AN EXACT PENALTY APPROACH FOR OPTIMIZATION WITH NONNEGATIVE ORTHOGONALITY CONSTRAINTS

BO JIANG∗, XIANG MENG†, ZAIWEN WEN‡, AND XIAOJUN CHEN§

Abstract. Optimization with nonnegative orthogonality constraints has wide applications in machine learning and data sciences. It is NP-hard due to some combinatorial properties of the constraints. We first propose an equivalent optimization formulation with nonnegative and multiple spherical constraints and an additional single nonlinear constraint. Various constraint qualifications, the first- and second-order optimality conditions of the equivalent formulation are discussed. We design a class of exact penalty models in which the nonnegative and multiple spherical constraints are kept. The penalty models are exact if the penalty parameter is sufficient large other than going to infinity. A practical penalty algorithm with rounding technique is then developed. It uses a second-order method to approximately solve a series of subproblems with nonnegative and multiple spherical constraints. Extensive numerical results on the projection problem, orthogonal nonnegative matrix factorization problems and the K-indicators model show the effectiveness of our proposed approach.

Key words. exact penalty, nonnegative orthogonality constraint, second-order method, constraint qualification, optimality condition

AMS subject classifications. 65K05, 90C30, 90C46, 90C90

1. Introduction. In this paper, we consider optimization with nonnegative orthogonality constraints, namely,

\[
\min_{X \in \mathbb{R}^{n \times k}} f(X) \quad \text{s.t.} \quad X \top X = I_k, \ X \succeq 0,
\]

where \(1 \leq k \leq n\), \(I_k\) is the \(k\)-by-\(k\) identity matrix and \(f: \mathbb{R}^{n \times k} \to \mathbb{R}\) is continuously differentiable. The feasible set of (1.1) is denoted as \(S_{n,k}^+ := \{X \in \mathbb{R}^{n \times k} : X \top X = I_k, \ X \succeq 0\}\). The non-negativity in \(S_{n,k}^+\) destroys the smoothness of \(S_{n,k} := \{X \in \mathbb{R}^{n \times k} : X \top X = I_k\}\) and introduces some combinatorial features to \(S_{n,k}^+\). Specifically, a matrix \(X \in S_{n,k}^+\) means that each row of \(X\) has at most one positive element and each column of \(X\) takes the unit norm.

Due to the combinatorial features, solving (1.1) is generally NP-hard. Actually, problem (1.1) covers the NP-hard quadratic assignment problem and the more general optimization over permutation matrices [23] as special cases. Besides, the constraint \(X \in S_{n,k}^+\) also appears in the \(k\)-means clustering [11, 14], the min-cut problem [34], etc. Several typical instances of problem (1.1) are briefly reviewed as follows.

1.1. Applications. We mainly introduced three classes of problem (1.1). The first one is the so-called trace minimization with nonnegative orthogonality constraints, formulated as

\[
\min_{X \in S_{n,k}^+} \text{tr}(X \top M X),
\]

∗School of Mathematical Sciences, Key Laboratory for NSLSCS of Jiangsu Province, Nanjing Normal University, CHINA (jiangbo@njnu.edu.cn). Research supported in part by the Young Elite Scientists Sponsorship Program by CAST (2017QNRC001), the NSFC grants 11501298 and 11671036.
†School of Mathematical Sciences, Peking University, CHINA (1700010614@pku.edu.cn). Research supported in part by the NSFC grants 11421101 and 11831002.
‡Beijing International Center for Mathematical Research, Peking University, CHINA (wenzw@pku.edu.cn). Research supported in part by the NSFC grants 11421101 and 11831002.
§Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong. (xiaojun.chen@polyu.edu.hk). Research supported in part by the Hong Kong Research Grant Council PolyU153099/14P.
where $M \in \mathbb{R}^{n \times n}$ is symmetric. If $M = -AA^\top$ with $A \in \mathbb{R}^{n \times r}$ being some data matrix, (1.2) is known as nonnegative principal component analysis [30, 44]. If $M = D - W$ with $W$ being a similarity matrix corresponding to $n$ objects and $D$ is a diagonal matrix having the same main diagonal as $We$, where $e$ is the all-one vector, (1.2) is known as the nonnegative Laplacian embedding [28]. If $W = D - W + \mu R$ with some particularly chosen matrix $R$ and nonnegative regularization parameter $\mu$, (1.2) is known as the discriminative nonnegative spectral clustering [41].

The second one is the orthogonal nonnegative matrix factorization (ONMF) [17]. Given the data matrix $A \in \mathbb{R}^{n \times r}$, ONMF solves

$$\min_{X \in S_n^{+,k}, Y \in \mathbb{R}^{r \times k}} \|A - XY^\top\|_F^2.$$  

Based on the idea of approximating the data matrix $A$ by its nonnegative subspace projection, Yang and Oja [42] proposed the orthonormal projective nonnegative matrix factorization (OPNMF) model as follows:

$$\min_{X \in S_n^{+,k}} \|A - XX^\top A\|_F^2.$$  

Models (1.3) and (1.4) are equivalent since the optimal solutions $\hat{X}$ and $\hat{Y}$ of (1.3) satisfy the relation $\hat{Y} = A^\top \hat{X}$. Yang and Oja [42] also proposed a special OPNMF model by replacing the Frobenius norm in (1.4) by the Kullback-Leibler divergence of $A$ and $XX^\top A$. Besides, [25, 32] considered the orthogonal symmetric non-negative matrix factorization models.

The third one is an efficient K-indicators model for data clustering proposed by Chen et al. [15]. Let $U \in S_n^{k,k}$ be the features matrix extracted from the data matrix $A$, the K-indicators model in [15] reads

$$\min_{X \in S_n^{+,k}, Y \in S^{k,k}} \|UY - X\|_F^2 \quad \text{s.t.} \quad \|X_{i,:}\|_0 = 1, i \in [n],$$

where $X_{i,:}$ is the $i$-th row of $X$ and $\|X_{i,:}\|_0$ is the total number of nonzero elements in $X_{i,:}$.

### 1.2. Related works

The existing works rarely considered the general problem (1.1), and most of them focused on some special formulations of (1.1). We briefly review some main existing methods. For solving ONMF model (1.3), motivated by the multiplicative update methods for nonnegative matrix factorization, Ding et al. [17] and Yoo and Choi [43] gave two different multiplicative update schemes; by establishing the equivalence of ONMF with a weighted variant of spherical $k$-means, Pompoli et al. [33] proposed an EM-like algorithm; Pompoli et al. [33] also designed an augmented Lagrangian method via penalizing the nonnegative constraints but keeping the orthogonality constraints; Li et al. [26] and Wang et al. [36] considered the nonconvex penalty approach by keeping the nonnegative constraints. Some theoretical properties of the nonconvex penalty model were investigated in [36]. For solving OPNMF model (1.4), Yang and Oja [42] designed a specific multiplicative update method; Pan and Ng [31] introduced a convex relaxation model, wherein the relaxed model is solved by the alternating direction method of multipliers. We remark that the multiplicative update scheme for solving problem (1.3) or (1.4) highly depends on the specific formulation of the objective function, so it is not easy to extend this class of methods to solve the general problem (1.1). Besides, Wen and Yin [37] designed an
augmented Lagrangian method by penalizing the nonnegative constraints but keeping the orthogonality constraints for solving the quadratic assignment problem; Chen et al. [15] proposed a semi-convex relaxation model and construct a double-layered alternating projection scheme to solve the K-indicators model (1.5).

1.3. Our contribution. In this paper, a general class of exact penalty models is proposed to solve problem (1.1). Specifically, we first give a new characterization of $S_{n,k}^+$, which consists of the nonnegative, multiple spherical constraints and a simple nonlinear constraint. Based on this equivalent characterization, we obtain an equivalent optimization formulation as

$$
\min_{X \in \mathcal{OB}_{n,k}^+} f(X) \quad \text{s.t.} \quad \|XV\|_F = 1,
$$

where $\mathcal{OB}_{n,k}^+ = \{X \in \mathbb{R}^{n \times k} : \|x_j\| = 1, x_j \geq 0, j \in [k]\}$. The various constraint qualifications (CQs), and the first- and second-order optimality conditions are investigated for problem (1.6). Particularly, only the Abadie CQ (ACQ) or Guignard CQ (GCQ) holds in most cases. Then we propose a useful rounding procedure to approximate the projection of a matrix onto $S_{n,k}^+$ when it is near to $S_{n,k}^+$. This rounding procedure can also give an upper bound estimation of the distance between a matrix and $S_{n,k}^+$. Based on this estimation, we propose a general class of exact penalty models, where we keep the simple nonnegative and multiple spherical constraints. We show that if the penalty parameters are chosen to be larger than a positive constant, the optimal solution of the exact penalty problem (possibly a rounding procedure and a postprocessing will be invoked) is also optimal for the original problem. Then we develop a practical exact penalty algorithm which approximately solves a series of penalty subproblems and performs the rounding procedure to generate some orthogonal nonnegative matrices. We also discuss how to use the proposed penalty algorithmic framework to solve a more general model

$$
\min_{X \in S_{n,k}^+, Y \in \mathcal{Y}} f(X, Y),
$$

where $\mathcal{Y}$ is some simple set such as the projection to the set $\mathcal{Y}$ is easy to compute. To solve the subproblem efficiently, we develop a second-order algorithm for solving optimization over $\mathcal{OB}_{n,k}^+$, which is of independent interest. Finally, numerical results on the projection problem and ONMF on synthetic data, text clustering, hyperspectral unmixing and K-indicators model demonstrate the efficiency of our approach.

1.4. Organization. The rest of this paper is organized as follows. A reformulation of problem (1.1) is given in section 2. The various constraint qualifications are discussed in section 2.1, and the first- and second-order optimality conditions are given in section 2.2 and section 2.3, respectively. We give a useful rounding procedure to approximate the projection of $X \in \mathcal{OB}_{n,k}^+$ onto $S_{n,k}^+$ in section 3.1, then propose a general exact penalty model in section 3.2 and develop a practical algorithm in section 3.3. We investigate a second-order method for optimization over $\mathcal{OB}_{n,k}^+$ in section 4. A variety of numerical results are presented in section 5. Finally, we make some concluding remarks in section 6.

1.5. Notations. For a positive integer $n$, denote $[n] := \{1, \ldots, n\}$. The $j$-th column (resp. $i$-th row) of a matrix $Z$ with appropriate dimension is denoted by $Z_{\cdot,j}$ (resp. $Z_{i,\cdot}$). For simplicity, we also denote $z_j := Z_{\cdot,j}$. Define the matrix $\text{sgn}(Z)$ of the
same size of $Z$ with $\sgn(Z)_{ij} = |Z_{ij}|/Z_{ij}$ if $Z_{ij} \neq 0$ and $\sgn(Z)_{ij} = 0$ otherwise. Define the matrix $\Pi_+(Z)$ of the same size as $Z$ with $(\Pi_+(Z))_{ij} = \max(Z_{ij}, 0)$. Define the support set $\text{supp}(Z) = \{(i, j) : Z_{ij} \neq 0\}$. The total number of nonzero elements of $Z$ is $\|Z\|_0$. The Frobenius norm of the matrix $Z$ is $\|Z\|_F$ while the 2-norm of the vector $z$ is $\|z\|$. For $z \in \mathbb{R}^n$, $\text{Diag}(z) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the main diagonal being $z$. For $Z \in \mathbb{R}^{n \times n}$, $\text{diag}(Z) \in \mathbb{R}^n$ is the main diagonal of $Z$. For simplicity, we use $\text{Diag}(Z)$ to denote $\text{Diag}(\text{diag}(Z))$. Let $\text{Off}(Z) = Z - \text{Diag}(Z)$. The inner product between two matrices $A$ and $B$ with the same sizes is $\langle A, B \rangle = \text{tr}(A^\top B)$. The notation $0 \preceq A \perp B \succeq 0$ means that $A \succeq 0$ and $B \succeq 0$ component-wisely and $A \circ B = 0$, where $\circ$ means the Hadamard product operation. Let $e_i$ be the unit vector along component $i$, its dimension is always clear from the context.

Throughout this paper, the matrix $V \in \mathbb{R}^{k \times r}$ $(1 \leq r \leq k)$ with $\|V\|_F = 1$ is fixed, and the corresponding global constant $\omega := \min_{i,j \in [k]} (VV^\top)_{ij}$ is positive.

2. An equivalent formulation of problem (1.1). We first give an equivalent algebraic characterization of $S^{n,k}_+$ and then a reformulation of problem (1.1).

**Lemma 2.1.** For any $X \in \mathcal{OB}^{n,k}_+$, there holds that

\begin{equation}
\|XV\|_F \geq 1,
\end{equation}

where the equality holds if and only if $X \in S^{n,k}_+$. Furthermore, we have

\begin{equation}
S^{n,k}_+ = \mathcal{X}_V := \mathcal{OB}^{n,k}_+ \cap \{X \in \mathbb{R}^{n \times k} : \|XV\|_F = 1\}.
\end{equation}

**Proof.** With $\|V\|_F = 1$ and $X \in \mathcal{OB}^{n,k}_+$, we have $\|XV\|_F^2 - 1 = \langle VV^\top, X^\top X - I_p \rangle = \sum_{i,j \in [k], i \neq j} (VV^\top)_{ij} (x_i^\top x_j)$, which with $VV^\top > 0$ implies that $\|XV\|_F^2 - 1 \geq 0$. The equality holds if and only if $x_i^\top x_j = 0$ for $i, j \in [k]$ and $i \neq j$, which with $X \in \mathcal{OB}^{n,k}_+$ means that $X \in S^{n,k}_+$. Hence (2.2) follows directly. The proof is completed. \hfill \Box

With the equivalent characterization (2.2) of $S^{n,k}_+$, we rewrite problem (1.1) as (1.6). Throughout this paper, we mainly focus on the formulation (1.6). By using the Lagrangian type method and keeping the simple constraint $X \in \mathcal{OB}^{n,k}_+$, it is more reasonable to consider (1.6) rather than (1.1) since we only need to estimate a Lagrange multiplier other than a symmetric Lagrange multiplier matrix of size $k \times k$.

2.1. Constraint qualifications. We first define some notations which will be used later. A key observation for $X \in \mathcal{X}_V$ is that each row of $X$ has at most one positive elements. Define the sets $\Omega_0(X) = \{(i, j) \in [n] \times [k] : X_{ij} = 0\}$, we have

\begin{equation}
\text{supp}(X) \cup \Omega_0(X) = [n] \times [k], \quad \text{supp}(X) \cap \Omega_0(X) = \emptyset.
\end{equation}

Define $\Omega_0'(X) = \{(i, j) \in \Omega_0(X) : |X_{i,:}| > 0\}$, $\Omega_0''(X) = \{(i, j) \in \Omega_0(X) : |X_{i,:}| = 0\}$. Clearly, we have $\Omega_0'(X) \neq \emptyset$ and

\begin{equation}
\Omega_0(X) = \Omega_0'(X) \cup \Omega_0''(X), \quad \Omega_0'(X) \cap \Omega_0''(X) = \emptyset.
\end{equation}

Problem (1.6) is actually a nonlinear programming (A.1) with $x = \text{vec}(X)$, $\mathcal{E} = \{1, \ldots, k + 1\}$ and $c_j(x) = x_j^\top x_j - 1, j \in [k], c_{k+1}(x) = \|XV\|_F - 1$, and $c_j(x) = x_{j-(k+1)}^\top x_j, j \in \mathcal{I} = \{k + 2, \ldots, nk + (k + 1)\}$. As to the definitions of LICQ, MFCQ, RCPLD, ACQ and GCQ for general nonlinear programming, one can refer to Definition A.1.
Lemma 2.2. Consider a feasible $X \in \mathcal{X}_l$. If $k = 1$, then LICQ holds at $X$; if $2 \leq k < n$ and $\|X\|_0 = n$, then ACQ holds but RPCLD fails to hold; if $2 \leq k = n$, then RPCLD holds but MFQ fails to hold; if $2 \leq k < n$ and $\|X\|_0 < n$, then GCQ holds but ACQ fails to hold.

Proof. Case I. $k = 1$. It is easy to see that the vectors $X$ and $e_i$, $i \notin \text{supp}(X)$ are linearly independent, which means that the LICQ holds at $X$.

Case II. $2 \leq k < n$ and $\|X\|_0 = n$, namely, each row of $X$ has exactly one positive element. In this case $\Omega_0'(X) = \Omega_0(X)$ and $\Omega_0''(X) = \emptyset$. Define the set $D_0(X) = \{D \in \mathbb{R}^{n \times k} : x_j^T d_j = 0, j \in [k], D_{ij} = 0, (i, j) \in \Omega_0(X)\}$. According to (A.3), we have

$$\text{LFD}_X(X) = \left\{ D \in \mathbb{R}^{n \times k} : x_j^T d_j = 0, j \in [k], D_{ij} \geq 0, (i, j) \in \Omega_0(X) \right\} = D_0(X),$$

where the last equality uses the fact that $VV^T > 0$ and $X \in \mathcal{X}_l$. On the other hand, for any $0 \neq D \in \text{LFD}_X(X)$, define the $j$-th column of $X^{(i)}$ as

$$x_j^{(i)} = x_j + \delta d_j \frac{X_{ij}}{\|x_j + \delta d_j\|}, \quad \delta_i = \min_{l \in [k]} \frac{X_{ij}}{|D_{ij}|}.$$

We hence know that $X^{(i)} \in \mathcal{X}_l$, $X^{(i)} \rightarrow X$ and $(X^{(i)} - X)\|X^{(i)} - X\|_F \rightarrow D/\|D\|_F$. This, together with (A.2) means that $D \in T\mathcal{C}_X(X)$ and further that $T\mathcal{C}_X(X) = \text{LFD}_X(X)$. Following from Definition A.1-(i), we know that ACQ holds at $X$.

Define $M_{eq}(X) := [\text{BlDiag}(X) \text{vec}(XVV^T)]/\|XV\|_F \in \mathbb{R}^{nk \times (k+1)}$ with its $j$-th column being $\nabla c_j(x)$, $j \in E$, where the elements of $\text{BlDiag}(X) \in \mathbb{R}^{nk \times k}$ are all zeros except that $\text{BlDiag}(X)_{n(j-1)+1:nj,j} = x_j$, $j \in [k]$. Besides, we know that $\{\nabla c_j(x)\}_{j \in I(X)}$ is $\{e_{n(i-1)+j}\}_{(i, j) \in \Omega_0(X)}$. Clearly, we have rank($M_{eq}(X)$) = $k + 1$, and there exists a neighborhood $\mathcal{N}(X)$ of $X$ such that rank($M_{eq}(Y)$) = $k + 1$ for all $Y \in \mathcal{N}(X)$. Noting that $\|XV\|_F = 1$, we have

$$\text{vec}(XVV^T) + \sum_{j \in [k]} \alpha_j[M_{eq}(X)]_{:,j} - \sum_{(i, j) \in \Omega_0(X)} \beta_{ij} e_{n(i-1)+j} = 0,$$

where $\alpha_j = -\frac{1}{2} (VV^T)_{jj}$, $\beta_{ij} = (X\text{Off}(VV^T))_{ij} > 0$. This means that the vectors $\{[M_{eq}(X)]_{:,j}\}_{j \in [k]}$ and $\{-e_{n(i-1)+j}\}_{(i, j) \in \Omega_0(X)}$ are positive-linearly dependent. Recalling $\|X\|_0 > k$, without loss of generality, we assume $\|x_1\|_0 \geq 2$. Take $Y$ with $y_2 = x_2 + 6\delta x_1$, where $w = [1 \ 2 \ \cdots \ n]^T$ and $y_j = x_j$, $j \in [k]/\{2\}$. Here the constant $\delta > 0$ is chosen such that $Y \in \mathcal{N}(X)$. It is not hard to verify that the vectors $\{[M_{eq}(Y)]_{:,j}\}_{j \in [k]}$ and $\{-e_{n(i-1)+j}\}_{(i, j) \in \Omega_0(X)}$ are linearly dependent. Hence, we know from Definition A.1-(iii) that RPCLD does not hold at $X$.

Case III. $2 \leq k = n$. In this case, $X$ is a permutation matrix. Clearly, we know that rank($M_{eq}(X)$) = $n + 1$, and there exist a neighborhood $\mathcal{N}(X)$ of $X$ such that rank($M_{eq}(Y)$) = $n + 1$ for all $Y \in \mathcal{N}(X)$. Similar to Case II, it is not hard to verify that the vectors $\{[M_{eq}(X)]_{:,j}\}_{j \in [n+1]}$ and $\{-e_{n(i-1)+j}\}_{(i, j) \in \Omega_0(X)}$ are positive-linearly dependent. And for any $j \subseteq \Omega_0(X)$, the vectors $\{[M_{eq}(X)]_{:,j}\}_{j \in [n+1]}$ and $\{-e_{n(i-1)+j}\}_{(i, j) \in \Omega_0(X)}$ are not positive-linearly dependent. On the other hand, consider any neighborhood $\mathcal{N}(X)$ of $X$, for any $Y \in \mathcal{N}(X)$, we always know that the vectors $\{[M_{eq}(X)]_{:,j}\}_{j \in [n+1]}$ and $\{-e_{n(i-1)+j}\}_{(i, j) \in \mathcal{J}}$ are linearly dependent since the number of vectors are larger than the dimension of the vector. This means RPCLD holds.

On the other hand, it is easy to see that $\text{LFD}_X(X) = \{0\}$ and the set obtained via replacing $D_{ij} \geq 0$ by $D_{ij} > 0$ therein is empty. This means MFCQ does not hold.
Case IV. $2 \leq k < n$ and $\|X\|_0 < n$. In this case, $\Omega_0''(X) \neq \emptyset$. Let $\mathcal{D}_1(X) = \{ D \in \mathbb{R}^{n \times k} : D_{ij} \geq 0, (i, j) \in \Omega_0''(X) \}$ and

$$\mathcal{D}_2(X) = \mathcal{D}_1(X) \cap \{ D \in \mathbb{R}^{n \times k} : \|D_{i,:}\|_0 \leq 1, \|X_{i,:}\| = 0 \}.$$  

Similar to the derivation in Case II, we have

$$(2.5) \quad \text{LFD}_{XV}(X) = \mathcal{D}_0(X) \cap \mathcal{D}_1(X), \quad \text{TC}_{XV}(X) = \text{LFD}_{XV}(X) \cap \mathcal{D}_2(X).$$

By some easy calculations, we have

$$\mathcal{D}_0(X)^\circ = \{ D \in \mathbb{R}^{n \times k} : d_j = \alpha_jx_j \text{ with } \alpha_j \in \mathbb{R} \ \forall j \in [k] \},$$

and $\mathcal{D}_1(X)^\circ = \mathcal{D}_2(X)^\circ$ with

$$\mathcal{D}_2(X)^\circ = \{ D \in \mathbb{R}^{n \times k} : D_{ij} = 0, (i, j) \in \text{supp}(X) \cup \Omega_0'(X); D_{ij} \leq 0, (i, j) \in \Omega_0''(X) \}.$$  

With (2.5) and the above results, using the property of the polar cone \(^1\), we obtain

$$\text{TC}_{XV}(X)^\circ = \text{LFD}_{XV}(X)^\circ = \text{cl}(\mathcal{D}_0(X)^\circ \cup \mathcal{D}_1(X)^\circ)$$

$$= \left\{ D \in \mathbb{R}^{n \times k} : D_{ij} = \alpha_jx_{ij}, (i, j) \in \text{supp}(X) \cup \Omega_0'(X) \text{ with } \alpha_j \in \mathbb{R}, D_{ij} \leq 0, (i, j) \in \Omega_0''(X) \right\},$$

where the last equality uses (2.3) and (2.4). Therefore, the GCQ holds at $X$ in this case. On the other hand, it follows from (2.5) that $\text{TC}_{XV}(X) \subset \text{LFD}_{XV}(X)$ which with Definition A.1-(ii) means ACQ does not hold at $X$. The proof is completed. \(\square\)

Finally, we summarize in Table 1 the CQ results mentioned above. Note that $\text{LICQ} \implies \text{MFCQ} \implies \text{RCPLD} \implies \text{ACQ} \implies \text{GCQ}$.

| Cases          | LICQ | MFCQ | RCPLD | ACQ | GCQ |
|----------------|------|------|-------|-----|-----|
| $k = 1$        | $\sqrt{\ }$ | $\sqrt{\ }$ | $\sqrt{\ }$ | $\sqrt{\ }$ | $\sqrt{\ }$ |
| $2 \leq k < n$, $|X|_0 = n$ | $\times$ | $\times$ | $\sqrt{\ }$ | $\sqrt{\ }$ | $\sqrt{\ }$ |
| $2 \leq k \leq n$, $|X|_0 = n$ | $\times$ | $\times$ | $\times$ | $\sqrt{\ }$ | $\sqrt{\ }$ |
| $2 \leq k < n$, $|X|_0 < n$ | $\times$ | $\times$ | $\times$ | $\times$ | $\sqrt{\ }$ |

### 2.2. First-order optimality conditions of problem (1.6)

Let $\Lambda \in \mathbb{R}^k$ be the Lagrange multiplier vector corresponding to $\|x_j\|^2 = 1, j \in [k]$, $Z \in \mathbb{R}^{n \times k}$ the Lagrange multiplier matrix corresponding to $X \geq 0$, and $\lambda \in \mathbb{R}$ the Lagrange multiplier corresponding to $\|XV\|_F = 1$. Define the Lagrangian function for problem (1.6) as

$$(2.6) \quad \mathcal{L}(X, \Lambda, Z, \lambda) = f(X) - \sum_{j \in [k]} \Lambda_j(\|x_j\|^2 - 1) - \langle Z, X \rangle + \lambda(\|XV\|_F - 1).$$

Define the Riemannian gradient with respect to the oblique manifold $\mathcal{OB}^{n,k} := \{ X \in \mathbb{R}^{n \times k} : \|x_j\| = 1, j \in [k] \}$, see [2] for more details, as

$$(2.7) \quad \text{grad} f(X) = \nabla f(X) - X \text{Diag}(X^T \nabla f(X)).$$

\(^1\text{Let } \mathcal{K}_i, i = 1, 2 \text{ be closed convex cone. The polar cone of } \mathcal{K}_i \text{ is given by } \mathcal{K}_i^\circ = \{ v : v^T d \leq 0 \ \forall d \in \mathcal{K}_i \}. \text{ Then } (\mathcal{K}_1 \cap \mathcal{K}_2)^\circ = \text{cl}((\mathcal{K}_1^\circ \cup \mathcal{K}_2^\circ)), \text{ where cl( ) is the closure of a set. See for instance [10, pp. 70].} \)
THEOREM 2.3 (First-order necessary conditions of problem (1.6)). Suppose that \( \bar{X} \in X_V \) is a local minimizer of problem (1.6). Then \( \bar{X} \) is a stationary point of problem (1.6). That is, there exists \( \bar{\lambda} \) such that

\[
0 \leq \bar{X} \perp \nabla f(\bar{X}) + \bar{\lambda} \text{Off}(VV^\top) \geq 0.
\]

**Proof.** Lemma 2.2 tells that GCQ holds at \( \bar{X} \). Therefore there exist \( \bar{\lambda} \in \mathbb{R}^k, \bar{\lambda} \in \mathbb{R} \) and \( \bar{Z} \in \mathbb{R}^{n \times k} \) such that \( 0 \leq \bar{X} \perp \bar{Z} \geq 0 \) and \( \nabla X \mathcal{L}(\bar{X}, \bar{\lambda}, \bar{Z}) = 0 \), which with (2.6) and \( \bar{X} \in X_V \) implies that

\[
\nabla f(\bar{X}) - \bar{X} \text{Diag}(2\bar{\lambda} - \bar{\lambda} \text{diag}(VV^\top)) - \bar{Z} + \bar{\lambda} \text{Off}(VV^\top) = 0.
\]

(2.9)

For \( \bar{X} \in X_V \), it follows from (2.2) that \( \bar{X}^\top \bar{X} = I_k \). Multiplying \( \bar{X}^\top \) on both sides of (2.9) and then performing the \( \text{diag}() \) operator, we have

\[
2\bar{\lambda} - \bar{\lambda} \text{diag}(VV^\top) = \text{diag}(\bar{X}^\top \nabla f(\bar{X})),
\]

which again with (2.9) and (2.7) implies that

\[
\bar{Z} = \nabla f(\bar{X}) + \bar{\lambda} \text{Off}(VV^\top).
\]

This fact with \( 0 \leq \bar{X} \perp \bar{Z} \geq 0 \) completes the proof. \( \Box \)

**Remark 2.4.** Since \( \bar{X} \in X_V \), it is not hard to verify that \( [\text{Off}(VV^\top)]_{ij} = 0 \) \( \forall (i, j) \in \supp(\bar{X}) \cup \Omega''_0(\bar{X}), \) \[\text{Off}(VV^\top)]_{ij} > 0 \forall (i, j) \in \Omega''_0(\bar{X}) \) and \( [\nabla f(\bar{X})]_{ij} = [\nabla f(\bar{X})]_{ij} \forall (i, j) \in \Omega''_0(\bar{X}) \). Hence, from (2.10), we have

\[
\bar{Z}_{ij} = \begin{cases} 
[\nabla f(\bar{X})]_{ij} & (i, j) \in \supp(\bar{X}), \\
[\nabla f(\bar{X})]_{ij} + \bar{\lambda}[\text{Off}(VV^\top)]_{ij} & (i, j) \in \Omega''_0(\bar{X}), \\
[\nabla f(\bar{X})]_{ij} & (i, j) \in \Omega''_0(\bar{X}).
\end{cases}
\]

(2.11)

Therefore, (2.8) is equivalent to

\[
[\nabla f(\bar{X})]_{ij} = 0 \forall (i, j) \in \supp(\bar{X}), 
[\nabla f(\bar{X})]_{ij} \geq 0 \forall (i, j) \in \Omega''(\bar{X}),
\]

since we can always choose

\[
\bar{\lambda} \geq \Lambda(\bar{X}) := \max_{(i, j) \in \Omega''_0(\bar{X})} \left\{ \frac{-[\nabla f(\bar{X})]_{ij}}{[\text{Off}(VV^\top)]_{ij}} \right\}
\]

(2.13)

such that \( \bar{Z}_{ij} \geq 0 \) holds for \( (i, j) \in \Omega''_0(\bar{X}) \).

2.3. Second-order optimality conditions of problem (1.6). We now discuss the second-order optimality conditions of problem (1.6) and we assume that \( f \) is twice continuously differentiable. The proof of Theorem 2.3 tells \( 2\bar{\lambda} - \bar{\lambda} \text{diag}(VV^\top) = \text{diag}(\bar{X}^\top \nabla f(\bar{X})) \). With the definition of the Lagrangian function (2.6), we have

\[
\nabla_{X, \bar{X}}^2 \mathcal{L}(\bar{X}, \bar{\lambda}, \bar{Z})[D] \cdot \text{Hess} f(\bar{X})[D] + \bar{\lambda} \text{Off}(VV^\top),
\]

where \( D \) satisfies \( x_j^\top d_j = 0 \) for \( j \in [k] \) and

\[
\text{Hess} f(\bar{X})[D] := \nabla^2 f(\bar{X})[D] - D \text{Diag}(\bar{X}^\top \nabla f(\bar{X})).
\]

Specializing (A.4) to (1.6), we have with

\[
\text{SNCD}_{X_V}(\bar{X}, \bar{Z}) = \left\{ D \in \mathbb{R}^{n \times k} : \begin{array}{ll}
\bar{X} + \alpha^{(l)} D^{(l)} & \in X_V, \alpha^{(l)} > 0, \alpha^{(l)} \rightarrow 0, D^{(l)} \rightarrow D, \\
\text{if } \bar{Z}_{ij} > 0, \alpha^{(l)} D^{(l)} \end{array} \right\}
\]

(2.14)
With (2.11), (2.12) and the expression (2.5) of $\text{TC}_{XV}(\bar{X})$, we know $\text{SNCD}_{XV}(\bar{X},\bar{Z})$ is always the same for different $\bar{Z}$. Hence, we write

\begin{equation}
\text{SNCD}_{XV}(\bar{X}) := \text{SNCD}_{XV}(\bar{X},\bar{Z}) = \text{TC}_{XV}(\bar{X}) \cap D_{3}(\bar{X})
\end{equation}

with $D_{3}(\bar{X}) = \{D \in \mathbb{R}^{n \times k} : D_{ij} = 0 \text{ if } |\nabla f(\bar{X})|_{ij} > 0, (i,j) \in \Omega_{0}^{n}(\bar{X}) \}$. Similarly, specializing (A.5) to problem (1.6), we have $\text{LNCD}_{XV}(\bar{X}) = \text{LFD}_{XV}(\bar{X}) \cap \{D \in \mathbb{R}^{n \times k} : D_{ij} = 0 \text{ if } Z_{ij} > 0, (i,j) \in \Omega_{0}(\bar{X}) \}$. Using (2.5), (2.11) and (2.12), we further have

\begin{equation}
\text{LNCD}_{XV}(\bar{X}) = \text{LFD}_{XV}(\bar{X}) \cap D_{3}(\bar{X}).
\end{equation}

We are now ready to establish the second-order optimality conditions as follows.

**Theorem 2.5 (Second-order necessary conditions of problem (1.6)).** If $\bar{X} \in X_{V}$ is a local minimizer of problem (1.6) then

\begin{equation}
\langle D, \nabla f(\bar{X}) | D \rangle \geq 0, \quad \text{for all } D \in \text{SNCD}_{XV}(\bar{X}).
\end{equation}

**Proof.** Let $\bar{\lambda}$ be the Lagrange multiplier corresponding to $\|XV\|_{F} = 1$. Note that $\bar{\lambda}$ should satisfy (2.13). Following from [35, Theorem 8.3.3] and the fact that $\bar{X}$ is a local minimizer of problem (1.6), we have from (2.14) that

\begin{equation}
\langle D, \nabla f(\bar{X}) | D + \bar{\lambda} \text{DOff}(VV^{T}) \rangle \geq 0, \quad \text{for all } D \in \text{SNCD}_{XV}(\bar{X}).
\end{equation}

If $D \in \text{SNCD}_{XV}(\bar{X})$, we know from (2.16) that $D^{T}D$ must be diagonal. Then we have

\begin{equation}
\langle D, \text{DOff}(VV^{T}) \rangle = \text{tr}(D^{T} \text{DOff}(VV^{T})) = 0,
\end{equation}

which with (2.19) implies (2.18). The proof is completed. \qed

**Theorem 2.6 (Second-order sufficient conditions of problem (1.6)).** Suppose that $\bar{X} \in X_{V}$ is a stationary point of problem (1.6). Suppose also that there exists a Lagrange multiplier $\bar{\lambda}$ corresponding to $\|XV\|_{F} = 1$ with $\bar{\lambda} \geq A(\bar{X})$ such that

\begin{equation}
\langle D, \nabla f(\bar{X}) | D + \bar{\lambda} \text{DOff}(VV^{T}) \rangle > 0, \quad \text{for all } D \in \text{LNCD}_{XV}(\bar{X})/\{0\}.
\end{equation}

Then $\bar{X}$ is a strict local minimizer of (1.6).

**Proof.** It follows directly from, for instance [35, Theorems 8.3.4]. \qed

To end this section, we give a remark on the second-order conditions.

**Remark 2.7.** Consider the case when $\Omega^{n}_{0}(X) = \emptyset$, namely, $\|\bar{X}\|_{0} = n$. Following from (2.16), (2.17) and $\text{TC}_{XV}(\bar{X}) = \text{LFD}_{XV}(\bar{X})$ (see Case II in the proof of Lemma 2.2), we have $\text{SNCD}_{XV}(\bar{X}) = \text{LNCD}_{XV}(\bar{X}) = \text{LFD}_{XV}(\bar{X})$. Recalling (2.20), we thus know that (2.18) and (2.21) become $\langle D, \nabla f(\bar{X}) | D \rangle \geq 0, \forall D \in \text{LFD}_{XV}(\bar{X})$ and $\langle D, \nabla f(\bar{X}) | D \rangle > 0, \forall D \in \text{LFD}_{XV}(\bar{X})/\{0\}$, respectively.

3. **An exact penalty approach.** We first give an assumption which will be used in this section. Let $f^{*}$ and $\Theta_{f}$ be the optimal value and optimal solution set of problem (1.6), respectively. Again note that $X_{V} = S^{n,k}_{+}$. We define

$$\text{sgn}(S^{n,k}_{+}) := \bigcup_{X \in S^{n,k}_{+}} \text{sgn}(X), \quad \text{sgn}(\Theta_{f}) := \bigcup_{X \in \Theta_{f}} \text{sgn}(X).$$

It is not hard to see that the cardinality of $\text{sgn}(S^{n,k}_{+})$ is finite. It is now ready to introduce the following assumption.
Assumption 3.1. We assume that $\text{sgn}(S_{n,k}^+)/\text{sgn}(\Theta_f) \neq \emptyset$, namely, the constant
\begin{equation}
\chi_f := \hat{f}^* - f^* > 0,
\end{equation}
where
\begin{equation}
\hat{f}^* = \min_{X \in S_{n,k}^+} f(X) \quad \text{s.t.} \quad \text{sgn}(X) \in \text{sgn}(S_{n,k}^+)/\text{sgn}(\Theta_f).
\end{equation}

If Assumption 3.1 does not hold, then $\text{sgn}(S_{n,k}^+)/\text{sgn}(\Theta_f) = \emptyset$. This means that $f$ will be a constant over $S_{n,k}^+$.

3.1. A rounding procedure. For $X \in O\mathcal{B}_{n,k}^+$, computing $\Pi_{S_{n,k}^+}(X)$ is generally hard. However, when $X$ is near to $X_V$, namely, $\zeta(X) := \|XV\|_F - 1$ is relatively small, we can construct a matrix $X^R \in S_{n,k}^+$ near to $\Pi_{S_{n,k}^+}(X)$. The basic idea for rounding is that we simply keep one largest element in each row and set the remaining elements to be zeros, and then do normalization such that each column takes the unit norm. The complete way for generating $X^R$ is presented in Procedure 3.1.

**Procedure 3.1** A procedure for rounding $X \in O\mathcal{B}_{n,k}^+$ to be $X^R \in S_{n,k}^+$.

Initialization: Set $H \in \mathbb{R}^{n \times k}$ as a zero matrix.

For $i \in [n]$, set
\begin{equation}
H_{ij^*} = 1, \quad \text{if } j^* \text{ is the smallest index in the set } \arg\max_{j \in [k]} X_{ij}.
\end{equation}

Set the $j$-th column of $X^R$ as
\begin{equation}
x_j^R = \frac{x_j \circ h_j}{\|x_j \circ h_j\|}, \quad j \in [k].
\end{equation}

if $X^R \notin S_{n,k}^+$ then
Set $X^R = I_{n,k}$.

Define $\zeta_q(X) := \|XV\|_F^q - 1$. We now estimate the quality of $X^R$ below. It plays a key role in establishing our exact penalty results.

**Lemma 3.2.** For any $X \in O\mathcal{B}_{n,k}^+$, we have $X^R \in S_{n,k}^+$ and
\begin{equation}
\text{dist}(X, X_V) \leq \|X^R - X\|_F \leq \varrho_q \sqrt{\zeta_q(X)},
\end{equation}
where $\varrho_q = \sqrt{2k\tilde{\varrho}_q}/\omega$. Here, $\tilde{\varrho}_q$ is 1 if $q \geq 2$, and is $\sqrt{q+1}/q$ if $1 \leq q < 2$, and is $2\sqrt{(q+1)}/q$ if $0 < q \leq 1$.

**Proof.** We first focus on $q = 2$ and thus $\zeta_q(X) = \zeta(X)$. Recalling $\|V\|_F = 1$ and $\omega = \min_{j,i \in [k]} (VV^\top)_{ij}$, we have
\begin{equation}
\zeta(X) = \sum_{j \in [k]} x_j^T \left( \sum_{l \in [k]/\{j\}} (VV^\top)_{jl} x_l \right) \geq \omega \sum_{j \in [k]} x_j^T \left( \sum_{l \in [k]/\{j\}} x_l \right).
\end{equation}

To prove (3.4), we consider two cases.
Case I. \( \zeta(X) \geq \omega \). Then \( X^R \) generated by (3.3) is either in \( S^n_{+} \), or \( \text{reset to be } I_{n,k} \). In both cases, we always have \( ||X^R||_F^2 = 2k \). Noticing that \( X \in \mathcal{OB}^{n,k}_+ \), we have \( ||X - X^R||_F^2 \leq 2k \). Hence, there holds \( ||X - X^R||_F \leq \sqrt{2k} \leq \varnothing \sqrt{\zeta(X)} \).

Case II. \( \zeta(X) < \omega \). First, we prove that \( X^R \) generated by (3.3) lies in \( S^n_{+} \). Clearly, from (3.2), we know that each row of \( H \) has at most one element being 1. We now claim that each column of \( H \) has at least one element being 1. Otherwise, without loss of generality, we assume \( h_1 = 0 \). This with (3.2) implies that \( X_{i1} = \max \{ e_{i} \} X_{i1}, \forall i \in [n] \), which with (3.5) tells that \( \zeta(X) \geq \omega \sum_{i \in [n]} X_{i1} \max \{ e_{i} \} X_{i1} \geq \omega \sum_{i \in [n]} X_{i1}^2 = \omega \|x_1\|^2 = \omega \). This gives a contradiction with \( \zeta(X) < \omega \). In summary, we know that \( \|h_j\|_0 \geq 1, \forall j \in [k] \) and \( h_j^\top h_j = 0, \forall i, j \in [k] \) and \( i \neq j \) and

\[
(3.6) \quad x_j \circ h_j \neq 0, \quad (x_j \circ h_j)^\top (x_j \circ (e - h_j)) = 0, \quad \forall j \in [k].
\]

Therefore, with the construction (3.3) of \( X^R \), we must have \( X^R \in \mathcal{S}^{n,k}_+ \). Using (3.3), (3.6), and the decomposition \( x_j = x_j \circ h_j + x_j \circ (e - h_j) \), we have \( ||x_j - x_j^R||_F^2 \leq 2 ||x_j \circ (e - h_j)||_F^2 \).

It is ready to prove (3.4) for general \( q \). For \( X \in \mathcal{OB}^{n,k}_+ \), there holds that \( 1 \leq \|XV\|_F \leq \|X\|_2 \|V\|_F \leq \sqrt{k} \). We consider three cases. Case I. \( q \in [2, +\infty) \). It is easy to have \( \zeta_q(X) \geq \zeta(X) \). Case II. \( q \in [1, 2) \). We first have \( \zeta_1(X) = \frac{\zeta(X)}{\|XV\|_{F+1}} \geq \frac{\zeta(X)}{\sqrt{k+1}} \).

Then we have \( \zeta_q(X) = (1 + \zeta_1(X))^q - 1 \geq q\zeta(X) \geq \frac{q}{\sqrt{k+1}} \zeta(X) \), where the first inequality uses the fact that \( (1 + a)^q - 1 > qa \) for \( a \in (0, +\infty) \) and \( q \in [1, 2) \). Case III. \( q \in (0, 1) \). Since \( \|XV\|_{F+1} = 1 + \zeta_1(X) \geq 1 + \frac{\zeta(X)}{\sqrt{k}} \), we have

\[
\zeta_q(X) \geq \left( 1 + \frac{\zeta_1(X)}{\sqrt{k}} \right)^q - 1 \geq \frac{q(q+1)}{2\sqrt{k}} \zeta_1(X) \geq \frac{q(q+1)}{2\sqrt{k}(\sqrt{k} + 1)} \zeta(X),
\]

where the second inequality uses the fact that \( (1 + a)^q - 1 > \frac{qa(q+1)}{2} \) for \( a \in (0, 1) \), \( q \in (0, 1) \). Combining the above three cases, we have \( \zeta(X) \leq \varnothing \zeta_q(X) \), which with (3.4) being true for \( q = 2 \) implies that (3.4) holds for general \( q \). The proof is completed.

3.2. A general exact penalty model. Let \( L_f > 0 \) be the Lipschitz constant of \( f \), namely,

\[
(3.7) \quad |f(X_1) - f(X_2)| \leq L_f \|X_1 - X_2\|_F \quad \forall X_1, X_2 \in \mathcal{OB}^{n,k}_+.
\]

Suppose that there is a function \( Q : \mathcal{OB}^{n,k}_+ \rightarrow \mathbb{R}_+ \) satisfying

\[
(3.8) \quad \text{dist}(X, A_X) \leq \|X - X^R\|_F \leq Q(X) \quad \forall X \in \mathcal{OB}^{n,k}_+;
\]

\[
(3.9) \quad Q(X) \equiv Q_0 \quad \forall X \in A_X; \quad Q(X) \geq Q_0 \quad \forall X \in \mathcal{OB}^{n,k}_+,
\]

where \( Q_0 \geq 0 \) is a constant. Consider the partial penalty model

\[
(3.10) \quad \min_{X \in \mathcal{OB}^{n,k}_+} f(X) + \sigma \Psi(Q(X)),
\]

where \( \sigma > 0 \) is the penalty parameter and \( \Psi(\cdot) \) satisfies the following assumption.
Assumption 3.3. We assume that the function $\Psi : \mathbb{R}_+ \to \mathbb{R}_+$ is strictly increasing on $[Q_0, +\infty)$.

Let $X_{\sigma, \Psi}$ be a global minimizer of (3.10). We use $X_{\sigma, \Psi}^R$ to denote the matrix returned by Procedure 3.1 with input $X_{\sigma, \Psi}$.

Lemma 3.4. Suppose Assumption 3.3 holds and $\sigma > 0$. We have

$$f(X^*) \leq f(X_{\sigma, \Psi}^R) \leq f(X^*) + L_f \Upsilon_{\sigma, Q_0, \Psi},$$

where $X^*$ is one global minimizer of problem (1.6) and

$$\Upsilon_{\sigma, Q_0, \Psi} := \max_{z \in \mathbb{R}} z \text{ s.t. } \Psi(z) \leq \Psi(Q_0) + \frac{L_f}{\sigma} z, 0 \leq \Psi^{-1}(\Psi(Q_0) + \sqrt{2kL_f/\sigma}).$$

Proof. Using the Lipschitz continuity of $f$, we have

$$f(X_{\sigma, \Psi}^R) \leq f(X_{\sigma, \Psi}) + L_f \|X_{\sigma, \Psi}^R - X_{\sigma, \Psi}\|_F \leq f(X_{\sigma, \Psi}) + L_f Q(X_{\sigma, \Psi}),$$

where the second inequality due to (3.8). By the optimality of $X_{\sigma, \Psi}$, we have

$$f(X_{\sigma, \Psi}) + \sigma \Psi(Q(X_{\sigma, \Psi})) \leq f(X) + \sigma \Psi(Q(X)) = f(X) + \sigma \Psi(Q_0) \quad \forall X \in \mathcal{X}_V.$$

Taking $X = X^*$ in (3.14), with Assumption 3.3, we have $f(X_{\sigma, \Psi}^R) \leq f(X^*)$. Hence, we know from (3.13) that

$$f(X^*) \leq f(X_{\sigma, \Psi}^R) \leq f(X^*) + L_f Q(X_{\sigma, \Psi}).$$

The remaining is to estimate $Q(X_{\sigma, \Psi})$. Taking $X$ to be $X_{\sigma, \Psi}^R$ in (3.14), we have

$$\Psi(Q(X_{\sigma, \Psi})) \leq \Psi(Q_0) + \frac{f(X_{\sigma, \Psi}^R) - f(X_{\sigma, \Psi})}{\sigma} \leq \Psi(Q_0) + \frac{L_f \|X_{\sigma, \Psi}^R - X_{\sigma, \Psi}\|_F}{\sigma},$$

where the second inequality is due to (3.7). Since $X_{\sigma, \Psi} \in \mathcal{O}_{\mathcal{B}_+^{n,k}}$, it is easy to see that $\|X_{\sigma, \Psi}^R - X_{\sigma, \Psi}\|_F \leq \sqrt{2kL_f}$. Thus we have from (3.16) that $\Psi(Q(X_{\sigma, \Psi})) \leq \Psi(Q_0) + \sqrt{2kL_f}/\sigma$, which with Assumption 3.3 implies that

$$Q(X_{\sigma, \Psi}) \leq \Psi^{-1}\left(\Psi(Q_0) + \sqrt{2kL_f}/\sigma\right).$$

On the other hand, recalling (3.8), we have from (3.16) that $\Psi(Q(X_{\sigma, \Psi})) \leq \Psi(Q_0) + \frac{L_f}{\sigma} Q(X_{\sigma, \Psi})$, which together with (3.17) and (3.15) establishes (3.11). The proof is completed.

We now give the exact penalty results. Define

$$X_{\sigma, \Psi}^\hat{\circ} \in \arg\min_{X \in \mathcal{O}_{\mathcal{B}_+^{n,k}}} f(X) \text{ s.t. } X_{ij} = 0 \text{ if } (i, j) \notin \text{supp}(X_{\sigma, \Psi}^R).$$

Note that for some special $f$, $X_{\sigma, \Psi}^\hat{\circ}$ is very easy to compute.

Theorem 3.5. Suppose Assumption 3.1 and Assumption 3.3 hold and the parameters $Q_0 \geq 0$ and $\sigma > 0$ are chosen such that

$$Q_0 \leq \Upsilon_{\sigma, Q_0, \Psi} < \kappa_f := \chi_f/L_f,$$

Then $X_{\sigma, \Psi}^\hat{\circ}$ is the minimizer of problem (1.6), namely, $f(X_{\sigma, \Psi}^\hat{\circ}) = f(X^*)$.
Proof. Clearly, $Q_0 \leq \Upsilon_{\sigma,Q_0,\Psi}$ follows directly from (3.12). We first claim that $\text{supp}(X_{\sigma,\Psi}^R) \in \text{supp}(\Theta_f)$. Otherwise, it follows from Assumption 3.1 that $f(X_{\sigma,\Psi}^R) \geq f(X^*) + \chi_f$. By using (3.11), we thus have $\Upsilon_{\sigma,Q_0,\Psi} \geq \kappa_f$, which makes a contradiction with (3.19). Using $\text{supp}(X_{\sigma,\Psi}^R) \in \text{supp}(\Theta_f)$ and the definition (3.18), we know that $X_{\sigma,\Psi}^\diamond$ is the global minimizer of problem (1.6). The proof is completed.

It follows from (3.12) that $\Upsilon_{\sigma,Q_0,\Psi} \leq \Psi^{-1}(\Psi(Q_0) + \sqrt{2k}L_f/\sigma)$. To make (3.19) hold, we can choose

$$0 \leq Q_0 < \kappa_f, \quad \sigma > \sigma := \sqrt{2k}(\Psi(\kappa_f) - \Psi(Q_0))^{-1}L_f.$$

For particular $\Psi(\cdot)$, we next show that this lower bound $\sigma$ can be improved.

Consider the following choices of $\sigma, Q(X)$ and $\Psi(z)$:

$$\Psi(z) = (z/\varrho_q)^{2p}, \quad p \in (0, +\infty); \quad Q(X) = \varrho_q\sqrt{\zeta_q}(X) + \epsilon, \quad Q_0 = \varrho_q\sqrt{\epsilon}$$

with $0 \leq \epsilon < \kappa_f^2/\varrho_q^2$. The penalty model (3.10) becomes

$$(3.20) \quad \min_{X \in \mathcal{B}_{n,k}} \left\{ P_{\sigma,p,q,\epsilon}(X) := f(X) + \sigma (\|XV\|_F^2 - 1 + \epsilon)^p \right\}.$$

Let $X_{\sigma,p,q,\epsilon}$ be a global minimizer of (3.20) and denote $X_{\sigma,p,q,\epsilon}^R$ as the matrix returned by Procedure 3.1 with input $X_{\sigma,p,q,\epsilon}$. Let $X_{\sigma,p,q,\epsilon}^\diamond$ be a global minimizer of the problem in (3.18) with $X_{\sigma,\Psi}^R$ being $X_{\sigma,p,q,\epsilon}^R$. According to Theorem 3.5, by some tedious calculations, we have the following result. To save space, we omit the proof.

**Lemma 3.6.** If we choose

$$\sigma > \sigma = \begin{cases} \sqrt{2k} \varrho_q \kappa_f & \text{if } 0 < p \leq 1/2 \text{ and } \epsilon = 0, \\ \min \left\{ \sqrt{2k} \kappa_f, (\varrho_q/\kappa_f)^{2p} \right\} & \text{if } p > 1/2 \text{ and } \epsilon = 0, \\ \sqrt{2k}((\kappa_f/\varrho_q)^{2p} - \epsilon^{2p})^{-1}L_f & \text{if } p > 0 \text{ and } 0 < \epsilon < \kappa_f^2, \end{cases}$$

then $X_{\sigma,p,q,\epsilon}^\diamond$ is a global minimizer of problem (1.6).

A few remarks on the exact penalty model (3.20) are listed in order. First, to make the objective function in problem (3.20) smooth, we need to choose $\epsilon \in (0, \kappa_f^2)$ for $p \in (0, 1)$. As for $p \in [1, +\infty)$, we can simply choose $\epsilon = 0$. Second, by directly using the results in [16, Lemma 3.1], we can show that the global minimizer of (3.20) with $p = 1/2$ and $\epsilon = 0$ is also a global minimizer of (1.1) under the condition that $\sigma > \varrho_q L_f$. However, the results therein does not apply to the general $\Psi(\cdot)$ and $Q_0$. By contrast, our results in Theorem 3.5 or Lemma 3.6 allow more flexible choices of $\Psi(\cdot)$ and $Q_0$. Third, the multiple spherical constraints $\|x_j\| = 1, j \in [k]$ in model (3.20) are not only important to establish the exact penalty property but also make model (3.20) as optimization over a compact set. It should be mentioned that for the ONMF formulation (1.3), [36] proposed an exact penalty model without keeping the multiple spherical constraints. However, their results only work for this special formulation (1.3) other than the general problem (1.1). Besides, Gao et al. [18] considered to use a customized augmented Lagrangian type method to solve optimization with orthogonality constraints. The multiple spherical constraints are also kept therein to make their algorithms more robust.
3.3. A practical exact penalty algorithm. Based on the above exact penalty results, theoretically, we only need to solve a series of subproblems (3.20) with dynamically increasing $\sigma$. However, solving (3.20) for fixed $\sigma$ is still hard. Considering that we mainly aim to find a orthogonal nonnegative matrix of high quality, we propose the following practical algorithm, outlined in Algorithm 3.2. In each iteration, we find an approximate stationary point of (3.20); see Definition 4.2 for its definition. To do that, we can use the nonconvex gradient projection method (see for instance in [5]) or the second-order method developed in section 4.2. We adopt the way in [16] to choose a feasible initial point for solving (3.20). Similar to the analysis therein, we can show that any limit point of the sequence $\{X^t\}$ (setting $t_{\text{max}} = \infty$) is an orthogonal nonnegative matrix. However, whether the limit point is a KKT point of (1.6) is still not clear. One possible reason is that generally only the weakest CQ, such as GCQ or ACQ holds for (1.6) while a stronger CQ condition is needed or assumed in [16, 36].

Algorithm 3.2 A practical exact penalty method for solving problem (1.6)

Initialization: Choose $X^0 \in \mathcal{OB}_+^{n,k}$, $X_{\text{feas}} \in \mathcal{S}_+^{n,k}$, $\sigma_0 > 0$, $p, q \in (0, +\infty)$, $\gamma_1 \in (0, 1)$ if $p \in (0, 1)$, $\gamma_1 = 0$ if $p \geq 1$, $\gamma_2 > 1$, $\eta, \text{tol}^{\text{feas}}, \text{tol}^{\text{sub}}_{\text{min}} \in (0, 1)$, a positive integer $t_{\text{max}}$. Set $X^{0,0} = X^0$.

For $t = 0, 1, 2, \ldots, t_{\text{max}}$ do

If $P_{\sigma, p, q, \epsilon}(X^{t,0}) > P_{\sigma, p, q, \epsilon}(X_{\text{feas}})$, set $X^{t,0} = X_{\text{feas}}$.

Starting from $X^{t,0}$, we find an approximate stationary point $X^t$ of (3.20) with $\sigma = \sigma_t$ and $\epsilon = \epsilon_t$ such that

$$\|X^t - \Pi_+ (X^t - \text{grad} P_{\sigma_t, p, q, \epsilon_t}(X^t))\|_F \leq \text{tol}_t^{\text{sub}}$$

and

$$P_{\sigma_t, p, q, \epsilon_t}(X^t) \leq P_{\sigma_t, p, q, \epsilon_t}(X^{t,0}).$$

If $\|X^t V\|_F^2 - 1 \leq \text{tol}^{\text{feas}}$, then

- Rounding: Set $X^R = (X^t)^R$.
- Postprocessing: Solve (3.18) with $X_{\sigma, \Phi}^R$ to get $X^\Phi$.
- break

Set $\epsilon_{t+1} = \gamma_1 \epsilon_t$, $\sigma_{t+1} = \gamma_2 \sigma_t$, $\text{tol}^{\text{sub}}_{t+1} = \max\{\eta \text{tol}^{\text{sub}}_t, \text{tol}^{\text{sub}}_{\text{min}} \}$, and $X^{t+1,0} = X^t$.

Moreover, our exact penalty approach also works for the general problem (1.7). The subproblem (3.20) becomes

$$\min_{X \in \mathcal{OB}_+^{n,k}, Y \in \mathcal{Y}} \left\{ \tilde{P}_{\sigma, p, q, \epsilon}(X, Y) := f(X, Y) + \sigma (\|XV\|_F^2 - 1 + \epsilon)^p \right\}.$$

The practical penalty algorithm is almost the same to Algorithm 3.2 except that (3.21) is replaced by

$$\|X^t - \Pi_+ (X^t - \text{grad} X \tilde{P}_{\sigma_t, p, q, \epsilon_t}(X^t, Y^t))\|_F \leq \text{tol}_t^{\text{sub}},$$

and (3.25) becomes $\tilde{P}_{\sigma_t, p, q, \epsilon_t}(X^t, Y^t) \leq \tilde{P}_{\sigma_t, p, q, \epsilon_t}(X^{t,0}, Y^{t,0})$. To obtain an approximate stationary point satisfying (3.24) and (3.25), we employ the proximal alternating
linearized minimization (PALM) method in [9]. The main iterations are given as
\[ Y_{l+1} = \Pi_Y \left( Y_l - \beta \nabla_X \tilde{P}_{\sigma_t, p, q, \varepsilon_t} (X_l, Y_l) \right), \]
\[ (3.26) \]
\[ X_{l+1} = \Pi_{OB_{n, k}^+} \left( X_l - \alpha \nabla_X \tilde{P}_{\sigma_t, p, q, \varepsilon_t} (X_l, Y_l) \right), \]
\[ (3.27) \]
Note that [36] also used PALM to solve their relaxation model for ONMF. Besides, one can also use the proximal alternating minimization scheme [4], wherein the \( X \)-subproblem can be approximately solved by the second-order method Algorithm 4.1.

4. Optimization over \( OB_{n, k}^+ \). In this section, we first investigate the optimality conditions for the following optimization problem over \( OB_{n, k}^+ \):
\[ (4.1) \min_{X \in OB_{n, k}^+} h(X), \]
where \( h : \mathbb{R}^{n \times k} \rightarrow \mathbb{R} \) is twice continuously differentiable. Then we introduce a second-order method for solving problem (4.1).

4.1. Optimality conditions of problem (4.1). Let us define the Lagrangian function for problem (4.1) as
\[ \hat{L}(X, \Lambda, Z) = h(X) - \sum_{j \in [k]} \Lambda_j (\|x_j\|^2 - 1) - \langle Z, X \rangle, \]
where \( \Lambda \in \mathbb{R}^k \) is the Lagrange multiplier corresponding to \( \|x_j\|^2 = 1, j \in [k] \) and \( Z \in \mathbb{R}^{n \times k}_+ \) is the Lagrange multiplier corresponding to \( X \geq 0 \).

**Lemma 4.1 (First-order necessary conditions of (4.1)).** Let \( \bar{X} \in OB_{n, k}^+ \) be a local minimizer of problem (4.1), then \( \bar{X} \) is a stationary point, namely, there holds that
\[ 0 \leq \bar{X} - \Pi_+ (\bar{X} - \text{grad } h(\bar{X})) \]
\[ (4.3) \]
which is further equivalent to
\[ \bar{X} - \Pi_+ (\bar{X} - \text{grad } h(\bar{X})) = 0. \]
\[ (4.4) \]
**Proof.** Similar to Case I in the proof of Lemma 2.2, it is easy to prove the LICQ holds at \( \bar{X} \). The remaining proof is similar to that of Theorem 2.3, we omit the details. \( \square \)

Based on the optimality condition (4.4), we define the \( \varepsilon \)-stationary point of problem (4.1) as follows.

**Definition 4.2 (\( \varepsilon \)-stationary point of (4.1)).** Let \( X \in OB_{n, k}^+ \), we call \( X \) an \( \varepsilon \)-stationary point of problem (4.1) if
\[ \| \bar{X} - \Pi_+ (\bar{X} - \text{grad } h(\bar{X})) \|_F \leq \varepsilon. \]
\[ (4.5) \]
Since the LICQ holds at \( \bar{X} \), we have \( \text{LNCD}_{OB_{n, k}^+} (\bar{X}) = \text{SNCD}_{OB_{n, k}^+} (\bar{X}) \) and
\[ \text{LNCD}_{OB_{n, k}^+} (\bar{X}) = \left\{ D \in \text{LFD}_{OB_{n, k}^+} (\bar{X}) : D_{ij} = 0 \text{ if } |\nabla f(\bar{X})|_{ij} > 0, (i, j) \notin \text{supp}(\bar{X}) \right\}, \]
where \( \text{LFD}_{OB_{n, k}^+} (\bar{X}) = \{ D \in \mathbb{R}^n : x_j^T d_j = 0, j \in [k], D_{ij} \geq 0 \text{ if } x_{ij} = 0 \} \). Recalling (4.2) and with (4.3), we have \( \nabla^2 \hat{L}(\bar{X}, \bar{\Lambda}, \bar{Z})[D] = \text{Hess } h(\bar{X})[D], \) where the Hess operator is defined in (2.15). Similar to the derivation of Theorems 2.5 and 2.6, we have the following second-order optimality conditions for problem (4.1).
THEOREM 4.3 (Second-order necessary condition of problem (4.1)). If $\bar{X}$ is a local minimizer of problem (4.1) then

$$\langle D, \text{Hess } h(\bar{X})[D] \rangle \geq 0, \quad \text{for all } D \in \text{LNCD}_{\Omega B_{\gamma}^{n,k}}(\bar{X}),$$

where $\text{Hess } h(\bar{X})[D]$ is obtained by specializing (2.15) to $h(\bar{X})$.

THEOREM 4.4 (Second-order sufficient conditions of problem (1.6)). Suppose that $\bar{X}$ is a stationary point of problem (4.1), namely, satisfying (4.3). Suppose also that

$$\langle D, \text{Hess } h(\bar{X})[D] \rangle > 0, \quad \text{for all } D \in \text{LNCD}_{\Omega B_{\gamma}^{n,k}}(\bar{X})/\{0\}.$$

Then $\bar{X}$ is a strict local minimizer of (1.6).

4.2. A second-order method. We consider the adaptive quadratically regularized Newton method [22] for solving (4.1). At the $l$-th iteration, we perform a single Newton step to inexactly solve the quadratic regularized subproblem

$$\min_{X \in \Omega B_{\gamma}^{n,k}} m_l(X),$$

where $m_l(X) := \langle \nabla f(X^l), X - X^l \rangle + \frac{1}{2} \langle X - X^l, \nabla^2 f(X^l)[X - X^l] \rangle + \frac{\tau}{2} \|X - X^l\|_2^2.$

To be specific, we solve the following subproblem to obtain a Newton step:

$$\begin{aligned}
\min_{D \in \mathbb{R}^{n \times k}} & \quad \langle \text{grad } m_l(X^l), D \rangle + \frac{1}{2} \langle D, \text{Hess } m_l(X^l)[D] \rangle \\
\text{s.t.} & \quad (x^l)^{\top} d_j = 0, \quad x^l_j + d_j \geq 0, \quad j \in [k].
\end{aligned}$$

(4.6)

By some easy calculations, we have $\text{grad } m_l(X^l) = \text{grad } h(X^l)$ and $\text{Hess } m_l(X^l)[D] = \text{Hess } h(X^l)[D] + \tau D$. Setting $D = Z - X^l$, we get a reformulation of problem (4.6) as

$$\begin{aligned}
\min_{Z \in \Delta(X^l)} & \quad \langle \text{grad } m_l(X^l), Z - X^l \rangle + \frac{1}{2} \langle Z - X^l, \text{Hess } m_l(X^l)[Z - X^l] \rangle, \\
\text{s.t.} & \quad \Delta(X^l) = \{Z \in \mathbb{R}^{n \times k} : (x^l)^{\top} z_j = 1, \quad z_j \geq 0, \quad j \in [k]\}. \\
\end{aligned}$$

(4.7)

Instead of solving (4.7) directly, we consider its first-order optimality condition, which can be formulated as the following nonsmooth equation:

$$\mathcal{F}(Z) := Z - \Pi_{\Delta(X^l)} \left( Z - \alpha (\text{grad } m_l(X^l) + \text{Hess } m_l(X^l)[Z - X^l]) \right) = 0.$$

(4.8)

We employ the adaptive semi-smooth Newton (ASSN) method proposed in [38] to solve (4.8). Thanks to [27], we can efficiently compute the the HS generalized Jacobian $\mathcal{P}_C(\cdot)$ of $\Pi_{\Delta(X^l)}(\cdot)$ efficiently. Denote $C := Z - \alpha (\text{grad } f(X^l) + \text{Hess } m_l(X^l)[Z - X^l])$ for simplicity. Let $\tilde{C} := \Pi_{\Delta(X^l)}(C)$. Define the linear operator $\Xi[D] : \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{n \times k}$ by $(\Xi(D))_{ij} = 0$ if $\tilde{C}_{ij} = 0$ and $(\Xi(D))_{ij} = D_{ij}$ otherwise. We simply denote $\Xi[d_j] = (\Xi(D))_{j,\cdot}$, $\forall j \in [k]$. Following Proposition 3 in [27], by some calculations, we have the HS-Jacobian of $\Pi_{\Delta(X^l)}(\cdot)$ at $C$ as $\mathcal{P}_C(D) = \Xi(D) - \Xi(X)M$, where $M$ is diagonal with $M_{jj} = x_j^{\top} \Xi[d_j]/x_j^{\top} \Xi[x_j], \quad \forall j \in [k]$. Hence, we have the HS-Jacobian of $\mathcal{F}$ as $\partial \mathcal{F}(D) = D - \mathcal{P}_C(D)$.

The complete second-order algorithm is presented in Algorithm 4.1. For sake of saving space, some details are omitted here. We refer the reader to [22, 38] for further information. The proposed algorithm can generate high-quality searching directions,
but its computational cost may be relatively expensive. In practice, we combine the second-order method with the projection gradient method whose main iteration is

\[
X^{l+1} = \Pi_{OB_{+}^{n,k}}(X^l - \alpha^l \nabla h(X^l)), \quad \alpha^l > 0,
\]

to achieve higher efficiency. Specifically, we utilize the projection gradient method if \(\zeta(X^{t,0}) = \|X^{t,0}V\|_F^2 - 1 > \zeta > 0\) and switch to the second-order method otherwise.

**Algorithm 4.1** An adaptive quadratically regularized Newton method for (4.1)

Initialization: Choose \(X^0 \in OB_{+}^{n,k}\), a tolerance \(\epsilon > 0\) and an initial regularization parameter \(\tau^0 > 0\). Choose \(0 < \eta_1 \leq \eta_2 < 1\), \(0 < \beta_0 < 1 < \beta_1 < \beta_2\). Set \(l := 0\).

while \(\|\bar{X} - \Pi_{+}(X - \text{grad } h(X))\|_F > \epsilon\) do

1. Solve problem (4.8) inexactly via the ASSN (Algorithm 1 in [38]) to obtain \(Z^l\).
2. Set \(\bar{Z}^l = \Pi_{OB_{+}^{n,k}}(Z^l)\) and calculate \(\rho_l = \left(h(\bar{Z}^l) - h(X^l))/m_l(\bar{Z}^l)\right)\).
3. Set \(X^{l+1} = \bar{Z}^l\) if \(\rho_l \geq \eta_1\) and set \(X^{l+1} := X^l\) otherwise.
4. Update \(\tau_{l+1} = \begin{cases} (0, \beta_0 \tau_l), & \text{if } \rho_l \geq \eta_2, \\ [\beta_0 \tau_l, \beta_1 \tau_l], & \text{if } \eta_1 \leq \rho_l \leq \eta_2, \\ [\beta_1 \tau_l, \beta_2 \tau_l], & \text{otherwise.} \end{cases}\)

Set \(l = l + 1\).

5. Numerical experiments. In this section, we present a variety of numerical results to evaluate the performance of our proposed method. All experiments are performed in Windows 10 on an Intel Core 4 Quad CPU at 2.30 GHZ with 8 GB of RAM. All codes are written in MATLAB R2018b. The matrix \(V\) is simply taken as \(V = e/\sqrt{k}\), and the choice of parameters for Algorithm 3.2 are set as follows: \(p = 1\), \(q = 2\), \(\epsilon = 0\), \(\gamma_1 = 0\), \(\text{tol}^{\text{feas}}_{\text{min}} = 10^{-7}\), \(t_{\text{max}} = 300\); \(\gamma_2\) is adjusted dynamically; the choices of \(\sigma_0\), \(\eta\), \(\text{tol}^{\text{feas}}\) and \(X^0\) are given in each subsection. In our implementation, instead of using (3.21), we use the stopping condition when the distance between two consecutive iterations is small, namely, \(\|X^{l+1} - X^l\|_F \leq \text{tol}^{\text{feas}}\).

5.1. Computing projection onto \(S_{+}^{n,k}\). Given a matrix \(C \in \mathbb{R}^{n \times k}\), we consider to compute its projection onto \(S_{+}^{n,k}\), which is formulated as

\[
\min_{X \in S_{+}^{n,k}} \|X - C\|_F^2.
\]

The exact penalty model (3.20) with \(p = 1\), \(q = 2\), and \(\epsilon = 0\) is equivalent to

\[
\min_{X \in S_{+}^{n,k}} \left\{ P_\sigma(X) := -\frac{1}{\sigma} \langle C, X \rangle + \frac{1}{2}\|XV\|_F^2 \right\}.
\]

Note that \(VV^T \succeq I_k\) since \(\|V\|_F = 1\). We have the Lipschitz constants of \(\nabla P_\sigma(X)\) being 1. Thus we know from Theorem 5.3 in [3] that the sequence \(\{X^l\}\) generated by the nonconvex gradient projection scheme

\[
X^{l+1} = \Pi_{OB_{+}^{n,k}}(X^l - \alpha^l \left( X^lVV^T - C/\sigma \right)), \quad \alpha^l \equiv \alpha \in (0,1)
\]

converges to a stationary point of (5.2). In our tests, we simply choose \(\alpha^l \equiv 0.99\). Note that we do not invoke the second-order method Algorithm 4.1 to solve (5.2).
It is always difficult to seek the projection globally for a general matrix \( C \). Thanks to Proposition B.1, we can construct a family of matrices with unique and known projection. For a given \( B \in S^{n,k}_{+} \), the MATLAB codes for generating \( C \) is given as

\[
X = (B>0).*(1+\text{rand}(n, k)); \\
xstar = X./\text{sqrt}(\text{sum}(X.*X)); \\
d = 0.5+3*\text{rand}(k, 1); \\
L = \xi*((d*d').^0.5).*\text{rand}(k, k); \\
L(\text{sub2ind}([k, k], 1:k, 1:k))=d; \\
C=xstar*L;
\]

The parameter \( \xi \in [0, 1] \) controls the magnitude of noise level. Larger \( \xi \) makes it more difficult to find the ground truth \( X^* = \Pi_{S^{n,k}_{+}}(C) \). Let \( X^\diamond \) be the solution generated by Algorithm 3.2. Note that the postprocessing problem (3.18) has closed form solution. We define gap = \( \|X^\diamond - C\|_F/\|X^* - C\|_F - 1 \) as a measure of the solution quality.

for each fixed \( \xi, n \) and \( k \), we run 50 times of our algorithms, and the initial point is generated by rounding \( C \) through Procedure 3.1. We choose \( \text{tol}^{\text{feas}} = 10^{-8}, \sigma_0 = 10^{-2}, \eta = 0.8 \). The averaged results are reported in Table 2, wherein the “suc” means the total number of instances for which the gap is zero. From this table, we can see that for small \( \xi \) our Algorithm 3.2 can solve all 50 instances to a zero gap, while for large \( \xi \) we can only solve some instances to a zero gap. However, for all cases, Algorithm 3.2 can always return an orthogonal nonnegative matrix with satisfactory quality.

| \( \xi \) | \( n = 2000, k = 10 \) | \( n = 2000, k = 50 \) | \( n = 2000, k = 100 \) |
|---|---|---|---|
| 0.50 | 50 | 0.0e0 | 0.01 | 20.5 | 50 | 0.0e0 | 0.04 | 38.3 | 50 | 0.0e0 | 0.32 | 53.9 |
| 0.70 | 50 | 0.0e0 | 0.01 | 22.9 | 50 | 0.0e0 | 0.05 | 50.9 | 50 | 0.0e0 | 0.43 | 76.5 |
| 0.80 | 50 | 0.0e0 | 0.01 | 24.5 | 50 | 0.0e0 | 0.06 | 62.7 | 50 | 0.0e0 | 0.52 | 95.7 |
| 0.90 | 50 | 0.0e0 | 0.01 | 28.7 | 50 | 0.0e0 | 0.07 | 82.1 | 50 | 0.0e0 | 0.66 | 134.6 |
| 0.95 | 49 | 7.2e-5 | 0.01 | 31.9 | 46 | 2.1e-4 | 0.09 | 112.2 | 49 | 6.6e-7 | 0.87 | 184.8 |
| 0.98 | 43 | 8.9e-4 | 0.01 | 33.8 | 22 | 5.0e-4 | 0.11 | 156.3 | 19 | 8.0e-4 | 1.23 | 268.2 |
| 1.00 | 37 | 1.2e-3 | 0.01 | 38.1 | 0 | 2.6e-3 | 0.12 | 170.3 | 0 | 2.6e-3 | 1.43 | 317.5 |

| \( \xi \) | \( n = 2000, k = 200 \) | \( n = 2000, k = 300 \) | \( n = 2000, k = 400 \) |
|---|---|---|---|
| 0.50 | 50 | 0.0e0 | 0.77 | 73.7 | 50 | 0.0e0 | 1.34 | 89.9 | 50 | 0.0e0 | 1.96 | 99.8 |
| 0.70 | 50 | 0.0e0 | 1.13 | 113.5 | 50 | 0.0e0 | 2.01 | 137.9 | 50 | 0.0e0 | 2.99 | 157.8 |
| 0.80 | 50 | 0.0e0 | 1.38 | 144.7 | 50 | 0.0e0 | 2.62 | 186.7 | 50 | 0.0e0 | 3.79 | 211.7 |
| 0.90 | 50 | 0.0e0 | 1.6 | 207.6 | 50 | 0.0e0 | 3.43 | 276.0 | 50 | 0.0e0 | 5.39 | 328.7 |
| 0.95 | 50 | 0.0e0 | 2.42 | 295.1 | 50 | 0.0e0 | 5.13 | 424.9 | 50 | 0.0e0 | 7.74 | 483.0 |
| 0.98 | 23 | 4.5e-4 | 3.93 | 489.2 | 20 | 2.5e-4 | 8.60 | 718.6 | 24 | 1.7e-4 | 15.42 | 962.2 |
| 1.00 | 0 | 1.9e-3 | 5.07 | 636.9 | 0 | 1.8e-3 | 11.31 | 951.3 | 0 | 1.6e-3 | 20.86 | 1324.0 |

5.2. Orthogonal nonnegative matrix factorization. We compare our proposed method with uni-orthogonal NMF (U-onnf) [17], orthonormal projective nonnegative matrix factorization (OPNMF) [42], orthogonal nonnegatively penalized matrix factorization (ONP-MF) [33] and EM-like algorithm for ONMF (EM-onmf) [33]. In addition to the above methods, we also compare our method with K-means, which is considered as a benchmark in clustering problems. We implement U-onmf by ourselves since the original code is not available. We adopt the implementation of OPNMF from https://github.com/asotiras/brainparts. The codes of ONP-MF and EM-onmf can be downloaded from https://github.com/filippo-p/onmf. As to K-means, we call the MATLAB function kmeans directly. Note that our proposed method and OPNMF solve the equivalent formulation (1.4) while the remaining methods solve...
directly (1.3). Considering that the objective function in (1.4) is quartic, to make the subproblem (3.20) easier to solve, one can consider the Gauss-Newton technique as

$$\|A - XX^T A\|_F^2 = \|A - XX^T A - \tilde{X} S^T A - SS^T A\|_F^2 \approx \|A - XX^T A - \tilde{X} S^T A\|_F^2,$$

where $S = X - \tilde{X}$. By neglecting the term $\tilde{X} S^T A$, we obtain a partial Gauss-Newton approximation, namely, $\|A - XX^T A\|_F^2 \approx \|A - XX^T A\|_F^2$. Moreover, if $X \in S_+^{n,k}$, we know that $\|A - XX^T A\|_F^2 = \|A - X(XX^T X)^{-1}X^T A\|_F^2$. Hence, to make the approximation robust, we consider $\|A - XX^T A\|_F^2 \approx \|A - X(XX^T X)^{-1}X^T A\|_F^2$.

The subproblem (3.20) at $t$-th iteration with $p = 1$, $q = 2$, and $\epsilon = 0$ becomes

$$\min_{X \in \mathbb{B}_+^{n,k}} \left\{ \tilde{F}_{\sigma, p, q, \epsilon}(X) := \|A - X(Y^t)^T\|_F^2 + \sigma \|XV\|_F^2 \right\}$$

with $Y^{t-1} = \Pi_+(A^T \tilde{X}^t((\tilde{X}^t)^T \tilde{X}^t)^{-1})$.

In some datasets, the matrix $A$ maybe degenerated, namely, there exists a row (column) of $A$ with all zero entries. This causes a division by zero error when running the ON-unmf method. Thus we will first remove such degenerate rows and columns of $A$. For K-means and EN-onmf, the initial points are chosen randomly. The other methods adopt the SVD-based initializations [12]. In practice, the time cost of generating initial points is relatively low compared to that of the rest parts. We set $\sigma_0 = 10^{-3}$, $\eta_1 = 0.98$, and choose $\text{tof}\text{eas} = 0.3$, $\zeta = 0.6$ for hyperspectral datasets and $\text{tof}\text{eas} = 10^{-8}$, $\zeta = 5$ for other datasets. The main parameters of Algorithm 4.1 are chosen as $\eta_1 = 0.01$, $\eta_2 = 0.9$, $\beta_0 = 0.98$, $\beta_1 = 1$, and $\beta_2 = 1.3$. We adopt the Barzilai-Borwein stepsize [6] and use the nonmonotone line search [45] in the gradient projection iteration (4.9). Define $S_t^{-1} = X_t - X_t^{-1}$ and $Z_t^{-1} = \nabla h(X_t^t) - \nabla h(X_t^{t-1})$. We compute $\alpha_t = \max\{10^{-10}, \min(\alpha_{t-1}^0, 10^{10})\}$ with $\alpha_{t-1}^0 = \frac{(s^{t-1}, s^{t-1})}{(s^{-1}, s^{-1})}$. Since we aim to show in sections 5.2.1 and 5.2.2 that our algorithm can generate a solution with high quality and small feasibility violation, we remove the rounding procedure and postprocessing in Algorithm 3.2 to give a fair comparison in sections 5.2.1 and 5.2.2.

### 5.2.2. Synthetic data

Our main aim in this part is to compare the performance of solving the ONMF problem itself, so EN-onmf and K-means will be excluded in the comparison since they can only provide the results of clustering other than a meaningful orthogonal nonnegative matrix factorization.

Given a random generated matrix $B \in S_+^{n,k}$, a positive integer $r$ and a real number $\xi$ which controls the magnitude of noise, we construct the matrix $A$ by the following MATLAB codes:

```matlab
C = rand(k,r); D = rand(n,r); A = B*C;
A = A/norm(A,’fro’); A = A + xi/norm(D,’fro’)*D;
```

Let $\tilde{X}$ be the solution generated by algorithms, we calculate the feasibility violation as $\text{feas} := \|\tilde{X}^T \tilde{X} - I_k\|_F + \min(\tilde{X}, 0)\|_F$. Performing a rounding procedure on $\tilde{X}$ to obtain a feasible $X^R$, we take $\text{resi} := \|A - X^R(X^R)^T A\|_F$ to measure the quality of the solution. The results are presented in Table 3, where $n = 1000$, $r = 3000$, $k = 10$. From this table, we can see that the orthogonality and nonnegativity of the solutions given by our method are well kept, while the solutions generated by U-onmf, ONP-MF and OPNMF have relatively large violation. Besides, the solution quality of our proposed method is also better than that of other methods. In summary, our proposed method outperforms the other methods for the synthetic datasets.

### 5.2.2. Text and image clustering

We evaluate algorithms on text and image datasets adopted from [13], they are available at [http://www.cad.zju.edu.cn/home/](http://www.cad.zju.edu.cn/home/)
Table 3: ONMF results on synthetic data with different noise magnitude, “time” is in seconds.

| method       | \( \xi = 0 \) | \( \xi = 0.01 \) | \( \xi = 0.1 \) |
|--------------|----------------|----------------|----------------|
|              | feasi resi    | feasi resi    | feasi resi    |
| our method   | 4.3e-16       | 3.8e-16       | 5.4e-3        |
| OPNMF        | 2.6e-2        | 1.6e-15       | 2.1e-3        |
| U-onmf       | 5.7e-2        | 4.9e-16       | 5.6e-3        |
| ONP-MF       | 3.1e-3        | 5.0e-1        | 5.0e-1       |

\( \xi = 1 \), \( \xi = 10 \), \( \xi = 100 \)

| method       | \( \xi = 1 \) | \( \xi = 10 \) | \( \xi = 100 \) |
|--------------|----------------|----------------|----------------|
|              | feasi resi    | feasi resi    | feasi resi    |
| our method   | 1.2e-15       | 5.1e-1        | 5.0e-1        |
| OPNMF        | 3.9e-1        | 5.9e-1        | 7.4e-1        |
| U-onmf       | 3.6e-1        | 5.8e-1        | 1.2e0         |
| ONP-MF       | 3.2e-3        | 7.1e-1        | 3.2e-3        |

dengcai/Data/data.html. Since the original text dataset is too huge and disproportionate, we extract some subsets from original data to make it suitable for testing clustering algorithms. The details of modification are provided as follows.

- Reuters-t10(-t20): For the 10 (20) classes with the largest number of texts in the dataset Reusters, we collect 5 percent of texts from the 1st class with the most texts, 10 percent from the 2nd, and all the texts from 3rd-10th (3rd-20th) classes.
- TDT2-110(-120): We use all texts in the 10 (20) classes with the smallest number of texts in the dataset TDT2.
- TDT2-t10(-t20): We take 20 percent of texts of 10 (20) classes with the largest number of texts in the dataset TDT2.
- NewsG-t5: We take 50 percent of texts of 5 classes with the largest number of texts in the dataset Newsgroup.

For text datasets, every article is assigned with a vector, which reflects the frequency of each word in the article. While for image datasets, a vector represents the gray level of each pixel in a picture. The data matrix \( A \) is comprised of these vectors. Any solution \( X^* \in S_n^{n,k} \) of ONMF indicates a partition (clustering result) of the dataset. The scale of each dataset is given in Table 4, in which “data” denotes the number of rows of data matrix \( A \) and “features” stands for the number of columns.

Table 4: Description of each dataset. In the table, “d”, “f”, “c” mean “data”, “features” and “clusters”, respectively.

| Name        | d    | f    | c    | Name        | d    | f    | c    | Name        | d    | f    | c    |
|-------------|------|------|------|-------------|------|------|------|-------------|------|------|------|
| Reuters-t10 | 1897 | 12444| 10   | Reuters-t20 | 2402 | 13568| 20   | TDT2-110    | 653  | 13684| 10   |
| TDT2-t10    | 1477 | 22181| 10   | TDT2-120    | 1938 | 20845| 20   | TDT2-t20    | 1721 | 23674| 20   |
| NewsG-t5    | 2344 | 14475| 5    | MNIST       | 4000 | 784  | 10   | Yale        | 165  | 1024 | 5    |

We consider three criteria to compare the performance of clustering results: purity, entropy and NMI. We denote \( k \) as the number of clusters, and \( n \) the total number of data points. Suppose that \( C = \bigcup_{i=1}^{k} C_i \) and \( C' = \bigcup_{j=1}^{k} C'_j \) are clustering results given by ground truth and certain test algorithm. Let \( n_i = |C_i| \), \( n_j = |C'_j| \) and \( n_{ij} = |C_i \cap C'_j| \). The purity \([17]\) is computed as purity := \( \sum_{i=1}^{k} \max_j \{n_{ij}\}/n \). Purity gives a measure of the predominance of the largest category per cluster, better
clustering results leads to larger purity. The entropy [47] and normalized mutual information (NMI) [39] are computed as entropy := $-\frac{1}{n \log_2 k} \sum_{j=1}^k \sum_{i=1}^n n_{ij} \log_2 \frac{n_{ij}}{n_j'}$ and NMI := $\frac{\max(H(C), H(C'))}{\sum_{i=1}^k \sum_{j=1}^n n_{ij} \frac{n_{ij}}{n_j'}},$ where $H(C) = -\sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$ and $H(C')$ was defined similarly. A better clustering result has smaller entropy and larger NMI. Note that we will not calculate feas for K-means and EN-onmf, as they only generate the clustering results instead of solutions of ONMF problem. For random algorithm, their results are averaged over 10 runs.

In Table 5, we report text and images clustering results. We can observe from this table that our proposed method performs very well. Specifically, the clustering results given by our proposed method has the highest purity and NMI in most of cases (being close for the rest dataset). As to the speed, our method is faster than U-onmf and ONP-MF for most of cases, and it is especially efficient on text dataset. Besides, the feasibility violation of the solution returned by our method is very small, while those returned by the other methods are always very large. On the other hand, K-means is the fastest among all algorithms and performs well on image datasets MNIST and Yale, but it results poorly when applying to text dataset; EM-onmf and OPNMF are efficient but their performance is slightly worse than ours.

Table 5: Text clustering results on real datasets. In the table, “c1”, “c2” and “c3” stand for “purity” (%), “NMI” (%) and “entropy” (%), respectively; “t” means the time in seconds. Results marked in bold mean better performance in the corresponding index.

| datasets       | our method | U-onmf | K-means |
|----------------|------------|--------|---------|
| c1 | c2 | c3 | feas | t | c1 | c2 | c3 | feas | t | c1 | c2 | c3 | t |
| Ret-t10 | 73.1 | 60.7 | 37.9 | 2e-15 | 9 | 72.7 | 59.2 | 39.4 | 0.6 | 55 | 36.9 | 22.2 | 75.1 | 4 |
| Ret-t20 | 65.5 | 56.3 | 38.4 | 2e-15 | 25 | 60.6 | 52.7 | 41.7 | 0.9 | 149 | 33.9 | 17.4 | 79.8 | 4 |
| TDT2-l10 | 84.5 | 79.9 | 20.1 | 9e-16 | 4 | 81.8 | 76.0 | 24.0 | 0.4 | 7 | 35.2 | 26.2 | 71.3 | 0.8 |
| TDT2-t10 | 85.7 | 70.0 | 20.8 | 2e-15 | 9 | 80.9 | 65.7 | 22.8 | 0.5 | 115 | 41.1 | 17.8 | 70.5 | 4 |
| TDT2-l20 | 83.1 | 84.2 | 15.5 | 1e-15 | 17 | 81.9 | 82.0 | 17.7 | 0.4 | 60 | 23.8 | 17.6 | 80.7 | 6 |
| TDT2-t20 | 82.3 | 69.6 | 18.1 | 1e-15 | 18 | 79.3 | 64.3 | 21.2 | 0.7 | 299 | 39.1 | 18.6 | 65.8 | 7 |
| NewsG-t5 | 41.5 | 22.8 | 77.1 | 2e-15 | 7 | 39.3 | 14.9 | 85.0 | 0.2 | 44 | 35.7 | 15.4 | 84.5 | 2 |
| MNIST | 60.1 | 48.9 | 51.0 | 1e-15 | 26 | 50.0 | 41.9 | 58.0 | 1.0 | 14 | 55.4 | 45.2 | 54.7 | 0.9 |
| Yale | 44.8 | 47.9 | 52.1 | 6e-16 | 2 | 43.7 | 45.9 | 54.0 | 1.2 | 2 | 40.8 | 44.1 | 55.9 | 0.1 |

| datasets       | OPNMF | ONP-MF | EM-onmf |
|----------------|-------|--------|---------|
| c1 | c2 | c3 | feas | t | c1 | c2 | c3 | feas | t | c1 | c2 | c3 | t |
| Ret-t10 | 72.0 | 58.7 | 39.9 | 1.1 | 15 | 66.9 | 52.8 | 45.6 | 3e-3 | 82 | 71.3 | 58.6 | 39.9 | 17 |
| Ret-t20 | 62.9 | 54.6 | 40.0 | 1.8 | 24 | 62.0 | 53.5 | 41.6 | 4e-3 | 386 | 64.1 | 57.4 | 37.8 | 30 |
| TDT2-l10 | 82.4 | 77.3 | 22.6 | 1.1 | 1 | 81.3 | 75.5 | 24.4 | 3e-3 | 77 | 78.0 | 78.5 | 21.4 | 4 |
| TDT2-t10 | 82.2 | 64.3 | 24.4 | 0.9 | 10 | 82.9 | 65.3 | 23.8 | 3e-3 | 133 | 85.0 | 71.3 | 20.1 | 21 |
| TDT2-l20 | 83.4 | 82.5 | 17.2 | 1.5 | 6 | 82.6 | 83.1 | 16.5 | 4e-3 | 450 | 80.4 | 82.0 | 17.7 | 27 |
| TDT2-t20 | 79.1 | 62.5 | 21.4 | 1.1 | 14 | 81.1 | 65.0 | 20.4 | 4e-3 | 542 | 80.8 | 67.2 | 19.3 | 25 |
| NewsG-t5 | 37.1 | 13.1 | 86.7 | 0.4 | 11 | 42.9 | 22.6 | 77.2 | 2e-3 | 44 | 35.7 | 15.4 | 84.5 | 14 |
| MNIST | 55.1 | 44.1 | 55.9 | 1.3 | 218 | 57.4 | 46.1 | 53.8 | 5e-2 | 61 | 56.3 | 47.8 | 52.2 | 4 |
| Yale | 43.7 | 45.4 | 54.6 | 1.4 | 4 | 40.0 | 43.6 | 56.6 | 1e-2 | 10 | 38.1 | 41.7 | 58.3 | 0.1 |

5.2.3. Hyperspectral unmixing. A set of images taken on the same object at different wave lengths is called a hyperspectral image. At a given wavelength, images are generated by surveying reflectance on each single pixel. Hyperspectral unmixing plays an essential role in hyperspectral image analysis [8, 24]. It assumes that each pixel spectrum $a \in \mathbb{R}^r_+$ is a composite of $k$ spectral bases $\{y_i\}_{i=1}^k \in \mathbb{R}^r_+$. Each spectral
base is denoted as an endmember, which represents the pure spectrum. For example, a spectral base could be the spectrum of “rock”, “tree” etc.

Linear mixture model [24] approximates the pixel spectrum \( \mathbf{a} \) by a linear combination of endmembers as

\[
\mathbf{a} = \mathbf{Y} \mathbf{x} + \mathbf{r},
\]

where \( \mathbf{x} \in \mathbb{R}_+^k \) is called the abundance vector corresponding to pixel \( \mathbf{a} \), \( \mathbf{r} \in \mathbb{R}^r \) is a residual term and \( \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_k] \in \mathbb{R}_{+}^{r \times k} \) is the endmember matrix. When ONMF is applied to hyperspectral unmixing, we assume that both endmember and abundances remain unknown. In addition, each pixel only corresponds to one material. That is to say, \( \mathbf{x} \) only has one non-zero element. For all the pixels combined together, the ONMF formulation of hyperspectral image unmixing becomes (1.3), where

\[
\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \top \in \mathbb{R}_+^{n \times r} \text{ is a hyperspectral image matrix}
\]

with row vectors correspond to its pixels and \( \mathbf{X} \in S_{+}^{n, k} \) is the abundance matrix with \( \mathbf{X}_i \) representing the \( i \)-th abundance vector for \( i \in [n] \).

We test algorithms on three hyperspectral image datasets, Samson, Jasper Ridge and Urban [48]. They are widely used datasets in the hyperspectral unmixing study and can be downloaded at [http://www.escience.cn/people/feiyunZHU/Dataset_GT.html](http://www.escience.cn/people/feiyunZHU/Dataset_GT.html). Since the sizes of the first two images are huge, we choose a region in each image. This process is common in the context of hyperspectral unmixing. For Samson, a region which contains 95 \times 95 pixels is chosen, starting from the (252, 332)-th pixel in original image. We choose a subimage of Jasper Ridge with 100 \times 100 pixels, whose first pixel corresponds to the (105, 269)-th pixel in the original image. The size of refined Samson is 156 \times 95 \times 95, which contains three endmembers: water, tree and rock. The size of refined Jasper Ridge is 198 \times 100 \times 100, and its endmembers include water, tree, dirt and road. Urban is the largest hyperspectral data with 307 \times 307 pixels observed at 162 wavelengths, and there are four endmembers: asphal, grass, tree and roof. Figure 1 gives an illustration of these datasets.

\[
\begin{align*}
(a) \text{Samson} & & (b) \text{Jasper Ridge} & & (c) \text{Urban}
\end{align*}
\]

Fig. 1: Three real hyperspectral images

Since the groundtruth of abundance matrix \( \mathbf{X} \) does not satisfy the orthogonality constraints, the criteria utilized in the preceding subsection are not appropriate to measure the quality of hyperspectral unmixing. Here we consider spectral angle distance (SAD) (see for instance [48]) to evaluate the performance of algorithms. SAD uses an angle distance between groundtruth and estimated endmembers to measure the accuracy of endmember estimation. It is defined as

\[
\text{SAD} := \frac{1}{k} \sum_{i=1}^{k} \arccos \left( \frac{\mathbf{y}_i \cdot \hat{\mathbf{y}}_i}{\| \mathbf{y}_i \| \| \hat{\mathbf{y}}_i \|} \right),
\]

where \( \hat{\mathbf{y}}_i \) and \( \mathbf{y}_i \) are estimation of \( i \)-th endmember and its corresponding groundtruth. Smaller SAD corresponds to better performance. Since other algorithms cannot generate a solution of problem (1.3) with small feasibility violation, in order to keep a fair comparison, we perform the rounding procedure and postprocessing on the solution generated by each method. Note that the postprocessing problem (3.18) is easy to solve, it mainly needs to find the maximum singular
The unmixing results of Samson, Jasper Ridge and Urban are illustrated in Figure 2, Figure 3 and Figure 4, respectively. For Samson image, our method and ONP-MF are able to separate three endmembers, while the rest methods mix them together. For Jasper Ridge image, none of the methods can identify the road endmember, while our method and K-means can split water from other endmembers completely. All of algorithms perform relatively well on Urban dataset except for K-means, being able to separate four endmembers.

Fig. 2: Unmixing results of Samson, from top to bottom: rock, tree, water.

Fig. 3: Unmixing results of Jasper Ridge, from top to bottom: tree, water, dirt, road

Finally, we report in Table 6 the SAD and time cost for the three hyperspectral image datasets. From this table, we know that the efficiency of the proposed method is competitive to other algorithms. Particularly, our method achieves satisfying SAD among all algorithms. Besides, although EM-onmf is faster than our method on these datasets, the unmixing quality given by EM-onmf is unstable.
Fig. 4: Unmixing results of Urban, from top to bottom: asphal, grass, tree, roof

Table 6: Results on the hyperspectral image datasets.

| method       | Samson |          | Jasper Ridge |          | Urban |
|--------------|--------|----------|--------------|----------|-------|
|              | SAD    | time(s)  | SAD          | time(s)  | SAD   | time(s) |
| our method   | 0.081  | 1.0      | 0.150        | 1.3      | 0.114 | 22      |
| U-onnmf      | 0.365  | 10       | 0.306        | 19       | 0.128 | 99      |
| OPNMF        | 0.348  | 44       | 0.336        | 85       | 0.132 | 545     |
| K-means      | 0.296  | 0.2      | 0.174        | 0.4      | 0.266 | 4       |
| ONP-MF       | 0.085  | 16       | 0.276        | 34       | 0.112 | 339     |
| EM-onnmf     | 0.196  | 0.4      | 0.192        | 0.8      | 0.091 | 17      |

5.3. K-indicators model. We first remove the zero norm constraints from (1.5). The exact penalty model (3.23) with $p = 1$, $q = 2$, and $\epsilon = 0$ for solving the K-indicator model becomes

$$
\min_{X \in S_{n,k}^{+}, Y \in S_{k,k}} \left\{ P_\sigma(X, Y) := \|UY - X\|_F^2 + \sigma\|XV\|_F^2 \right\},
$$

which is further equivalent to

$$
\min_{X \in S_{n,k}^{+}, Y \in S_{k,k}} \left\{ \tilde{P}_\sigma(X, Y) := -\frac{1}{\sigma}(UY, X) + \frac{1}{2}\|XV\|_F^2 \right\}.
$$

With a fixed $Y$, (5.4) is exactly (5.2) with $C = UY$. Similar to the discussion therein, we obtain the main PALM iterations [9] for solving (5.4) in Algorithm 3.2 as

$$
Y^{l+1} = \Pi_{S_{k,k}} (Y^l + \beta U^T X^{l+1}), \quad \beta > 0,
$$

$$
X^{l+1} = \Pi_{\Theta_{n,k}} (X^l - \alpha (X^l V V^T - UY^l / \sigma)), \quad 0 < \alpha < 1.
$$

As to the two projectors, see Appendix B.2. Theorem 1 in [9] tells that the sequence $\{(X^l, Y^l)\}$ generated by (5.5) and (5.6) converges to a critical point of (5.4). However,
we find the convergence is slow if we fix the constant stepsizes $\alpha$ and $\beta$. Noting that (5.4) with $X = X^l$ has closed form solution, $\Pi_{S_{n,k}}(U^TX^l)$, we choose $\beta = +\infty$ in (5.5). For the tested problem, by some easy calculations, we can see $\alpha^l_{LBB} \geq 1$. The practical PALM iterations for solving (5.4) is thus given as

\begin{align}
Y^{l+1} &= \Pi_{S_{k,k}}(U^TX^l), \\
X^{l+1} &= \Pi_{OB_{n,k}}(X^l - \alpha^l (X^lVV^T - UY^l/\sigma)) , \quad \alpha^l = \min\{\alpha^l_{LBB}, 10^k\}
\end{align}

(5.7) (5.8)

Note that the flops for (5.7) and (5.8) are $2nk^2 + O(nk)$ and $2nk^2 + O(k^3)$, respectively.

Chen et al. [15] proposed a semi-convex relaxation model to solve (1.5). Their model corresponds to (5.3) with $\sigma = 0$ and $X \in S_{n,k}^+$ replaced by $X \in \{X \in \mathbb{R}^{n \times k} : 0 \leq X \leq 1\}$. A double-layered alternating projection framework was investigated in [15] to solve the semi-convex relaxation model. The method was named KindAP. To evaluate the efficiency of our method, we compare it with KindAP (downloaded from https://github.com/yangyuchen0340/Kind) on data clustering problems.

Table 7: Comparison of KindAP and our methods on data clustering problems. In the table, “a” and “b” stand for KindAP and our method, respectively. Results marked in bold mean better performance in the corresponding index.

| datasets     | n   | k   | purity(%) | NMI(%) | entropy(%) | time(s) |
|--------------|-----|-----|-----------|--------|------------|---------|
|              | a   | b   | a         | b      | a          |         |
| catsndogs    | 4000| 2   | 96.20     | 96.23  | 76.80      | 0.03    |
|              | 2   |     |           |        |            | 0.02    |
| ORL          | 400 | 40  | 87.75     | 88.00  | 92.94      | 7.06    |
|              | 2   |     |           |        |            | 7.24    |
| CIFAR100-test| 10000| 100 | 69.42     | 69.44  | 71.34      | 28.66   |
|              | 2   |     |           |        |            | 28.64   |
| CIFAR100-train| 50000| 100 | 99.63     | 99.63  | 99.57      | 0.43    |
|              | 2   |     |           |        |            | 0.43    |
| COIL100      | 7200| 100 | 91.93     | 91.93  | 97.30      | 2.70    |
|              | 2   |     |           |        |            | 2.59    |
| flower       | 2040| 102 | 44.95     | 44.90  | 63.52      | 36.48   |
|              | 2   |     |           |        |            | 36.60   |
| omniglot     | 17853| 1623| 21.95     | 21.97  | 70.86      | 29.14   |
|              | 2   |     |           |        |            | 29.06   |
| UKBench      | 10200| 2550| 90.64     | 91.04  | 97.64      | 2.36    |
|              | 2   |     |           |        |            | 2.24    |

We adopt eight image datasets, including catsndogs, ORL, CIFAR (train and test), COIL100, flower, omniglot, and UKBench. We set $\sigma_0 = 10$, $\eta = 0.5$ and tol\textsuperscript{feas} = 0.1 in our algorithm. The initial points of KindAP and our method are set as $X^0 = U_+$ and $X^0 = \Pi_{OB_{n,k}}(U_+)$, respectively. Similar as in section 5.2.2, purity, entropy and NMI are adopted to judge the performance of proposed algorithms. The results are presented in Table 7. It shows that the clustering results given by our methods are comparable to that provided by KindAP, which means both methods are able to solve (1.5) with a relatively high quality. On the other hand, our algorithm is generally faster than KindAP. Our algorithm is especially efficient on datasets omniglot and UKbench, in which the number of clusters is relatively large. Besides, it should be mentioned that although we relax the zero norm constraints from problem (1.5), the matrix $X$ we obtained is always feasible to (1.5). By contrast, the matrix $X$ returned by KindAP may not be an orthogonal nonnegative matrice although it always satisfies the zero norm constraints.

6. Concluding remarks. In this paper, we consider optimization with nonnegative and orthogonality constraints. We focus on an equivalent formulation of the concerned problem via giving a characterization of the feasible set $S_{n,k}^+$. We investigate some theoretical properties of the new formulation, including the constraint
Appendix A. Constraint qualifications. Consider a general nonlinear programming

\[ \min_{x \in \mathbb{R}^n} g(x) \quad \text{s.t.} \quad c_i(x) = 0, i \in \mathcal{E}, \quad c_i(x) \geq 0, i \in \mathcal{I}, \]

where \( \mathcal{E} = \{1, \ldots, m_e\}, \mathcal{I} = \{m_e + 1, \ldots, r\} \) and the functions \( g \) and \( c_i, i \in \mathcal{E} \cup \mathcal{I} \) are continuously differentiable. With slight of abused notations, we use \( \mathcal{X} \) to denote the feasible set of (A.1). Consider a feasible point \( x \in \mathcal{X} \), the tangent cone of \( \mathcal{X} \) at \( x \) is

\[ \mathcal{T}_\mathcal{X}(x) = \{0\} \cup \left\{ d \in \mathbb{R}^n : \exists \{x^{(l)}\} \subset \mathcal{X}, x^{(l)} \neq x, x^{(l)} \to x \text{ such that } \frac{x^{(l)} - x}{\|x^{(l)} - x\|} \to \frac{d}{\|d\|} \right\} \]

and the set of linearized feasible directions is

\[ \mathcal{LFD}_\mathcal{X}(x) = \{d \in \mathbb{R}^n : d^T \nabla c_i(x) = 0, i \in \mathcal{E}, d^T \nabla c_i(x) \geq 0, i \in \mathcal{I}(x)\}, \]

where \( \mathcal{I}(x) = \{i : c_i(x) = 0, i \in \mathcal{I}\} \). It is well known that \( \mathcal{T}_\mathcal{X}(x) \subseteq \mathcal{LFD}_\mathcal{X}(x) \); see [35] for instance. The constraint qualification (CQ) is used to guarantee that a local minimizer of (A.1) is a stationary point of (A.1). (Suppose that problem (A.1) has at least one stationary point.) We introduce five CQs here.

**Definition A.1.** Consider a feasible point \( x \in \mathcal{X} \).

(i) The Guignard CQ (GCQ) \([20]\) holds at \( x \) if \( \mathcal{T}_\mathcal{X}(x)^o = \mathcal{LFD}_\mathcal{X}(x)^o \).

(ii) The Abadie CQ (ACQ) \([1]\) holds at \( x \) if \( \mathcal{T}_\mathcal{X}(x) = \mathcal{LFD}_\mathcal{X}(x) \).

(iii) The relaxed constant positive linear dependence CQ (RCPLD) \([3]\) holds at \( x \) if there exist a neighborhood \( \mathcal{N}(x) \) of \( x \) such that \( a\{\nabla c_i(y)\} \in \mathcal{E} \) has the same rank for every \( y \in \mathcal{N}(x) \), where \( \hat{\mathcal{E}} \subseteq \mathcal{E} \) is chosen such that \( \{\nabla c_i(x)\}_{i \in \hat{\mathcal{E}}} \) is a basis for \( \text{span}\{\nabla c_i(x)\}_{i \in \mathcal{E}} \); \( b \) for every \( \mathcal{J} \subseteq \mathcal{I}(x) \), if the vectors \( \{\nabla c_i(x)\}_{i \in \mathcal{E}}, \{-\nabla c_i(x)\}_{i \in \mathcal{J}} \) are positive-linearly dependent \(^2\), then the vectors \( \{\nabla c_i(x)\}_{i \in \mathcal{E}}, \{-\nabla c_i(y)\}_{i \in \mathcal{J}} \) are linearly dependent for every \( y \in \mathcal{N}(x) \).

(iv) The Mangasarian-Fromovitz CQ (MFCQ) \([29]\) holds at \( x \) if \( \nabla c_i(x), i \in \mathcal{E} \) are linearly independent and \( \{d \in \mathbb{R}^n : d^T \nabla c_i(x) = 0, i \in \mathcal{E}, d^T \nabla c_i(x) > 0, i \in \mathcal{I}(x)\} \neq \emptyset \).

(v) The linear independence CQ (LICQ) \([21, p. 29]\) holds at \( x \) if the set \( \{\nabla c_i(x), i \in \mathcal{E} \cup \mathcal{I}(x)\} \) is linearly independent.

Gould and Tolle [19] showed that the GCQ is the weakest CQ. We also have the implications: LICQ \( \implies \) MFCQ \( \implies \) RCPLD \( \implies \) ACQ \( \implies \) GCQ; see for example [3].

Let \( \hat{x} \) be a stationary point of problem (A.1) and \( \hat{\lambda}, i \in \mathcal{E} \cup \mathcal{I} \) be the corresponding Lagrange multipliers. The set of all sequential null constraint directions is given as

\[ \mathcal{SNCD}_\mathcal{X}(\hat{x}, \hat{\lambda}) = \left\{ d \in \mathbb{R}^n : \begin{array}{l}
\text{x}^{(l)} = \hat{x} + \alpha^{(l)}d^{(l)} \in \mathcal{X}, \alpha^{(l)} > 0, \alpha^{(l)} \to 0, d^{(l)} \to d, \\
c_i(x^{(l)}) = 0, i \in \mathcal{E} \cup \mathcal{I}_+(\hat{x}), \\
c_i(x^{(l)}) \geq 0, i \in \mathcal{I}(x)/\mathcal{I}_+(\hat{x})
\end{array} \right\} \]

\(^2\)That is to say, there exists \( \{\alpha_i \in \mathbb{R}, i \in \mathcal{E}\} \) and \( \{\beta_i \in \mathbb{R}_+, i \in \mathcal{I}(x)\} \) not all zero such that \( \sum_{i \in \mathcal{E}} \alpha_i \nabla c_i(x) + \sum_{i \in \mathcal{I}(x)} \beta_i (-\nabla c_i(x)) = 0 \).
where \( \mathcal{I}_+(\hat{x}) = \{ i : i \in \mathcal{I}(\hat{x}) \text{ with } \hat{\lambda}_i > 0 \} \). The set of all linearized null constraint directions is given as

\[ \text{LNC}_X(\hat{x}, \hat{\lambda}) = \{ d \in \mathbb{R}^n : d \in \text{LFD}_X(\hat{x}), d^\top \nabla c_i(\hat{x}) = 0, i \in \mathcal{I}_+(\hat{x}) \} . \]

Note that there always hold that \( \text{SNCD}_X(\hat{x}, \hat{\lambda}) \subseteq \text{LNC}_X(\hat{x}, \hat{\lambda}) \). We simply write \( \text{SNCD}_X(\hat{x}) \) (resp. \( \text{LNC}_X(\hat{x}) \)) if \( \text{SNCD}_X(\hat{x}, \hat{\lambda}) \) (resp. \( \text{LNC}_X(\hat{x}, \hat{\lambda}) \)) is always the same for different \( \hat{\lambda} \). For more details, one can refer to [35].

Appendix B. Several projection issues.

B.1. Construction of problem (5.1) with unique and known solution.

**Proposition B.1.** Choose \( X^* \in S_+^{n,k} \) and \( L \in \mathbb{R}^{k \times k} \) with positive diagonal elements satisfying

\[ L_{ii}L_{jj} > \max\{L_{ij}, L_{ji}, 0\}^2, \quad \forall i, j \in [k], i \neq j. \]

Then the optimal solution of (5.1) is equivalent to \( \max_i \langle C, X \rangle \) and exactly \( X^* \).

**Proof.** For simplicity of notation, we use \( \sum_i \) to denote \( \sum_{i \in [k]} \) in the proof. It is clear that problem (5.1) is equivalent to \( \max_{X \in S_+^{n,k}} \langle C, X \rangle \). Hence we only need to show that \( \langle C, Y \rangle < \langle C, X^* \rangle = \sum_i L_{ii} \) for all \( Y \in S_+^{n,k} \) and \( Y \neq X^* \). Let \( Z = \text{sgn}(Y) \) and \( P = \Pi_+(L) \). We have

\[ \langle C, Y \rangle = \text{tr}(L(X^*)^\top Y) = \sum_i \sum_j L_{ij} Y_{i}^\top x_i^* \leq \sum_i \sum_j P_{ji} Y_{i}^\top (x_i^* \circ z_i). \]

Define \( w_{ji} = \|x_i^* \circ z_i\|^2 \). With \( X^* \in S_+^{n,k} \), we have \( \| \sum_j P_{ji}(x_i^* \circ z_i) \| = (\sum_j P_{ji}^2 w_{ji})^{1/2} \) and \( \sum_i w_{ji} \leq 1 \). Using the Cauchy-Schwarz inequality, \( \|y_i\| = 1 \) and (B.1), we have

\[ \sum_j P_{ji} Y_{i}^\top (x_i^* \circ z_i) \leq \left( \sum_j P_{ji}^2 w_{ji} \right)^{1/2} \leq P_{ii} \left( \sum_j P_{ji}^2 / P_{ii} w_{ji} \right)^{1/2}. \]

With (B.2) and \( \langle C, X^* \rangle = \sum_i L_{ii} = \sum_i P_{ii} \), we further have

\[ \langle C, Y \rangle \leq \sum_i P_{ii} \left( \sum_j P_{ji}^2 / P_{ii} w_{ji} \right)^{1/2} \leq \left( \sum_i P_{ii} \right)^{1/2} \left( \sum_i \sum_j P_{ji}^2 w_{ji} \right)^{1/2} \leq \langle C, X^* \rangle, \]

where the second inequality uses the fact that \( \sum_i a_i x_i^2 \leq \left( \sum_i a_i \right)^2 \left( \sum_i a_i x_i \right)^2 \) for \( a_i > 0 \) and \( x_i \geq 0 \), and the third inequality uses \( \sum_i w_{ji} \leq 1 \). Obviously, the equalities in (B.3) and (B.4) hold if and only if \( Y = X^* \). The proof is completed.

B.2. Projection onto \( S_+^{k,k} \) and \( \mathcal{OB}_+^{n,k} \). Given \( K \in \mathbb{R}^{k \times k} \) with its SVD as \( K = M \Sigma N^\top \), we have \( \Pi_{S_+^{k,k}}(K) = MN^\top \). Given \( C \in \mathbb{R}^{n \times k} \), we have \( \Pi_{\mathcal{OB}_+^{n,k}}(C) = \Pi_{S_+^{n-1}}(c_j), \forall j \in [k] \) with \( S_+^{n-1} = \{ x \in \mathbb{R}^n : \|x\| = 1, x \geq 0 \} \). For \( a \in \mathbb{R}^n \), letting \( a_{\max} := \max_{i \in [n]} a_i \) and \( \mathcal{A}(a) := \{ i \in [n] : a_i = a_{\max} \} \), we have

\[ \Pi_{S_+^{n-1}}(a) = \begin{cases} \Pi_{+}(a)/\| \Pi_{+}(a) \|, & \text{if } a_{\max} > 0, \\ \left\{ \sum_{i \in \mathcal{A}(a)} \alpha_i e_i : \alpha_i \in \mathbb{R}_+, \sum_i \alpha_i^2 = 1 \right\}, & \text{if } a_{\max} = 0, \\ \{ e_i : i \in \mathcal{A}(a) \}, & \text{if } a_{\max} < 0. \end{cases} \]

The above formulation appears in Example 8.9 in [7] and is also discussed in [40, 46]. Thanks to (B.5), \( \Pi_{\mathcal{OB}_+^{n,k}}(C) \) can be computed explicitly in \( O(nk) \) flops.
REFERENCES

[1] J. Abadie, On the Kuhn-Tucker theorem, in Nonlinear Programming, J. Abadie, ed., Wiley, New York, 1967, pp. 21–36.
[2] P.-A. Absil, R. Mahony, and R. Sepulchre, Optimization algorithms on matrix manifolds, Princeton University Press, 2009.
[3] R. Andreani, G. Haeser, M. L. Schuverdt, and P. J. Silva, A relaxed constant positive linear dependence constraint qualification and applications, Math. Program., 135 (2012), pp. 255–273.
[4] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the curdya-lojasiewicz inequality, Math. Oper. Res., 35 (2010), pp. 438–457.
[5] H. Attouch, J. Bolte, and B. F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods, Math. Program., 137 (2013), pp. 91–129.
[6] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, IMA J. Numer. Anal., 8 (1988), pp. 141–148.
[7] H. Baitschke, M. Bai, and X. Wang, Projecting onto the intersection of a cone and a sphere, SIAM J. Optim., 28 (2018), pp. 2158–2188.
[8] J. M. Bioucasdias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., 5 (2012), pp. 354–379.
[9] J. Bolte, S. Sabach, and M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Math. Program., 146 (2014), pp. 459–494.
[10] J. Borwein and A. S. Lewis, Convex analysis and nonlinear optimization: theory and examples, Springer Science & Business Media, 2010.
[11] C. Boutsidis, P. Drineas, and M. W. Mahoney, Unsupervised feature selection for the k-means clustering problem, in Advances in Neural Information Processing Systems, 2009, pp. 153–161.
[12] C. Boutsidis and E. Gallopoulos, SVD based initialization: A head start for nonnegative matrix factorization, Pattern Recogn., 41 (2008), pp. 1350–1362.
[13] D. Cai, Q. Mei, J. Han, and C. Zhai, Modeling hidden topics on document manifold, in Proceedings of the 17th ACM conference on Information and knowledge management, ACM, 2008, pp. 911–920.
[14] T. Carson, D. G. Mixon, and S. Villar, Manifold optimization for k-means clustering, in International Conference on Sampling Theory and Applications, IEEE, 2017, pp. 73–77.
[15] F. Chen, Y. Yang, L. Xu, T. Zhang, and Y. Zhang, Big-data clustering: K-means or k-indicators?, arXiv:1906.00938, (2019).
[16] X. Chen, Z. Lu, and T. K. Pong, Penalty methods for a class of non-Lipschitz optimization problems, SIAM J. Optim., 26 (2016), pp. 1465–1492.
[17] C. Ding, T. Li, W. Peng, and H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 126–135.
[18] B. Gao, X. Liu, and Y.-X. Yuan, Parallelizable algorithms for optimization problems with orthogonality constraints, to appear in SIAM J. Sci. Comput., (2018).
[19] F. Gould and J. W. Tolle, A necessary and sufficient qualification for constrained optimization, SIAM J. Appl. Math., 20 (1971), pp. 164–172.
[20] M. Guignard, Generalized Kuhn-Tucker conditions for mathematical programming problems in a Banach space, SIAM J. Control, 7 (1969), pp. 232–241.
[21] M. R. Hestenes, Calculus of variations and optimal control theory, Wiley, 1966.
[22] J. He, A. Mijatovic, Z. Wen, and Y. Yuan, Adaptive quadratically regularized Newton method for Riemannian optimization, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1181–1207.
[23] B. Jiang, Y.-F. Liu, and Z. Wen, 1p-norm regularization algorithms for optimization over permutation matrices, SIAM J. Optim., 26 (2016), pp. 2284–2313.
[24] N. Keshava and J. F. Mustard, Spectral unmixing, IEEE Signal Process. Mag., 19 (2002), pp. 44–57.
[25] D. Kuang, C. Ding, and H. Park, Symmetric nonnegative matrix factorization for graph clustering, in Proceedings of the 2012 SIAM international conference on data mining, SIAM, 2012, pp. 106–117.
[26] B. Li, G. Zhou, and A. Cichocki, Two efficient algorithms for approximately orthogonal nonnegative matrix factorization, IEEE Signal Process. Lett., 22 (2015), pp. 843–846.
[27] X. Li, D. Sun, and K.-C. Toh, On the efficient computation of a generalized Jacobian of the projector over the Birkhoff polytope, Math. Program., (2018), pp. 1–28.

[28] D. Luo, C. Ding, H. Huang, and T. Li, Non-negative Laplacian embedding, in 2009 Ninth IEEE International Conference on Data Mining, IEEE, 2009, pp. 337–346.

[29] O. L. Mangasarian and S. Fromovitz, The Fritz John necessary optimality conditions in the presence of equality and inequality constraints, J. Math. Anal. Appl., 17 (1967), pp. 37–47.

[30] A. Montanari and E. Richard, Non-negative principal component analysis: Message passing algorithms and sharp asymptotics, IEEE Trans. Inf. Theory, 62 (2016), pp. 1458–1484.

[31] J. Pan and M. K. Ng, Orthogonal nonnegative matrix factorization by sparsity and nuclear norm optimization, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 856–875.

[32] S. Paul and Y. Chen, Orthogonal symmetric non-negative matrix factorization under the stochastic block model, arXiv:1605.05349, (2016).

[33] F. Pomponi, N. Gillis, P.-A. Absil, and F. Glineur, Two algorithms for orthogonal non-negative matrix factorization with application to clustering, Neurocomputing, 141 (2014), pp. 15–25.

[34] J. Povh and F. Rendl, A copositive programming approach to graph partitioning, SIAM J. Optim., 18 (2007), pp. 223–241.

[35] W. Sun and Y.-X. Yuan, Optimization theory and methods: nonlinear programming, vol. 1, Springer Science & Business Media, 2006.

[36] S. Wang, T.-H. Chang, Y. Cui, and J.-S. Pang, Clustering by orthogonal NMF model and non-convex penalty optimization, arXiv:1906.00570, (2019).

[37] Z. Wen and W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program., 142 (2013), pp. 397–434.

[38] X. Xiao, Y. Li, Z. Wen, and L. Zhang, A regularized semi-smooth Newton method with projection steps for composite convex programs, J. Sci. Comput., (2016), pp. 1–26.

[39] W. Xu, X. Liu, and Y. Gong, Document clustering based on non-negative matrix factorization with application to clustering, Neurocomputing, 141 (2014), pp. 15–25.

[40] Y. Xu and W. Yin, A globally convergent algorithm for nonconvex optimization based on block coordinate update, J. Sci. Comput., 72 (2017), pp. 700–734.

[41] Y. Yang, Y. Yang, H. T. Shen, Y. Zhang, X. Du, and X. Zhou, Discriminative nonnegative spectral clustering with out-of-sample extension, IEEE Trans. Knowl. Data Eng., 25 (2012), pp. 1760–1771.

[42] Z. Yang and E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Trans. Neural Netw., 21 (2010), pp. 734–749.

[43] J. Yoo and S. Choi, Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds, in International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2008, pp. 140–147.

[44] R. Zass and A. Shashua, Nonnegative sparse PCA, in Advances in neural information processing systems, 2007, pp. 1561–1568.

[45] H. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim., 14 (2004), pp. 1043–1056.

[46] J. Zhang, H. Liu, Z. Wen, and S. Zhang, A sparse completely positive relaxation of the modularity maximization for community detection, SIAM J. Sci. Comput., 40 (2018), pp. A3091–A3120.

[47] Y. Zhao and G. Karypis, Empirical and theoretical comparisons of selected criterion functions for document clustering, Mach. Learn., 55 (2004), pp. 311–331.

[48] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, Spectral unmixing via data-guided sparsity, IEEE Trans. Image Process., 23 (2014), pp. 5412–5427.