A BLOCK COORDINATE DESCENT METHOD FOR REGULARIZED MULTI-CONVEX OPTIMIZATION WITH APPLICATIONS TO NONNEGATIVE TENSOR FACTORIZATION AND COMPLETION

YANGYANG XU* AND WOTAO YIN*

Abstract. This paper considers regularized block multi-convex optimization, where the feasible set and objective function are generally non-convex but convex in each block of variables. We review some of its interesting examples and propose a generalized block coordinate descent method. Under certain conditions, we show that any limit point satisfies the Nash equilibrium conditions. Furthermore, we establish its global convergence and estimate its asymptotic convergence rate by assuming a property based on the Kurdyka-Łojasiewicz inequality. The proposed algorithms are adapted for factorizing nonnegative matrices and tensors, as well as completing them from their incomplete observations. The algorithms were tested on synthetic data, hyperspectral data, as well as image sets from the CBCL and ORL databases. Compared to the existing state-of-the-art algorithms, the proposed algorithms demonstrate superior performance in both speed and solution quality. The Matlab code of nonnegative matrix/tensor decomposition and completion, along with a few demos, are accessible from the authors' homepages.

Key words. block multi-convex, block coordinate descent method, Kurdyka-Łojasiewicz inequality, Nash equilibrium, nonnegative matrix and tensor factorization, matrix completion, tensor completion, proximal gradient method

1. Introduction.

In this paper, we consider the optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}_1,\cdots,\mathbf{x}_s) \equiv f(\mathbf{x}_1,\cdots,\mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i), \tag{1.1}$$

where variable $\mathbf{x}$ is decomposed into s blocks $\mathbf{x}_1,\cdots,\mathbf{x}_s$, the set $\mathcal{X}$ of feasible points is assumed to be a closed and *block multi-convex* subset of $\mathbb{R}^n$, f is assumed to be a differentiable and *block multi-convex* function, and r_i, $i=1,\cdots,s$, are extended-value convex functions. Set $\mathcal{X}$ and function f can be non-convex over $\mathbf{x}=(\mathbf{x}_1,\cdots,\mathbf{x}_s)$.

We call a set $\mathcal{X}$ *block multi-convex* if its projection to each block of variables is convex, namely, for each i and fixed $(s-1)$ blocks $\mathbf{x}_1,\cdots,\mathbf{x}_{i-1},\mathbf{x}_{i+1},\cdots,\mathbf{x}_s$, the set

$$\mathcal{X}_i(\mathbf{x}_1,\cdots,\mathbf{x}_{i-1},\mathbf{x}_{i+1},\cdots,\mathbf{x}_s) \triangleq \{\mathbf{x}_i\in\mathbb{R}^{n_i} : (\mathbf{x}_1,\cdots,\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1},\cdots,\mathbf{x}_s)\in\mathcal{X}\} \tag{1.2}$$

is convex. We call a function f is *block multi-convex* if for each i, f is a convex function of $\mathbf{x}_i$ while all the other blocks are fixed. Therefore, when all but one blocks are fixed, (1.1) over the free block is a convex problem. (Later, using the proximal update (1.3b), we allow f to be non-convex over a block.)

Extended value means $r_i(\mathbf{x}_i)=\infty$ if $\mathbf{x}_i\notin\mathrm{dom}(r_i)$, $i=1,\cdots,s$. In particular, r_i (or a part of it) can be indicator functions of convex sets. We use $\mathbf{x}\in\mathcal{X}$ to model joint constraints and $r_1,\ldots,r_s$ to include individual constraints of $\mathbf{x}_1,\cdots,\mathbf{x}_s$, when they are present. In addition, r_i can include nonsmooth functions.

Our main interest is the *block coordinate descent* (BCD) method of the Gauss-Seidel type, which minimizes F cyclically over each of $\mathbf{x}_1,\cdots,\mathbf{x}_s$ while fixing the remaining blocks at their last updated values. Let $\mathbf{x}_i^k$ denote the value of $\mathbf{x}_i$ after its kth update, and

$$f_i^k(\mathbf{x}_i) \triangleq f(\mathbf{x}_1^k,\cdots,\mathbf{x}_{i-1}^k,\mathbf{x}_i,\mathbf{x}_{i+1}^{k-1},\cdots,\mathbf{x}_s^{k-1}),\ \text{for all } i \text{ and } k.$$

*yangyang.xu@rice.edu and wotao.yin@rice.edu. Department of Applied and Computational Mathematics, Rice University, Houston, Texas.

At each step, we consider three different updates

$$\textit{Original}: \quad \mathbf{x}_i^k = \underset{\mathbf{x}_i \in \mathcal{X}_i^k}{\operatorname{argmin}} f_i^k(\mathbf{x}_i) + r_i(\mathbf{x}_i), \tag{1.3a}$$

$$\textit{Proximal}: \quad \mathbf{x}_i^k = \underset{\mathbf{x}_i \in \mathcal{X}_i^k}{\operatorname{argmin}} f_i^k(\mathbf{x}_i) + \frac{L_i^{k-1}}{2}\|\mathbf{x}_i - \mathbf{x}_i^{k-1}\|^2 + r_i(\mathbf{x}_i), \tag{1.3b}$$

$$\textit{Prox-linear}: \quad \mathbf{x}_i^k = \underset{\mathbf{x}_i \in \mathcal{X}_i^k}{\operatorname{argmin}} \langle \hat{\mathbf{g}}_i^k, \mathbf{x}_i - \hat{\mathbf{x}}_i^{k-1}\rangle + \frac{L_i^{k-1}}{2}\|\mathbf{x}_i - \hat{\mathbf{x}}_i^{k-1}\|^2 + r_i(\mathbf{x}_i), \tag{1.3c}$$

where $\|\cdot\|$ denotes the ℓ_2 norm, $L_i^{k-1} > 0$,

$$\mathcal{X}_i^k = \mathcal{X}_i(\mathbf{x}_1^k, \cdots, \mathbf{x}_{i-1}^k, \mathbf{x}_{i+1}^{k-1}, \cdots, \mathbf{x}_s^{k-1})$$

and in the last type of update (1.3c),

$$\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^{k-1} + \omega_i^{k-1}(\mathbf{x}_i^{k-1} - \mathbf{x}_i^{k-2}) \tag{1.4}$$

denotes an extrapolated point, $\omega_i^{k-1} \geq 0$ is the extrapolation weight, $\hat{\mathbf{g}}_i^k = \nabla f_i^k(\hat{\mathbf{x}}_i^{k-1})$ is the block-partial gradient of f at $\hat{\mathbf{x}}_i^{k-1}$. We consider extrapolation (1.4) for update (1.3c) since it significantly accelerates the convergence of BCD in our applications. The framework of BCD is given in Alg. 1, which allows each $\mathbf{x}_i$ to be updated by (1.3a), (1.3b), or (1.3c).

Algorithm 1 Block coordinate descent method for solving (1.1)

Initialization: choose initial two points $(\mathbf{x}_1^{-1}, \cdots, \mathbf{x}_s^{-1}) = (\mathbf{x}_1^0, \cdots, \mathbf{x}_s^0)$
for $k = 1, 2, \cdots$ **do**
  for $i = 1, 2, \cdots, s$ **do**
    $\mathbf{x}_i^k \leftarrow$ (1.3a), (1.3b), or (1.3c).
  end for
  if stopping criterion is satisfied **then**
    return $(\mathbf{x}_1^k, \cdots, \mathbf{x}_s^k)$.
  end if
end for

Since $\mathcal{X}$ and f are block multi-convex, all three subproblems in (1.3) are convex. In general, the three updates generate different sequences and can thus cause BCD to converge to different solutions. We found in many tests, applying (1.3c) on all or some blocks give solutions of lower objective values, for a possible reason that its local prox-linear approximation help avoid the small regions around certain local minima. In addition, it is generally more time consuming to compute (1.3a) and (1.3b) than (1.3c) though each time the former two tend to make larger objective decreases than applying (1.3c) without extrapolation. We consider all of the three updates since they fit different applications, and also different blocks in the same application, yet their convergence can be analyzed in a unified framework.

To ensure the convergence of Alg. 1, for every block i to which (1.3a) is applied, we require $f_i^k(\mathbf{x}_i)$ to be strongly convex, and for every block i to which (1.3c) is applied, we require $\nabla f_i^k(\mathbf{x}_i)$ to be Lipschitz continuous. The parameter L_i^k in both (1.3b) and (1.3c) can be fixed for all k. For generality and faster convergence, we allow it to change during the iterations. Use of (1.3b) only requires L_i^k to be uniformly lower bounded from *zero* and uniformly upper bounded. In fact, f_i^k in (1.3b) can be *nonconvex*, and our proof still goes through. (1.3b) is a good replacement of (1.3a) if f_i^k is not strongly convex. Use of (1.3c) requires more conditions on L_i^k; see Lemmas 2.2 and 2.6. (1.3c) is relatively easy to solve and often allows

closed form solutions. For block i, (1.3c) is prefered over (1.3a) and (1.3b) when they are expensive to solve and f_i^k has Lipschitz continuous gradients. Overall, the three choices cover a large number of cases.

Original subproblem (1.3a) is the most-used form in BCD and has been extensively studied. It dates back to methods in [52] for solving equation systems and to works [5, 24, 61, 70], which analyze the method assuming F to be convex (or quasiconvex or hemivariate), differentiable, and have bounded level sets except for certain classes of convex functions. When F is non-convex, BCD may cycle and stagnate [56]. However, subsequence convergence can be obtained for special cases such as quadratic function [48], strictly pseudo-convexity in each of $(s-2)$ blocks [22], unique minimizer per block [47], p.195. If F is non-differentiable, BCD can get stuck at a non-stationary point; see [5] p.94. However, subsequence convergence can be obtained if the non-differentiable part is separable; see works [23, 50, 65, 66] for results on different forms of F. In our objective function, f is differentiable and possibly non-convex, and the nonsmooth part is block-separable functions r_i.

Proximal subproblem (1.3b) has been used with BCD in [22]. For $\mathcal{X} = \mathbb{R}^n$, their work shows that every limit point is a critical point. Recently, this method is revisited in [4] for only two blocks and shown to converge globally via the Kurdyka-Łojasiewicz (KL) inequality.

Prox-linear subproblem (1.3c) with extrapolation is new but very similar to the update in the block-coordinate gradient descent (BGD) method of [67], which identifies a block descent direction by gradient projection and then performs an Armijo-type line search. [67] does not use extrapolation (1.4). Their work considers more general functions f which are smooth but not necessarily multi-convex, but it does not consider joint constraints. While we are preparing the paper, [57] provides a unified convergence analysis of coordinatewise successive minimization methods for nonsmooth nonconvex optimization. Those methods update block variables by minimizing a surrogate function that dominates the original objective around the current iterate. They do not use extrapolation either and only have subsequence convergence.

There are examples of r_i that make (1.3c) easier to compute than (1.3a) and (1.3b). For instance, if $r_i = \delta_{\mathcal{D}_i}$ the indicator function of convex set $\mathcal{D}_i$ (equivalent to $\mathbf{x}_i \in \mathcal{D}_i$), (1.3c) reduces to $\mathbf{x}_i^k = \mathcal{P}_{\mathcal{X}_i^k \cap \mathcal{D}_i}\left(\hat{\mathbf{x}}_i^{k-1} - \hat{\mathbf{g}}_i^{k-1}/L_i^{k-1}\right)$, where $\mathcal{P}_{\mathcal{X}_i^k \cap \mathcal{D}_i}$ is the project to set $\mathcal{X}_i^k \cap \mathcal{D}_i$. If $r_i(\mathbf{x}_i) = \lambda_i \|\mathbf{x}_i\|_1$ and $\mathcal{X}_i^k = \mathbb{R}^{n_i}$, (1.3c) reduces to $\mathbf{x}_i^k = \mathcal{S}_{L_i^{k-1}/\lambda_i}\left(\hat{\mathbf{x}}_i^{k-1} - \hat{\mathbf{g}}_i^{k-1}/L_i^{k-1}\right)$, where $\mathcal{S}_\nu(\cdot)$ is soft-thresholding defined component-wise as $\mathcal{S}_\nu(t) = \text{sign}(t)\max(|t|-\nu, 0)$. More examples arise in joint/group ℓ_1 and nuclear norm minimization, total variation, etc.

1.1. Contributions. We propose Alg. 1 and establish its global convergence. The algorithm is applied to two classes problems (i) nonnegative matrix/tensor *factorization* and (ii) nonnegative matrix/tensor *completion* from incomplete observations, and is demonstrated superior than the state-of-the-arts on both synthetic and real data in both speed and solution quality.

Our convergence analysis takes two steps. Under certain assumptions, the first step establishes the square summable result $\sum_k \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 < \infty$ and obtains subsequence convergence to Nash equilibrium points, as well as global convergence to a single Nash point if the sequence is bounded and the Nash points are isolated. The second step assumes the KL inequality [13, 14] and improves the result to $\sum_k \|\mathbf{x}^k - \mathbf{x}^{k+1}\| < \infty$, which gives the algorithm global convergence, as well as asymptotic rates of convergence. The classes of functions that obey the KL inequality are reviewed. Despite the popularity of BCD, very few works establish global convergence without the (quasi)convexity assumption on F; works [48, 67] have obtained global convergence by assuming a local Lipschitzian error bound and the isolation of the isocost surfaces of F. Some very interesting problems satisfy their assumptions. Their and our assumptions do not contain each other, though there are problems satisfying both.

1.2. Applications. A large number of practical problems can be formulated in the form of (1.1) such as convex problems: (group) Lasso [64,75] or the basis pursuit (denoising) [15], low-rank matrix recovery [58], hybrid huberized support vector machine [69], and so on. We give some non-convex examples as follows.

Blind source separation and sparse dictionary learning. Let $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathbb{R}^{1\times p}$ be a set of source signals. Given m sensor signals $\mathbf{x}_i = \sum_{j=1}^n a_{ij}\mathbf{s}_j + \boldsymbol{\eta}_i, i = 1, \cdots, m$, where $\mathbf{A} = [a_{ij}]_{m\times n} \in \mathbb{R}^{m\times n}$ is an *unknown* mixing matrix and $\boldsymbol{\eta}_i$ is noise, blind source separation (BSS) [27] aims to estimate both $\mathbf{A}$ and $\mathbf{S} = [\mathbf{s}_1^\top, \ldots, \mathbf{s}_n^\top]^\top$. It has found applications in many areas such as artifact removal [26] and image processing [28]. Two classical approaches for BSS are principle component analysis (PCA) [62] and independent component analysis (ICA) [18]. If $m < n$ and no prior information on $\mathbf{A}$ and $\mathbf{S}$ is given, these methods will fail. Assuming $\mathbf{s}_1, \cdots, \mathbf{s}_n$ are sparse under some dictionary $\mathbf{B} \in \mathbb{R}^{T\times p}$, namely, $\mathbf{s}_i = \mathbf{y}_i\mathbf{B}$ and $\mathbf{y}_i \in \mathbb{R}^{1\times T}$ is sparse for $i = 1, \ldots, n$, [12, 79] use the sparse BSS model

$$\min_{\mathbf{A},\mathbf{Y}} \frac{\lambda}{2}\|\mathbf{AYB} - \mathbf{X}\|_F^2 + r(\mathbf{Y}), \text{ subject to } \mathbf{A} \in \mathcal{D} \tag{1.5}$$

where $\mathbf{Y} = [\mathbf{y}_1^\top, \cdots, \mathbf{y}_n^\top]^\top \in \mathbb{R}^{n\times T}$, $r(\mathbf{Y})$ is a sparsity regularizer such as $r(\mathbf{Y}) = \|\mathbf{Y}\|_1$, $\mathcal{D}$ is a convex set to control the scale of $\mathbf{A}$ such as $\|\mathbf{A}\|_F \le 1$, and λ is a balancing parameter. Note that model (1.5) is block multi-convex in $\mathbf{A}$ and $\mathbf{Y}$ each but jointly non-convex. A similar model appears in cosmic microwave background analysis [10] which solves

$$\min_{\mathbf{A},\mathbf{Y}} \frac{\lambda}{2}\text{trace}\left((\mathbf{AYB} - \mathbf{X})^\top\mathbf{C}^{-1}(\mathbf{AYB} - \mathbf{X})\right) + r(\mathbf{Y}), \text{ subject to } \mathbf{A} \in \mathcal{D} \tag{1.6}$$

for a certain covariance matrix $\mathbf{C}$. Algorithms for (sparse) BSS include online learning algorithm [2], feature extraction method [43], feature sign algorithm [40], and so on.

Model (1.5) with $\mathbf{B} = \mathbf{I}$ also arises in sparse dictionary training [1, 49], where the goal is to build a dictionary $\mathbf{A}$ that sparsely represented the signals in $\mathbf{X}$.

Nonnegative matrix factorization. Nonnegative matrix factorization (NMF) was first proposed by Paatero and his coworkers in the area of environmental science [53]. The later popularity of NMF can be partially attributed to the publication of [38] in Nature. It has been widely applied in data mining such as text mining [55] and image mining [41], dimension reduction and clustering [16,73], hyperspectral endmember extraction, as well as spectral data analysis [54]. A widely used model for (regularized) NMF is

$$\min_{\mathbf{X}\ge 0,\mathbf{Y}\ge 0} \frac{1}{2}\|\mathbf{XY} - \mathbf{M}\|_F^2 + \alpha r_1(\mathbf{X}) + \beta r_2(\mathbf{Y}), \tag{1.7}$$

where $\mathbf{M}$ is the input nonnegative matrix, r_1, r_2 are some regularizers promoting solution structures, and α, β are weight parameters. Two early popular algorithms for NMF are the projected alternating least squares method [53] and multiplicative updating method [39]. Due to the bi-convexity of the objective in (1.7), a series of alternating nonnegative least square (ANLS) methods have been proposed such as [30, 32, 42]; they are BCDs with update (1.3a). Recently, the classic alternating direction method (ADM) has been applied in [78]. We compare the proposed algorithms to them in Sec. 4 below.

Similar models also arise in low-rank matrix recovery, such as the one considered in [58]

$$\min_{\mathbf{X},\mathbf{Y}} \frac{1}{2}\|\mathcal{A}(\mathbf{XY}) - \mathbf{b}\|^2 + \alpha\|\mathbf{X}\|_F^2 + \beta\|\mathbf{Y}\|_F^2, \tag{1.8}$$

where $\mathcal{A}$ is a linear operator. The method of multipliers is employed in [58] to solve (1.8) with no convergence guarantees. Since the objective of (1.8) is coercive and real analytic, our algorithm is guranteed to produce a sequence of points that globally converge to a critical point; see Theorems 2.8 and 2.9.

Nonnegative tensor factorization. Nonnegative tensor factorization (NTF) is a generalization of NMF to multi-dimensional arrays. One commonly used model for NTF is based on CANDECOMP/PARAFAC tensor decomposition [71]

$$\min_{\mathbf{A}_1,\cdots,\mathbf{A}_N\geq 0} \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1\circ\mathbf{A}_2\circ\cdots\circ\mathbf{A}_N\|_F^2 + \sum_{n=1}^{N}\lambda_n r_n(\mathbf{A}_n); \tag{1.9}$$

and another one is based on Tucker decomposition [34]

$$\min_{\boldsymbol{\mathcal{G}},\mathbf{A}_1,\cdots,\mathbf{A}_N\geq 0} \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{G}}\times_1\mathbf{A}_1\times_2\mathbf{A}_2\cdots\times_N\mathbf{A}_N\|_F^2 + \lambda r(\mathbf{G}) + \sum_{n=1}^{N}\lambda_n r_n(\mathbf{A}_n). \tag{1.10}$$

where $\boldsymbol{\mathcal{M}}$ is a given nonnegative tensor, $r, r_1, \cdots, r_N$ are regularizers, $\lambda, \lambda_1, \cdots, \lambda_N$ are weight parameters, and "$\circ$" and "$\times_n$" represent outer product and tensor-matrix multiplication, respectively. (The necessary background of tensor is reviewed in Sec. 3) Most algorithms for solving NMF have been directly extended to NTF. For example, the multiplicative update in [53] is extended to solving (1.9) in [63]. The ANLS methods in [30,32] are extended to solving (1.9) in [31,33]. Algorithms for solving (1.10) also include the column-wise coordinate descent method [44] and the alternating least square method [21]. More about NTF algorithms can be found in [76].

1.3. Organization. The rest of the paper is organized as follows. Sec. 2 studies the convergence of Alg. 1. In Sec. 3, Alg. 1 is applied to both the nonnegative matrix/tensor *factorization* problem and the *completion* problem. The numerical results are presented in Sec. 4. Finally, Sec. 5 concludes the paper.

2. Convergence analysis. In this section, we analyze the convergence of Alg. 1 under the following assumptions.

ASSUMPTION 1. *F is continuous in $dom(F)$ and $\inf_{\mathbf{x}\in dom(F)} F(\mathbf{x}) > -\infty$. Problem (1.1) has a Nash point (see below for definition).*

ASSUMPTION 2. *Each block i is updated by the same scheme among (1.3a)–(1.3c) for all k. Let $\mathcal{I}_1, \mathcal{I}_2$ and $\mathcal{I}_3$ denote the set of blocks updated by (1.3a), (1.3b) and (1.3c), respectively. In addition, there exist constants $0 < \ell_i \leq L_i < \infty, i = 1, \cdots, s$ such that*

1. *for $i \in \mathcal{I}_1$, f_i^k is* strongly convex *with modulus $\ell_i \leq L_i^{k-1} \leq L_i$, namely,*

$$f_i^k(\mathbf{u}) - f_i^k(\mathbf{v}) \geq \langle \nabla f_i^k(\mathbf{v}), \mathbf{u}-\mathbf{v}\rangle + \frac{L_i^{k-1}}{2}\|\mathbf{u}-\mathbf{v}\|^2, \text{ for all } \mathbf{u},\mathbf{v}\in\mathcal{X}_i^k; \tag{2.1}$$

2. *for $i \in \mathcal{I}_2$, parameters L_i^{k-1} obey $\ell_i \leq L_i^{k-1} \leq L_i$;*
3. *for $i \in \mathcal{I}_3$, ∇f_i^k is Lipschitz continuous and parameters L_i^{k-1} obey $\ell_i \leq L_i^{k-1} \leq L_i$ and*

$$f_i^k(\mathbf{x}_i^k) \leq f_i^k(\hat{\mathbf{x}}_i^{k-1}) + \langle \hat{\mathbf{g}}_i^k, \mathbf{x}_i^k - \hat{\mathbf{x}}_i^{k-1}\rangle + \frac{L_i^{k-1}}{2}\|\mathbf{x}_i^k - \hat{\mathbf{x}}_i^{k-1}\|^2. \tag{2.2}$$

REMARK 2.1. *The same notation L_i^{k-1} is used in all three schemes for the simplicity of unified convergence analysis, but we want to emphasize that it has different meanings in the three different schemes. For $i \in \mathcal{I}_1$, L_i^{k-1} is determined by the objective and the current values of all other blocks, while for $i \in \mathcal{I}_2 \cup \mathcal{I}_3$, we have some freedom to choose L_i^{k-1}. For $i \in \mathcal{I}_2$, L_i^{k-1} can be simply fixed to a positive constant or selected by a pre-determined rule to be uniformly lower bounded from* zero *and upper bounded. For $i \in \mathcal{I}_3$, L_i^{k-1} is selected to satisfy (2.2). Taking L_i^{k-1} as the Lipschitz constant of ∇f_i^k can satisfy (2.2). However, we allow smaller L_i^{k-1}, which can speed up the algorithm.*

In addition, we want to emphasize that we make different assumptions on the three different schemes. The use of (1.3a) requires block strong convexity with modulus uniformly away from zero *and upper bounded,*

and the use of (1.3c) *requires block Lipschitz continuous gradient. The use of* (1.3b) *requires neither strong convexity nor Lipschitz continuity. Even the block convexity is unnecessary for* (1.3b)*, and our proof still goes through. Each assumption on the corresponding scheme guarantees sufficient decrease of the objective and makes square summable; see Lemma 2.2, which plays the key role in our convergence analysis.*

For our analysis below, we need the Nash equilibrium condition of (1.1): for $i = 1, \cdots, s$,

$$F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \bar{\mathbf{x}}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) \le F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \mathbf{x}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s), \ \forall \mathbf{x}_i \in \bar{\mathcal{X}}_i, \tag{2.3}$$

or equivalently

$$\langle \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}) + \bar{\mathbf{p}}_i, \mathbf{x}_i - \bar{\mathbf{x}}_i \rangle \ge 0, \ \text{for all } \mathbf{x}_i \in \bar{\mathcal{X}}_i \text{ and for some } \bar{\mathbf{p}}_i \in \partial r_i(\bar{\mathbf{x}}_i), \tag{2.4}$$

where $\bar{\mathcal{X}}_i = \mathcal{X}_i(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s)$ and $\partial r(\mathbf{x}_i)$ is the limiting subdifferential (e.g., see [60]) of r at $\mathbf{x}_i$. We call $\bar{\mathbf{x}}$ a Nash point or block coordinate-wise minimizer. Let $\mathcal{N}$ be the set of all Nash points, which we assume to be nonempty.

REMARK 2.2. *As shown in [4], it holds that*

$$\partial F(\mathbf{x}) = \{\nabla_{\mathbf{x}_1} f(\mathbf{x}) + \partial r_1(\mathbf{x}_1)\} \times \cdots \times \{\nabla_{\mathbf{x}_s} f(\mathbf{x}) + \partial r_s(\mathbf{x}_s)\}.$$

Therefore, if $\mathcal{X} = \mathbb{R}^n$ or $\bar{\mathbf{x}}$ is an interior point of $\mathcal{X}$, (2.4) *reduces to the first-order optimality condition $\mathbf{0} \in \partial F(\bar{\mathbf{x}})$, and $\bar{\mathbf{x}}$ is a* critical point *(or* stationary point*) of* (1.1)*. In general, the condition* (2.4) *is weaker than first-order optimality condition. For problem* (1.1)*, a critical point must be a Nash point, but a Nash point is not necessarily a critical point. An example can be found in Section 4 of [72]. The same example also reveals that, when $\mathcal{X} \neq \mathbb{R}^n$, the iterates given by BCD do not necessarily converge to a stationary point, even if the objective is strongly convex and $\mathcal{X}$ is convex.*

2.1. Preliminary result. The analysis in this subsection follows the following steps. First, we show sufficient descent at each step (inequality (2.8) below), from which we establish the square summable result (Lemma 2.2 below). Next, the square summable result is exploited to show that any limit point is a Nash point in Theorem 2.3 below. Finally, with the additional assumptions of isolated Nash points and bounded $\{\mathbf{x}^k\}$, global convergence is obtained in Corollary 2.4 below. The first step is essential while the last two steps use rather standard arguments. We begin with the following lemma similar to Lemma 2.3 of [8]. Since the proof in [8] does not consider constraints, we include a slightly changed proof for completeness.

LEMMA 2.1. *Let $\xi_1(\mathbf{u})$ and $\xi_2(\mathbf{u})$ be two convex functions defined on the convex set $\mathcal{U}$ and $\xi_1(\mathbf{u})$ be differentiable. Let $\xi(\mathbf{u}) = \xi_1(\mathbf{u}) + \xi_2(\mathbf{u})$ and $\mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}\in\mathcal{U}} \langle \nabla\xi_1(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{L}{2}\|\mathbf{u} - \mathbf{v}\|^2 + \xi_2(\mathbf{u})$. If*

$$\xi_1(\mathbf{u}^*) \le \xi_1(\mathbf{v}) + \langle \nabla\xi_1(\mathbf{v}), \mathbf{u}^* - \mathbf{v}\rangle + \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2, \tag{2.5}$$

then we have

$$\xi(\mathbf{u}) - \xi(\mathbf{u}^*) \ge \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2 + L\langle \mathbf{v} - \mathbf{u}, \mathbf{u}^* - \mathbf{v}\rangle \ \ \textit{for any } \mathbf{u} \in \mathcal{U}. \tag{2.6}$$

Proof. Since $\mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}\in\mathcal{U}} \langle \nabla\xi_1(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{L}{2}\|\mathbf{u} - \mathbf{v}\|^2 + \xi_2(\mathbf{u})$, the first-order optimality condition holds

$$\langle \nabla\xi_1(\mathbf{v}) + L(\mathbf{u}^* - \mathbf{v}) + \mathbf{g}, \mathbf{u} - \mathbf{u}^*\rangle \ge 0, \ \text{for any } \mathbf{u} \in \mathcal{U}, \tag{2.7}$$

for some $\mathbf{g} \in \partial\xi_2(\mathbf{u}^*)$. For any $\mathbf{u} \in \mathcal{U}$, we have

$$\begin{aligned}
\xi(\mathbf{u}) - \xi(\mathbf{u}^*) \geq & \xi(\mathbf{u}) - \left(\xi_1(\mathbf{v}) + \langle \nabla\xi_1(\mathbf{v}), \mathbf{u}^* - \mathbf{v}\rangle + \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2\right) - \xi_2(\mathbf{u}^*) \\
= & \xi_1(\mathbf{u}) - \xi_1(\mathbf{v}) - \langle \nabla\xi_1(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \langle \nabla\xi_1(\mathbf{v}), \mathbf{u} - \mathbf{u}^*\rangle + \xi_2(\mathbf{u}) - \xi_2(\mathbf{u}^*) - \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2 \\
\geq & \xi_2(\mathbf{u}) - \xi_2(\mathbf{u}^*) - \langle \mathbf{g}, \mathbf{u} - \mathbf{u}^*\rangle - L\langle \mathbf{u}^* - \mathbf{v}, \mathbf{u} - \mathbf{u}^*\rangle - \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2 \\
\geq & -L\langle \mathbf{u}^* - \mathbf{v}, \mathbf{u} - \mathbf{u}^*\rangle - \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2 \\
= & \frac{L}{2}\|\mathbf{u}^* - \mathbf{v}\|^2 + L\langle \mathbf{v} - \mathbf{u}, \mathbf{u}^* - \mathbf{v}\rangle,
\end{aligned}$$

where the first inequality uses (2.5), the second inequality is obtained from the convexity of ξ_1 and (2.7), and the last inequality uses the convexity of ξ_2 and the fact $\mathbf{g} \in \partial\xi_2(\mathbf{u}^*)$. This completes the proof. □

Based on this lemma, we can show our key lemma below.

LEMMA 2.2 (Square summable $\|\mathbf{x}^k - \mathbf{x}^{k+1}\|$). *Under Assumptions 1 and 2, let $\{\mathbf{x}^k\}$ be the sequence generated by Alg. 1 with $0 \leq \omega_i^{k-1} \leq \delta_\omega\sqrt{\frac{L_i^{k-2}}{L_i^{k-1}}}$ for $\delta_\omega < 1$ uniformly over all $i \in \mathcal{I}_3$ and k. Then $\sum_{k=0}^{\infty}\|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 < \infty$.*

Proof. For $i \in \mathcal{I}_3$, we have the inequality (2.2) and use Lemma 2.1 by letting $F_i^k \triangleq f_i^k + r_i$ and taking $\xi_1 = f_i^k, \xi_2 = r_i, \mathbf{v} = \hat{\mathbf{x}}_i^{k-1}$ and $\mathbf{u} = \mathbf{x}_i^{k-1}$ in (2.6) to have

$$\begin{aligned}
F_i^k(\mathbf{x}_i^{k-1}) - F_i^k(\mathbf{x}_i^k) \geq & \frac{L_i^{k-1}}{2}\|\hat{\mathbf{x}}_i^{k-1} - \mathbf{x}_i^k\|^2 + L_i^{k-1}\langle \hat{\mathbf{x}}_i^{k-1} - \mathbf{x}_i^{k-1}, \mathbf{x}_i^k - \hat{\mathbf{x}}_i^{k-1}\rangle \\
= & \frac{L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2 - \frac{L_i^{k-1}}{2}(\omega_i^{k-1})^2\|\mathbf{x}_i^{k-2} - \mathbf{x}_i^{k-1}\|^2 \\
\geq & \frac{L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2 - \frac{L_i^{k-2}}{2}\delta_\omega^2\|\mathbf{x}_i^{k-2} - \mathbf{x}_i^{k-1}\|^2.
\end{aligned} \tag{2.8}$$

For $i \in \mathcal{I}_1 \cup \mathcal{I}_2$ we have $F_i^k(\mathbf{x}_i^{k-1}) - F_i^k(\mathbf{x}_i^k) \geq \frac{L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2$, and thus the inequality (2.8) still holds (regard $\omega_i^k \equiv 0$ for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$). Therefore,

$$\begin{aligned}
F(\mathbf{x}^{k-1}) - F(\mathbf{x}^k) = & \ \sum_{i=1}^s \left(F_i^k(\mathbf{x}_i^{k-1}) - F_i^k(\mathbf{x}_i^k)\right) \\
\geq & \ \sum_{i=1}^s \left(\tfrac{L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2 - \tfrac{L_i^{k-2}\delta_\omega^2}{2}\|\mathbf{x}_i^{k-2} - \mathbf{x}_i^{k-1}\|^2\right).
\end{aligned}$$

Summing the above inequality over k from 1 to K, we have

$$\begin{aligned}
F(\mathbf{x}^0) - F(\mathbf{x}^K) \geq & \sum_{k=1}^K\sum_{i=1}^s \left(\frac{L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2 - \frac{L_i^{k-2}}{2}\delta_\omega^2\|\mathbf{x}_i^{k-2} - \mathbf{x}_i^{k-1}\|^2\right) \\
\geq & \sum_{k=1}^K\sum_{i=1}^s \frac{(1-\delta_\omega^2)L_i^{k-1}}{2}\|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2 \geq \sum_{k=1}^K \frac{(1-\delta_\omega^2)\ell_i}{2}\|\mathbf{x}^{k-1} - \mathbf{x}^k\|^2.
\end{aligned}$$

Since F is lower bounded, taking $K \to \infty$ completes the proof. □

Now, we can establish the following preliminary convergence result.

THEOREM 2.3 (Limit point is Nash point). *Define the difference measure of two sets $\mathcal{X}, \mathcal{Y}$ by*

$$\mathrm{diff}(\mathcal{X}, \mathcal{Y}) = \max\left(\sup_{\mathbf{x}\in\mathcal{X}}\inf_{\mathbf{y}\in\mathcal{Y}}\|\mathbf{x} - \mathbf{y}\|, \ \sup_{\mathbf{y}\in\mathcal{Y}}\inf_{\mathbf{x}\in\mathcal{X}}\|\mathbf{x} - \mathbf{y}\|\right).$$

Assume the set map $\mathcal{X}_i(\cdot)$ defined in (1.2) continuously changes, namely, $\mathbf{x}^{k'}, \mathbf{x} \in \mathcal{X}$ and $\mathbf{x}^{k'} \to \mathbf{x}$ imply

$$\lim_{k\to\infty}\mathrm{diff}\left(\mathcal{X}_i(\mathbf{x}_1^{k'}, \ldots, \mathbf{x}_{i-1}^{k'}, \mathbf{x}_{i+1}^{k'}, \ldots, \mathbf{x}_s^{k'}), \mathcal{X}_i(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_s)\right) = 0, \ \forall i.$$

Then if the assumptions of Lemma 2.2 hold, any limit point of $\{\mathbf{x}^k\}$ is a Nash point, namely, satisfying the Nash equilibrium condition (2.4).

Proof. Let $\bar{\mathbf{x}}$ be a limit point of $\{\mathbf{x}^k\}$ and $\{\mathbf{x}^{k_j}\}$ be the subsequence converging to $\bar{\mathbf{x}}$. The closedness of $\mathcal{X}$ implies $\bar{\mathbf{x}} \in \mathcal{X}$. Since $\{L_i^k\}$ is bounded, passing another sequence if necessary, we have $L_i^{k_j} \to \bar{L}_i$ for $i = 1, \cdots, s$ as $j \to \infty$. Lemma 2.2 implies that $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \to 0$, so $\{\mathbf{x}^{k_j+1}\}$ also converges to $\bar{\mathbf{x}}$.

For $i \in \mathcal{I}_1$, we have

$$F_i^{k_j+1}(\mathbf{x}_i^{k_j+1}) \leq F_i^{k_j+1}(\mathbf{x}_i), \ \forall \mathbf{x}_i \in \mathcal{X}_i^{k_j+1}. \tag{2.9}$$

Letting $j \to \infty$, we can show (2.3) by the continuity of F and the set map $\mathcal{X}_i(\cdot)$ by the following arguments. For any block $i_0 \in \mathcal{I}_1$ and any $\mathbf{y}_{i_0} \in \bar{\mathcal{X}}_{i_0}$, since $\lim_j \operatorname{diff}\left(\mathcal{X}_{i_0}^{k_j+1}, \bar{\mathcal{X}}_{i_0}\right) = 0$, there is $\mathbf{y}_{i_0}^j \in \mathcal{X}_{i_0}^{k_j+1}$ such that $\lim_j \mathbf{y}_{i_0}^j = \mathbf{y}_{i_0}$. From the continuity of F, we have

$$\lim_{j\to\infty} F(\mathbf{x}_1^{k_j+1}, \ldots, \mathbf{x}_{i_0-1}^{k_j+1}, \mathbf{y}_{i_0}^j, \mathbf{x}_{i_0+1}^{k_j}, \ldots, \mathbf{x}_s^{k_j}) = F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i_0-1}, \mathbf{y}_{i_0}, \bar{\mathbf{x}}_{i_0+1}, \cdots, \bar{\mathbf{x}}_s). \tag{2.10}$$

Note that (2.9) implies

$$F(\mathbf{x}_1^{k_j+1}, \ldots, \mathbf{x}_{i_0-1}^{k_j+1}, \mathbf{x}_{i_0}^{k_j+1}, \mathbf{x}_{i_0+1}^{k_j}, \ldots, \mathbf{x}_s^{k_j}) \leq F(\mathbf{x}_1^{k_j+1}, \ldots, \mathbf{x}_{i_0-1}^{k_j+1}, \mathbf{y}_{i_0}^j, \mathbf{x}_{i_0+1}^{k_j}, \ldots, \mathbf{x}_s^{k_j}).$$

Letting $j \to \infty$ and using (2.10), we get

$$F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i_0-1}, \bar{\mathbf{x}}_{i_0}, \bar{\mathbf{x}}_{i_0+1}, \cdots, \bar{\mathbf{x}}_s) \leq F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i_0-1}, \mathbf{y}_{i_0}, \bar{\mathbf{x}}_{i_0+1}, \cdots, \bar{\mathbf{x}}_s).$$

Hence, (2.3) holds. Similarly, for $i \in \mathcal{I}_2$, we have

$$F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \bar{\mathbf{x}}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) \leq F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \mathbf{x}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) + \frac{\bar{L}_i}{2}\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2, \forall \mathbf{x}_i \in \bar{\mathcal{X}}_i,$$

namely,

$$\bar{\mathbf{x}}_i = \operatorname*{argmin}_{\mathbf{x}_i \in \bar{\mathcal{X}}_i} F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \mathbf{x}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) + \frac{\bar{L}_i}{2}\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2. \tag{2.11}$$

Thus, $\bar{\mathbf{x}}_i$ satisfies the first-order optimality condition of (2.11), which is precisely (2.4). For $i \in \mathcal{I}_3$, we have

$$\mathbf{x}_i^{k_j+1} = \operatorname*{argmin}_{\mathbf{x}_i \in \mathcal{X}_i^{k_j+1}} \langle \nabla f_i^{k_j+1}(\hat{\mathbf{x}}_i^{k_j}), \mathbf{x}_i - \hat{\mathbf{x}}_i^{k_j} \rangle + \frac{L_i^{k_j}}{2}\|\mathbf{x}_i - \hat{\mathbf{x}}_i^{k_j}\|^2 + r_i(\mathbf{x}_i).$$

The convex proximal minimization is continuous in the sense that the output $\mathbf{x}_i^{k_j+1}$ depends continuously on the input $\hat{\mathbf{x}}_i^{k_j}$ [59]. Letting $j \to \infty$, from $\mathbf{x}_i^{k_j+1} \to \bar{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_i^{k_j} \to \bar{\mathbf{x}}_i$, we get

$$\bar{\mathbf{x}}_i = \operatorname*{argmin}_{\mathbf{x}_i \in \bar{\mathcal{X}}_i} \langle \nabla_{\mathbf{x}_i} f(\bar{\mathbf{x}}), \mathbf{x}_i - \bar{\mathbf{x}}_i \rangle + \frac{\bar{L}_i}{2}\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2 + r_i(\mathbf{x}_i). \tag{2.12}$$

Hence, $\bar{\mathbf{x}}_i$ satisfies the first-order optimality condition of (2.12), which is precisely (2.4). This completes the proof. ∎

REMARK 2.3. *If $\mathcal{X}$ is convex, then the set map $\mathcal{X}_i(\cdot)$ is continuous; see Theorem 4.32 in [60]. A special case is $\mathcal{X} = \mathbb{R}^n$, namely, there is no joint constraints.*

COROLLARY 2.4 (Global convergence given isolated Nash points). *Under the assumptions of Theorem 2.3, if $\{\mathbf{x}^k\}$ is bounded, we have $\operatorname{dist}(\mathbf{x}^k, \mathcal{N}) \to 0$. If further $\mathcal{N}$ contains uniformly isolated points, namely, there is $\eta > 0$ such that $\|\mathbf{x} - \mathbf{y}\| \geq \eta$ for any distinct points $\mathbf{x}, \mathbf{y} \in \mathcal{N}$, then $\{\mathbf{x}^k\}$ converges to a point in $\mathcal{N}$.*

Proof. Suppose $\mathrm{dist}(\mathbf{x}^k,\mathcal{N})$ does not converge to 0. Then there exists $\varepsilon > 0$ and a subsequence $\{\mathbf{x}^{k_j}\}$ such that $\mathrm{dist}(\mathbf{x}^{k_j},\mathcal{N}) \geq \varepsilon$ for all j. However, the boundedness of $\{\mathbf{x}^{k_j}\}$ implies that it must have a limit point $\bar{\mathbf{x}} \in \mathcal{N}$ according to Theorem 2.3, which is a contradiction.

From $\mathrm{dist}(\mathbf{x}^k,\mathcal{N}) \to 0$, it follows that there is an integer $K_1 > 0$ such that $\mathbf{x}^k \in \cup_{\mathbf{y}\in\mathcal{N}} B(\mathbf{y}, \frac{\eta}{3})$ for all $k \geq K_1$, where $B(\mathbf{y}, \frac{\eta}{3}) \triangleq \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x}-\mathbf{y}\| < \frac{\eta}{3}\}$. In addition, Lemma 2.2 implies that there exists another integer $K_2 > 0$ such that $\|\mathbf{x}^k - \mathbf{x}^{k+1}\| < \frac{\eta}{3}$ for all $k \geq K_2$. Take $K = \max(K_1, K_2)$ and assume $\mathbf{x}^K \in B(\bar{\mathbf{x}}, \frac{\eta}{3})$ for some $\bar{\mathbf{x}} \in \mathcal{N}$. We claim that for any $\mathbf{y} \in \mathcal{N}$ and $\mathbf{y} \neq \bar{\mathbf{x}}$, $\|\mathbf{x}^k - \mathbf{y}\| > \frac{\eta}{3}$ holds for all $k \geq K$. This claim can be shown by induction on $k \geq K$. If some $\mathbf{x}^k \in B(\bar{\mathbf{x}}, \frac{\eta}{3})$, then $\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}\| \leq \|\mathbf{x}^{k+1} - \mathbf{x}^k\| + \|\mathbf{x}^k - \bar{\mathbf{x}}\| < \frac{2\eta}{3}$, and

$$\|\mathbf{x}^{k+1} - \mathbf{y}\| \geq \|\bar{\mathbf{x}} - \mathbf{y}\| - \|\mathbf{x}^{k+1} - \bar{\mathbf{x}}\| > \frac{\eta}{3}, \text{ for any } \bar{\mathbf{x}} \neq \mathbf{y} \in \mathcal{N}.$$

Therefore, $\mathbf{x}^k \in B(\bar{\mathbf{x}}, \frac{\eta}{3})$ for all $k \geq K$ since $\mathbf{x}^k \in \cup_{\mathbf{y}\in\mathcal{N}} B(\mathbf{y}, \frac{\eta}{3})$, and thus $\{\mathbf{x}^k\}$ has the unique limit point $\bar{\mathbf{x}}$, which means $\mathbf{x}^k \to \bar{\mathbf{x}}$. □

REMARK 2.4. *The boundedness of $\{\mathbf{x}^k\}$ is guaranteed if the level set $\{\mathbf{x} \in \mathcal{X} : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ is bounded. However, the isolation assumption does not hold, or holds but is difficult to verify, for many functions. This motivates another approach below for global convergence.*

2.2. Kurdyka-Łojasiewicz inequality. Before proceeding with our analysis, let us briefly review the Kurdyka-Łojasiewicz inequality, which is central to the global convergence analysis in the next subsection.

DEFINITION 2.5. *A function $\psi(\mathbf{x})$ satisfies the Kurdyka-Łojasiewicz (KL) property at point $\bar{\mathbf{x}} \in \mathrm{dom}(\partial\psi)$ if there exists $\theta \in [0, 1)$ such that*

$$\frac{|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|^\theta}{\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x}))} \tag{2.13}$$

is bounded around $\bar{\mathbf{x}}$ under the notational conventions: $0^0 = 1, \infty/\infty = 0/0 = 0$. In other words, in a certain neighborhood $\mathcal{U}$ of $\bar{\mathbf{x}}$, there exists $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in [0, 1)$ such that the KL inequality holds

$$\phi'(|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|)\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x})) \geq 1, \text{ for any } \mathbf{x} \in \mathcal{U} \cap \mathrm{dom}(\partial\psi) \text{ and } \psi(\mathbf{x}) \neq \psi(\bar{\mathbf{x}}), \tag{2.14}$$

where $\mathrm{dom}(\partial\psi) \triangleq \{\mathbf{x} : \partial\psi(\mathbf{x}) \neq \emptyset\}$ *and* $\mathrm{dist}(\mathbf{0}, \partial\psi(\mathbf{x})) \triangleq \min\{\|\mathbf{y}\| : \mathbf{y} \in \partial\psi(\mathbf{x})\}$.

This property was introduced by Łojasiewicz [46] on real analytic functions, for which the term with $\theta \in [\frac{1}{2}, 1)$ in (2.13) is bounded around any critical point $\bar{\mathbf{x}}$. Kurdyka extended this property to functions on the o-minimal structure in [36]. Recently, the KL inequality was extended to nonsmooth sub-analytic functions [13].

The KL inequality (2.14) is usually weaker than the condition of isolated Nash points used in Corollary 2.4. In (2.14), we require $\psi(\mathbf{x}) \neq \psi(\bar{\mathbf{x}})$, so a point obeying the KL inequality need not to be an isolated Nash point. The function $\psi(x, y) = (xy - 1)^2$ is an example, where $(x, y) = (1, 1)$ is a minimizer meeting the KL inequality but not isolated. Since it is not trivial to check the conditions in the definition, we give some examples below that satisfy the KL inequality.

Real analytic functions. A smooth function $\varphi(t)$ on $\mathbb{R}$ is analytic if $\left(\frac{\varphi^{(k)}(t)}{k!}\right)^{\frac{1}{k}}$ is bounded for all k and on any compact set $\mathcal{D} \subset \mathbb{R}$. One can verify whether a real function $\psi(\mathbf{x})$ on $\mathbb{R}^n$ is analytic by checking the analyticity of $\varphi(t) \triangleq \psi(\mathbf{x} + t\mathbf{y})$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. For example, any polynomial function is real analytic such as $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ and the first terms in the objectives of (1.9) and (1.10). In addition, it is not difficult to verify that the non-convex function $L_q(\mathbf{x}, \varepsilon, \lambda) = \sum_{i=1}^n (x_i^2 + \varepsilon^2)^{q/2} + \frac{1}{2\lambda}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ with $0 < q < 1$ considered in [37] for sparse vector recovery is a real analytic function (the first term is the ε-smoothed ℓ_q semi-norm).

The logistic loss function $\psi(t) = \log(1 + e^{-t})$ is also analytic. Therefore, all the above functions satisfy the KL inequality with $\theta \in [\frac{1}{2}, 1)$ in (2.13).

Locally strongly convex functions. A function $\psi(\mathbf{x})$ is strongly convex in a neighborhood $\mathcal{D}$ with constant μ, if

$$\psi(\mathbf{y}) \geq \psi(\mathbf{x}) + \langle \gamma(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \text{ for all } \gamma(\mathbf{x}) \in \partial\psi(\mathbf{x}) \text{ and for any } \mathbf{x}, \mathbf{y} \in \mathcal{D}.$$

According to the definition and using the Cauchy-Schwarz inequality, we have

$$\psi(\mathbf{y}) - \psi(\mathbf{x}) \geq \langle \gamma(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2 \geq -\frac{1}{\mu}\|\gamma(\mathbf{x})\|^2, \text{ for all } \gamma(\mathbf{x}) \in \partial\psi(\mathbf{x}).$$

Hence, $\mu(\psi(\mathbf{x}) - \psi(\mathbf{y})) \leq \big(\text{dist}(\mathbf{0}, \partial\psi(\mathbf{x}))\big)^2$, and ψ satisfies the KL inequality (2.14) at any point $\mathbf{y} \in \mathcal{D}$ with $\phi(s) = \frac{2}{\mu}\sqrt{s}$ and $\mathcal{U} = \mathcal{D} \cap \{\mathbf{x} : \psi(\mathbf{x}) \geq \psi(\mathbf{y})\}$. For example, the logistic loss function $\psi(t) = \log(1 + e^{-t})$ is strongly convex in any bounded set $\mathcal{D}$.

Semi-algebraic functions. A set $\mathcal{D} \subset \mathbb{R}^n$ is called *semi-algebraic* [11] if it can be represented as

$$\mathcal{D} = \bigcup_{i=1}^{s} \bigcap_{j=1}^{t} \{\mathbf{x} \in \mathbb{R}^n : p_{ij}(\mathbf{x}) = 0, q_{ij}(\mathbf{x}) > 0\},$$

where p_{ij}, q_{ij} are real polynomial functions for $1 \leq i \leq s$, $1 \leq j \leq t$. A function ψ is called *semi-algebraic* if its graph $\text{Gr}(\psi) \triangleq \{(\mathbf{x}, \psi(\mathbf{x})) : \mathbf{x} \in \text{dom}(\psi)\}$ is a *semi-algebraic* set.

Semi-algebraic functions are sub-analytic, so they satisfy the KL inequality according to [13, 14]. We list some known elementary properties of semi-algebraic sets and functions below as they help identify semi-algebraic functions.

1. If a set $\mathcal{D}$ is semi-algebraic, so is its closure $\text{cl}(\mathcal{D})$.
2. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are both semi-algebraic, so are $\mathcal{D}_1 \cup \mathcal{D}_2$, $\mathcal{D}_1 \cap \mathcal{D}_2$ and $\mathbb{R}^n \backslash \mathcal{D}_1$.
3. Indicator functions of semi-algebraic sets are semi-algebraic.
4. Finite sums and products of semi-algebraic functions are semi-algebraic.
5. The composition of semi-algebraic functions is semi-algebraic.

From items 1 and 2, any polyhedral set is semi-algebraic such as the nonnegative orthant $\mathbb{R}^n_+ = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0, \forall i\}$. Hence, the indicator function $\delta_{\mathbb{R}^n_+}$ is a semi-algebraic function. The absolute value function $\varphi(t) = |t|$ is also semi-algebraic since its graph is $\text{cl}(\mathcal{D})$, where

$$\mathcal{D} = \{(t, s) : t + s = 0, -t > 0\} \cup \{(t, s) : t - s = 0, t > 0\}.$$

Hence, the ℓ_1-norm $\|\mathbf{x}\|_1$ is semi-algebraic since it is the finite sum of absolute functions. In addition, the sup-norm $\|\mathbf{x}\|_\infty$ is semi-algebraic, which can be shown by observing

$$\text{Graph}(\|\mathbf{x}\|_\infty) = \{(\mathbf{x}, t) : t = \max_j |x_j|\} = \bigcup_i \{(\mathbf{x}, t) : |x_i| = t, |x_j| \leq t, \forall j \neq i\}.$$

Further, the Euclidean norm $\|\mathbf{x}\|$ is shown to be semi-algebraic in [11]. According to item 5, $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$, $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty$ and $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ are all semi-algebraic functions.

Sum of real analytic and semi-algebraic functions. Both real analytic and semi-algebraic functions are sub-analytic. According to [11], if ψ_1 and ψ_2 are both sub-analytic and ψ_1 maps bounded sets to bounded sets, then $\psi_1 + \psi_2$ is also sub-analytic. Since real analytic functions map bounded set to bounded set, the sum of a real analytic function and a semi-algebraic function is sub-analytic, so the sum satisfies the KL property. For example, the sparse logistic regression function

$$\psi(\mathbf{x}, b) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-c_i(\mathbf{a}_i^\top \mathbf{x} + b)\right)\right) + \lambda\|\mathbf{x}\|_1$$

is sub-analytic and satisfies the KL inequality.

2.3. Global convergence and rate. If $\{\mathbf{x}^k\}$ is bounded, then Theorem 2.3 guarantees that there exists one subsequence converging to a Nash point of (1.1). In this subsection, we assume $\mathcal{X} = \mathbb{R}^n$ and strengthen this result for problems with F obeying the KL inequality. Our analysis here was motivated by [4], which applies the inequality to establish the global convergence of the alternating proximal point method — the special case of BCD with two blocks and using update (1.3b).

We make the following modification to Alg. 1.

(**M1**). Whenever $F(\mathbf{x}^k) \geq F(\mathbf{x}^{k-1})$, we re-do the kth iteration with $\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^{k-1}$ (i.e., no extrapolation) for all $i \in \mathcal{I}_3$.

REMARK 2.5. *From the proof of Lemma 2.2, we can see that this modification makes $F(\mathbf{x}^k)$ strictly less than $F(\mathbf{x}^{k-1})$ as long as $\mathbf{x}^k \neq \mathbf{x}^{k-1}$. To show this, observe the proof of Lemma 2.2 implies that updates (1.3a) and (1.3b) both make the objective decrease at least $\frac{L_i^{k-1}}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2$ and update (1.3c) also makes the objective decrease at least $\frac{L_i^{k-1}}{2}\|\mathbf{x}_i^k - \mathbf{x}_i^{k-1}\|^2$ when $\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^{k-1}$, namely, the modification step (**M1**) occurs. Hence, if $\mathbf{x}^{k_0} = \mathbf{x}^{k_0-1}$ for some k_0, then $F(\mathbf{x}^k) = F(\mathbf{x}^{k_0})$ and $\mathbf{x}^k = \mathbf{x}^{k_0}$ for all $k \geq k_0$.*

In the sequel, we use the notion $F_k = F(\mathbf{x}^k)$ and $\bar{F} = F(\bar{\mathbf{x}})$. First, we establish the following pre-convergence result, the proof of which is given in the Appendix.

LEMMA 2.6. *Under Assumptions 1 and 2, let $\{\mathbf{x}^k\}$ be the sequence of Alg. 1 with (**M1**) and its parameters satisfying: $L_i^k \geq \ell^{k-1} \triangleq \min_{i\in\mathcal{I}_3} L_i^{k-1}$ and $\omega_i^k \leq \delta_\omega\sqrt{\frac{\ell^{k-1}}{L_i^k}}$, $\delta_\omega < 1$, for all $i \in \mathcal{I}_3$ and k. Assume*

1. *∇f is Lipschitz continuous on any bounded set;*
2. *F satisfies the KL inequality (2.14) at $\bar{\mathbf{x}}$;*
3. *$\mathbf{x}_0$ is* sufficiently close *to $\bar{\mathbf{x}}$, and $F_k > \bar{F}$ for $k \geq 0$.*

Then there is some $\mathcal{B} \subset \mathcal{U} \cap \mathrm{dom}(\partial\psi)$ with $\psi = F$ in (2.14) such that $\{\mathbf{x}^k\} \subset \mathcal{B}$ and $\mathbf{x}^k$ converges to a point in $\mathcal{B}$.

REMARK 2.6. *In the lemma, the required closeness of $\mathbf{x}^0$ to $\bar{\mathbf{x}}$ depends on $\mathcal{U}, \phi$ and $\psi = F$ in (2.14) (see the inequality in (A.1)). The extrapolation weight ω_i^k must be smaller than it is in Lemma 2.2 in order to guarantee sufficient decrease at each iteration.*

The following corollary is a straightforward application of Lemma 2.6.

COROLLARY 2.7. *Under the assumptions of Lemma 2.6, $\{\mathbf{x}^k\}$ converges to a global minimizer of (1.1) if the initial point $\mathbf{x}^0$ is sufficiently close to any global minimizer $\bar{\mathbf{x}}$.*

Proof. Suppose $F(\mathbf{x}^{k_0}) = F(\bar{\mathbf{x}})$ at some k_0. Then $\mathbf{x}^k = \mathbf{x}^{k_0}$, for all $k \geq k_0$, according to Remark 2.5. Now consider $F(\mathbf{x}^k) > F(\bar{\mathbf{x}})$ for all $k \geq 0$, and thus Lemma 2.6 implies that $\mathbf{x}^k$ converges to some critical point $\mathbf{x}^*$ if $\mathbf{x}^0$ is sufficiently close to $\bar{\mathbf{x}}$, where $\mathbf{x}^0, \mathbf{x}^*, \bar{\mathbf{x}} \in \mathcal{B}$. If $F(\mathbf{x}^*) > F(\bar{\mathbf{x}})$, then the KL inequality (2.14) indicates $\phi'\left(F(\mathbf{x}^*) - F(\bar{\mathbf{x}})\right)\mathrm{dist}\left(\mathbf{0}, \partial F(\mathbf{x}^*)\right) \geq 1$, which is impossible since $\mathbf{0} \in \partial F(\mathbf{x}^*)$. ∎

Next, we give the convergence result of Alg. 1.

THEOREM 2.8 (Global convergence). *Under the assumptions of Lemma 2.6 and that $\{\mathbf{x}^k\}$ has a finite limit point $\bar{\mathbf{x}}$ where F satisfies the KL inequality (2.14), the sequence $\{\mathbf{x}^k\}$ converges to $\bar{\mathbf{x}}$, which is a critical point of (1.1).*

Proof. Note that $F(\mathbf{x}^k)$ is monotonically nonincreasing and converges to $F(\bar{\mathbf{x}})$. If $F(\mathbf{x}^{k_0}) = F(\bar{\mathbf{x}})$ at some k_0, then $\mathbf{x}^k = \mathbf{x}^{k_0} = \bar{\mathbf{x}}$ for all $k \geq k_0$ accroding to Remark 2.5. It remains to consider $F(\mathbf{x}^k) > F(\bar{\mathbf{x}})$ for all $k \geq 0$. Since $\bar{\mathbf{x}}$ is a limit point and $F(\mathbf{x}^k) \to F(\bar{\mathbf{x}})$, there must exist an integer k_0 such that $\mathbf{x}^{k_0}$ is sufficiently close to $\bar{\mathbf{x}}$ as required in Lemma 2.6 (see the inequality in (A.1)). Hence, the entire sequence $\{\mathbf{x}^k\}$ converges according to Lemma 2.6. Since $\bar{\mathbf{x}}$ is a limit point of $\{\mathbf{x}^k\}$, we have $\mathbf{x}^k \to \bar{\mathbf{x}}$. ∎

We can also estimate the rate of convergence, and the proof is given in the Appendix.

THEOREM 2.9 (Convergence rate). *Assume the assumptions of Lemma 2.6, and suppose that* $\mathbf{x}^k$ *converges to a critical point* $\bar{\mathbf{x}}$*, at which* F *satisfies the KL inequality with* $\phi(s) = cs^{1-\theta}$ *for* $c > 0$ *and* $\theta \in [0, 1)$*. We have:*

1. *If* $\theta = 0$, $\mathbf{x}^k$ *converges to* $\bar{\mathbf{x}}$ *in finite iterations;*
2. *If* $\theta \in (0, \frac{1}{2}]$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \leq C\tau^k,\ \forall k \geq k_0$*, for certain* $k_0 > 0,\ C > 0,\ \tau \in [0, 1)$*;*
3. *If* $\theta \in (\frac{1}{2}, 1)$, $\|\mathbf{x}^k - \bar{\mathbf{x}}\| \leq Ck^{-(1-\theta)/(2\theta-1)},\ \forall k \geq k_0$*, for certain* $k_0 > 0,\ C > 0$*.*

3. Factorization and completion of nonnegative matrices and tensors. In this section, we apply Alg. 1 with modification (**M1**) to the factorization and the completion of nonnegative matrices and tensors. Since a matrix is a two-way tensor, we present the algorithm for tensors. We first overview tensor and its two popular factorizations.

3.1. Overview of tensor. A *tensor* is a multi-dimensional array. For example, a *vector* is a first-order *tensor*, and a *matrix* is a second-order *tensor*. The *order* of a tensor is the number of dimensions, also called *way* or *mode*. For an N-way *tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we let its $(i_1, i_2, \cdots, i_N)$th element be denoted by $x_{i_1 i_2 \cdots i_N}$. Below we list some concepts related to tensor. For more details about tensor, the reader is referred to the review paper [35].

1. **fiber:** a *fiber* of a tensor $\boldsymbol{\mathcal{X}}$ is a vector obtained by fixing all indices of $\boldsymbol{\mathcal{X}}$ except one. For example, a row of a matrix is a mode-2 fiber (the 1st index is fixed), and a column is a mode-1 fiber (the 2nd index is fixed). We use $\mathbf{x}_{i_1 \cdots i_{n-1} : i_{n+1} \cdots i_N}$ to denote a mode-n fiber of an Nth-order tensor $\boldsymbol{\mathcal{X}}$.
2. **slice:** a *slice* of a tensor $\boldsymbol{\mathcal{X}}$ is a matrix obtained by fixing all indices of $\boldsymbol{\mathcal{X}}$ except two. Take a third-order tensor $\boldsymbol{\mathcal{X}}$ for example. $\mathbf{X}_{i::}, \mathbf{X}_{:j:}$, and $\mathbf{X}_{::k}$ denote horizontal, lateral, and frontal slices of $\boldsymbol{\mathcal{X}}$, respectively.
3. **matricization:** the mode-n *matricization* of a tensor $\boldsymbol{\mathcal{X}}$ is a matrix whose columns are the mode-n fibers of $\boldsymbol{\mathcal{X}}$ in the lexicographical order. We let $\mathbf{X}_{(n)}$ denote the mode-n matricization of $\boldsymbol{\mathcal{X}}$.
4. **tensor-matrix product:** the mode-n product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is a tensor of size $I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ defined as

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{A})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} a_{j i_n}. \tag{3.1}$$

In addition, we briefly review the matrix Kronecker, Khatri-Rao and Hadamard products below, which we use to derive tensor-related computations.

The *Kronecker product* of matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ is an $mp \times nq$ matrix defined by $\mathbf{A} \otimes \mathbf{B} = [a_{ij}\mathbf{B}]_{mp \times nq}$. The *Khatri-Rao product* of matrices $\mathbf{A} \in \mathbb{R}^{m \times q}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ is an $mp \times q$ matrix: $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1,\ \mathbf{a}_2 \otimes \mathbf{b}_2,\ \cdots,\ \mathbf{a}_q \otimes \mathbf{b}_q]$, where $\mathbf{a}_i, \mathbf{b}_i$ are the ith columns of $\mathbf{A}$ and $\mathbf{B}$, respectively. The *Hadamard product* of matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ is the componentwise product defined by $\mathbf{A} * \mathbf{B} = [a_{ij} b_{ij}]_{m \times n}$.

Two important tensor decompositions are the CANDECOMP/PARAFAC (CP) [29] and Tucker [68] decompositions. The former one decomposes a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ in the form of $\boldsymbol{\mathcal{X}} = \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N$, where $\mathbf{A}_n \in \mathbb{R}^{I_n \times r}, n = 1, \cdots, N$ are factor matrices, r is the tensor rank of $\boldsymbol{\mathcal{X}}$, and the outer product "$\circ$" is defined as

$$x_{i_1 i_2 \cdots i_N} = \sum_{j=1}^{r} a^{(1)}_{i_1 j} a^{(2)}_{i_2 j} \cdots a^{(N)}_{i_N j},\ \text{for } i_n \in [I_n], n = 1, \cdots, N,$$

where $a^{(n)}_{ij}$ is the (i, j)th element of $\mathbf{A}_n$ and $[I] \triangleq \{1, 2, \cdots, I\}$. The latter Tucker decomposition decomposes a tensor $\boldsymbol{\mathcal{X}}$ in the form of $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_N \mathbf{A}_N$, where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is called the core tensor and $\mathbf{A}_n \in \mathbb{R}^{I_n \times J_n}, n = 1, \cdots, N$ are factor matrices.

3.2. An algorithm for nonnegative tensor factorization. One can obtain a nonnegative CP decomposition of a nonnegative tensor $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ by solving

$$\min \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2, \text{ subject to } \mathbf{A}_n \in \mathbb{R}_+^{I_n \times r}, n = 1, \cdots, N \tag{3.2}$$

where r is a specified order and the Frobenius norm of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined as $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{i_1, i_2, \cdots, i_N} x_{i_1 i_2 \cdots i_N}^2}$. Similar models based on the CP decomposition can be found in [19,31,33]. One can obtain a nonnegative Tucker decomposition of $\boldsymbol{\mathcal{M}}$ by solving

$$\min \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_N \mathbf{A}_N\|_F^2, \text{ subject to } \boldsymbol{\mathcal{G}} \in \mathbb{R}_+^{J_1 \times \cdots \times J_N}, \mathbf{A}_n \in \mathbb{R}_+^{I_n \times J_n}, \forall n, \tag{3.3}$$

as in [34, 44, 51]. Usually, it is computationally expensive to update $\boldsymbol{\mathcal{G}}$. Since applying Alg. 1 to problem (3.3) involves lots of computing details, we focus on applying Alg. 1 with update (1.3c) to problem (3.2).

Let $\mathbf{A} = (\mathbf{A}_1, \cdots, \mathbf{A}_N)$ and

$$F(\mathbf{A}) = F(\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_N) = \frac{1}{2}\|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N\|_F^2$$

be the objective of (3.2). Consider updating $\mathbf{A}_n$ at iteration k. Using the fact that if $\boldsymbol{\mathcal{X}} = \mathbf{A}_1 \circ \mathbf{A}_2 \circ \cdots \circ \mathbf{A}_N$, then $\mathbf{X}_{(n)} = \mathbf{A}_n \left(\mathbf{A}_N \odot \cdots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \cdots \mathbf{A}_1\right)^\top$, we have

$$F(\mathbf{A}) = \frac{1}{2}\left\|\mathbf{M}_{(n)} - \mathbf{A}_n \left(\mathbf{A}_N \odot \cdots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \cdots \mathbf{A}_1\right)^\top\right\|_F^2,$$

and

$$\nabla_{\mathbf{A}_n} F = \left(\mathbf{A}_n \left(\mathbf{A}_N \odot \cdots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \cdots \mathbf{A}_1\right)^\top - \mathbf{M}_{(n)}\right)\left(\mathbf{A}_N \odot \cdots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \cdots \mathbf{A}_1\right).$$

Let

$$\mathbf{B}_n^{k-1} = \mathbf{A}_N^{k-1} \odot \cdots \mathbf{A}_{n+1}^{k-1} \odot \mathbf{A}_{n-1}^{k} \cdots \mathbf{A}_1^{k}. \tag{3.4}$$

We take $L_n^{k-1} = \max(\ell^{k-2}, \|(\mathbf{B}_n^{k-1})^\top \mathbf{B}_n^{k-1}\|)$, where $\ell^{k-2} = \min_n L_n^{k-2}$ and $\|\mathbf{A}\|$ is the spectral norm of $\mathbf{A}$. Let

$$\omega_n^{k-1} = \min\left(\hat{\omega}_{k-1}, \delta_\omega \sqrt{\frac{\ell^{k-2}}{L_n^{k-1}}}\right) \tag{3.5}$$

where $\delta_\omega < 1$ is pre-selected and $\hat{\omega}_{k-1} = \frac{t_{k-1}-1}{t_k}$ with $t_0 = 1$ and $t_k = \frac{1}{2}\left(1 + \sqrt{1 + 4t_{k-1}^2}\right)$. In addition, let $\hat{\mathbf{A}}_n^{k-1} = \mathbf{A}_n^{k-1} + \omega_n^{k-1}(\mathbf{A}_n^{k-1} - \mathbf{A}_n^{k-2})$, and $\hat{\mathbf{G}}_n^{k-1} = \left(\hat{\mathbf{A}}_n^{k-1}(\mathbf{B}_n^{k-1})^\top - \mathbf{M}_{(n)}\right)\mathbf{B}_n^{k-1}$ be the gradient. Then we derive the update (1.3c):

$$\mathbf{A}_n^k = \underset{\mathbf{A}_n \geq 0}{\operatorname{argmin}} \left\langle \hat{\mathbf{G}}_n^{k-1}, \mathbf{A}_n - \hat{\mathbf{A}}_n^{k-1} \right\rangle + \frac{L_n^{k-1}}{2}\left\|\mathbf{A}_n - \hat{\mathbf{A}}_n^{k-1}\right\|_F^2,$$

which can be written in the closed form

$$\mathbf{A}_n^k = \max\left(0, \hat{\mathbf{A}}_n^{k-1} - \hat{\mathbf{G}}_n^{k-1}/L_n^{k-1}\right). \tag{3.6}$$

At the end of iteration k, we check whether $F\left(\mathbf{A}^k\right) \geq F\left(\mathbf{A}^{k-1}\right)$. If so, we re-update $\mathbf{A}_n^k$ by (3.6) with $\hat{\mathbf{A}}_n^{k-1} = \mathbf{A}_n^{k-1}$, for $n = 1, \cdots, N$.

Algorithm 2 Alternating proximal gradient method for solving (3.2)

Input: nonnegative N-way tensor $\mathcal{M}$ and rank r.
Output: nonnegative factors $\mathbf{A}_1, \cdots, \mathbf{A}_N$.
Initialization: choose a positive number $\delta_\omega < 1$ and randomize $\mathbf{A}_n^{-1} = \mathbf{A}_n^0, n = 1, \cdots, N$, as nonnegative matrices of appropriate sizes.
for $k = 1, 2, \cdots$ **do**
  for $n = 1, 2, \cdots, N$ **do**
    Compute L_n^{k-1} and set ω_n^{k-1} according to (3.5);
    Let $\hat{\mathbf{A}}_n^{k-1} = \mathbf{A}_n^{k-1} + \omega_n^{k-1}(\mathbf{A}_n^{k-1} - \mathbf{A}_n^{k-2})$;
    Update $\mathbf{A}_n^k$ according to (3.6).
  end for
  if $F\left(\mathbf{A}^k\right) \geq F\left(\mathbf{A}^{k-1}\right)$ **then**
    Re-update $\mathbf{A}_n^k$ according to (3.6) with $\hat{\mathbf{A}}_n^{k-1} = \mathbf{A}_n^{k-1}, n = 1, \cdots, N$
  end if
  if stopping criterion is satisfied **then**
    Return $\mathbf{A}_1^k, \cdots, \mathbf{A}_N^k$.
  end if
end for

REMARK 3.1. *In (3.6), $\hat{\mathbf{G}}_n^{k-1}$ is most expensive to compute. To efficiently compute it, we write $\hat{\mathbf{G}}_n^{k-1} = \hat{\mathbf{A}}_n^{k-1}(\mathbf{B}_n^{k-1})^\top \mathbf{B}_n^{k-1} - \mathbf{M}_{(n)}\mathbf{B}_n^{k-1}$. Using $(\mathbf{A} \odot \mathbf{B})^\top(\mathbf{A} \odot \mathbf{B}) = (\mathbf{A}^\top\mathbf{A}) * (\mathbf{B}^\top\mathbf{B})$, we compute $(\mathbf{B}_n^{k-1})^\top \mathbf{B}_n^{k-1}$ by*

$$(\mathbf{B}_n^{k-1})^\top \mathbf{B}_n^{k-1} = \left((\mathbf{A}_1^k)^\top \mathbf{A}_1^k\right) * \cdots * \left((\mathbf{A}_{n-1}^k)^\top \mathbf{A}_{n-1}^k\right) * \left((\mathbf{A}_{n+1}^{k-1})^\top \mathbf{A}_{n+1}^{k-1}\right) * \cdots * \left((\mathbf{A}_N^{k-1})^\top \mathbf{A}_N^{k-1}\right).$$

Then, $\mathbf{M}_{(n)}\mathbf{B}_n^{k-1}$ can be obtained by the matricized-tensor-times-Khatri-Rao-product [6].

Alg. 2 summarizes how to apply Alg. 1 with update (1.3c) to problem (3.2).

REMARK 3.2. *When $N = 2$, $\mathcal{M}$ becomes a matrix, and Alg. 2 solves nonnegative matrix factorization.*

3.3. Convergence results. Since problem (3.2) is a special case of problem (1.1), the convergence results in Sec. 2 apply to Alg. 2. Let $\mathcal{D}_n = \mathbb{R}_+^{I_n \times r}$ and $\delta_{\mathcal{D}_n}(\cdot)$ be the indicator function on $\mathcal{D}_n$ for $n = 1, \cdots, N$. Then (3.2) is equivalent to

$$\min_{\mathbf{A}_1, \cdots, \mathbf{A}_N} Q(\mathbf{A}) \equiv F(\mathbf{A}) + \sum_{n=1}^N \delta_{\mathcal{D}_n}(\mathbf{A}_n). \tag{3.7}$$

According to the discussion in Sec. 2.2, Q is a semi-algebraic function and satisfies the KL property (2.13) at any feasible point. Further, we get $\theta \neq 0$ in (2.13) for Q at any critical point. By writing the first-order optimality conditions of (3.7), one can find that if $\left(\bar{\mathbf{A}}_1, \cdots, \bar{\mathbf{A}}_N\right)$ is a critical point, then so is $\left(t\bar{\mathbf{A}}_1, \frac{1}{t}\bar{\mathbf{A}}_2, \bar{\mathbf{A}}_3, \cdots, \bar{\mathbf{A}}_N\right)$ for any $t > 0$. Therefore, from Theorems 2.8 and 2.9 and the above discussions, we have

THEOREM 3.1. *Let $\{\mathbf{A}^k\}$ be the sequence generated by Alg. 2. Assume $\{\mathbf{A}^k\}$ is bounded and there is a positive constant ℓ such that $\ell \leq \ell^k$ for all k. Then $\{\mathbf{A}^k\}$ converges to a critical point $\bar{\mathbf{A}}$, and the asymptotic convergence rates in parts 2 and 3 of Theorem 2.9 apply.*

REMARK 3.3. *The boundedness of $\{\mathbf{A}^k\}$ guarantees that L_n^k is upper bounded. A simple way to make $\{\mathbf{A}^k\}$ bounded is to scale $(\mathbf{A}_1, \cdots, \mathbf{A}_N)$ so that $\|\mathbf{A}_1\|_F = \cdots = \|\mathbf{A}_N\|_F$ after each iteration. The existence of a positive ℓ can be satisfied if one changes L_n^k to $\max(L_n^k, L_{\min})$ for a positive constant $L_{\min}$.*

3.4. An algorithm for nonnegative tensor completion. Alg. 2 can be easily modified for solving the nonnegative tensor completion problem

$$\min_{\mathbf{A}_1,\cdots,\mathbf{A}_N\geq 0} \frac{1}{2}\|\mathcal{P}_\Omega(\boldsymbol{\mathcal{M}} - \mathbf{A}_1\circ\mathbf{A}_2\circ\cdots\circ\mathbf{A}_N)\|_F^2, \tag{3.8}$$

where $\Omega \subset [I_1]\times[I_2]\times\cdots[I_N]$ is the index set of the observed entries of $\boldsymbol{\mathcal{M}}$ and $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}})$ keeps the entries of $\boldsymbol{\mathcal{X}}$ in Ω and sets the remaining ones to zero. Nonnegative matrix completion (corresponding to $N = 2$) has been proposed in [74], where it is demonstrated that a low-rank and nonnegative matrix can be recovered from a small set of its entries by taking advantages of both low-rankness and nonnegative factors . To solve (3.8), we transform it into the equivalent problem

$$\min_{\boldsymbol{\mathcal{X}},\mathbf{A}_n\geq 0,n=1,\cdots,N} G(\mathbf{A},\boldsymbol{\mathcal{X}}) \equiv \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \mathbf{A}_1\circ\mathbf{A}_2\circ\cdots\circ\mathbf{A}_N\|_F^2, \text{ subject to } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}). \tag{3.9}$$

Our algorithm shall cycle through the decision variables $\mathbf{A}_1,\cdots,\mathbf{A}_N$ and $\boldsymbol{\mathcal{X}}$. To save space, we describe a modification to Alg. 2. At the k-th iteration of Alg. 2, we use $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}}^{k-1}$ wherever $\boldsymbol{\mathcal{M}}$ is referred to. Specifically, we use $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}}^{k-1}$ for the computation of $\hat{\mathbf{G}}_n^{k-1}$ in Line 8 and for the evaluation of F in Line 10 of Alg. 2. After Line 12, perform update (1.3a) on $\boldsymbol{\mathcal{X}}$ as

$$\boldsymbol{\mathcal{X}}^k = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}) + \mathcal{P}_{\Omega^c}(\mathbf{A}_1^k\circ\cdots\circ\mathbf{A}_N^k), \tag{3.10}$$

where Ω^c is the complement of Ω. Note that for a fixed $\mathbf{A}$, $G(\mathbf{A},\boldsymbol{\mathcal{X}})$ is a strongly convex function of $\boldsymbol{\mathcal{X}}$ with modulus 1, namely, the condition in item 1 of Assumption 2 is satisfied. Hence, according to Theorem 2.8, the convergence result for Alg. 2 still holds for this algorithm with extra update (3.10).

4. Numerical results. In this section, we test Alg. 2 for nonnegative matrix and three-way tensor factorization, as well as their completion. In our implementations, we choose $\delta_\omega = 0.9999$. The algorithm is terminated whenever $\frac{F_k - F_{k+1}}{1+F_k} \leq tol$ holds for three iterations in a row or $\frac{F_k}{\|\boldsymbol{\mathcal{M}}\|_F} \leq tol$ is met, where F_k is the objective value after iteration k and tol is specified below. We compare

- APG-MF: nonnegative matrix factorization (NMF) by Alg. 2 in Sec. 3.2;
- APG-TF: nonnegative tensor factorization (NTF) by Alg. 2 in Sec. 3.2;
- APG-MC: nonnegative matrix completion (NMC) by modified Alg. 2 in Sec. 3.4;
- APG-TC: nonnegative tensor completion (NTC) by modified Alg. 2 in Sec. 3.4.

All the tests were performed on a laptop with an i7-620m CPU and 3GB RAM and running 32-bit Windows 7 and Matlab 2010b with Tensor Toolbox of version 2.5 [7].

4.1. Nonnegative matrix factorization. We choose to compare the most popular and recent algorithms. The first two compared ones are the alternating least square method (Als-MF) [9, 53] and multiplicative updating method (Mult-MF) [39], which are available as MATLAB's function `nnmf` with specifiers `als` and `mult`, respectively. The recent ANLS method Blockpivot-MF is compared since it outperforms all other compared ANLS methods in both speed and solution quality [32]. Another compared algorithm is the recent ADM-based method ADM-MF [78]. Although both Blockpivot-MF and ADM-MF have superior performance than Als-MF and Mult-MF, we include them in the first two tests below due to their popularity.

We set $tol = 10^{-4}$ for all the compared algorithms except ADM-MF, for which we set $tol = 10^{-5}$ since it is a dual algorithm and 10^{-4} is too loose. The maximum number of iterations is set to 2000 for all the compared algorithms. The same random starting points are used for all the algorithms except for Mult-MF. Since Mult-MF is very sensitive to initial points, we set the initial point by running Mult-MF 10 iterations for 5 independent times and choose the best one. All the other parameters for Als-MF, Mult-MF, Blockpivot-MF and ADM-MF are set to their default values.

TABLE 4.1

*Comparison on nonnegative random $m \times n$ matrices for $n = 1000$; **bold** are **large error** or **slow time**.*

		APG-MF† (prop'd)		ADM-MF		Blockpivot-MF		Als-MF		Mult-MF	
m	r	relerr	time	relerr	time	relerr	time	relerr	time	relerr	time
200	10	9.98e-5	7.16e-1	**2.24e-3**	1.04e+0	5.36e-4	1.30e+0	**7.39e-3**	1.04e+0	**3.61e-2**	**2.67e+0**
200	20	9.97e-5	2.09e+0	**3.02e-3**	2.80e+0	**1.02e-3**	4.71e+0	**1.01e-2**	2.33e+0	**4.64e-2**	3.61e+0
200	30	9.97e-5	4.72e+0	**4.55e-3**	5.70e+0	**1.75e-3**	1.06e+1	**1.04e-2**	4.54e+0	**4.09e-2**	5.53e+0
500	10	9.98e-5	1.61e+0	**2.26e-3**	2.39e+0	5.11e-4	2.38e+0	**1.15e-2**	2.99e+0	**3.58e-2**	**7.76e+0**
500	20	9.98e-5	3.66e+0	**2.82e-3**	4.38e+0	5.53e-4	6.86e+0	**1.08e-2**	6.31e+0	**4.96e-2**	**7.99e+0**
500	30	9.98e-5	7.75e+0	**3.51e-3**	8.34e+0	5.75e-4	1.37e+1	**1.29e-2**	9.95e+0	**4.42e-2**	1.20e+1
1000	10	9.98e-5	2.86e+0	**2.11e-3**	3.44e+0	4.99e-4	3.18e+0	**1.54e-3**	8.04e+0	**3.25e-2**	**1.55e+1**
1000	20	9.98e-5	7.44e+0	**2.82e-3**	7.19e+0	5.46e-4	1.05e+1	**1.74e-2**	**1.75e+1**	**4.96e-2**	**1.61e+1**
1000	30	9.98e-5	1.27e+1	**3.01e-3**	1.28e+1	5.76e-4	2.00e+1	**1.99e-2**	**2.61e+1**	**4.57e-2**	2.21e+1

†: the relerr values of APG-MF are nearly the same due to the use of the same stopping tolerence.

TABLE 4.2

*Comparison on 2000 selected images from the CBCL face database; **bold** are **large error** or **slow time**.*

	APG-MF (proposed)		ADM-MF		Blockpivot-MF		Als-MF		Mult-MF	
r	relerr	time	relerr	time	relerr	time	relerr	time	relerr	time
30	1.91e-1	3.68	1.92e-1	7.33	1.90e-1	**21.5**	**3.53e-1**	3.15	**2.13e-1**	6.51
60	1.42e-1	12.5	1.43e-1	19.5	1.40e-1	**63.2**	**4.59e-1**	1.80	**1.74e-1**	12.1
90	1.13e-1	26.7	1.15e-1	34.2	1.12e-1	**111**	**6.00e-1**	2.15	**1.52e-1**	18.4

4.1.1. Synthetic data. Each matrix in this test is exactly low-rank and can be written in the form of $\mathbf{M} = \mathbf{LR}$, where $\mathbf{L}$ and $\mathbf{R}$ are generated by MATLAB commands `max(0,randn(m,q))` and `rand(q,n)`, respectively. It is worth mentioning that generating $\mathbf{R}$ by `rand(q,n)` makes the problems more difficult than `max(0,randn(q,n))` or `abs(randn(q,n))`. The algorithms are compared with fixed $n = 1000$ and m chosen from $\{200, 500, 1000\}$, q from $\{10, 20, 30\}$. The parameter r is set to q in (3.2). We use relative error relerr $= \|\mathbf{A}_1\mathbf{A}_2 - \mathbf{M}\|_F/\|\mathbf{M}\|_F$ and CPU time (in seconds) to measure performance. Table 4.1 lists the average results of 20 independent trials. From the table, we can see that APG-MF outperforms all the other algorithms in both CPU time and solution quality.

4.1.2. Image data. In this subsection, we compare APG-MF (proposed), ADM-MF, Blockpivot-MF, Als-MF and Mult-MF on the CBCL and ORL image databases used in [25, 42]. There are 6977 face images in the training set of CBCL, each having 19×19 pixels. Multiple images of each face are taken with varying illuminations and facial expressions. The first 2000 images are used for test. We vectorize every image and obtain a matrix $\mathbf{M}$ of size 361×2000. Rank r is chosen from $\{30, 60, 90\}$. The average results of 10 independent trials are given in Table 4.2. We can see that APG-MF outperforms ADM-MF in both speed and solution quality. APG-MF is as accurate as Blockpivot-MF but runs much faster. Als-MF and Mult-MF fail this test, and Als-MF stagnates at solutions of low quality at the very beginning. Due to the poor performance of Als-MF and Mult-MF, we only compare APG-MF, ADM-MF and Blockpivot-MF in the remaining tests.

The ORL database has 400 images divided into 40 groups. Each image has 112×92 pixels, and each group has 10 images of one face taken from 10 different directions and with different expressions. All the images are used for test. We vectorize each image and obtain a matrix $\mathbf{M}$ of size 10304×400. As in last test, we choose r from $\{30, 60, 90\}$. The average results of 10 independent trials are listed in Table 4.3. From the results, we can see again that APG-MF is better than ADM-MF in both speed and solution quality, and in far less time APG-MF achieves comparable relative errors as Blockpivot-MF.

TABLE 4.3
Comparison on the images from the ORL face database; ***bold*** *are* ***slow time****.*

	APG-MF (proposed)		ADM-MF		Blockpivot-MF	
r	relerr	time	relerr	time	relerr	time
30	1.67e-1	15.8	1.71e-1	46.5	1.66e-1	**74.3**
60	1.41e-1	42.7	1.45e-1	88.0	1.40e-1	**178**
90	1.26e-1	76.4	1.30e-1	127	1.25e-1	**253**

FIG. 4.1. *Hyperspectral data of* $150 \times 150 \times 163$*: four selected slices are shown*

4.1.3. Hyperspectral data. It has been shown in [54] that NMF can be applied to spectral data analysis. In [54], a regularized NMF model is also considered with penalty terms $\alpha\|\mathbf{A}_1\|_F^2$ and $\beta\|\mathbf{A}_2\|_F^2$ added in the objective of (3.2). The parameters α and β can be tuned for specific purposes in practice. Here, we focus on the original NMF model to show the effectiveness of our algorithm. However, our method can be easily modified for solving the regularized NMF model. In this test, we use a $150 \times 150 \times 163$ hyperspectral cube to test the compared algorithms. Each slice of the cube is reshaped as a column vector, and a 22500×163 matrix $\mathbf{M}$ is obtained. In addition, the cube is scaled to have a unit maximum element. Four selected slices before scaling are shown in Figure 4.1 corresponding to the 1st, 50th, 100th and 150th columns of $\mathbf{M}$. The dimension r is chosen from $\{20, 30, 40, 50\}$, and Table 4.4 lists the average results of 10 independent trials. We can see from the table that APG-MF is superior to ADM-MF and Blockpivot-MF in both speed and solution quality.

4.1.4. Nonnegative matrix completion. In this subsection, we compare APG-MC and the ADM-based algorithm (ADM-MC) proposed in [74] on the hyperspectral data used in last test. It is demonstrated in [74] that ADM-MC outperforms other matrix completion solvers such as APGL and LMaFit on recovering nonnegative matrices because ADM-MC takes advantages of data nonnegativity while the latter two do not. We fix the dimension $r = 40$ in (3.8) and choose sample ratio $\text{SR} \triangleq \frac{|\Omega|}{mn}$ from $\{0.20, 0.30, 0.40\}$, where the samples in Ω are chosen at random. The parameter δ_ω for APG-MC is set to 1, and all the parameters for ADM-MC are set to their default values. To make the comparison consistent, we let both of the algorithms run to a maximum time (sec) $T = 50, 100$, and we employ peak-signal-to-noise-ratio (PSNR) and mean squared error (MSE) to measure the performance of the two algorithms. Table 4.5 lists the average results of 10 independent trials. From the table, we can see that APG-MC is significantly better than ADM-MC in all cases.

4.2. Nonnegative three-way tensor factorization. To the best of our knowledge, all the existing algorithms for nonnegative tensor factorizations are extensions of those for nonnegative matrix factorization including multiplicative updating method [71], hierachical alternating least square algorithm [19], alternaing Poisson regression algorithm [17] and alternating nonnegative least square (ANLS) methods [31, 33]. We compare APG-TF with two ANLS methods AS-TF [31] and Blockpivot-TF [33], which are also proposed based on the CP decomposition and superior over many other algorithms. We set $tol = 10^{-4}$ and $maxit = 2000$ for all the compared algorithms. All the other parameters for Blockpivot-TF and AS-TF are set to their default values.

TABLE 4.4
Comparison on hyperspectral data of size $150 \times 150 \times 163$*;* ***bold*** *are* ***large error*** *or* ***slow time****.*

	APG-MF (proposed)		ADM-MF		Blockpivot-MF	
r	relerr	time	relerr	time	relerr	time
20	1.18e-2	34.2	**2.34e-2**	**87.5**	1.38e-2	62.5
30	9.07e-3	63.2	**2.02e-2**	116	1.10e-2	**143**
40	7.56e-3	86.2	**1.78e-2**	140	9.59e-3	**194**
50	6.45e-3	120	**1.58e-2**	182	8.00e-3	**277**

TABLE 4.5
Comparison on a hyperspectral data at stopping time $T = 50, 100$ *(sec);* ***bold*** *are* ***large error****.*

$T = 50$	APG-MC (proposed)		ADM-MC		$T = 100$	APG-MC (proposed)		ADM-MC	
Smpl. Rate	PSNR	MSE	PSNR	MSE	Smpl. Rate	PSNR	MSE	PSNR	MSE
0.20	32.30	5.89e-4	**28.72**	**1.35e-3**	0.20	32.57	5.54e-4	**28.80**	**1.33e-3**
0.30	40.65	8.62e-5	**33.58**	**4.64e-4**	0.30	41.19	7.61e-5	**33.69**	**4.52e-4**
0.40	45.77	2.66e-5	**38.52**	**1.46e-4**	0.40	46.03	2.50e-5	**38.69**	**1.41e-4**

4.2.1. Synthetic data. We compare APG-TF, Blockpivot-TF and AS-TF on randomly generated three-way tensors. Each tensor is $\boldsymbol{\mathcal{M}} = \mathbf{L} \circ \mathbf{C} \circ \mathbf{R}$, where $\mathbf{L}, \mathbf{C}$ are generated by MATLAB commands `max(0,randn(N1,q))` and `max(0,randn(N2,q))`, respectively, and $\mathbf{R}$ by `rand(N3,q)`. The algorithms are compared with two sets of (N_1, N_2, N_3) and $r = q = 10, 20, 30$. The relative error relerr $= \|\boldsymbol{\mathcal{M}} - \mathbf{A}_1 \circ \mathbf{A}_2 \circ \mathbf{A}_3\|_F / \|\boldsymbol{\mathcal{M}}\|_F$ and CPU time (sec) measure the performance of the algorithms. The average results of 10 independent runs are shown in Table 4.6, from which we can see that all the algorithms give similar results.

4.2.2. Image test. NMF does not utilize the spatial redundancy, and the matrix decomposition is not unique. Also, NMF factors tend to form the invariant parts of all images as ghosts while NTF factors can correctly resolve all the parts [63]. We compare APG-TF, Blockpivot-TF and AS-TF on two nonnegative three-way tensors in [63]. Each slice of the tensors corresponds to an image. The first tensor is $19 \times 19 \times 2000$ and is formed from 2000 images in the CBCL database, used in Sec. 4.1.2. The average performance of 10 independent runs with $r = 40, 50, 60$ are shown in Table 4.7. Another one has the size of $32 \times 32 \times 256$ and is formed with the 256 images in the Swimmer dataset [20]. The results of 10 independent runs with $r = 40, 50, 60$ are listed in Table 4.8. Both tests show that APG-TF is consistently faster than Blockpivot-TF and AS-TF. In particular, APG-TF is much faster than Blockpivot-TF and AS-TF with better solution quality in the second test.

4.2.3. Hyperspectral data. NTF is employed in [77] for hyperspectral unmixing. It is demonstrated that the cubic data can be highly compressed and NTF is efficient to identify the material signatures. We compare APG-TF with Blockpivot-TF and AS-TF on the $150 \times 150 \times 163$ hyperspectral cube, which is used in Sec. 4.1.3. For consistency, we let them run to a maximum time T and compare the relative errors. The parameter r is chosen from $\{30, 40, 50, 60\}$. The relative errors corresponding to $T = 10, 25, 50, 100$ are shown in Table 4.9, as the average of 10 independent trials. We can see from the table that APG-TF achieves the same accuracy much earlier than Blockpivot-TF and AS-TF.

4.2.4. Nonnegative tensor completion. Recently, [45] proposes tensor completion based on minimizing tensor n-rank, the matrix rank of mode-n matricization of a tensor. Using the matrix nuclear norm instead of matrix rank, they solve the convex program

$$\min_{\boldsymbol{\mathcal{X}}} \sum_{n=1}^{N} \alpha_n \|\mathbf{X}_{(n)}\|_*, \text{ subject to } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{M}}), \tag{4.1}$$

TABLE 4.6
Comparison on synthetic three-way tensors; ***bold*** *are* ***large error*** *or* ***slow time****.*

Problem Setting				APG-TF (proposed)		AS-TF		Blockpivot-TF	
N_1	N_2	N_3	q	relerr	time	relerr	time	relerr	time
80	80	80	10	8.76e-005	4.39e-001	7.89e-005	**8.64e-001**	8.62e-005	8.19e-001
80	80	80	20	9.47e-005	1.26e+000	**1.97e-004**	1.45e+000	**1.77e-004**	1.21e+000
80	80	80	30	9.65e-005	**2.83e+000**	**2.05e-004**	2.13e+000	**2.07e-004**	1.95e+000
50	50	500	10	9.15e-005	1.27e+000	1.07e-004	**1.91e+000**	9.54e-005	**1.87e+000**
50	50	500	20	9.44e-005	3.42e+000	**1.86e-004**	3.17e+000	**1.77e-004**	3.47e+000
50	50	500	30	9.74e-005	**7.11e+000**	**1.89e-004**	5.04e+000	**1.88e-004**	4.54e+000

TABLE 4.7
Comparison results on CBCL database; ***bold*** *are* ***slow time****.*

	APG-TF (proposed)		AS-TF		Blockpivot-TF	
r	relerr	time	relerr	time	relerr	time
40	1.85e-001	9.95e+000	1.86e-001	**2.99e+001**	1.85e-001	**2.04e+001**
50	1.68e-001	1.65e+001	1.68e-001	**4.55e+001**	1.69e-001	2.47e+001
60	1.53e-001	2.13e+001	1.56e-001	**4.16e+001**	1.56e-001	2.85e+001

where α_n's are pre-specified weights satisfying $\sum_n \alpha_n = 1$ and $\|\mathbf{A}\|_*$ is the nuclear norm of $\mathbf{A}$ defined as the sum of singular values of $\mathbf{A}$. Meanwhile, they proposed some algorithms to solve (4.1) or its relaxed versions, including simple low-rank tensor completion (SiLRTC), fast low-rank tensor completion (FaLRTC) and high accuracy low-rank tensor completion (HaLRTC). We compare APG-TC with FaLRTC on synthetic three-way tensors since FaLRTC is more efficient and stable than SiLRTC and HaLRTC. Each tensor is generated similarly as in Sec. 4.2.1. Rank q is chosen from $\{10, 20, 30\}$ and sampling ratio $\text{SR} = |\Omega|/(N_1N_2N_3)$ from $\{0.10, 0.30, 0.50\}$. For APG-TC, we use $r = q$ and $r = \lfloor 1.25q \rfloor$ in (3.8). We set $tol = 10^{-4}$ and $maxit = 2000$ for both algorithms. The weights α_n's in (4.1) are set to $\alpha_n = \frac{1}{3}, n = 1, 2, 3$, and the smoothing parameters for FaLRTC are set to $\mu_n = \frac{5\alpha_n}{N_n}, n = 1, 2, 3$. Other parameters of FaLRTC are set to their default values. The average results of 10 independent trials are shown in Table 4.10. We can see that APG-TC produces much more accurate solutions within less time.

4.3. Summary. Although our test results are obtained with a given set of parameters, it is clear from the results that, compared to the existing algorithms, the proposed ones can return solutions of similar or better quality in less time. Tuning the parameters of the compared algorithms can hardly obtain much improvement in both solution quality and time. We believe that the superior performance of the proposed algorithms is due to the use of prox-linear steps, which are not only easy to compute but also, as a local approximate, help avoid the small regions around certain local minima.

5. Conclusions. We have proposed a block coordinate descent method with three choices of update schemes for multi-convex optimization, with subsequence and global convergence guarantees under certain assumptions. Numerical results on both synthetic and real image data illustrate the high efficiency of the proposed algorithm.

Acknowledgements. This work is partly supported by ARL and ARO grant W911NF-09-1-0383, NSF grants DMS-0748839 and ECCS-1028790, and ONR Grant N00014-08-1-1101. The authors would like to thank three referees for their careful reviews and helpful comments. Also, they would like to thank Zhi-Quan (Tom) Luo for very helpful discussions and sharing his manuscript [57], and Michael Ulbrich for his

TABLE 4.8
Comparison results on Swimmer database; ***bold*** *are* ***large error*** *or* ***slow time****.*

	APG-TF (proposed)		AS-TF		Blockpivot-TF	
r	relerr	time	relerr	time	relerr	time
40	2.43e-001	2.01e+000	2.71e-001	**2.09e+001**	2.53e-001	**2.50e+001**
50	1.45e-001	3.21e+000	**2.00e-001**	**5.54e+001**	1.87e-001	**3.23e+001**
60	3.16e-002	6.91e+000	**1.10e-001**	**3.55e+001**	7.63e-002	**3.74e+001**

TABLE 4.9
Relative errors on hyperspectral data.

	APG-TF (proposed)				AS-TF				Blockpivot-TF			
$r \setminus T$	10	25	50	100	10	25	50	100	10	25	50	100
30	2.56e-1	2.53e-1	2.53e-1	2.53e-1	2.60e-1	2.56e-1	2.54e-1	2.53e-1	2.60e-1	2.56e-1	2.54e-1	2.53e-1
40	2.32e-1	2.27e-1	2.26e-1	2.26e-1	2.37e-1	2.30e-1	2.28e-1	2.26e-1	2.36e-1	2.29e-1	2.28e-1	2.27e-1
50	2.14e-1	2.07e-1	2.04e-1	2.04e-1	2.20e-1	2.11e-1	2.07e-1	2.06e-1	2.17e-1	2.10e-1	2.07e-1	2.05e-1
60	2.00e-1	1.91e-1	1.87e-1	1.86e-1	2.04e-1	1.95e-1	1.91e-1	1.88e-1	2.01e-1	1.94e-1	1.90e-1	1.88e-1

careful reading and pointing out one mistake in our proofs.

Appendix A. Proofs of Lemma 2.6 and Theorem 2.9.

A.1. Proof of Lemma 2.6. Without loss of generality, we assume $\bar{F} = 0$. Otherwise, we can consider $F - \bar{F}$. Let $B(\bar{\mathbf{x}}, \rho) \triangleq \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\| \le \rho\} \subset \mathcal{U}$ for some $\rho > 0$ where $\mathcal{U}$ is the neighborhood of $\bar{\mathbf{x}}$ in (2.14) with $\psi = F$, and let L_G be the global Lipschitz constant for $\nabla_{\mathbf{x}_i} f(\mathbf{x}), i = 1, \cdots, s$ within $B(\bar{\mathbf{x}}, \sqrt{10}\rho)$, namely,

$$\|\nabla_{\mathbf{x}_i} f(\mathbf{x}) - \nabla_{\mathbf{x}_i} f(\mathbf{y})\| \le L_G \|\mathbf{x} - \mathbf{y}\|, \ i = 1, \cdots, s$$

for any $\mathbf{x}, \mathbf{y} \in B(\bar{\mathbf{x}}, \sqrt{10}\rho)$.

The proof will follow two steps. The first step will show

CLAIM A.1. *Let* $\ell = \min_i \ell_i$, $L = \max_i L_i$ *and*

$$C_1 = \frac{9(L + sL_G)}{2\ell(1 - \delta_\omega)^2}, \ C_2 = 2\sqrt{\frac{2}{\ell}} + \frac{3}{1 - \delta_\omega}\sqrt{\frac{2 + 2\delta_\omega^2}{\ell}},$$

where ℓ_i, L_i*'s are the constants in Assumption 2. If* $F_k > \bar{F}$ *and*

$$C_1 \phi(F_0 - \bar{F}) + C_2 \sqrt{F_0 - \bar{F}} + \|\mathbf{x}^0 - \bar{\mathbf{x}}\| < \rho, \tag{A.1}$$

then

$$\mathbf{x}^k \in B(\bar{\mathbf{x}}, \rho), \ \forall k. \tag{A.2}$$

Note that (A.1) quantifies how close to $\bar{\mathbf{x}}$ the initial point $\mathbf{x}^0$ is required. The second step will establish

CLAIM A.2.

$$\sum_{k=N}^{\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| \le C_1 \phi(F_N - \bar{F}) + \|\mathbf{x}^{N-1} - \mathbf{x}^{N-2}\| + \frac{2 + \delta_\omega}{1 - \delta_\omega}\|\mathbf{x}^N - \mathbf{x}^{N-1}\|, \forall N \ge 2, \tag{A.3}$$

where C_1 *is specified in Claim A.1.*

Note (A.3) implies $\{\mathbf{x}^k\}$ is a Cauchy sequence, and thus $\mathbf{x}^k$ converges. Hence, if (A.2) and (A.3) both hold, then letting $\mathcal{B} = B(\bar{\mathbf{x}}, \rho)$ will prove the results of Lemma 2.6.

TABLE 4.10

Comparison results on synthetic nonnegative tensor completion; ***bold*** *are* ***bad*** *or* ***slow****.*

Problem Setting					APG-TC (prop'd) $r = q$		APG-TC (prop'd) $r = \lfloor 1.25q \rfloor$		FaLRTC	
N_1	N_2	N_3	q	SR	relerr	time	relerr	time	relerr	time
80	80	80	10	0.10	2.02e-004	4.09e+000	6.08e-004	6.88e+000	**4.61e-001**	**3.17e+001**
80	80	80	10	0.30	1.18e-004	2.52e+000	3.29e-004	5.85e+000	**1.96e-002**	**1.96e+001**
80	80	80	10	0.50	9.54e-005	2.22e+000	2.45e-004	5.28e+000	**1.13e-002**	**1.52e+001**
80	80	80	20	0.10	1.50e-004	9.55e+000	4.84e-004	**1.60e+001**	4.41e-001	**2.47e+001**
80	80	80	20	0.30	1.15e-004	6.08e+000	2.64e-004	**1.23e+001**	**1.43e-001**	**1.17e+001**
80	80	80	20	0.50	9.65e-005	5.01e+000	1.72e-004	**1.27e+001**	**1.46e-002**	**1.95e+001**
80	80	80	30	0.10	3.14e-003	1.64e+001	4.23e-004	**2.66e+001**	**4.00e-001**	**2.08e+001**
80	80	80	30	0.30	1.04e-004	1.12e+001	1.94e-004	**2.11e+001**	**2.22e-001**	8.12e+000
80	80	80	30	0.50	1.14e-004	9.91e+000	1.47e-004	**2.00e+001**	**5.60e-002**	1.28e+001
50	50	500	10	0.10	2.76e-004	1.16e+001	4.69e-004	2.03e+001	**5.52e-001**	**1.83e+002**
50	50	500	10	0.30	9.81e-005	6.24e+000	2.12e-004	**1.62e+001**	**8.58e-002**	**9.69e+001**
50	50	500	10	0.50	9.51e-005	5.34e+000	1.74e-004	**1.63e+001**	**1.25e-002**	**9.63e+001**
50	50	500	20	0.10	1.80e-004	2.45e+001	3.50e-004	**4.37e+001**	**4.82e-001**	**1.32e+002**
50	50	500	20	0.30	3.95e-003	1.34e+001	1.59e-004	**3.91e+001**	**2.76e-001**	**5.82e+001**
50	50	500	20	0.50	5.32e-003	1.18e+001	1.15e-004	**3.53e+001**	**9.44e-002**	**5.59e+001**
50	50	500	30	0.10	7.09e-003	3.90e+001	5.08e-004	6.76e+001	**4.32e-001**	**1.18e+002**
50	50	500	30	0.30	1.03e-004	2.54e+001	1.26e-004	**6.32e+001**	**2.76e-001**	**5.04e+001**
50	50	500	30	0.50	3.28e-003	2.30e+001	1.03e-004	**5.56e+001**	**1.62e-001**	**4.30e+001**

Proof of Claim A.1. We will prove $\mathbf{x}^k \in B(\bar{\mathbf{x}}, \rho)$ by induction on k.

Obviously, $\mathbf{x}^0 \in B(\bar{\mathbf{x}}, \rho)$ from (A.1). Hence, (A.2) holds for $k = 0$.

For $k = 1$, we have from (2.8) that

$$F_0 \geq F_0 - F_1 \geq \sum_{i=1}^{s} \frac{L_i^0}{2} \|\mathbf{x}_i^0 - \mathbf{x}_i^1\|^2 \geq \frac{\ell}{2} \|\mathbf{x}^0 - \mathbf{x}^1\|^2.$$

Hence, $\|\mathbf{x}^0 - \mathbf{x}^1\| \leq \sqrt{\frac{2}{\ell} F_0}$, and

$$\|\mathbf{x}^1 - \bar{\mathbf{x}}\| \leq \|\mathbf{x}^0 - \mathbf{x}^1\| + \|\mathbf{x}^0 - \bar{\mathbf{x}}\| \leq \sqrt{\frac{2}{\ell} F_0} + \|\mathbf{x}^0 - \bar{\mathbf{x}}\|,$$

which indicates $\mathbf{x}^1 \in B(\bar{\mathbf{x}}, \rho)$.

For $k = 2$, we have from (2.8) that (regard $\omega_i^k \equiv 0$ for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$)

$$F_0 \geq F_1 - F_2 \geq \sum_{i=1}^{s} \frac{L_i^1}{2} \|\mathbf{x}_i^1 - \mathbf{x}_i^2\|^2 - \sum_{i=1}^{s} \frac{L_i^1}{2} (\omega_i^1)^2 \|\mathbf{x}_i^0 - \mathbf{x}_i^1\|^2.$$

Note $L_i^1 (\omega_i^1)^2 \leq \delta_\omega^2 \ell^0$ for $i = 1, \cdots, s$. Thus, it follows from the above inequality that

$$\frac{\ell^1}{2} \|\mathbf{x}^1 - \mathbf{x}^2\|^2 \leq \sum_{i=1}^{s} \frac{L_i^1}{2} \|\mathbf{x}_i^1 - \mathbf{x}_i^2\|^2 \leq F_0 + \frac{\ell^0}{2} \delta_\omega^2 \|\mathbf{x}^0 - \mathbf{x}^1\|^2 \leq (1 + \frac{\ell^0}{\ell} \delta_\omega^2) F_0,$$

which implies $\|\mathbf{x}^1 - \mathbf{x}^2\| \leq \sqrt{\frac{2+2\delta_\omega^2}{\ell} F_0}$. Therefore,

$$\|\mathbf{x}^2 - \bar{\mathbf{x}}\| \leq \|\mathbf{x}^1 - \mathbf{x}^2\| + \|\mathbf{x}^1 - \bar{\mathbf{x}}\| \leq \left(\sqrt{\frac{2}{\ell}} + \sqrt{\frac{2 + 2\delta_\omega^2}{\ell}} \right) \sqrt{F_0} + \|\mathbf{x}^0 - \bar{\mathbf{x}}\|,$$

and thus $\mathbf{x}^2 \in B(\bar{\mathbf{x}}, \rho)$.

Suppose $\mathbf{x}^k \in B(\bar{\mathbf{x}}, \rho)$ for $0 \le k \le K$. We go to show $\mathbf{x}^{K+1} \in B(\bar{\mathbf{x}}, \rho)$. For $k \le K$, note

$$\begin{aligned}
-\nabla f_i^k(\mathbf{x}_i^k) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k) &\in \partial r_i(\mathbf{x}_i^k) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k),\ i \in \mathcal{I}_1,\\
-L_i^{k-1}(\mathbf{x}_i^k - \mathbf{x}_i^{k-1}) - \nabla f_i^k(\mathbf{x}_i^k) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k) &\in \partial r_i(\mathbf{x}_i^k) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k),\ i \in \mathcal{I}_2,\\
-L_i^{k-1}(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^{k-1}) - \nabla f_i^k(\hat{\mathbf{x}}_i^{k-1}) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k) &\in \partial r_i(\mathbf{x}_i^k) + \nabla_{\mathbf{x}_i} f(\mathbf{x}^k),\ i \in \mathcal{I}_3,
\end{aligned}$$

and

$$\partial F(\mathbf{x}^k) = \left\{\partial r_1(\mathbf{x}_1^k) + \nabla_{\mathbf{x}_1} f(\mathbf{x}^k)\right\} \times \cdots \times \left\{\partial r_s(\mathbf{x}_s^k) + \nabla_{\mathbf{x}_s} f(\mathbf{x}^k)\right\},$$

so (for $i \in \mathcal{I}_1 \cup \mathcal{I}_2$, regard $\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^{k-1}$ in $\mathbf{x}_i^k - \hat{\mathbf{x}}_i^{k-1}$ and $\hat{\mathbf{x}}_i^{k-1} = \mathbf{x}_i^k$ in $\nabla f_i^k(\hat{\mathbf{x}}_i^{k-1}) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^k)$, respectively)

$$\begin{aligned}
&\operatorname{dist}\left(0, \partial F(\mathbf{x}^k)\right)\\
\le\ &\left\|\left(L_1^{k-1}(\mathbf{x}_1^k - \hat{\mathbf{x}}_1^{k-1}), \cdots, L_s^{k-1}(\mathbf{x}_s^k - \hat{\mathbf{x}}_s^{k-1})\right)\right\|\\
&+ \textstyle\sum_{i=1}^s \left\|\nabla f_i^k(\hat{\mathbf{x}}_i^{k-1}) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^k)\right\|.
\end{aligned} \tag{A.4}$$

For the first term in (A.4), plugging in $\hat{\mathbf{x}}_i^{k-1}$ and recalling $L_i^{k-1} \le L, \omega_i^{k-1} \le 1$ for $i = 1, \cdots, s$, we can easily get

$$\begin{aligned}
&\left\|\left(L_1^{k-1}(\mathbf{x}_1^k - \hat{\mathbf{x}}_1^{k-1}), \cdots, L_s^{k-1}(\mathbf{x}_s^k - \hat{\mathbf{x}}_s^{k-1})\right)\right\|\\
\le\ &L\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right).
\end{aligned} \tag{A.5}$$

For the second term in (A.4), it is not difficult to verify

$$\left(\mathbf{x}_1^k, \cdots, \mathbf{x}_{i-1}^k, \hat{\mathbf{x}}_i^{k-1}, \cdots, \mathbf{x}_s^{k-1}\right) \in B(\bar{\mathbf{x}}, \sqrt{10}\rho).$$

In addition, note

$$\nabla f_i^k(\hat{\mathbf{x}}_i^{k-1}) = \nabla_{\mathbf{x}_i} f\left(\mathbf{x}_1^k, \cdots, \mathbf{x}_{i-1}^k, \hat{\mathbf{x}}_i^{k-1}, \cdots, \mathbf{x}_s^{k-1}\right).$$

Hence,

$$\begin{aligned}
&\sum_{i=1}^s \left\|\nabla_{\mathbf{x}_i} f_i^k(\hat{\mathbf{x}}_i^{k-1}) - \nabla_{\mathbf{x}_i} f(\mathbf{x}^k)\right\|\\
\le &\sum_{i=1}^s L_G \left\|\left(\mathbf{x}_1^k, \cdots, \mathbf{x}_{i-1}^k, \hat{\mathbf{x}}_i^{k-1}, \cdots, \mathbf{x}_s^{k-1}\right) - \mathbf{x}^k\right\|\\
\le &\, sL_G\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right).
\end{aligned} \tag{A.6}$$

Combining (A.4), (A.5) and (A.6) gives

$$\operatorname{dist}(\mathbf{0}, \partial F(\mathbf{x}^k)) \le (L + sL_G)\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right),$$

which together with the KL inequality (**??**) implies

$$\phi'(F_k) \ge (L + sL_G)^{-1}\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right)^{-1}. \tag{A.7}$$

Note that ϕ is concave and $\phi'(F_k) > 0$. Thus it follows from (2.8) and (A.7) that

$$\begin{aligned}
\phi(F_k) - \phi(F_{k+1}) \ge &\phi'(F_k)(F_k - F_{k+1})\\
\ge &\frac{\sum_{i=1}^s \left(L_i^k \|\mathbf{x}_i^k - \mathbf{x}_i^{k+1}\|^2 - \ell^{k-1}\delta_\omega^2 \|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2\right)}{2(L + sL_G)\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right)},
\end{aligned}$$

or equivalently

$$\sum_{i=1}^{s} L_i^k \|\mathbf{x}_i^k - \mathbf{x}_i^{k+1}\|^2 \leq 2(L+sL_G)\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right)(\phi(F_k) - \phi(F_{k+1})) + \sum_{i=1}^{s} \ell^{k-1}\delta_\omega^2 \|\mathbf{x}_i^{k-1} - \mathbf{x}_i^k\|^2.$$

Recalling $\ell \leq \ell^{k-1} \leq \ell^k \leq L_i^k \leq L$ for all i, k, we have from the above inequality that

$$\begin{aligned} & \|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 \\ \leq\ & \tfrac{2(L+sL_G)}{\ell}\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right)(\phi(F_k) - \phi(F_{k+1})) \\ & + \delta_\omega^2 \|\mathbf{x}^{k-1} - \mathbf{x}^k\|^2. \end{aligned} \tag{A.8}$$

Using inequalities $a^2 + b^2 \leq (a+b)^2$ and $ab \leq ta^2 + \frac{b^2}{4t}$ for $t > 0$, we get from (A.8) that

$$\begin{aligned} & \|\mathbf{x}^k - \mathbf{x}^{k+1}\| \\ \leq & \left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right)^{\frac{1}{2}} \left(\frac{2(L+sL_G)}{\ell}(\phi(F_k) - \phi(F_{k+1}))\right)^{\frac{1}{2}} \\ & + \delta_\omega \|\mathbf{x}^{k-1} - \mathbf{x}^k\| \\ \leq & \frac{1-\delta_\omega}{3}\left(\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\|\right) + \frac{3(L+sL_G)}{2\ell(1-\delta_\omega)}(\phi(F_k) - \phi(F_{k+1})) \\ & + \delta_\omega \|\mathbf{x}^{k-1} - \mathbf{x}^k\|, \end{aligned}$$

or equivalently

$$\begin{aligned} & 3\|\mathbf{x}^k - \mathbf{x}^{k+1}\| \\ \leq\ & (1+2\delta_\omega)\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + (1-\delta_\omega)\|\mathbf{x}^{k-1} - \mathbf{x}^{k-2}\| \\ & + \tfrac{9(L+sL_G)}{2\ell(1-\delta_\omega)}(\phi(F_k) - \phi(F_{k+1})). \end{aligned} \tag{A.9}$$

Summing up (A.9) over k from 2 to K and doing some eliminations give

$$\begin{aligned} & \sum_{k=2}^{K}(1-\delta_\omega)\|\mathbf{x}^k - \mathbf{x}^{k+1}\| + (2+\delta_\omega)\|\mathbf{x}^K - \mathbf{x}^{K+1}\| + (1-\delta_\omega)\|\mathbf{x}^{K-1} - \mathbf{x}^K\| \\ \leq & (1-\delta_\omega)\|\mathbf{x}^0 - \mathbf{x}^1\| + (2+\delta_\omega)\|\mathbf{x}^1 - \mathbf{x}^2\| + \frac{9(L+sL_G)}{2\ell(1-\delta_\omega)}(\phi(F_2) - \phi(F_{K+1})). \end{aligned}$$

Recalling $\|\mathbf{x}^0 - \mathbf{x}^1\| \leq \sqrt{\frac{2}{\ell}F_0}$ and $\|\mathbf{x}^1 - \mathbf{x}^2\| \leq \sqrt{\frac{2+2\delta_\omega^2}{\ell}F_0}$, we have from the above inequality that

$$\begin{aligned} \|\mathbf{x}^{K+1} - \bar{\mathbf{x}}\| \leq & \sum_{k=2}^{K}\|\mathbf{x}^k - \mathbf{x}^{k+1}\| + \|\mathbf{x}^2 - \bar{\mathbf{x}}\| \\ \leq & \sqrt{\frac{2}{\ell}F_0} + \frac{2+\delta_\omega}{1-\delta_\omega}\sqrt{\frac{2+2\delta_\omega^2}{\ell}F_0} + \frac{9(L+sL_G)}{2\ell(1-\delta_\omega)^2}(\phi(F_2) - \phi(F_{K+1})) \\ & + \|\mathbf{x}^2 - \bar{\mathbf{x}}\| \\ \leq & \frac{9(L+sL_G)}{2\ell(1-\delta_\omega)^2}\phi(F_0) + \left(2\sqrt{\frac{2}{\ell}} + \frac{3}{1-\delta_\omega}\sqrt{\frac{2+2\delta_\omega^2}{\ell}}\right)\sqrt{F_0} + \|\mathbf{x}^0 - \bar{\mathbf{x}}\|. \end{aligned}$$

Hence, $\mathbf{x}^{K+1} \in B(\bar{\mathbf{x}}, \rho)$, and this completes the proof of Claim A.1.

Proof of Claim A.2. We will prove (A.3) from (A.9). Indeed, (A.9) holds for all $k \geq 0$. Summing it over k from N to T and doing some eliminations yield

$$\begin{aligned}&\sum_{k=N}^{T}(1-\delta_\omega)\|\mathbf{x}^k-\mathbf{x}^{k+1}\|+(2+\delta_\omega)\|\mathbf{x}^T-\mathbf{x}^{T+1}\|+(1-\delta_\omega)\|\mathbf{x}^{T-1}-\mathbf{x}^T\|\\ \leq&(1-\delta_\omega)\|\mathbf{x}^{N-2}-\mathbf{x}^{N-1}\|+(2+\delta_\omega)\|\mathbf{x}^{N-1}-\mathbf{x}^N\|+\frac{9(L+sL_G)}{2\ell(1-\delta_\omega)}(\phi(F_N)-\phi(F_{T+1})),\end{aligned}$$

which implies

$$\sum_{k=N}^{\infty}\|\mathbf{x}^k-\mathbf{x}^{k+1}\|\leq\|\mathbf{x}^{N-2}-\mathbf{x}^{N-1}\|+\frac{2+\delta_\omega}{1-\delta_\omega}\|\mathbf{x}^{N-1}-\mathbf{x}^N\|+\frac{9(L+sL_G)}{2\ell(1-\delta_\omega)^2}\phi(F_N)$$

by letting $T\to\infty$. This completes the proof of Claim A.2.

A.2. Proof of Theorem 2.9. If $\theta=0$, we must have $F(\mathbf{x}^{k_0})=F(\bar{\mathbf{x}})$ for some k_0. Otherwise, $F(\mathbf{x}^k)>F(\bar{\mathbf{x}})$ for all sufficiently large k. The Kurdyka-Łojasiewicz inequality gives $c\cdot\operatorname{dist}(\mathbf{0},\partial F(\mathbf{x}^k))\geq 1$ for all $k\geq 0$, which is impossible since $\mathbf{x}^k\to\bar{\mathbf{x}}$ and $\mathbf{0}\in\partial F(\bar{\mathbf{x}})$. The finite convergence now follows from the fact that $F(\mathbf{x}^{k_0})=F(\bar{\mathbf{x}})$ implies $\mathbf{x}^k=\mathbf{x}^{k_0}=\bar{\mathbf{x}}$ for all $k\geq k_0$.

For $\theta\in(0,1)$, we assume $F(\mathbf{x}^k)>F(\bar{\mathbf{x}})=0$ and use the same notation as in the proof of Lemma 3. Define $S_k=\sum_{i=k}^{\infty}\|\mathbf{x}^i-\mathbf{x}^{i+1}\|$. Then (A.3) can be written as

$$S_k\leq C_1\phi(F_k)+\frac{2+\delta_\omega}{1-\delta_\omega}(S_{k-1}-S_k)+S_{k-2}-S_{k-1},\ \text{for } k\geq 2,$$

which implies

$$S_k\leq C_1\phi(F_k)+\frac{2+\delta_\omega}{1-\delta_\omega}(S_{k-2}-S_k),\ \text{for } k\geq 2, \tag{A.10}$$

since $S_{k-2}-S_{k-1}\geq 0$. Using $\phi(s)=cs^{1-\theta}$, we have from (A.7) for sufficiently large k that

$$c(1-\theta)(F_k)^{-\theta}\geq(L+sL_G)^{-1}\left(\|\mathbf{x}^k-\mathbf{x}^{k-1}\|+\|\mathbf{x}^{k-1}-\mathbf{x}^{k-2}\|\right)^{-1},$$

or equivalently $(F_k)^\theta\leq c(1-\theta)(L+sL_G)(S_{k-2}-S_k)$. Then,

$$\phi(F_k)=c(F_k)^{1-\theta}\leq c\big(c(1-\theta)(L+sL_G)(S_{k-2}-S_k)\big)^{\frac{1-\theta}{\theta}}. \tag{A.11}$$

Letting $C_3=C_1c\big(c(1-\theta)(L+sL_G)\big)^{\frac{1-\theta}{\theta}}$ and $C_4=\frac{2+\delta_\omega}{1-\delta_\omega}$, we have from (A.10) and (A.11) that

$$S_k\leq C_3\left(S_{k-2}-S_k\right)^{\frac{1-\theta}{\theta}}+C_4\left(S_{k-2}-S_k\right). \tag{A.12}$$

When $\theta\in(0,\frac{1}{2}]$, i.e., $\frac{1-\theta}{\theta}\geq 1$, (A.12) implies that $S_k\leq(C_3+C_4)(S_{k-2}-S_k)$ for sufficiently large k since $S_{k-2}-S_k\to 0$, and thus $S_k\leq\frac{C_3+C_4}{1+C_3+C_4}S_{k-2}$. Note that $\|\mathbf{x}^k-\bar{\mathbf{x}}\|\leq S_k$. Therefore, item 2 holds with $\tau=\sqrt{\frac{C_3+C_4}{1+C_3+C_4}}<1$ and sufficiently large C.

When $\theta\in(\frac{1}{2},1)$, i.e., $\frac{1-\theta}{\theta}<1$, we get

$$S_N^\nu+S_{N-1}^\nu-S_{K+1}^\nu-S_K^\nu\geq\mu(N-K), \tag{A.13}$$

for $\nu=\frac{1-2\theta}{1-\theta}<0$, some constant $\mu>0$ and any $N>K$ with sufficiently large K by the same argument as in the proof of Theorem 2 of [3]. Note $S_N\leq S_{N-1}$ and $\nu<0$. Hence, (A.13) implies

$$S_N\leq\left(\frac{1}{2}\left(S_{K+1}^\nu+S_K^\nu+\mu(N-K)\right)\right)^{\frac{1}{\nu}}\leq CN^{-\frac{1-\theta}{2\theta-1}},$$

for sufficiently large C and N. This completes the proof.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, Signal Processing, IEEE Transactions on, 54 (2006), pp. 4311–4322.

[2] S. Amari, A. Cichocki, H.H. Yang, et al., *A new learning algorithm for blind signal separation*, Advances in neural information processing systems, (1996), pp. 757–763.

[3] H. Attouch and J. Bolte, *On the convergence of the proximal algorithm for nonsmooth functions involving analytic features*, Mathematical Programming, 116 (2009), pp. 5–16.

[4] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality*, Mathematics of Operations Research, 35 (2010), pp. 438–457.

[5] A. Auslender, *Optimisation: méthodes numériques*, Masson, 1976.

[6] B.W. Bader and T.G. Kolda, *Efficient matlab computations with sparse and factored tensors*, SIAM Journal on Scientific Computing, 30 (2009), pp. 205–231.

[7] B. W. Bader, T. G. Kolda, et al., *Matlab tensor toolbox version 2.5*, January 2012.

[8] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.

[9] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis, 52 (2007), pp. 155–173.

[10] J. Bobin, Y. Moudden, J.L. Starck, J. Fadili, and N. Aghanim, *SZ and CMB reconstruction using generalized morphological component analysis*, Statistical Methodology, 5 (2008), pp. 307–317.

[11] J. Bochnak, M. Coste, and M.F. Roy, *Real algebraic geometry*, vol. 36, Springer Verlag, 1998.

[12] P. Bofill and M. Zibulevsky, *Underdetermined blind source separation using sparse representations*, Signal processing, 81 (2001), pp. 2353–2362.

[13] J. Bolte, A. Daniilidis, and A. Lewis, *The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, SIAM Journal on Optimization, 17 (2007), pp. 1205–1223.

[14] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota, *Clarke subgradients of stratifiable functions*, SIAM Journal on Optimization, 18 (2007), pp. 556–572.

[15] S.S. Chen, D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit*, SIAM review, (2001), pp. 129–159.

[16] Y. Chen, M. Rege, M. Dong, and J. Hua, *Non-negative matrix factorization for semi-supervised data clustering*, Knowledge and Information Systems, 17 (2008), pp. 355–379.

[17] E.C. Chi and T.G. Kolda, *On tensors, sparsity, and nonnegative factorizations*, Arxiv preprint arXiv:1112.2414, (2011).

[18] S. Choi, A. Cichocki, H.M. Park, and S.Y. Lee, *Blind source separation and independent component analysis: A review*, Neural Information Processing-Letters and Reviews, 6 (2005), pp. 1–57.

[19] A. Cichocki and A.H. Phan, *Fast local algorithms for large scale nonnegative matrix and tensor factorizations*, IEICE transactions on fundamentals of electronics, communications and computer science, 92 (2009), pp. 708–721.

[20] D. Donoho and V. Stodden, *When does non-negative matrix factorization give a correct decomposition into parts*, Advances in neural information processing systems, 16 (2003).

[21] M.P. Friedlander and K. Hatz, *Computing non-negative tensor factorizations*, Optimisation Methods and Software, 23 (2008), pp. 631–647.

[22] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[23] S.P. Han, *A successive projection method*, Mathematical Programming, 40 (1988), pp. 1–14.

[24] C. Hildreth, *A quadratic programming procedure*, Naval Research Logistics Quarterly, 4 (1957), pp. 79–85.

[25] P.O. Hoyer, *Non-negative matrix factorization with sparseness constraints*, The Journal of Machine Learning Research, 5 (2004), pp. 1457–1469.

[26] T.P. Jung, S. Makeig, C. Humphries, T.W. Lee, M.J. Mckeown, V. Iragui, and T.J. Sejnowski, *Removing electroencephalographic artifacts by blind source separation*, Psychophysiology, 37 (2000), pp. 163–178.

[27] C. Jutten and J. Herault, *Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture*, Signal processing, 24 (1991), pp. 1–10.

[28] J. Karhunen, A. Hyvarinen, R. Vigário, J. Hurri, and E. Oja, *Applications of neural blind separation to signal and image processing*, in Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, vol. 1, IEEE, 1997, pp. 131–134.

[29] H.A.L. Kiers, *Towards a standardized notation and terminology in multiway analysis*, Journal of Chemometrics, 14 (2000), pp. 105–122.

[30] H. Kim and H. Park, *Non-negative matrix factorization based on alternating non-negativity constrained least squares*

and active set method, SIAM J. Matrix Anal. Appl, 30 (2008), pp. 713–730.

[31] H. Kim, H. Park, and L. Eldén, *Non-negative tensor factorization based on alternating large-scale non-negativity-constrained least squares*, in Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on, IEEE, 2007, pp. 1147–1151.

[32] J. Kim and H. Park, *Toward faster nonnegative matrix factorization: A new algorithm and comparisons*, in Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, IEEE, 2008, pp. 353–362.

[33] ———, *Fast nonnegative tensor factorization with an active-set-like method*, High-Performance Scientific Computing, (2012), pp. 311–326.

[34] Y.D. Kim and S. Choi, *Nonnegative Tucker decomposition*, in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[35] T.G. Kolda and B.W. Bader, *Tensor decompositions and applications*, SIAM review, 51 (2009), p. 455.

[36] K. Kurdyka, *On gradients of functions definable in o-minimal structures*, in Annales de l'institut Fourier, vol. 48, Chartres: L'Institut, 1950-, 1998, pp. 769–784.

[37] M. Lai and Y. Wang, *An unconstrained ℓ_q minimization with $0 < q < 1$ for sparse solution of under-determined linear systems*, SIAM J. Optimization, 21 (2011), pp. 82–101.

[38] D.D. Lee and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[39] ———, *Algorithms for Non-Negative Matrix Factorization*, Advances in Neural Information Processing Systems, 13 (2001), pp. 556–562.

[40] H. Lee, A. Battle, R. Raina, and A.Y. Ng, *Efficient sparse coding algorithms*, Advances in neural information processing systems, 19 (2007), pp. 801–808.

[41] S.Z. Li, X.W. Hou, H.J. Zhang, and Q.S. Cheng, *Learning spatially localized, parts-based representation*, in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, IEEE, 2001, pp. 207–212.

[42] C.J. Lin, *Projected gradient methods for nonnegative matrix factorization*, Neural Computation, 19 (2007), pp. 2756–2779.

[43] J.K. Lin, D.G. Grier, and J.D. Cowan, *Feature extraction approach to blind source separation*, in Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop, IEEE, 1997, pp. 398–405.

[44] J. Liu, J. Liu, P. Wonka, and J. Ye, *Sparse non-negative tensor factorization using columnwise coordinate descent*, Pattern Recognition, (2011).

[45] J. Liu, P. Musialski, P. Wonka, and J. Ye, *Tensor completion for estimating missing values in visual data*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2012).

[46] S. Łojasiewicz, *Sur la géométrie semi-et sous-analytique*, Ann. Inst. Fourier (Grenoble), 43 (1993), pp. 1575–1595.

[47] D.G. Luenberger, *Introduction to linear and nonlinear programming*, (1973).

[48] Z.Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Annals of Operations Research, 46 (1993), pp. 157–178.

[49] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 689–696.

[50] O.L. Mangasarian and R. Leone, *Parallel successive overrelaxation methods for symmetric linear complementarity problems and linear programs*, Journal of Optimization Theory and Applications, 54 (1987), pp. 437–446.

[51] M. Mørup, L.K. Hansen, and S.M. Arnfred, *Algorithms for sparse nonnegative Tucker decompositions*, Neural computation, 20 (2008), pp. 2112–2131.

[52] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, 1970.

[53] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[54] V.P. Pauca, J. Piper, and R.J. Plemmons, *Nonnegative matrix factorization for spectral data analysis*, Linear Algebra and its Applications, 416 (2006), pp. 29–47.

[55] V.P. Pauca, F. Shahnaz, M.W. Berry, and R.J. Plemmons, *Text mining using nonnegative matrix factorizations*, in Proc. SIAM Inter. Conf. on Data Mining, Orlando, FL, 2004.

[56] M.J.D. Powell, *On search directions for minimization algorithms*, Mathematical Programming, 4 (1973), pp. 193–201.

[57] M. Razaviyayn, M. Hong, and Z.Q. Luo, *A unified convergence analysis of coordinatewise successive minimization methods for nonsmooth optimization*, preprint, (2012).

[58] B. Recht, M. Fazel, and P.A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM review, 52 (2010), pp. 471–501.

[59] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[60] R.T. Rockafellar and R.J.B. Wets, *Variational analysis*, vol. 317, Springer Verlag, 1998.

[61] RWH Sargent and DJ Sebastian, *On the convergence of sequential minimization algorithms*, Journal of Optimization Theory and Applications, 12 (1973), pp. 567–575.

[62] C. Serviere and P. Fabry, *Principal component analysis and blind source separation of modulated sources for electro-mechanical systems diagnostic*, Mechanical systems and signal processing, 19 (2005), pp. 1293–1311.

[63] A. Shashua and T. Hazan, *Non-negative tensor factorization with applications to statistics and computer vision*, in Proceedings of the 22nd international conference on Machine learning, ACM, 2005, pp. 792–799.

[64] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological), (1996), pp. 267–288.

[65] P. Tseng, *Dual coordinate ascent methods for non-strictly convex minimization*, Mathematical Programming, 59 (1993), pp. 231–247.

[66] ———, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of Optimization Theory and Applications, 109 (2001), pp. 475–494.

[67] Paul Tseng and Sangwoon Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423.

[68] L.R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[69] L. Wang, J. Zhu, and H. Zou, *Hybrid huberized support vector machines for microarray classification and gene selection*, Bioinformatics, 24 (2008), pp. 412–419.

[70] J. Warga, *Minimizing certain convex functions*, Journal of the Society for Industrial and Applied Mathematics, 11 (1963), pp. 588–593.

[71] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recognition Letters, 22 (2001), pp. 1255–1261.

[72] Z. Wen, D. Goldfarb, and K. Scheinberg, *Block coordinate descent methods for semidefinite programming*, Handbook on Semidefinite, Conic and Polynomial Optimization, (2012), pp. 533–564.

[73] W. Xu, X. Liu, and Y. Gong, *Document clustering based on non-negative matrix factorization*, in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, 2003, pp. 267–273.

[74] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, *An alternating direction algorithm for matrix completion with nonnegative factors*, Journal of Frontiers of Mathematics in China, Special Issue on Computational Mathematics, 7 (2011), pp. 365–384.

[75] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68 (2006), pp. 49–67.

[76] S. Zafeiriou, *Algorithms for nonnegative tensor factorization*, Tensors in Image Processing and Computer Vision, (2009), pp. 105–124.

[77] Q. Zhang, H. Wang, R.J. Plemmons, and V. Pauca, *Tensor methods for hyperspectral data analysis: a space object material identification study*, JOSA A, 25 (2008), pp. 3001–3012.

[78] Y. Zhang, *An alternating direction algorithm for nonnegative matrix factorization*, Rice Technical Report, (2010).

[79] M. Zibulevsky and B.A. Pearlmutter, *Blind source separation by sparse decomposition in a signal dictionary*, Neural computation, 13 (2001), pp. 863–882.