A Stochastic Nonconvex Splitting Method for Symmetric Nonnegative Matrix Factorization

Songtao Lu
Iowa State University

Mingyi Hong
Iowa State University

Zhengdao Wang
Iowa State University

Abstract

Symmetric nonnegative matrix factorization (SymNMF) plays an important role in applications of many data analytics problems such as community detection, document clustering and image segmentation. In this paper, we consider a stochastic SymNMF problem in which the observation matrix is generated in a random and sequential manner. We propose a stochastic nonconvex splitting method, which not only guarantees convergence to the set of stationary points of the problem (in the mean-square sense), but further achieves a sublinear convergence rate. Numerical results show that for clustering problems over both synthetic and real world datasets, the proposed algorithm converges quickly to the set of stationary points.

1 Introduction

Symmetric nonnegative matrix factorization (SymNMF) approximates a given symmetric nonnegative matrix $\mathbf{Z} \in \mathbb{R}^{N\times N}$ by a low rank matrix $\mathbf{X}\mathbf{X}^T$, where the factor matrix $\mathbf{X} \in \mathbb{R}^{N\times K}$ is component-wise nonnegative, typically with $K \ll N$ [1–3]. Finding an exact factorization (i.e., $\exists\ \mathbf{X} \geq 0$ such that $\mathbf{X}\mathbf{X}^T = \mathbf{Z}$) is NP hard [4], where such factors are called *completely positive matrices* [5]. In recent years, SymNMF has found many applications in document clustering, community detection, image segmentation and pattern clustering in bioinformatics [1, 3, 6]. In particular, SymNMF shows better clustering results than the well-known eigen-value decomposition based spectral clustering method [7–9], when the clusters are placed within a cluttered background [3, 10]. SymNMF is not only able to provide a good interpretation of the resulting data, but also outperforms spectral clustering and nonnegative matrix factorization (NMF) when the data set exhibits certain nonlinear structures [3].

Classical SymNMF problems are deterministic, where the observation matrix $\mathbf{Z}$ is completely known. However, in recent applications such as social network community detection, the matrix $\mathbf{Z}$ represents the relations among the clusters/communities, observed during a given time period. By nature such matrix is random, whose structure is determined by the dynamics of the network connections [11]. Furthermore, in many modern big-data related problems such as matrix completion [12], subspace tracking [13], community detection, the data are usually collected through some random sampling techniques. As a concrete example, in community detection problems the observed activities among the nodes can change over time hence is random. In these applications sampling the connectivity of the graph at a given time results in a random similarity matrix, such as stochastic block model [14]. Mathematically, the stochastic SymNMF problem can be formulated as the following stochastic optimization problem

$$\min_{\mathbf{X}\geq 0} \quad f(\mathbf{X}) = \frac{1}{2}\mathbb{E}_{\mathbf{Z}}[\|\mathbf{X}\mathbf{X}^T - \mathbf{Z}\|_F^2] \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, inequality constraint $\mathbf{X} \geq 0$ is component-wise, $\mathbf{Z}$ follows some distribution over a set $\Xi \in \mathbb{R}^{N\times N}$, and the expectation is taken over the random observation $\mathbf{Z}$. In clustering problems, the samples of matrix $\mathbf{Z}$ can be the similarity matrix which measures the connections among nodes over networks.

As we will see later, the problem in (1) is equivalent to $\min_{\mathbf{X}\geq 0}\|\mathbf{X}\mathbf{X}^T - \mathbb{E}_{\mathbf{Z}}[\mathbf{Z}]\|_F^2$. If we know the distribution of $\mathbf{Z}$, then we can computer $\mathbb{E}_{\mathbf{Z}}[\mathbf{Z}]$ first and the problem is converted to a classical SymNMF problem. However, in practice, we usually do not have access to the underlying distribution of $\mathbf{Z}$. Instead, we can obtain sequentially realizations of $\mathbf{Z}$, such as in the application of online streaming data

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

[15]. It is possible to use a batch of samples to compute the empirical mean of $\mathbf{Z}$ and implement the deterministic SymNMF algorithms. As more samples are collected, the empirical mean will converge to the ensemble mean, leading to a consistent estimator of the solution of the symmetric factor $\mathbf{X}$. There are two problems with such an approach. First, it may be desirable to have an estimate of the symmetric factor $\mathbf{X}$ at each time instant, namely when a new sample of $\mathbf{Z}$ is available. Running the complete SymNMF algorithm at each time instant may be computationally expensive. Second, even if the computational complexity is not a concern, existing analysis results and theoretical guarantees such as convergence rate are not applicable to the case where the matrix to be factorized is changing with time (although eventually converging to the ensemble mean). Therefore, it is desirable to develop efficient algorithms that produce online SymNMF updates based on sequential realizations of $\mathbf{Z}$.

Related Works. Recently, the stochastic projected gradient descent (SPGD) methods are proposed for dealing with stochastic nonconvex problems [16, 17]. However, there has been no convergence guarantee when directly applying SPGD to solve the stochastic SymNMF problem, since there is no global Lipschitz continuity of the gradient of the objection function. Classical stochastic approximation methods can also be used, but without convergence and rate of convergence guarantees. There have been a number of works that focus on designing customized algorithms for deterministic SymNMF such as projected gradient descent (PGD) and projected Newton (PNewton) [3, 18]. Both of then solve the deterministic version of problem (1). However, there has been no global convergence analysis since the objective function is a nonconvex fourth-order polynomial. A straightforward strategy of reducing the order of the polynomial with respective to $\mathbf{X}$ is to introduce a new variable $\mathbf{Y}$ and rewrite SymNMF equivalently as

$$\min_{\mathbf{Y}\geq 0, \mathbf{X}=\mathbf{Y}} \frac{1}{2}\|\mathbf{X}\mathbf{Y}^T - \mathbf{Z}\|_F^2. \tag{2}$$

Then, a simply way of solving problem (2) might ignore the equality constraint $\mathbf{X} = \mathbf{Y}$ first, and then update variables $\mathbf{X}$ and $\mathbf{Y}$ in an alternative way. The alternative nonnegative least squares (ANLS) algorithm was proposed in [3] for dealing with SymNMF, where a regularized term is added to the objective function as the penalty of the difference between the two matrices. Unfortunately, there is no guarantee that the $\mathbf{Y}$-iterate will converge to the $\mathbf{X}$-iterate. Alternating direction method of multipliers (ADMM) is one of the powerful tools on solving the optimization problems where the variables have linear coupling. It has also been applied to matrix factorization-type problems such as NMF [19–21]. Existing results such as [22–25] for analyzing ADMM for nonconvex problems do not apply for SymNMF either, because in these works the objective function is required to be separable over the block variables. Fast convergence rates of stochastic ADMM algorithms are presented recently [26, 27], however, these algorithms only work for stochastic convex optimization problems. In fact, none of the works has rigorous theoretical justification that they can be applied directly for SymNMF in the stochastic settings.

The most relevant algorithm that uses the nonconvex splitting method for solving SymNMF was proposed in [28], but the algorithm, called NS-SymNMF, only works for the case where the given data is deterministic. In this paper, we consider the stochastic setting of matrix factorization that potentially make the SymNMF more practical. The proposed algorithm is a generalization of the previous NS-SymNMF algorithm, which is able to factorize the realizations of the random observation matrix in each iteration. Further, actually the convergence proof of NS-SymNMF does not apply to that of SNS-SymNMF, since the iterates are coupled with the random data matrices as the algorithm proceeds such that the boundness of the iterates is not clear if the convergence proof of NS-SymNMF was used.

Contributions of This Paper. In this paper, a stochastic nonconvex splitting SymNMF (SNS-SymNMF) is proposed for problem (1), where the underlying distribution is unknown, but realizations of $\mathbf{Z}$ are available sequentially. Our algorithm is based upon reformulation (2), where we have introduced a new variable $\mathbf{Y}$. The advantage of doing so is that when adding the equality constraint to the objective as the quadratic penalty, the problem is strongly convex with respect to either $\mathbf{Y}$ and $\mathbf{X}$. A Lagrangian relaxation technique is further used to gradually enforce the equality constraint as the algorithm evolves. The proposed algorithm belongs to the class of stochastic algorithms, because at each iteration only a few samples of the observation matrix are used. Based on different ways in which the samples are utilized, we analyze the performance of the algorithm in terms of its convergence rates to the set of stationary solutions of problem (1). The main contributions of this paper are given below.

- The proposed algorithm possesses sublinear convergence rate guarantees. When an aggregate of the past samples is used (possibility with non-uniform weighting), the algorithm converges sublinearly to the stationary points of problem (1) in mean-square; when the instantaneous samples are used, the algorithm converges sublinearly to a neighborhood around the stationary solutions. To our best knowledge, this is the first stochastic algorithm that can possess a sublinear convergence rate for stochastic SymNMF.

- We demonstrate the performance of the proposed stochastic algorithm for clustering problems. It has been shown that SNS-SymNMF is much faster compared with some existing algorithms for generic stochastic nonconvex optimization problems numerically. Further, due to the use of non-uniform aggregate sampling, the proposed algorithm is capable of tracking changes of the community structure.

All proofs of this paper are provided in the supplemental materials.

2 Stochastic Nonconvex Splitting for SymNMF

2.1 Main Assumptions

The sequentially sampled data $\widehat{\mathbf{Z}}^{(i)}$ are assumed to be independent and identically distributed (*i.i.d.*) realizations of the random matrix $\mathbf{Z}$, where i denotes the index of the sample. Rather than assuming the unbiased gradient and bounded variance of the stochastic gradient in most stochastic gradient methods [17], we only need to make assumptions on samples for SymNMF. Specifically, we assume the following.

- A1) Unbiased sample: $\mathbb{E}[\widehat{\mathbf{Z}}^{(i)}] = \overline{\mathbf{Z}} \quad \forall i$;
- A2) Bounded variance: $\mathsf{Tr}[\mathsf{Var}[\widehat{\mathbf{Z}}^{(i)}]] = \mathbb{E}[\|\widehat{\mathbf{Z}}^{(i)} - \overline{\mathbf{Z}}\|_F^2] \leq \sigma^2 \quad \forall i$;
- A3) Bounded magnitude: $\|\widehat{\mathbf{Z}}^{(i)}\|_F \leq \mathcal{Z} < \infty \quad \forall i$.

In practice, the magnitude of samples is finite, so A3 is valid [3,17].

2.2 The Problem Formulation for Stochastic SymNMF

We start by considering the following reformulation of problem (1) to the following problem:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{Y}} \quad & \frac{1}{2}\|\mathbf{X}\mathbf{Y}^T - \mathbb{E}_{\mathbf{Z}}[\mathbf{Z}]\|_F^2 \qquad (3)\\ \text{s.t.} \quad & \mathbf{X} = \mathbf{Y},\ 0 \leq \mathbf{Y} \leq \tau \end{aligned}$$

where $\mathbf{Z}$ is a symmetric matrix; $\tau > 0$ is some given constant.

Under A1, it is easy to check that when τ is sufficiently large (with a lower bound dependent on $\overline{\mathbf{Z}}$), then problem (3) is *equivalent* to problem (1), in the sense that there is a one-to-one correspondence between the stationary points of problem (1) and (3), where the stationary condition of problem (1) is given by [29, Proposition 2.1.2]

$$\left\langle \left(\mathbf{X}^*(\mathbf{X}^*)^T - \overline{\mathbf{Z}}\right)\mathbf{X}^*, \mathbf{X} - \mathbf{X}^* \right\rangle \geq 0,\ \forall \mathbf{X},\ 0 \leq \mathbf{X} \leq \tau.$$

where $\mathbf{X}^*$ denotes the stationary points. To be precise, we have the following result.

Lemma 1 *Let $\overline{\mathbf{Z}}_{i,k}$ denote the (i,k)th entry of the matrix $\overline{\mathbf{Z}}$. Under A1 – A3, suppose $\tau > \theta_k, \forall k$ where*

$$\theta_k \triangleq \frac{\overline{\mathbf{Z}}_{k,k} + \sqrt{\sum_{i=1}^N \overline{\mathbf{Z}}_{i,k}^2}}{2}, \qquad (4)$$

then a point $\mathbf{X}^*$ is a stationary point of problem (1) if and only if $\mathbf{X}^*$ is a stationary point of problem (3).

Although the objective function does not have Lipschitz continuous gradient, Theorem 1 suggests that we can solve (1) within a compact set.

2.3 The Framework of SNS for SymNMF

To this end, let us construct the augmented Lagrangian for (3), given by

$$\begin{aligned} \mathcal{L}(\mathbf{X},\mathbf{Y};\mathbf{\Lambda}) = & \frac{1}{2}\|\mathbf{X}\mathbf{Y}^T - \overline{\mathbf{Z}}\|_F^2 + \langle \mathbf{Y} - \mathbf{X}, \mathbf{\Lambda}\rangle \\ & + \frac{\rho}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2 \qquad (5) \end{aligned}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{N \times K}$ is a matrix of dual variables (or Lagrange multipliers); $\langle\cdot\rangle$ denotes the inner product operator; $\rho > 0$ is a penalty parameter whose value will be determined later.

The proposed SNS-SymNMF algorithm alternates between the primal updates of variables $\mathbf{X}$ and $\mathbf{Y}$, and the dual update for $\mathbf{\Lambda}$. We split the data samples into two groups where $\widehat{\mathbf{Z}}_1^{(i)}$ is used for updating $\mathbf{Y}$ and $\widehat{\mathbf{Z}}_2^{(i)}$ is used for $\mathbf{X}$, respectively. Our algorithm is also capable of dealing with a few different ways of aggregating the samples at each iteration:

1. A Mini-Batch of L instantaneous samples are used;
2. An aggregate of the historical samples is used;
3. A special weighted aggregate of the historical samples is used.

See Table 1 for their mathematical descriptions. In the table, t denotes the tth iteration of the algorithm;

Table 1: Rules of Aggregating Samples

Mini-batch	Aggregate	Weighted Aggregate
$\mathbf{Z}_1^{(t)} = \frac{1}{L}\sum_{i=(t-1)L+1}^{tL} \widehat{\mathbf{Z}}_1^{(i)}$	$\mathbf{Z}_1^{(t)} = \frac{1}{t}\sum_{i=1}^{t} \widehat{\mathbf{Z}}_1^{(i)}$	$\mathbf{Z}_1^{(t)} = \frac{2}{t(t+1)}\sum_{i=1}^{t} i\widehat{\mathbf{Z}}_1^{(i)}$
$\mathbf{Z}_2^{(t)} = \frac{1}{L}\sum_{i=(t-1)L+1}^{tL} \widehat{\mathbf{Z}}_2^{(i)}$	$\mathbf{Z}_2^{(t)} = \frac{1}{t}\sum_{i=1}^{t} \widehat{\mathbf{Z}}_2^{(i)}$	$\mathbf{Z}_2^{(t)} = \frac{2}{t(t+1)}\sum_{i=1}^{t} i\widehat{\mathbf{Z}}_2^{(i)}$

$\mathbf{Z}_1^{(t)}$ and $\mathbf{Z}_2^{(t)}$ are the actual (aggregated) samples used in our algorithm.

In the following, we provide the main steps of the proposed algorithm. The implementation of each step will be provided shortly. At iteration $t+1$, we first compute the objective value evaluated at the previous sample, followed by the primal updates for $\mathbf{X}$ and $\mathbf{Y}$, finally the dual variable $\mathbf{\Lambda}$ is updated. Specifically,

$$\beta^{(t)} = \frac{8}{\rho}\|\mathbf{X}^{(t)}(\mathbf{Y}^{(t)})^T - \mathbf{Z}_2^{(t-1)}\|_F^2, \tag{6a}$$

$$\mathbf{Y}^{(t+1)} = \arg\min_{0\le\mathbf{Y}\le\tau} \widehat{\mathcal{L}}_{\mathbf{Y}}(\mathbf{X}^{(t)}, \mathbf{Y}; \mathbf{\Lambda}^{(t)}; \mathbf{Z}_1^{(t)}), \tag{6b}$$

$$\mathbf{X}^{(t+1)} = \arg\min_{\mathbf{X}} \widehat{\mathcal{L}}_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}^{(t+1)}; \mathbf{\Lambda}^{(t)}; \mathbf{Z}_2^{(t)}), \tag{6c}$$

$$\mathbf{\Lambda}^{(t+1)} = \mathbf{\Lambda}^{(t)} + \rho(\mathbf{X}^{(t+1)} - \mathbf{Y}^{(t+1)}) \tag{6d}$$

where we have defined

$$\widehat{\mathcal{L}}_{\mathbf{Y}}(\mathbf{X}^{(t)}, \mathbf{Y}; \mathbf{\Lambda}^{(t)}; \mathbf{Z}_1^{(t)}) \triangleq \frac{1}{2}\|\mathbf{X}^{(t)}\mathbf{Y}^T - \mathbf{Z}_1^{(t)}\|_F^2 + \frac{\rho}{2}\|\mathbf{X}^{(t)} - \mathbf{Y} + \mathbf{\Lambda}^{(t)}/\rho\|_F^2 + \frac{\beta^{(t)}}{2}\|\mathbf{Y} - \mathbf{Y}^{(t)}\|_F^2,$$

$$\widehat{\mathcal{L}}_{\mathbf{X}}(\mathbf{X}, \mathbf{Y}^{(t+1)}; \mathbf{\Lambda}^{(t)}; \mathbf{Z}_2^{(t)}) \triangleq \frac{1}{2}\|\mathbf{X}(\mathbf{Y}^{(t+1)})^T - \mathbf{Z}_2^{(t)}\|_F^2 + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Y}^{(t+1)} + \mathbf{\Lambda}^{(t)}/\rho\|_F^2.$$

We remark that this algorithm is somewhat similar in form to the standard ADMM method applied to problem (3). The ADMM based methods lack convergence guarantees for our problem, because they only work for nonconvex problems in which primal variables are separable in the objective, which is not satisfied in our SymNMF problem [23, 24]. The key difference compared with the aforementioned algorithms is the proximal term $\|\mathbf{Y} - \mathbf{Y}^{(t)}\|_F^2$ multiplied by an *iteration dependent* penalty parameter $\beta^{(t)} \ge 0$. This proximal term makes the objective function (6b) as a tight upper bound of the augmented Lagrangian for the original Y-subproblem that does not include this term. In the convergence analysis, we will see that introducing such proximal term is critical in guaranteeing the decrease of the augmented Lagrangian as the iteration proceeds.

We also mention that using independent samples for the $\mathbf{X}$ and $\mathbf{Y}$ update is critical in the convergence analysis of the algorithm.

2.4 Implementation of the SNS-SymNMF Algorithm

The X-Subproblem. The $\mathbf{X}$-subproblem in (6c) is equivalent to the following problem

$$\min_{\mathbf{X}} \|\mathbf{D}_{\mathbf{X}}^{(t+1)} - \mathbf{X}\mathbf{A}_{\mathbf{X}}^{(t+1)}\|_F^2 \tag{7}$$

where

$$\mathbf{D}_{\mathbf{X}}^{(t+1)} \triangleq \mathbf{Z}_2^{(t)}\mathbf{Y}^{(t+1)} - \mathbf{\Lambda}^{(t)} + \rho\mathbf{Y}^{(t+1)}$$
$$\mathbf{A}_{\mathbf{X}}^{(t+1)} \triangleq (\mathbf{Y}^{(t+1)})^T\mathbf{Y}^{(t+1)} + \rho\mathbf{I} \succ 0$$

are two fixed matrices for $\mathbf{X}$-iterate. This is a simple least-squares problem whose solution is given by

$$\mathbf{X}^{(t+1)} = \mathbf{D}_{\mathbf{X}}^{(t+1)}(\mathbf{A}_{\mathbf{X}}^{(t+1)})^{-1}. \tag{8}$$

Note that the $\mathbf{A}_{\mathbf{X}}^{(t+1)}$ is a $K \times K$ matrix, where K is usually small (e.g., K represents the number of clusters in graph clustering applications).

The Y-Subproblem. To solve the $\mathbf{Y}$-subproblem in (6b), we similarly define the fixed matrices for $\mathbf{Y}$-iterate as follows,

$$\mathbf{D}_{\mathbf{Y}}^{(t)} \triangleq (\mathbf{X}^{(t)})^T\mathbf{Z}_1^{(t)} + \rho(\mathbf{X}^{(t)})^T + (\mathbf{\Lambda}^{(t)})^T + \beta^{(t)}(\mathbf{Y}^{(t)})^T,$$
$$\mathbf{A}_{\mathbf{Y}}^{(t)} \triangleq (\mathbf{X}^{(t)})^T\mathbf{X}^{(t)} + (\rho + \beta^{(t)})\mathbf{I} \succ 0.$$

Note that the subproblem (6b) can be decomposed into N separable constrained least squares problems. We may just use the conventional gradient projection for solving each one of them, using iterations

$$\mathbf{Y}_i^{(n+1)} = \mathsf{proj}_{\mathcal{Y}}(\mathbf{Y}_i^{(n)} - \alpha(\mathbf{A}_{\mathbf{Y}}^{(t)}\mathbf{Y}_i^{(n)} - \mathbf{D}_{\mathbf{Y},i}^{(t)})) \tag{9}$$

where $\mathbf{D}_{\mathbf{Y},i}$ denotes the ith column of matrix $\mathbf{D}_{\mathbf{Y}}$, α is the step size, which is chosen as $1/\lambda_{\max}(\mathbf{A}_{\mathbf{Y}}^{(t)})$, $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a matrix, n denotes the iteration of the inner loop, $\mathsf{proj}_{\mathcal{Y}}(\mathbf{w})$ denotes the projection of a given vector $\mathbf{w}$ to the feasible set of $\mathbf{Y}_i$. Other efficient methods of solving (6b) can be also applied, such as active set algorithms [30].

The SNS-SymNMF Algorithm. Leveraging the efficient calculation of $\mathbf{Y}^{(t+1)}$ and $\mathbf{X}^{(t+1)}$, we summarize the algorithm as shown in Algorithm 1, where T denotes the total number of iterations.

Algorithm 1 The SNS-SymNMF Algorithm

```
Input: $\mathbf{Y}^{(1)}$, $\mathbf{X}^{(1)}$, $\mathbf{\Lambda}^{(1)}$, and $\rho$
for $t = 1, \ldots, T$ do
    Update $\beta^{(t)}$ according to (6a)
    Select data using Table 1
    Update $\mathbf{Y}^{(t+1)}$ by solving (6b)
    Update $\mathbf{X}^{(t+1)}$ using (8)
    Update $\mathbf{\Lambda}^{(t+1)}$ using (6d)
end for
Output: Iterate $\mathbf{Y}^{(r)}$ chosen uniformly random from $\{\mathbf{Y}^{(t)}\}_{t=1}^{T}$.
```

3 Convergence Analysis

The convergence analysis is built upon a series of lemmas (shown in the supplemental materials), which characterize the relationship among the augmented Lagrangian, the primal/dual variables as well as the random samples.

First we construct a function that measures the optimality of the iterates $\{\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}, \mathbf{\Lambda}^{(t)}\}$. Define the *proximal gradient* of the augmented Lagrangian function as

$$\widetilde{\nabla}\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{\Lambda}) \triangleq \begin{bmatrix} \mathbf{Y}^{T} - \text{proj}_{\mathcal{Y}}[\mathbf{Y}^{T} - \nabla_{\mathbf{Y}}(\mathcal{L}(\mathbf{Y}, \mathbf{X}, \mathbf{\Lambda})] \\ \nabla_{\mathbf{X}}\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{\Lambda}) \end{bmatrix}$$

where the operator

$$\text{proj}_{\mathcal{Y}}(\mathbf{W}) \triangleq \arg \min_{0 \le \mathbf{Y} \le \tau} \|\mathbf{W} - \mathbf{Y}\|_F^2 \qquad (10)$$

i.e., it is the projection operator that projects a given matrix $\mathbf{W}$ onto the feasible set of $\mathbf{Y}$. Here we propose to use the following quantity to measure the progress of the algorithm [17, 23]

$$\mathcal{P}(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}, \mathbf{\Lambda}^{(t)}) \triangleq \|\widetilde{\nabla}\mathcal{L}(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}, \mathbf{\Lambda}^{(t)})\|_F^2 + \|\mathbf{X}^{(t)} - \mathbf{Y}^{(t)}\|_F^2. \qquad (11)$$

It can be verified that if $\lim_{t\to\infty} \mathcal{P}(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}, \mathbf{\Lambda}^{(t)}) = 0$, then a stationary point of the problem (3) is obtained.

The key point of the proof is to quantify the optimality gap in (11). First, we bound the successive difference of the multipliers by that of the successive difference of the primal variables and samples. Second, we prove the augmented Lagrangian decreases in every iteration and lower bounded. Finally, we check the optimality gap that is quantified by some constant over T(or L). These steps represent a major departure from the more traditional steps for proving ADMM-type algorithms, which only work for convex cases.

We also remark the convergence proof of SNS-SymNMF is different from the work in [28]. Here we start from the proof of the boundness of the $\mathbf{X}$-iterate, then the convergence of the algorithm to stationary points can be characterized.

Theoretical Results. First, when a mini-batch of samples are used at each iteration, we have the following result.

Theorem 1 *Suppose A1 – A3 hold true. Then the iterates generated by the SNS-SymNMF algorithm with Mini-Batch samples satisfy the following relation*

$$\mathbb{E}[\mathcal{P}_{\textit{Mini-Batch}}(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}, \mathbf{\Lambda}^{(r)})] \le \frac{1}{T}\mathcal{C}(\mathcal{U} + \frac{\sigma^2}{L}) + \frac{\mathcal{W}\sigma^2}{L}$$

where $\mathcal{C}, \mathcal{U}, \mathcal{W}$ are some constants

Theorem 1 says that using the Mini-Batch samples the SNS-SymNMF algorithm converges sublinearly to a ball of size $\mathcal{W}\sigma^2/L$ around the stationary points of problem (3). Further, the radius of the ball can be reduced when increasing the number of samples L.

Second, if all the past samples are averaged using the same weight, then the algorithm can converge to the stationary points of the stochastic SymNMF problem.

Theorem 2 *Suppose A1 – A3 hold true and the following is satisfied*

$$\rho > 8NK\tau^2. \qquad (12)$$

Then the following statements are true for SNS-SymNMF with averaged samples:

1. *The equality constraint is satisfied in the limit, i.e.,* $$\lim_{t\to\infty} \mathbb{E}[\|\mathbf{X}^{(t)} - \mathbf{Y}^{(t)}\|_F^2] \to 0.$$
2. *The sequence $\{\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}, \mathbf{\Lambda}^{(t)}\}$ is bounded, and every limit point of the sequence is a stationary point of problem* (3)*.*

Below we show that the gap $\mathbb{E}[\mathcal{P}(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}, \mathbf{\Lambda}^{(r)})]$ goes to zero in mean-square sublinearly.

Theorem 3 *Suppose A1 – A3 hold true. Then the iterates generated by the SNS-SymNMF algorithm with aggregate samples satisfy the following relation*

$$\mathbb{E}[\mathcal{P}_{\textit{aggregate}}(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}, \mathbf{\Lambda}^{(r)})] \le \frac{\mathcal{C}\mathcal{S} + \mathcal{C}\sigma^2 + \mathcal{K}\sigma^2}{T}$$

where $\mathcal{C}, \mathcal{S}, \mathcal{K}$ are some constants.

Theorem 2 and Theorem 3 show that the stochastic SymNMF can converge to a stationary point of (3) in mean-square, and in a sublinear manner. Then, we have the following corollary directly.

Corollary 1 *Suppose A1 – A3 hold true. Then the iterates generated by the SNS-SymNMF algorithm with weighted aggregate samples satisfy the following relation*

$$\mathbb{E}[\mathcal{P}_{\mathit{weighted}}(\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}, \mathbf{\Lambda}^{(r)})] \leq \frac{\mathcal{CS} + \mathcal{C}\sigma^2 + \mathcal{K}'\sigma^2}{T}$$

where $\mathcal{K}' \geq \mathcal{K}$.

We remark that those constants, such as $\mathcal{C}, \mathcal{U}, \mathcal{W}, \mathcal{S}, \mathcal{K}$, mentioned in the theorems are only dependent on the initialization of the algorithm and parameters of given problems, such as $N, K, \tau, \mathcal{Z}$. The explicit expressions of the constants can be found in the supplemental materials.

It is worth noting that when $\sigma^2 = 0$, our convergence analysis of the SNS-SymNMF algorithm still holds true for the deterministic case [28].

We also remark that given a required error, when the dimension of the problems increases, the stochastic algorithms need a more total number of iterations to achieve this error.

4 Numerical Results

4.1 Synthetic Data Set

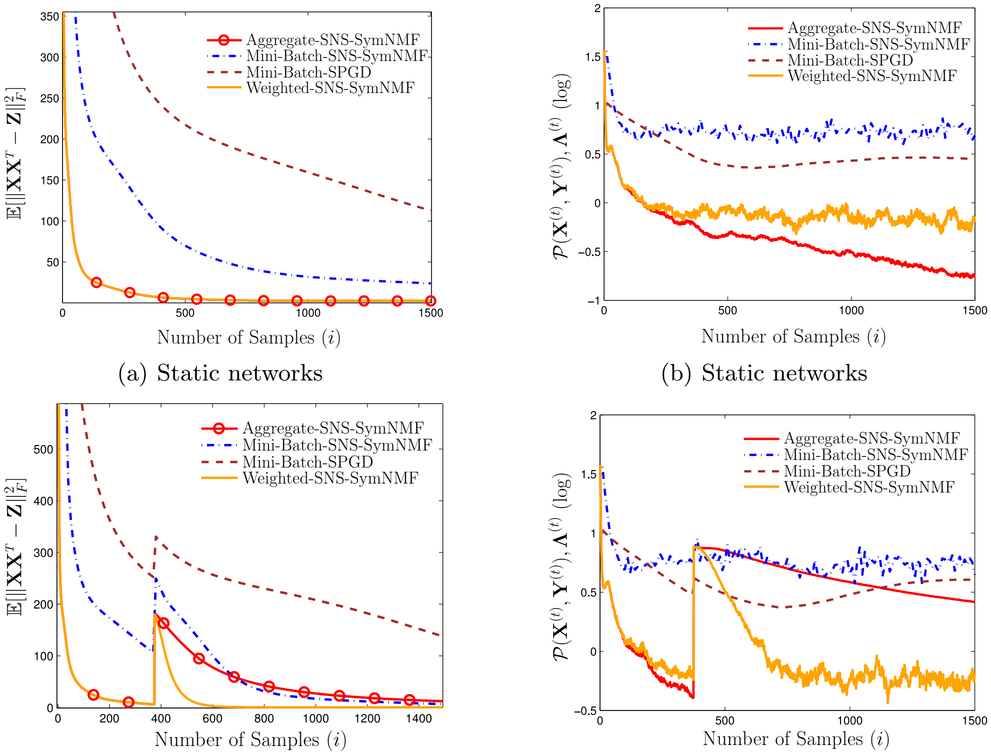
Data Set Description. We use a similar random graph as adopted in [8] for spectral clustering. The graph is generated as follows. For each time slot, data points $\{x_i\} \in \mathbb{R}$, $i = 1, \ldots, N$, are generated in one dimension. We specify 4 clusters. The numbers of data points in each cluster are 12, 24, 48 and 36. Within each cluster, data points follow an *i.i.d.* Gaussian distribution. The means of the random variables in these 4 clusters are $2, 4, 6, 8$, respectively, and the variance is 0.5 for all distributions. Then, construct the similarity matrix $\widehat{\mathbf{Z}}_1^{(i)} \in \mathbb{R}^{N\times N}$ (or $\widehat{\mathbf{Z}}_2^{(i)}$), whose (i, j)th entry is determined by the Gaussian function $\exp(-(x_i - x_j)^2/(2\sigma^2))$ where $\sigma^2 = 0.5$. Finally, we repeat the process mentioned above to generate a series of adjacency matrices for the community detection problem. The mean of the adjacency matrix represents the ground truth of the connections among the nodes and variance measures the uncertainty of each sample. Based on this model, we know that the weights between two points which belong to the same cluster are very likely higher than the weights between two points which belong to different clusters.

Algorithms Comparison. Each point in Figure 1 is an average of 20 independent Monte Carlo (MC) trials. All algorithms are started with the same initial point each time, and the entries of the initialized $\mathbf{X}$ (or $\mathbf{Y}$) follow an *i.i.d.* uniform distribution in the range $[0, \tau]$. Mini-Batch SPGD [17] is applied to solve problem (3) where the step-size α is 0.01. Note that this algorithm cannot be directly applied to solve problem (1) due to the lack of Lipschitz continuous gradient. The proposed SNS-SymNMF uses two groups of data at each iteration, while Mini-Batch-SPGD only needs one. For fair comparison, in the simulation Mini-Batch-SPGD uses $(\mathbf{Z}_1^{(t)} + \mathbf{Z}_2^{(t)})/2$ as the input sample. Also, when the Mini-Batch strategy is used, the algorithms perform updates every L independent samples, where L is fixed.

We remark that in the implementation of SNS-SymNMF we let $\tau = \max_k \theta_k$, and gradually increase the value of ρ from an initial value to meet condition (12) for accelerating the convergence rate [31]. Here, the choice of ρ follows $\rho^{(t+1)} = \min\{\rho^{(t)}/(1 - \epsilon/\rho^{(t)}), 8.1NK\tau^2\}$ where $\epsilon = 10^{-3}$ as suggested in [32], and $\rho^{(1)} = N\tau$. To update $\mathbf{Y}$, we use the block pivoting method [30].

The SNS-SymNMF algorithm is performed using different data sampling rules. From Figure 1(a), it is shown that the aggregate-SNS-SymNMF algorithm converges faster than Mini-Batch-SPGD and Mini-Batch-SNS-SymNMF since the variance of samples is reduced by the aggregated data. The weighted-SNS-SymNMF algorithm is slightly slower than aggregate-SNS-SymNMF, but still presents a sublinear convergence rate. As shown in Figure 1(b), the optimality gap plateaus in Mini-Batch-SNS-SymNMF and Mini-Batch-SPGD due to the sample aggregation rules, which is consistent with the theoretical analysis shown in Theorem 1. The optimality gap of Mini-Batch-SNS-SymNMF is larger than that of Mini-Batch-SPGD, since the number of samples used for each block is only a half of Mini-Batch-SPGD. Here, to get rid of the effect of the dimension of $\mathbf{Z}$, we use $\|\mathbf{X} - \mathsf{proj}_+[\mathbf{X} - \nabla_{\mathbf{X}}(f(\mathbf{X}))]\|_\infty$ as the optimality gap, where proj_+ denotes the nonnegative projection operator.

The convergence behaviors for dynamic networks are shown in Figure 1(c) and Figure 1(d), where the means of the random variables in the 4 clusters are changed to $1, 7, 3, 5$ at the 400th sample. Aggregate-SNS-SymNMF performs worse than weighted-SNS-SymNMF because of the aggregated errors. Although Mini-Batch-SNS-SymNMF and Mini-Batch-SPGD can adapt to the network topology variation, constant optimality gaps still remain as can be observed in Figure 1(d). For the weighted-SNS-SymNMF algorithm, since more weights are given to the current data samples, the change of the network topology can be tracked. Therefore, weighted-SNS-SymNMF can still give a very low objective value after the 400th sample



(a) Static networks

(b) Static networks

(c) Dynamic networks

(d) Dynamic networks

Figure 1: The convergence behaviors. The parameters are $K = 4$; $N = 120$; $L = 10$. The x-axis represents the total number of observed samples.

compared with other algorithms.

We also compare the performance of the SNS-SymNMF algorithm and the deterministic SymNMF algorithm where the samples are replaced by $\overline{\mathbf{Z}}$ in SNS-SymNMF. The results are shown in Figure 2. It can be observed that the SNS-SymNMF algorithm has a similar convergence rate with NS-SymNMF in terms of the objective values. However, deterministic SymNMF has a faster convergence rate than SNS-SymNMF with respective to the optimality gap, which is expected, since deterministic SymNMF uses the mean of the adjacency matrix without any uncertainty.

4.2 Real Data Set

Data Set Description. we use the 6th subset of the processed topic detection and tracking (TDT2) data set with 10 classes[1] which includes 3050 documents and each of them has 36771 features. The adjacency matrix is constructed by the self-tuning method [33], where the weight between the ith sample and the jth one is given by $w_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/(\sigma_i\sigma_j)), \forall i \neq j$. The local scale σ_i is computed by the Euclidean distance between $\mathbf{x}_i$ and its $\widehat{k}$th neighbor, where $\mathbf{x}_i$ denotes the ith document vector which is normalized to have unit 2-norm and $i = 1, \ldots, N$. We use $\widehat{k} = 7$ as suggested in [33] and enforce $w_{i,i} = 0, \forall i$. Then the (i, j)th entry of the similarity matrix $\widehat{\mathbf{Z}}_1^{(i)}$ (or $\widehat{\mathbf{Z}}_2^{(i)}$) is computed as in the normalized cut [8] which is $d_i^{-1/2} w_{i,j} d_j^{-1/2}$ where $d_i = \sum_{i'}^N w_{i,i'}, \forall i'$.

In order to mimic the stochastic setting, we select 5 classes that have larger number of documents than the others in the 6th subset of TDT2. The total numbers of documents in these 5 classes are 1843, 440, 226, 144, and 103. Then, for each time slot, we uniformly pick up 100, 50, 45, 15, 30 documents from the selected 5 classes to form $\widehat{\mathbf{Z}}_1^{(i)}$, and then independently perform the same sampling process again to form $\widehat{\mathbf{Z}}_2^{(i)}$. The average of all samples is considered as the true mean (i.e., $\overline{\mathbf{Z}}$) for NS-SymNMF. The variance of samples in this case is $\sigma^2 = 32.32$.

Algorithms Comparison. The simulation results shown in Figure 3 are based on 20 MC trials. It can be observed that Mini-Batch algorithms converge slowly compared with aggregated/weighted SNS-SymNMF and NS-SymNMF, since Mini-Batch algorithms only

[1] see http://www.cad.zju.edu.cn/home/dengcai/Data /TextData.html.

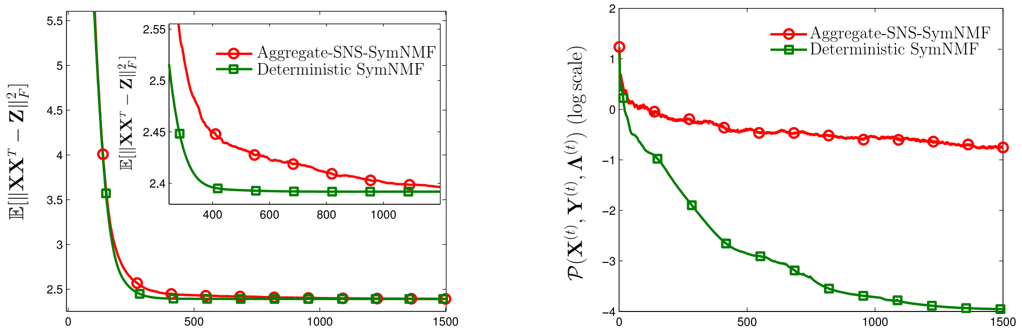


Figure 2: The convergence behaviors. The parameters are $K = 4$; $N = 120$; $L = 10$. The x-axis represents the total number of the observed samples for stochastic SymNMF and iterations for deterministic SymNMF.

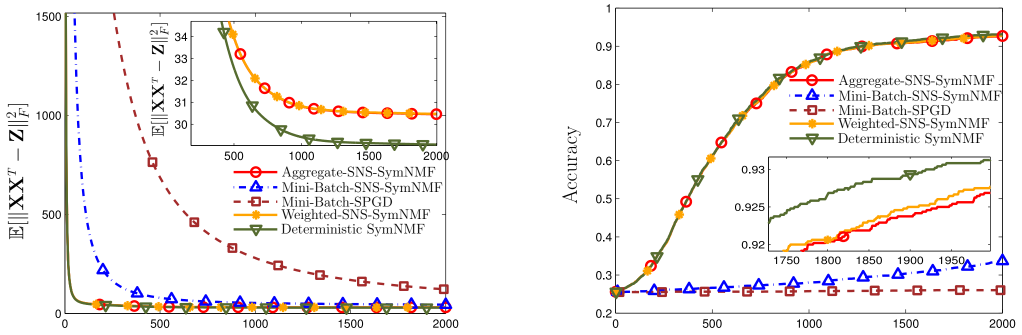


Figure 3: The convergence behaviors. The parameters are $K = 5$; $N = 240$; $L = 10$. The x-axis represents the total number of observed samples for stochastic SymNMF and iterations for deterministic SymNMF.

use a subset of samples. Although NS-SymNMF shows a lower objective value than SNS-SymNMF, it is interesting to see that SNS-SymNMF has a similar convergence rate as NS-SymNMF in terms of the objective values with only a small difference. Furthermore, the accuracy obtained by NS-SymNMF and aggregated/weighted SNS-SymNMF is only slightly different during the whole process as the algorithms proceed. Therefore, the new variant of SymNMF, SNS-SymNMF, can be considered as an online algorithm that deals with clustering problems, which is not only processing the real-time data sequentially but also can provide accurate clustering results[2].

Finally, we remark that the previous literatures [2, 3] have already shown the advantages of deterministic SymNMF in terms of clustering accuracy compared with classic methods, such as K-means variants, NMF variants, spectral clustering variants. Here, we focus on the stochastic setting for SymNMF and omit the accuracy results for other methods.

We also remark that in this paper we just adopt a very simple version of Mini-Batch methods. The main purpose is to take the Mini-Batch methods as the counterparts for the average/weighted aggregation rules and to show the impact of the variance of samples on performance of algorithms. Actually, there is a tradeoff on selecting the length L as the Mini-Batch algorithm proceeds. A more reasonable way of choosing L is discussed in [17] and more variants of Mini-Batch algorithms for stochastic SymNMF could be considered as the future work.

5 Conclusion

In this paper, the stochastic SymNMF problem is considered in the areas of clustering and community detection. We show that the proposed stochastic nonconvex splitting algorithm converges to the set of stationary points of SymNMF in a sublinear manner. Numerical experiments show that the proposed method has a similar convergence rate and clustering accuracy as deterministic SymNMF does.

Acknowledgements

Research in this work was supported in part by NSF under Grants No. 1523374 and No. 1526078, and by AFOSR under Grant No. 15RT0767.

[2] More simulations related to the computational time, impact of sample variance, and parameter tuning are shown in the supplemental materials, where the numerical results with larger networks are also included.

References

[1] Zhaoshui He, Shengli Xie, R. Zdunek, Guoxu Zhou, and A. Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, Dec. 2011.

[2] Kejun Huang, Nicholas D. Sidiropoulos, and Ananthram Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, Jan. 2014.

[3] Da Kuang, Sangwoon Yun, and Haesun Park. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, July 2015.

[4] Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.

[5] Marshall Hall and Morris Newman. Copositive and completely positive quadratic forms. *Mathematical Proceedings of the Cambridge Philosophical Society*, 59(02):329–339, 1963.

[6] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, May 2011.

[7] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Proc. of Neural Information Processing Systems (NIPS)*, 2:849–856, 2002.

[8] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[9] Songtao Lu and Zhengdao Wang. Accelerated algorithms for eigen-value decomposition with application to spectral clustering. In *Proc. of Asilomar Conf. Signals, Systems and Computers*, pages 355–359, Nov. 2015.

[10] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In *Proc. of Neural Information Processing Systems (NIPS)*, pages 1601–1608, 2005.

[11] Daniel L Sussman, Minh Tang, Donniell E Fishkind, and Carey E Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.

[12] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proc. of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pages 665–674, 2013.

[13] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proc. of Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 704–711, 2010.

[14] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 670–688, 2015.

[15] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *Journal of Machine Learning Research*, 11(May):1601–1604, 2010.

[16] Meisam Razaviyayn, Maziar Sanjabi, and Zhi-Quan Luo. A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. *Mathematical Programming*, 157(2):515–545, 2016.

[17] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

[18] D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proc. of SIAM Int. Conf. Data Mining*, pages 106–117, 2012.

[19] D.L. Sun and C. Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Proc. of IEEE Int. Conf. Acoustics Speech and Signal Process (ICASSP)*, pages 6201–6205, May 2014.

[20] Kejun Huang, Nicholas D. Sidiropoulos, and Athanasios P. Liavas. A flexible and effcient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, June 2016.

[21] R. Zhao and V. Y. F. Tan. Online nonnegative matrix factorization with outliers. *IEEE Transactions on Signal Processing*, 65(3):555–570, Feb. 2017.

[22] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

[23] M. Hong, Z.-Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[24] G. Li and T.-K Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

[25] Y. Wang and J. Zeng W. Yin. Global convergence of ADMM in nonconvex nonsmooth optimization. *arXiv Preprint, arXiv:1511.06324*, 2015.

[26] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *Proc. of the 30th International Conference on Machine Learning*, pages 80–88, 2013.

[27] Samaneh Azadi and Suvrit Sra. Towards an optimal stochastic alternating direction method of multipliers. In *Proc. of the 31st International Conference on Machine Learning*, pages 620–628, 2014.

[28] Songtao Lu, Mingyi Hong, and Zhengdao Wang. A nonconvex splitting method for symmetric nonnegative matrix factorization: Convergence analysis and optimality. *IEEE Transactions on Signal Processing*, 2017.

[29] D. P. Bertsekas. *Nonlinear Programming, 2nd ed.* Athena Scientific, Belmont, MA, 1999.

[30] Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

[31] Meisam Razaviyayn, Mingyi Hong, Zhi-Quan Luo, and Jong-Shi Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. In *Proc. of Neural Information Processing Systems (NIPS)*, pages 1440–1448, 2014.

[32] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J. S. Pang. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Transactions on Signal Processing*, 62(3):641–656, Feb. 2014.

[33] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Proc. of Neural Information Processing Systems (NIPS)*, 2004.