., the number of gradient evaluation units, to fairly compare different algorithms.

tou, 2010; Gemulla et al., 2011), because one only needs to calculate a mini-batch of the stochastic gradients each time. However, due to the intrinsic variance introduced by the stochastic gradient, these methods often have a slower convergence rate compared with full gradient methods. Therefore, a lot of variance reduction techniques have been proposed to reduce the variance of the stochastic gradient and pursue a faster convergence rate. One of the most popular methods is the stochastic variance-reduced gradient (SVRG) (Johnson & Zhang, 2013). Inspired by this method, various machine learning tasks (Li et al., 2016; Chen & Gu, 2016; Garber & Hazan, 2015) have utilized the stochastic variance reduction technique to provide improved performance of nonconvex optimization with univariate structures. Recently, Reddi et al. (2016); Allen-Zhu & Hazan (2016) also analyzed SVRG for the general univariate nonconvex finite-sum optimization. Motivated by all of these SVRG methods, one natural question is that, can we accelerate gradient based EM algorithms using SVRG? We show in this work that the answer is in the affirmative. Since all the aforementioned SVRG methods can not be applied to the special bivariate structure of the Q-function, in order to incorporate the variance reduction technique into stochastic gradient based EM algorithms, we need to construct a new semi-stochastic gradient.

3. Methodology

In this section, we present our proposed algorithm. We first introduce the general framework of the EM method, and then give two representative high-dimensional latent variable models as examples before going into the details of our algorithm.

3.1. Background

We now briefly review the latent variable model and the conventional EM algorithm. Let $\boldsymbol{Y} \in \mathcal{Y}$ be the observed random variable and $\boldsymbol{Z} \in \mathcal{Z}$ be the latent random variable with joint distribution $f_{\boldsymbol{\beta}}(\mathbf{y}, \mathbf{z})$ and conditional distribution $p_{\boldsymbol{\beta}}(\mathbf{z}|\mathbf{y})$, with the model parameter $\boldsymbol{\beta} \in \mathbb{R}^d$. Given N observations $\{\mathbf{y}_i\}_{i=1}^N$ of $\boldsymbol{Y}$, the EM algorithm aims at maximizing the Q-function

$$\bar{Q}_N(\boldsymbol{\beta}; \boldsymbol{\beta}') = \frac{1}{N} \sum_{i=1}^{N} \int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_i) \cdot \log f_{\boldsymbol{\beta}}(\mathbf{y}_i, \mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

Particularly, in the l-th iteration of EM algorithm, we evaluate $\bar{Q}_N(\boldsymbol{\beta}; \boldsymbol{\beta}^{(l)})$ in the E-step, and perform the maximization of $\bar{Q}_N(\boldsymbol{\beta}; \boldsymbol{\beta}^{(l)})$ on $\boldsymbol{\beta}$ in the M-step. For example, in the standard gradient ascent implementation of EM algorithm, the M-step is given by

$$\boldsymbol{\beta}^{(l+1)} = \boldsymbol{\beta}^{(l)} + \eta \nabla_1 \bar{Q}_N(\boldsymbol{\beta}^{(l)}; \boldsymbol{\beta}^{(l)}),$$

where $\nabla_1 \bar{Q}_N(\cdot;\cdot)$ denotes the gradient on the first variable and η is the learning rate.

In the high-dimensional regime, we assume $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is sparse with $\|\boldsymbol{\beta}\|_0 \leq s^*$. In order to ensure the sparsity of the estimator, Wang et al. (2014) proposed to use a truncation step (i.e., T-step) following the M-step.

3.2. Illustrative Examples

We now introduce two representative latent variable models as running examples for high-dimensional EM algorithms.

Example 3.1 (Sparse Gaussian Mixture Model)**.** The random variable $\boldsymbol{Y} \in \mathbb{R}^d$ is given by

$$\boldsymbol{Y} = Z \cdot \boldsymbol{\beta}^* + \boldsymbol{V},$$

where Z is a random variable with $\mathbb{P}(Z = 1) = \mathbb{P}(Z = -1) = 1/2$, and $\boldsymbol{V} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ is a Gaussian random vector, with $\boldsymbol{\Sigma}$ being the covariance matrix, $\boldsymbol{V}$ and Z are independent, and $\|\boldsymbol{\beta}^*\|_0 \leq s^*$. We assume $\boldsymbol{\Sigma}$ is known for simplicity.

Example 3.2 (Mixture of Sparse Linear Regression)**.** Let $\boldsymbol{X} \in \mathbb{R}^d \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ be a Gaussian random vector, and $V \sim N(0, \sigma^2)$ be a univariate normal random variable. The random variable $Y \in \mathbb{R}$ is given by

$$Y = Z \cdot \boldsymbol{X}^\top \boldsymbol{\beta}^* + V,$$

where Z is a random variable with $\mathbb{P}(Z = 1) = \mathbb{P}(Z = -1) = 1/2$. Here $\boldsymbol{X}$, V and Z are independent, and $\|\boldsymbol{\beta}\|_0 \leq s^*$. In addition, we assume that σ is known.

3.3. Proposed Algorithm

Now we present our high-dimensional EM algorithm based on stochastic variance-reduced gradient ascent. The outline of the proposed algorithm is described in Algorithm 1.

Since our algorithm is based on stochastic gradient, we divide the N samples into n mini-batches $\{\mathcal{D}_i\}_{i=1}^n$, and define function $\{q_i\}_{i=1}^n$ on these mini-batches, i.e., $q_i(\boldsymbol{\beta}; \boldsymbol{\beta}') = 1/b \sum_{j \in \mathcal{D}_i} \int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_j) \cdot \log f_{\boldsymbol{\beta}}(\mathbf{y}_j, \mathbf{z}) \, \mathrm{d}\mathbf{z}$, where we let b be the mini-batch size, and $N = nb$. Let $Q_n(\boldsymbol{\beta}; \boldsymbol{\beta}') = 1/n \sum_{i=1}^n q_i(\boldsymbol{\beta}; \boldsymbol{\beta}')$. It is easy to show that $Q_n(\boldsymbol{\beta}; \boldsymbol{\beta}') = \bar{Q}_N(\boldsymbol{\beta}; \boldsymbol{\beta}')$.

Note that in Algorithm 1, to ensure the sparsity of the output estimator , we use the hard thresholding operator (Blumensath & Davies, 2009), $\mathcal{H}_s(\mathbf{v}) = \mathbf{v}_{\mathrm{supp}(\mathbf{v},s)}$, which only keeps the largest s entries in magnitude of a vector $\mathbf{v} \in \mathbb{R}^d$. The sparsity parameter s controls the sparsity level of the estimated parameter, and is critical to the estimation error as we will show later.

We can see that there are two layers of iterations in our algorithm. For each outer iteration, we first conduct E-step,

Algorithm 1 Variance Reduced Stochastic Gradient EM Algorithm (VRSGEM)

Parameter: Sparsity Parameter s, Maximum Number of Outer Iterations m, Number of Inner Iterations T, learning rate η
Initialization:
$\widetilde{\boldsymbol{\beta}}^{(0)} = \mathcal{H}_s(\boldsymbol{\beta}^{\text{init}})$,
For $l = 0$ to $m-1$
  E-step:
  Evaluate $Q_n\big(\boldsymbol{\beta}; \widetilde{\boldsymbol{\beta}}^{(l)}\big)$ with the dataset
  $\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(l)}, \quad \widetilde{\mu} = \nabla_1 Q_n(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}})$
  M-step:
  $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$
  Randomly select j_l uniformly from $\{0, \ldots, T-1\}$
  For $t = 0$ to j_l
    Randomly select i from $[n]$ uniformly
    $\mathbf{v}^{(t)} = \nabla_1 q_i\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) - \nabla_1 q_i\big(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}\big) + \widetilde{\mu}$,
    $\boldsymbol{\beta}^{(t+0.5)} = \boldsymbol{\beta}^{(t)} + \eta \mathbf{v}^{(t)}$,
    T-step: $\boldsymbol{\beta}^{(t+1)} = \mathcal{H}_s(\boldsymbol{\beta}^{(t+0.5)})$
  End For
  $\widetilde{\boldsymbol{\beta}}^{(l+1)} = \boldsymbol{\beta}^{(j_l+1)}$
End For
Output: $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$

where we compute the averaged gradient $\widetilde{\mu}$ based on the whole dataset and the model parameter from last outer iteration. This averaged gradient will be used repetitively in the M-step for variance reduction. In M-step, we have the inner iterations. We first determine the number of inner iterations, which is randomly selected from $[T]$ uniformly. At each inner iteration, we make use of the variance reduction technique. Note that we extend the variance reduction idea originally proposed by Johnson & Zhang (2013) to the bivariate structure of the Q-function. Specifically, we design a novel semi-stochastic gradient on mini-batches as $\mathbf{v}^{(t)} = \nabla_1 q_i(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}) - \nabla_1 q_i(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}) + \widetilde{\mu}$, which fixes the second variable within each outer iteration for the sake of convergence guarantee. While the standard gradient implementation of EM algorithm (Wang et al., 2014) uses $\nabla_1 \bar{Q}_N(\boldsymbol{\beta}^{(t)}; \boldsymbol{\beta}^{(t)})$ to update the parameter at each iteration, our newly designed semi-stochastic gradient is proved to better reduce the variance and attain a lower gradient complexity. After finishing all the inner iterations, we use the output from the last inner iteration as the output of this outer iteration. We use the output from the last outer iteration as the final estimator.

We believe our newly proposed semi-stochastic gradient is of independent interest for the stochastic optimization of functions with bivariate structures, to prove a faster rate of convergence.

4. Main Theoretical Results

In this section, we show the main theory on the theoretical guarantees of our proposed Algorithm 1. We also present the implications of our algorithm applied to two models described in Section 3.2.

To facilitate the technical analysis of our algorithm, we focus on the resampling version of Algorithm 1 following the convention of previous work (Wang et al., 2014; Yi & Caramanis, 2015). The key difference between the resampling version and Algorithm 1 is that we split the whole dateset into m subsets and use one subset for each outer iteration. The details of the resampling version of our algorithm is provided in the longer version of this paper. It is worth noting that the resampling version of our algorithm is able to decouple the dependence between consecutive outer iterations, and it is only used to simplify the technical proof. In practice including our experiment, we use Algorithm 1 rather than the resampling version.

Before we present the main results, we introduce three conditions that are essential for our analysis.

Condition 4.1 (Smoothness). For any $\boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, for any $i \in [n]$, $q_i(\cdot;\cdot)$ in Algorithm 1 satisfies the smoothness condition with respect to the first variable with parameter L:

$$\big\|\nabla_1 q_i(\boldsymbol{\beta}_1; \boldsymbol{\beta}) - \nabla_1 q_i(\boldsymbol{\beta}_2; \boldsymbol{\beta})\big\|_2 \leq L\big\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\big\|_2.$$

Condition 4.1 says that the gradient of $q_i(\cdot;\cdot)$ we use in each inner iteration is Lipschitz continuous with respect to the first variable when the first and second variables are within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$. There exists a wide range of models with this condition holding.

Condition 4.2 (Concavity). For all $\boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, the function $Q_n(\cdot;\cdot)$ satisfies the strong concavity condition with parameter μ:

$$\big[\nabla_1 Q_n(\boldsymbol{\beta}_1; \boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}_2; \boldsymbol{\beta})\big]^\top (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2) \leq -\mu\|\boldsymbol{\beta}_2 - \boldsymbol{\beta}_1\|_2^2.$$

Condition 4.2 requires $Q_n(\cdot;\cdot)$ to be strongly concave with respect to the first variable when the first and second variables are within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$. This is a reasonable requirement when N is large enough.

Condition 4.3 (First-order stability). For the true model parameter $\boldsymbol{\beta}^*$ and any $\boldsymbol{\beta} \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, $Q_n\big(\cdot;\cdot\big)$ satisfies the first-order stability with parameter γ:

$$\big\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\big\|_2 \leq \gamma\big\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\big\|_2.$$

Condition 4.3 requires that the gradient $\nabla_1 Q_n(\boldsymbol{\beta}^*;\cdot)$ is stable with regard to the second variable, with the second variable within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$. There are actually

various versions of this condition in previous work (Yi & Caramanis, 2015; Balakrishnan et al., 2014) on population version $Q(\cdot;\cdot) = \mathbb{E}[Q_n(\cdot,\cdot)]$. Here we impose the condition on the sample Q-function, i.e., $Q_n(\cdot,\cdot)$, because our proof technique directly analyzes the sample Q-function. Intuitively, when the sample size N is sufficiently large, $Q_n(\cdot;\cdot)$ and $Q(\cdot;\cdot)$ should be close. Therefore, this condition should hold for $Q_n(\cdot;\cdot)$ as well.

Due to the space limit, we verify the above conditions for the two examples in the longer version of this paper. We use $\kappa = L/\mu$ to denote the *condition number* .

4.1. Theory for the Generic Model

With the above conditions on $q_i(\cdot;\cdot)$ and $Q_n(\cdot;\cdot)$, we have the following theorem to characterize the estimation error of our estimator $\widetilde{\boldsymbol{\beta}}^{(r)}$ returned by the resampling version of Algorithm 1.

Theorem 4.4. Suppose $q_i(\cdot;\cdot)$ satisfies Condition 4.1 and $Q_n(\cdot;\cdot)$ satisfies Conditions 4.2, 4.3. We also assume that $\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 \le p\big\|\boldsymbol{\beta}^*\big\|_2$, where $p \in (0,1)$. If $\eta \le \mu/(8L^2)$, and T and s are chosen such that

$$\rho = \frac{1}{T(1-\tau)} + \frac{2\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]}{1-\tau} < 1,$$

where $\tau = \alpha(1-\eta\mu+2\eta^2L^2)$ and $\alpha = 1+\sqrt{s^*}/\sqrt{s-s^*}$, then the estimator $\widetilde{\boldsymbol{\beta}}^{(r)}$ from the resampling version of Algorithm 1 satisfies

$$\mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(r)} - \boldsymbol{\beta}^*\big\|_2 \le \rho^{r/2}\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 + \sqrt{\frac{2\widetilde{s}\alpha\eta(2\eta + L/\mu^2)}{(1-\tau)(1-\rho)}}\big\|\nabla_1 Q_n\big(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*\big)\big\|_\infty, \tag{4.1}$$

where $\widetilde{s} = 2s + s^*$.

Remark 4.5. As suggested in Theorem 4.4 that by choosing an appropriate learning rate η, a sufficiently large number of inner iterations T, and sparsity parameter s such that $\rho < 1$, we can achieve a linear convergence rate. Here we give an example to show that such ρ is achievable. If we choose step size $\eta = \mu/(8L^2)$, and truncation parameter s satisfies

$$s > \left[\frac{4(1-K)^2}{K^2} + 1\right]s^*,$$

where

$$K = \frac{5\mu^2}{96L^2} - \frac{\mu^2\gamma^2}{12L^4} - \frac{\gamma^2}{3L\mu} > 0.$$

Then, we can get

$$\alpha < \frac{1}{1 - 5\mu^2/96L^2 + \mu^2\gamma^2/12L^4 + \gamma^2/3L\mu},$$

and the contraction parameter ρ in Theorem 4.4 can be simplified as

$$\rho \le \frac{1}{T(1-\tau)} + \frac{3}{4}.$$

Therefore, if we choose $T \ge 256\kappa^2/\big(3(\alpha-1)\big)$, we can obtain $\rho \le 7/8$, ensuring the linear convergence rate.

Remark 4.6. The right hand side of (4.1) in Theorem 4.4 consists of two terms. The first term stands for the optimization error. The second term is the statistical error. According to Remark 4.5, we can ensure the linear convergence rate of our algorithm. Thus for any error bound $\epsilon > 0$, we need $r \ge 2\log_{\rho^{-1}}[\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2/\epsilon]$ iterations to let the optimization error $\rho^{r/2}\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2 \le \epsilon$, which basically requires $O\big(\log(1/\epsilon)\big)$ outer iterations. For each outer iteration, we need to compute T gradients of $q_i(\cdot,\cdot)$, and one full gradient. Since we have $T = O(\kappa^2)$, which is suggested in Remark 4.5, the gradient complexity of our algorithm would be $O\big((N + b\kappa^2)\cdot\log(1/\epsilon)\big)$. Nevertheless, for the state-of-the-art gradient based high-dimensional EM algorithm (Wang et al., 2014), its gradient complexity is $O\big(\kappa N\log(1/\epsilon)\big)$. As long as $\kappa \le N/b$, the gradient complexity of our algorithm is less than that of Wang et al. (2014). Since in big data scenarios, N is always very large and b as the batch size is relatively small, this condition is naturally satisfied in most real applications.

The second term on the right-hand side of (4.1) stands for the upper bound of the statistical error, which depends on specific models as we will introduce later.

4.2. Implications for Specific Models

Now we apply our algorithm to the two examples introduced in Section 3.2.

4.2.1. Sparse Gaussian mixture model

The next corollary gives the implication of our main theory for sparse Gaussian mixture models.

Corollary 4.7. Under the same conditions of Theorem 4.4 and suppose $\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 \le (\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})/\lambda_{\max}(\boldsymbol{\Sigma})}/4)\big\|\boldsymbol{\beta}^*\big\|_2$. Then with probability at least $1 - 2e/d$, the estimator $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$ from the resampling version of Algorithm 1 satisfies

$$\mathbb{E}\big\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2 \le \rho^{m/2}\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 + C\Phi\kappa^{3/2}\sqrt{\frac{s^*\log d\cdot\log N}{N}}, \tag{4.2}$$

where $\Phi = \lambda_{\min}(\boldsymbol{\Sigma})\big(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sigma\lambda_{\min}^{-1/2}(\boldsymbol{\Sigma})\big)$ and $\kappa = L/\mu$.

Proof Sketch. For sparse Gaussian mixture model, we have Conditions 4.1 to 4.3 hold with parameters $L = 1/\lambda_{\min}(\mathbf{\Sigma})$, $\mu = 1/\lambda_{\max}(\mathbf{\Sigma})$, and $\gamma = 20(\xi^2 + \xi + 1 + \xi^{-2})e^{-\xi^2/64}/\lambda_{\min}(\mathbf{\Sigma})$, where $\xi = \|\mathbf{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ denotes the signal-to-noise ratio (SNR). Next, $\widetilde{s} = 2s + s^*$ is of the same order as s^*. For the term $\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\|_\infty$ in (4.1), we have the following inequality holds with probability at least $1 - 2e/d$

$$\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\|_\infty \leq C\bigg(\|\mathbf{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \frac{\sigma}{\sqrt{\lambda_{\min}(\mathbf{\Sigma})}}\bigg)\sqrt{\frac{\log d \cdot \log N}{N}}.$$

This completes the proof. □

Remark 4.8. We can see that the parameters in Conditions 4.1 and 4.2 are determined by the covariance matrix $\mathbf{\Sigma}$, which is reasonable because $\mathbf{\Sigma}$ actually denotes the variance of the data. For Condition 4.3, we need to introduce the signal-to-noise ratio (SNR). The concept of SNR in parameter estimation is also proposed in Balakrishnan et al. (2014); Dasgupta & Schulman (2007). Since we have extended the covariance matrix of noise from identity matrix in previous work to any positive definite matrix, our SNR is also a little bit different from their definition. Generally speaking, for GMM with lower SNR, the variance of the noise makes it harder or even impossible for the algorithm to converge. Therefore, it is always reasonable to have a requirement for the SNR of GMM to be large enough for reliable parameter estimation. Spectral method (Anandkumar et al., 2014) can be used to match the requirement on initialization for GMM, however, we find that random initialization also performs reasonably well in practice as we will show later.

According to Remark 4.5, by choosing appropriate learning rate η, inner iterations T, and sparsity parameter s, we can ensure linear convergence rate of our algorithm. Therefore, from Corollary 4.7, we know that after $O\big(\log\big(N/(s^* \log d \log N)\big)\big)$ number of iterations, the output of our algorithm attains $O(\sqrt{s^* \log d \cdot \log N/N})$ statistical error, which matches the best-known error bound (Wang et al., 2014; Yi & Caramanis, 2015) for Gaussian mixture model up to a logarithmic factor $\log N$. Note that the extra logarithmic factor is due to the resampling strategy.

4.2.2. Mixture of Sparse Linear Regression

The implication of our main theory for mixture of linear regression is presented in the following corollary.

Corollary 4.9. Under the same conditions of Theorem 4.4 and suppose $\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 \leq (\sqrt{\lambda_{\min}(\mathbf{\Sigma})/\lambda_{\max}(\mathbf{\Sigma})}/32)\big\|\boldsymbol{\beta}^*\big\|_2$. Then with probability at least $1 - 2e/d$, the estimator $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$ from the resampling version of Algorithm 1 satisfies

$$\mathbb{E}\big\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2 \leq \rho^{m/2}\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 + C\kappa^{3/2}\bigg(\|\boldsymbol{\beta}^*\|_2 + \frac{\sigma}{\sqrt{\lambda_{\max}(\mathbf{\Sigma})}}\bigg)\sqrt{\frac{s^* \log d \cdot \log N}{N}},$$

where $\kappa = L/\mu$.

Proof Sketch. For mixture of linear regression, we have Conditions 4.1 to 4.3 hold with parameters $L = 2\lambda_{\max}(\mathbf{\Sigma})$, $\mu = \lambda_{\min}(\mathbf{\Sigma})/2$, and $\gamma = \gamma_1\lambda_{\max}(\mathbf{\Sigma})$, where $\gamma_1 \in (0, 1/3)$ is a constant. We also show that $\widetilde{s}$ is of the same order as s^*. Next, for the term $\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\|_\infty$ in (4.1), we have the following inequality holds with probability at least $1 - 2e/d$

$$\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\|_\infty \leq C\big(\lambda_{\max}(\mathbf{\Sigma})\|\boldsymbol{\beta}^*\|_2 + \lambda_{\max}^{1/2}(\mathbf{\Sigma})\sigma\big)\sqrt{\frac{\log d \cdot \log N}{N}}.$$

This completes the proof. □

Remark 4.10. According to Remark 4.5, our algorithm can achieve a linear convergence rate with appropriate learning rate η, inner iterations T, and sparsity parameter s. Thus Corollary 4.9 tells us that after $O\big(\log\big(N/(s^* \log d \log N)\big)\big)$ number of outer iterations, the output of our algorithm achieves $O(\sqrt{s^* \log d \cdot \log N/N})$ statistical error, which matches the best-known statistical error (Yi & Caramanis, 2015) for mixture of linear regression up to a logarithmic factor from the resampling strategy. Specifically, the dependence on $\|\boldsymbol{\beta}^*\|_2$ is due to the fundamental limits of EM, which also appears in Balakrishnan et al. (2014); Yi & Caramanis (2015). There is also a spectral method (Chaganty & Liang, 2013) helping the initialization of MLR, but we use random initialization which also performs well in our experiments.

5. Experiments

In this section, we present experiment results to validate our theory. For parameter estimation, we use Gaussian mixture model and mixture of linear regression, and compare our proposed variance-reduced stochastic gradient EM algorithm (VRSGEM) with two state-of-the-art high-dimensional EM algorithms as baselines:

- (HDGEM) High-Dimensional Gradient EM algorithm proposed in Wang et al. (2014): the gradient variant of high-dimensional EM method enforcing sparsity structure.
- (HDREM) High-Dimensional Regularized EM algorithm proposed in Yi & Caramanis (2015): the method based on decaying regularization.

Since high-dimensional scenario is much more challenging, we only compare our algorithm with high-dimensional EM algorithms.

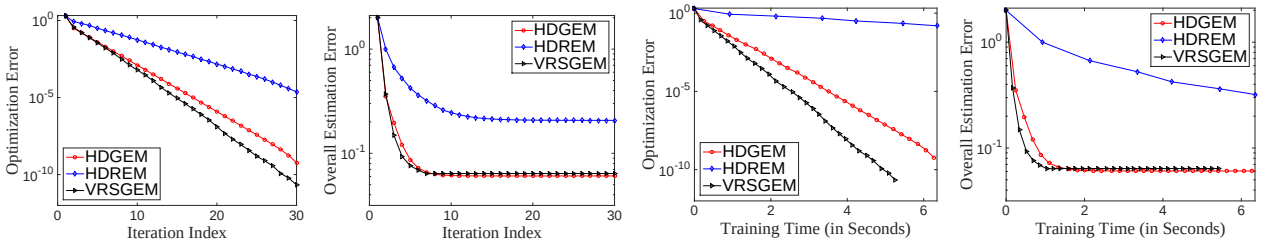


Figure 1. Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for GMM. $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$. (a) (b) errors against iterations, (c) (d) errors against training time.

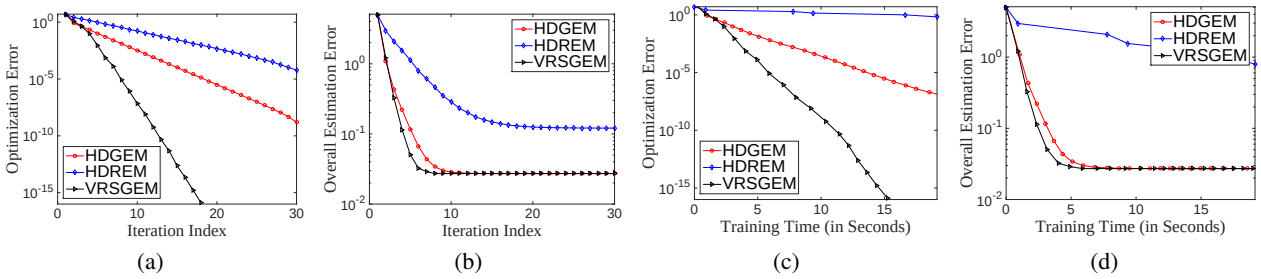


Figure 2. Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for GMM. $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. (a) (b) errors against iteration, (c) (d) errors against training time.

5.1. Experimental Setup

For each latent variable model, we compare both the optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ featuring the convergence of the estimator to the local optima, and the overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ featuring the overall estimation accuracy with regard to the true model parameter $\boldsymbol{\beta}^*$. We also show the convergence comparison in terms of training time. All the comparisons are under two different parameter settings: $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$ and $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. For VRSGEM, we choose $m = 30$, $n = 50$ and $T = 50$ across all settings and models. Besides the comparison of different algorithms, we also verify our statistical rate of convergence by plotting the statistical error $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|$ against $\sqrt{s^* \log d / N}$. Specifically, we fix $d = 512$ and show the plots of three cases $s^* = 5$, $s^* = 10$ and $s^* = 15$ with varying N.

In each experiment setting, we run 100 trials and show the averaged results. The learning rate η is tuned by grid search and s is chosen by cross validation. We use random initialization.

5.2. Gaussian Mixture Model

We test VRSGEM on Gaussian mixture models introduced in Section 3.2. For the sake of simplicity and better matching the problem setting of the baseline methods, the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{V}$ is chosen to be a diagonal matrix with all elements being 1. We randomly set two elements to $\lambda_{\max}(\boldsymbol{\Sigma}) = 10$, and another two elements to $\lambda_{\min}(\boldsymbol{\Sigma}) = 0.1$. The results are shown in Figures 1 and 2.

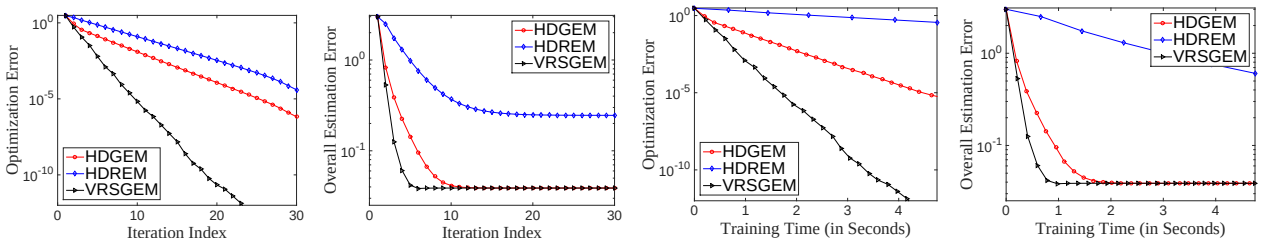
From Figures 1(a) and 2(a), we can see that all three algorithms have linear convergence as Corollary 4.7 suggests. VRSGEM clearly enjoys a faster convergence rate than the baselines. Moreover, as shown in Figures 1(b) and 2(b), the performance on overall estimation error of our algorithm is as good as HDGEM, which is far better than HDREM. In terms of time consumption, our algorithm also enjoys a remarkable advantage over the baselines as shown in Figures 1(c), 1(d), 2(c) and 2(d).

The statistical error results are shown in Figure 5. From Figure 5(a), we can see that statistical error of VRSGEM shows a linear dependency on $\sqrt{s^* \log d / N}$ across different settings of s^*, verifying results in Corollary 4.7.

5.3. Mixture of Linear Regression

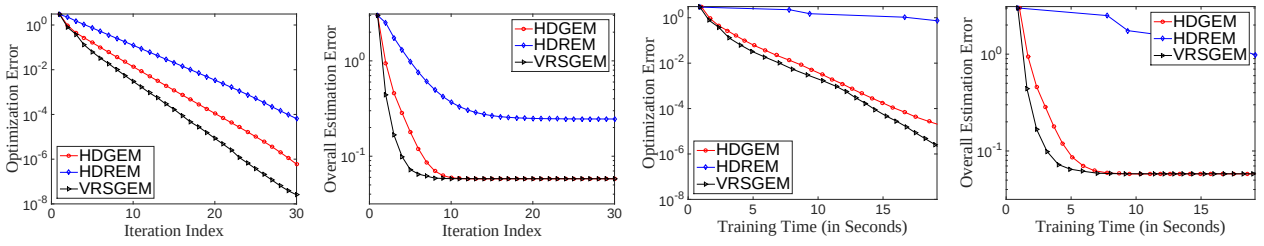
Similar to the setting for GMM, we use the same covariance matrix $\boldsymbol{\Sigma}$ in Section 5.2 for $\boldsymbol{X}$ here. For V, we let $\sigma = 1$. We show the results in Figures 3 and 4.

From Figures 3(a) and 4(a), we can see that VRSGEM achieves linear convergence which is consistent with Corollary 4.9, and our algorithm significantly outperforms the



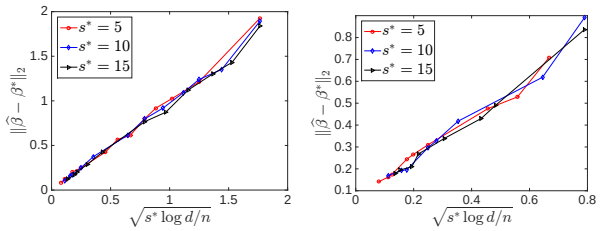
(a) (b) (c) (d)

Figure 3. Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for MLR. $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$. (a) (b) errors against iterations, (c) (d) errors against training time.



(a) (b) (c) (d)

Figure 4. Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for MLR. $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. (a) (b) errors against iteration, (c) (d) errors against training time.



(a) GMM (b) MLR

Figure 5. Statistical error $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$ of VRSGEM against $\sqrt{s^* \log d/N}$ with fixed $d = 512$ and varying s^* and N.

baselines in terms of optimization error. In terms of overall estimation error shown in Figures 3(b) and 4(b), VRSGEM is as good as HDGEM and beats HDREM by a remarkable margin. Our algorithm also beats the baselines in time consumption for convergence as we can see in Figures 3(c), 3(d), 4(c) and 4(d). Overall, VRSGEM achieves the best performance among all the methods.

In addition, from Figure 5(b), we can see that for MLR, the statistical error of VRSGEM is of order $\sqrt{s^* \log d/N}$, which supports Corollary 4.9.

6. Conclusions and Future Work

We propose a novel accelerated stochastic gradient EM algorithm based on a uniquely constructed semi-stochastic variance reduced gradient. We show that with an appropriate initialization, the proposed algorithm converges at a linear rate and attains the optimal statistical rate. We apply our proposed algorithm to two popular latent variable models in the high-dimensional regime and numerical experiments are provided to support our theory.

It is worth noting that, the proposed algorithm is directly applicable to the classical regime, by dropping the T-step. It will give rise to an accelerated stochastic extension of conventional EM algorithm, and the corresponding theory in this paper can be extended to the classical regime analogously (Balakrishnan et al., 2014). We will investigate this by-product in our future work. We also plan to extend our algorithm to the estimation of high-dimensional latent variable models with low-rank parameters (Yi & Caramanis, 2015).

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation under Grant Numbers CNS-1513939, CNS-1027965, IIS-1629161, IIS-1618948, IIS-1652539. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Allen-Zhu, Zeyuan and Hazan, Elad. Variance reduction for faster non-convex optimization. *arXiv preprint arXiv:1603.05643*, 2016.

Anandkumar, Animashree, Ge, Rong, Hsu, Daniel, Kakade, Sham M., and Telgarsky, Matus. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014.

Balakrishnan, Sivaraman, Wainwright, Martin J, and Yu, Bin. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.

Blumensath, Thomas and Davies, Mike E. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Chaganty, Arun Tejasvi and Liang, Percy. Spectral experts for estimating mixtures of linear regressions. *arXiv preprint arXiv:1306.3729*, 2013.

Chen, Jinghui and Gu, Quanquan. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 132–141. AUAI Press, 2016.

Dasgupta, Sanjoy and Schulman, Leonard. A probabilistic analysis of EM for mixtures of separated, spherical Gaussians. *Journal of Machine Learning Research*, 8: 203–226, 2007.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 39(1):1–38, 1977. ISSN 00359246.

Garber, Dan and Hazan, Elad. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Gemulla, Rainer, Nijkamp, Erik, Haas, Peter J, and Sismanis, Yannis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 69–77. ACM, 2011.

Han, Jiawei, Pei, Jian, and Kamber, Micheline. *Data mining: concepts and techniques*. Elsevier, 2011.

Jain, Anil K, Murty, M Narasimha, and Flynn, Patrick J. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Li, Xingguo, Zhao, Tuo, Arora, Raman, Liu, Han, and Haupt, Jarvis. Stochastic variance reduced optimization for nonconvex sparse learning. *arXiv preprint arXiv:1605.02711*, 2016.

Little, Roderick JA and Rubin, Donald B. *Statistical analysis with missing data*. John Wiley & Sons, 2014.

Reddi, Sashank J, Hefny, Ahmed, Sra, Suvrit, Póczós, Barnabás, and Smola, Alex. Stochastic variance reduction for nonconvex optimization. *arXiv preprint arXiv:1603.06160*, 2016.

Roux, Nicolas L, Schmidt, Mark, and Bach, Francis R. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2012.

Shalev Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb): 567–599, 2013.

Tseng, Paul. An analysis of the EM algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29(1):27–44, 2004. ISSN 0364765X.

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

Wu, C. F. Jeff. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 03 1983. doi: 10.1214/aos/1176346060.

Yi, Xinyang and Caramanis, Constantine. Regularized em algorithms: A unified framework and statistical guarantees. In *Advances in Neural Information Processing Systems*, pp. 1567–1575, 2015.

Zhang, Aston and Gu, Quanquan. Accelerated stochastic block coordinate descent with optimal sampling. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2035–2044. ACM, 2016.