Structured Low-Rank Matrix Factorization: Global Optimality, Algorithms, and Applications

Benjamin D. Haeffele, René Vidal, *Fellow, IEEE*

Abstract—Convex formulations of low-rank matrix factorization problems have received considerable attention in machine learning. However, such formulations often require solving for a matrix of the size of the data matrix, making it challenging to apply them to large scale datasets. Moreover, in many applications the data can display structures beyond simply being low-rank, e.g., images and videos present complex spatio-temporal structures that are largely ignored by standard low-rank methods. In this paper we study a matrix factorization technique that is suitable for large datasets and captures additional structure in the factors by using a particular form of regularization that includes well-known regularizers such as total variation and the nuclear norm as particular cases. Although the resulting optimization problem is non-convex, we show that if the size of the factors is large enough, under certain conditions, any local minimizer for the factors yields a global minimizer. A few practical algorithms are also provided to solve the matrix factorization problem, and bounds on the distance from a given approximate solution of the optimization problem to the global optimum are derived. Examples in neural calcium imaging video segmentation and hyperspectral compressed recovery show the advantages of our approach on high-dimensional datasets.

Index Terms—Low-rank matrix factorization, non-convex optimization, calcium imaging, hyperspectral compressed recovery.

1 INTRODUCTION

In many large datasets, relevant information often lies in a subspace of much lower dimension than the ambient space, and thus the goal of many learning algorithms can be broadly interpreted as trying to find or exploit this underlying “structure” that is present in the data. One structure that is particularly useful both due to its wide-ranging applicability and efficient computation is the linear subspace model. Generally speaking, if one is given $N$ data points from a $D$ dimensional ambient space, $Y = [Y_1, Y_2, \ldots, Y_N] \in \mathbb{R}^{D \times N}$, a linear subspace model simply implies that there exists matrices $(U, V)$ such that $Y \approx UV^T$. When one of the factors is known *a priori*, the problem of finding the other factor simplifies considerably, but if both factors are allowed to be arbitrary one can always find an infinite number of $(U, V)$ matrices that yield the same product. As a result, to accomplish anything meaningful, one must impose some restrictions on the factors. This idea leads to a variety of common matrix factorization techniques, which includes the following well-known examples:

- **Principal Component Analysis (PCA):** The number of columns, $r$, in $(U, V)$ is typically constrained to be small, e.g., $r \ll \min\{D, N\}$, and $U$ is constrained to have orthonormal columns, i.e., $U^T U = I$.
- **Nonnegative Matrix Factorization (NMF):** The number of columns in $(U, V)$ is constrained to be small, and $(U, V)$ are required to be non-negative [1], [2].
- **Sparse Dictionary Learning (SDL):** The number of columns in $(U, V)$ is allowed to be larger than $D$ or $N$, the columns of $U$ are required to have unit $l_2$ norm, and $V$ is required to be sparse as measured by, e.g., the $l_1$ norm or the $l_0$ pseudo-norm [3], [4].

Mathematically, the general problem of recovering structured linear subspaces from a dataset can be captured by a **structured matrix factorization problem** of the form

$$\min_{U, V} \ell(Y, UV^T) + \lambda \Theta(U, V), \quad (1)$$

where $\ell$ is some loss function that measures how well $Y$ is approximated by $UV^T$ and $\Theta$ is a regularizer that encourages or enforces specific properties in $(U, V)$. By taking an appropriate combination of $\ell$ and $\Theta$ one can formulate both unsupervised learning techniques, such as PCA, NMF, and SDL, or supervised learning techniques like discriminative dictionary learning [5], [6] and learning max-margin factorized classifiers [7]. However, while there are wide-ranging applications for structured matrix factorization methods that have achieved good empirical success, the associated optimization problem (1) is non-convex regardless of the choice of $\ell$ and $\Theta$ functions due to the presence of the matrix product $UV^T$. As a result, aside from a few special cases (such as PCA), finding solutions to (1) poses a significant challenge, which often requires one to instead consider approximate solutions that depend on a particular choice of initialization and optimization method.

Given the challenge of non-convex optimization, one possible approach to matrix factorization is to relax the non-convex problem into a problem which is convex on the product of the factorized matrices, $X = UV^T$, and then recover the factors of $X$ after solving the convex relaxation. As a concrete example, in low-rank matrix factorization, one might be interested in solving a problem of the form

$$\min_X \ell(Y, X) \text{ s.t. } \text{rank}(X) \leq r, \quad (2)$$

1. As a result, in SDL, one does not assume that there exists a single low-dimensional subspace to model the data, but rather that the data lie in a union of a large number of low-dimensional subspaces.
which is equivalently defined as a factorization problem
\[
\min_{U,V} \ell(Y, UV^T),
\]  
(3)
where the rank constraint is enforced by limiting the number of columns in \((U, V)\) to be equal to \(r\). However, aside from a few special choices of \(\ell\), solving (2) or (3) is in general an NP-hard problem. Instead, one can relax (2) into a convex problem by using a convex regularizer that promotes low-rank solutions, such as the nuclear norm \(\|X\|_*\) (sum of the singular values of \(X\)), and then solve
\[
\min_X \ell(Y, X) + \lambda \|X\|_*
\]  
(4)
which can often be done efficiently if \(\ell(Y, X)\) is convex with respect to \(X\) [8], [9]. Given a solution to (4), \(\hat{X}\), it is then simple to find a low-rank factorization \(UV^T = \hat{X}\) via a singular value decomposition. Unfortunately, while the nuclear norm provides a nice convex relaxation for low-rank matrix factorization problems, nuclear norm relaxation does not capture the full generality of problems such as (1) as it does not necessarily ensure that \(X\) can be factorized as \(\hat{X} = UV^T\) for some \((U, V)\) pair which has the desired structure encouraged by \(\Theta(U, V)\) (e.g., in non-negative matrix factorization we require \(U\) and \(V\) to be non-negative), nor does it provide a means to find the desired factors. Additionally, solving over the full matrix \(X\) also potentially requires significantly more parameters than solving over \((U, V)\) if one expects the solution to (4) to have rank significantly smaller than the dimensions of \(X\), as the iterates of many optimization algorithms, applied to (4), are not guaranteed to be low-rank even if the ultimate solution is low-rank.

Based on the above discussion, optimization problems in the factorized space, such as (1), versus problems in the product space, with (4) as a particular example, both present various advantages and disadvantages. Factorized problems attempt to solve for the desired factors \((U, V)\) directly, provide significantly increased modeling flexibility by permitting one to model structure on the factors (sparsity, non-negativity, etc.), and allow one to potentially work with a significantly reduced number of variables if the number of columns in \((U, V)\) is \(\ll \min\{D, N\}\); however, they suffer from the significant challenges associated with non-convex optimization. On the other hand, problems in the product space can be formulated to be convex, which affords many practical algorithms and analysis techniques, but one is required to optimize over a potentially large number of variables and solve a second factorization problem in order to recover the factors \((U, V)\) from the solution \(X\). These various pros and cons are briefly summarized in Table 1.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Product space</th>
<th>Factorized space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convex</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Problem size</td>
<td>Large</td>
<td>Small</td>
</tr>
<tr>
<td>Structured factors</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

To bridge this gap between the two classes of problems, here we explore the link between non-convex matrix factorization problems, which have the general form
\[
\min_{U,V} \ell(Y, UV^T) + \lambda \Theta(U, V),
\]  
(5)
and a closely related family of convex problems in the product space, given by
\[
\min_X \ell(Y, X) + \lambda \Omega_{\Theta}(X),
\]  
(6)
where the function \(\Omega_{\Theta}\) will be defined based on the choice of the regularization function \(\Theta\) and will have the desirable property of being a convex function of \(X\). Unfortunately, while the optimization problem in (6) is convex w.r.t. \(X\), it will typically be non-tractable to solve due to the fact that the definition of \(\Omega_{\Theta}\) requires one to solve another non-convex optimization problem. Moreover, even if a solution to (6) could be found, solving a convex problem in the product space does not necessarily achieve our goal, as we still must solve another matrix factorization problem to recover the \((U, V)\) factors with the desired properties encouraged by the \(\Theta\) function (sparsity, non-negativity, etc.). Nevertheless, the two problems given by (5) and (6) will be tightly coupled. Specifically, the convex problem in (6) will be shown to be a tight global lower-bound to the non-convex factorized problem in (5), and solutions \((U, V)\) to the factorized problem will yield solutions \(X = UV^T\) to the convex problem. As a result, we will tailor our results to the non-convex factorization problem (5) using the convex problem (6) as an analysis tool. While the optimization problem in the factorized space is not convex, by analyzing this tight interconnection between the two problems, we will show that if the number of columns in \((U, V)\) is large enough and can be adapted to the data instead of being fixed \textit{a priori}, local minima of the non-convex factorized problem will be global minima of both the convex and non-convex problems. This result will lead to a practical optimization strategy that is parallelizable and often requires a much smaller set of variables. Experiments in image processing applications will illustrate the effectiveness of the proposed approach.

2 Mathematical Background & Prior Work

As discussed in the introduction, relaxing low-rank matrix factorization problems via nuclear norm formulations fails to capture the full generality of factorized problems as it does not yield “structured” factors, \((U, V)\), with the desired properties encouraged by \(\Theta(U, V)\) (sparseness, non-negativity, etc.). To address this issue, several studies have explored a more general convex relaxation via the matrix norm given by
\[
\|X\|_{u,v} \equiv \inf_{r \in \mathbb{N}^+} \min_{U,V:UV^T = X} \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_v 
\]
\[
\equiv \inf_{r \in \mathbb{N}^+} \min_{U,V:UV^T = X} \sum_{i=1}^{r} \frac{1}{2}(\|U_i\|_u^2 + \|V_i\|_v^2),
\]  
(7)
where \((U_i, V_i)\) denote the \(i\)th columns of \(U\) and \(V\), respectively, \(\|\cdot\|_u\) and \(\|\cdot\|_v\) are arbitrary vector norms, and the number of columns, \(r\), in the \(U\) and \(V\) matrices is allowed to be variable [10]–[14]. The norm in (7) has appeared under
multiple names in the literature, including the projective tensor norm, decomposition norm, and atomic norm. It is worth noting that for particular choices of the $\| \cdot \|_u$ and $\| \cdot \|_v$ vector norms, $\|X\|_{u,v}$ reverts to several well known matrix norms and thus provides a generalization of many commonly used regularizers. Notably, when the vector norms are both $l_2$ norms, the form in (7) becomes the well known variational definition of the nuclear norm [9]:

$$\|X\|_* = \|X\|_{2,2} \equiv \inf_{r \in \mathbb{N}^+} \inf_{U,V:UV = X} \sum_{i=1}^r \|U_i\|_2 \|V_i\|_2$$

Moreover, by replacing the column norms in (7) with gauge functions one can incorporate additional regularization on $(U, V)$, such as non-negativity, while still being a convex function of $X$ [12]. Recall that given a closed, convex set $C$ containing the origin, a gauge function, $\sigma_C(x)$, is defined as,

$$\sigma_C(x) \equiv \inf_{\mu} \mu \text{ s.t. } \mu x \in C.$$  

Further, recall that all norms are gauge functions (as can be observed by choosing $C$ to be the unit ball of a norm), but gauge functions are a slight generalization of norms, since they satisfy all the properties of a norm except that they are not required to be invariant to non-negative scaling (i.e., it is not required that $\sigma_C(x)$ be equal to $\sigma_C(-x)$). Finally, note that given a gauge function, $\sigma_C$, its polar, $\sigma_C^\circ$, is defined as

$$\sigma_C^\circ(z) \equiv \sup_x \langle z, x \rangle \text{ s.t. } \sigma_C(x) \leq 1,$$

which itself is also a gauge function. In the case of a norm, its polar function is often referred to as the dual norm.

### 2.1 Matrix Factorization as Semidefinite Optimization

Due to the increased modeling opportunities it provides, several studies have explored structured matrix factorization formulations based on the $\| \cdot \|_{u,v}$ norm in a way that allows one to work with a highly reduced set of variables while still providing some guarantees of global optimality. In particular, it is possible to explore optimization problems over factorized matrices $(U, V)$ of the form

$$\min_{U,V} \ell(Y, UV^T) + \lambda \|UV^T\|_{u,v},$$  

(11)

While this problem is convex with respect to the product $X = UV^T$, it is still non-convex with respect to $(U, V)$ jointly due to the matrix product. However, if we define a matrix $\Gamma$ to be the concatenation of $U$ and $V$

$$\Gamma \equiv \begin{bmatrix} U \\ V \end{bmatrix} \implies \Gamma^T = \begin{bmatrix} U^T & UV^T \\ V^T & VV^T \end{bmatrix},$$

(12)

we see that $UV^T$ is a submatrix of the positive semidefinite matrix $M = \Gamma \Gamma^T$. After defining the function $H : S_n^+ \to \mathbb{R}$

$$H(\Gamma \Gamma^T) = \ell(Y, UV^T) + \lambda \|UV^T\|_{u,v},$$

(13)

it is clear that the proposed formulation (11) can be recast as an optimization problem over a positive semidefinite matrix.

At first the above discussion seems to be a circular argument, since while $H(M)$ is a convex function of $M$, this says nothing about finding $\Gamma$ (or $U$ and $V$). However, results for semidefinite programs in standard form [15] show that one can minimize $H(M)$ by solving for $\Gamma$ directly without introducing any additional local minima, provided that the rank of $\Gamma$ is larger than the rank of the true solution, $M$. Further, if the rank of $M$ is not known a priori and $H(M)$ is twice differentiable, then any local minima w.r.t. $\Gamma$ such that $\Gamma$ is rank-deficient give a global minimum of $H(\Gamma \Gamma^T)$ [11], and these results can also be extended to non-differentiable functions as shown in our prior work [13].

However, while the above results from semidefinite programming are sufficient if we only wish to find general factors such that $UV^T = X$, for the purposes of solving structured matrix factorizations, we are interested in finding factors $(U, V)$ that achieve the infimum in the definition of (7), which is not provided by a solution to (11), as the $(U, V)$ that one obtains is not necessarily the $(U, V)$ that minimize (7). Put differently, results from semidefinite optimization guarantee that we can find the optimal value of $X$ by reparameterizing $X$ as $UV^T$ and not introduce additional poor local minima if we solve a problem of form (13). However, solutions to (13) simply guarantee that $X$ can be factorized as $X = UV^T$ for some $(U, V)$ with the desired properties, but it does not guarantee that the $(U, V)$ that we obtained by the reparameterization will satisfy those properties. In this paper we will instead focus on solving problems with form (1) directly using convex relaxations merely as an analysis tool.

### 3 Structured Matrix Factorization Problem

To develop our analysis we will introduce a matrix regularization function, $\Omega_\theta$, that generalizes the $\| \cdot \|_{u,v}$ norm and is similarly defined in the product space but allows one to enforce structure in the factorized space. We will establish basic properties of the proposed regularization function, such as its convexity, and discuss several practical examples.

#### 3.1 Structured Matrix Factorization Regularizers

The proposed matrix regularization function, $\Omega_\theta$, will be constructed from a regularization function $\theta$ on rank-1 matrices, which can be defined as follows:

**Definition 1.** A function $\theta : \mathbb{R}^D \times \mathbb{R}^N \to \mathbb{R} \cup [\infty]$ is said to be a rank-1 regularizer if

1) $\theta(u, v)$ is positively homogeneous with degree 2, i.e., $\theta(\alpha u, \alpha v) = \alpha^2 \theta(u, v)$ for all $\alpha \geq 0, (u, v)$.

2) $\theta(u, v)$ is positive semi-definite, i.e., $\theta(0, 0) = 0$ and $\theta(u, v) \geq 0 \forall (u, v)$.

3) For any sequence $(u_n, v_n)$ such that $\|u_nv_n^T\| \to \infty$ we have that $\theta(u_n, v_n) \to \infty$.

The first two properties of the definition are straightforward and their necessity will become apparent when we derive properties of $\Omega_\theta$. The final property is necessary to ensure that $\theta$ is well defined as a regularizer for rank-1 matrices by ensuring that the rank-1 level sets, $\{uv^T : \theta(u, v) \leq k, k > 0\}$, are compact. Note also that these properties imply that $\theta(u, v) > 0$ for all $uv^T \neq 0$.

These three properties define a general set of requirements that are satisfied by a very wide range of rank-1 regularizers (see §3.2 for specific examples of regularizers that
can be used for well known problems). While we will prove our theoretical results using this general definition of a rank-1 regularizer, later, when discussing specific algorithms that can be used to solve structured matrix factorization problems in practice, we will require that \( \theta(u, v) \) satisfies a few additional requirements.

Using the notion of a rank-1 regularizer, we now define a regularization function on matrices of arbitrary rank:

**Definition 2.** Given a rank-1 regularizer \( \theta : \mathbb{R}^D \times \mathbb{R}^N \rightarrow \mathbb{R}_+ \cup \mathbb{R}_- \), the matrix factorization regularizer \( \Omega_\theta : \mathbb{R}^{D \times N} \rightarrow \mathbb{R}_+ \cup \mathbb{R}_- \) is defined as

\[
\Omega_\theta(X) = \inf_{r \in \mathbb{N}_+} \inf_{U \in \mathbb{R}^{D \times r}, V \in \mathbb{R}^{N \times r}} \sum_{i=1}^{r} \theta(U_i, V_i) \text{ s.t. } X = UV^T. \tag{14}
\]

We will further say that \( UV^T = X \) is an optimal factorization of \( X \) if \((U, V)\) achieves the infimum in (14).

The function defined in (14) is very closely related to other regularizers that have appeared in the literature. In particular, taking \( \theta(u, v) = ||u||_1 ||v||_1 \), or \( \theta(u, v) = \frac{1}{2}(||u||_2^2 + ||v||_2^2) \) for arbitrary vector norms \( ||\cdot||_u \) and \( ||\cdot||_v \), gives the \( ||\cdot||_u \) norm in (7). Note, however, there is no requirement for \( \theta(u, v) \) to be convex w.r.t. \((u, v)\) or to be composed of norms.

Based on Definition 2, then if \( \theta \) satisfies the requirements from Definition 1, one can show that \( \Omega_\theta \) satisfies the following proposition:

**Proposition 1.** Given a rank-1 regularizer \( \theta \), the matrix factorization regularizer \( \Omega_\theta \) satisfies the following properties:

1. \( \Omega_\theta(0) = 0 \) and \( \Omega_\theta(X) > 0 \forall X \neq 0. \)
2. \( \Omega_\theta(\alpha X) = \alpha \Omega_\theta(X) \forall \alpha \geq 0, \forall X. \)
3. \( \Omega_\theta(X + Z) \leq \Omega_\theta(X) + \Omega_\theta(Z) \forall (X, Z). \)
4. \( \Omega_\theta(X) \) is convex w.r.t. \( X. \)
5. The infimum in (14) is achieved with \( r \leq DN. \)
6. If \( \theta(-u, v) = \theta(u, v) \) or \( \theta(u, -v) = \theta(u, v) \) for all \((u, v)\), then \( \Omega_\theta \) is a norm on \( X. \)
7. The subgradient \( \partial \Omega_\theta(X) \) of \( \Omega_\theta(X) \) is given by:
   \[
   \{ W : (W, X) = \Omega_\theta(X), u^T W v \leq \theta(u, v) \forall (u, v) \}.
   \]
8. Given a factorization \( X = UV^T \), if there exists a matrix \( W \) such that \( \sum_{i=1}^{r} U_i^T W V_i = \sum_{i=1}^{r} \theta(U_i, V_i) \) and \( u^T W v \leq \theta(u, v) \forall (u, v) \), then \( UV^T \) is an optimal factorization of \( X \) and \( W \in \partial \Omega_\theta(X) \).

**Proof.** A full proof of the above Proposition can be found in Appendix C.1 and uses similar arguments to those found in [10]–[12], [14], [16] for related problems. ■

Note that the first 3 properties show that \( \Omega_\theta \) is a gauge function on \( X \) (and further it will be a norm if property 6 is satisfied). While this also implies that \( \Omega_\theta(X) \) must be a convex function of \( X \), note that it can still be very challenging to evaluate or optimize functions involving \( \Omega_\theta \) due to the fact that it requires solving a non-convex optimization problem by definition. However, by exploiting the convexity of \( \Omega_\theta \), we are able to use it to study the optimality conditions of many associated non-convex matrix factorization problems, some of which are discussed next.

### 3.2 Structured Matrix Factorization Problem Examples

The matrix factorization regularizer provides a natural bridge between convex formulations in the product space (6) and non-convex functions in the factorized space (5) due to the fact that \( \Omega_\theta(X) \) is a convex function of \( X \). Also, from the definition (14) one can induce a wide range of properties in \((U, V)\) by an appropriate choice of \( \theta(u, v) \) function. In what follows, we give a number of examples which lead to variants of several structured matrix factorization problems that have been studied previously in the literature.

**Low-Rank:** The first example of note is to relax low-rank constraints into nuclear norm regularized problems. Taking \( \theta(u, v) = \frac{1}{2}(||u||_2^2 + ||v||_2^2) \) gives the well known variational form of the nuclear norm, \( \Omega_\theta(X) = ||X||_*, \) and thus provides a means to solve problems in the factorized space where the size of the factorization gets controlled by regularization. In particular, we have the conversion

\[
\min_{X} \ell(Y, X) + \lambda ||X||_* \iff \min_{r, U, V} \ell(Y, UV^T) + \frac{\lambda}{2} \sum_{i=1}^{r} (||U_i||_2^2 + ||V_i||_2^2) \iff \tag{15}
\]

\[
\min_{r, U, V} \ell(Y, UV^T) + \lambda \sum_{i=1}^{r} ||U_i||_2 ||V_i||_2,
\]

where the \( \iff \) notation implies that solutions to all three objective functions will have identical values at the global minimum and any global minimum w.r.t. \((U, V)\) will give a global minimum w.r.t \( X = UV^T \). While the above equivalence is well known for the nuclear norm [9], [17], the factorization is “unstructured” in the sense that the Euclidean norms do not bias the columns of \( U \) and \( V \) to have any particular properties. Therefore, to find factors with additional structure, such as non-negativity, sparseness, etc., more general \( \theta(u, v) \) functions need to be considered.

**Non-Negative Matrix Factorization:** If we extend the previous example to now add non-negative constraints on \((u, v)\), we get \( \theta(u, v) = \frac{1}{2}(||u||_2^2 + ||v||_2^2) + \delta(u \geq 0) \delta(v \geq 0) \), where \( \delta(C) \) denotes the indicator function for a condition \( C \), which takes value 0 if \( C \) is satisfied and \( \infty \) otherwise. This choice of \( \theta \) acts similarly to the variational form of the nuclear norm in the sense that it limits the number of non-zero columns in \((U, V)\), but it also imposes the constraints that \( U \) and \( V \) must be non-negative. As a result, one gets a convex relaxation of traditional non-negative matrix factorization

\[
\min_{U, V} \ell(Y, UV^T) \text{ s.t. } U \geq 0, V \geq 0 \implies \tag{16}
\]

\[
\min_{r, U, V} \ell(Y, UV^T) + \frac{\lambda}{2} \sum_{i=1}^{r} (||U_i||_2^2 + ||V_i||_2^2) \text{ s.t. } U \geq 0, V \geq 0.
\]

The \( \implies \) notation is meant to imply that the two problems are not strictly equivalent as in the nuclear norm example. The key difference between the two problems is that in the first one the number of columns in \((U, V)\), \( r \), is fixed \textit{a priori}, while in the second one \( r \) is allowed to vary and is adapted to the data via the low-rank regularization induced by the squared Frobenius norms on \((U, V)\).

**Row or Columns Norms:** Taking \( \theta(u, v) = ||u||_1 ||v||_1 \) results in \( \Omega_\theta(X) = \sum_{i=1}^{D} ||(X^T)^i||_1 \), i.e., the sum of the \( ||\cdot||_1 \) norms of the rows of \( X \), while taking \( \theta(u, v) = ||u||_u ||v||_v \)
results in $\Omega_b(X) = \sum_{i=1}^{N} \|X_i\|_u$, i.e., the sum of the $\|\cdot\|_u$ norms of the columns of $X$ [10], [11]. As a result, the regularizer $\Omega_b(X)$ generalizes the $\|X\|_{1,1}$ and $\|X\|_{1,v}$ mixed norms. The reformulations into a factorized form give:

$$\min_X \ell(Y, X) + \lambda \|X\|_{1,v} \Leftrightarrow \min_{U,V} \ell(Y, UV^T) + \lambda \sum_{i=1}^{r} \|U_i\|_1 \|V_i\|_v$$

$$\min_X \ell(Y, X) + \lambda \|X\|_{u,1} \Leftrightarrow \min_{U,V} \ell(Y, UV^T) + \lambda \sum_{i=1}^{r} \|U_i\|_u \|V_i\|_1.$$  

However, the factorization problems in this case are relatively uninteresting as taking either $U$ or $V$ to be the identity (depending on whether the $l_1$ norm is on the columns of $U$ or $V$, respectively) and the other matrix to be $X$ (or $X^T$) results in one of the possible optimal factorizations [11].

**Sparse Dictionary Learning:** Similar to the non-negative matrix factorization case, convex relaxations of sparse dictionary learning can also be obtained by combining $l_2$ norms with sparsity-inducing regularization. For example, taking $\theta(u, v) = \frac{1}{2} (\|u\|^2_2 + \|v\|^2_2 + \gamma \|u\|_{1,v})$ results in a relaxation

$$\min_{U,V} \ell(Y, UV^T) + \lambda \|V\|_1 \text{ s.t. } \|U_i\|_2 = 1 \forall i \implies \min_{r,U,V} \ell(Y, UV^T) + \frac{\lambda}{2} \sum_{i=1}^{r} \|U_i\|_2^2 + \|V_i\|_2^2 + \gamma \|V_i\|_{1,v}^2.$$

which was considered as a potential formulation for sparse dictionary learning in [11], where now the number of atoms in the dictionary is fit to the dataset via the low-rank regularization induced by the Frobenius norms. A similar approach would be to take $\theta(u, v) = \|u\|_2 (\|v\|_2 + \gamma \|v\|_1)$.

**Sparse PCA:** If both the rows and columns of $U$ and $V$ are regularized to be sparse, then one can obtain convex relaxations of sparse PCA [18]. One example of this is to take $\theta(u, v) = \frac{1}{2} (\|u\|^2_2 + \gamma \|u\|_2)$. Alternatively, one can also place constraints on the number of elements in the non-zero support of each column in $(u, v)$ via a rank-1 regularizer of the form $\theta(u, v) = \frac{1}{2} (\|u\|^2_2 + \|v\|_2^2) + \delta(\|u\|_{1,v} \leq k) + \delta(\|v\|_{1,v} \leq \ell)$, where $\delta(x \leq y)$ denotes the number of non-zero elements in $z$. Such a form was analyzed in [19] and gives a relaxation of sparse PCA that regularizes the number of sparse components via the $l_2$ norm, while requiring that a given component have the specified level of sparseness.

**General Structure:** More generally, this theme of using a combination of $l_2$ norms and additional regularization on the factors can be used to model additional forms of structure on the factors. For example one can take $\theta(u, v) = \|u\|_2 \|v\|_2 + \gamma \delta(u, v)$ or $\theta(u, v) = \|u\|^2_2 + \|v\|_2^2 + \gamma \delta(u, v)$ with a function $\delta$ that promotes the desired structure in $U$ and $V$ provided that $\theta(u, v)$ satisfies the necessary properties in the definition of a rank-1 regularizer. Additional example problems can be found in [12], [13].

**Symmetric Factorizations:** Assuming that $X$ is a square matrix, it is also possible to learn symmetrical formulations with this framework, as the indicator function $\delta(u = v)$ that requires $u$ and $v$ to be equal is also positively homogeneous. As a result, one can use regularization such as $\theta(u, v) = \delta(u = v) + \|u\|^2_2$ to learn low-rank symmetrical factorizations of $X$, and add additional regularization to encourage additional structures. For example $\theta(u, v) = \delta(u = v) + \|u\|^2_2 + \|v\|^2_2 + \delta(u \geq 0)$ learns symmetrical factorizations where the factors are required to be non-negative and encouraged to be sparse.

**4 THEORETICAL ANALYSIS**

In this section we provide a theoretical analysis of the link between convex formulations (6), which offer guarantees of global optimality, and factorized formulations (5), which offer additional flexibility in modeling the data structure and recovery of features that can be used in subsequent analysis. Using the matrix factorization regularizer introduced in §3.1, we consider the convex optimization problem

$$\min_{X,Q} \{ F(X, Q) \equiv \ell(Y, X, Q) + \lambda \Omega_b(X) \}. \quad (18)$$

Here the term $Q$ can be used to model additional variables that will not be factorized. For example, in robust PCA (RPCA) [20] the term $Q$ accounts for sparse outlying entries, and a formulation in which the data is corrupted by both large corruptions and Gaussian noise is given by:

$$\min_{X,Q} \{ F_{RPCA}(X, Q) \equiv \frac{1}{2} \|Y - X - Q\|_{F}^2 + \gamma \|Q\|_1 + \lambda \|X\|_u \}. \quad (19)$$

In addition to the convex formulation (18), we will also consider the closely related non-convex factorized formulation

$$\min_{U,V,Q} \{ f(U, V, Q) \equiv \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) \}. \quad (20)$$

Note that in (19) we consider problems where $r$ is held fixed. If we additionally optimize over the factorization size, $r$, in (19), then the problem becomes equivalent to (18), as we shall see. We will assume throughout that $\ell(Y, X, Q)$ is jointly convex w.r.t. $(X, Q)$ and once differentiable w.r.t. $X$. Also, we will use the notation $[r]$ to denote the set $\{1, \ldots, r\}$.

**4.1 Conditions under which Local Minima Are Global**

Given the non-convex optimization problem (19), note from the definition of $\Omega_b(X)$ that for all $(X, U, V)$ such that $X = UV^T$ we must have $\Omega_b(X) \leq \sum_{i=1}^{r} \theta(U_i, V_i)$. This yields a global lower bound between the convex and non-convex objective functions, i.e., for all $(X, U, V)$ such that $X = UV^T$ $F(X, Q) = \ell(Y, X, Q) + \lambda \Omega_b(X) \leq \ell(Y, UV^T, Q) + \lambda \sum_{i=1}^{r} \theta(U_i, V_i) = f(U, V, Q). \quad (20)$

From this, if $\hat{X}$ denotes an optimal solution to the convex problem $\min_{X,Q} F(X, Q)$, then any factorization $UV^T = \hat{X}$ such that $\sum_{i=1}^{r} \theta(U_i, V_i) = \Omega_b(X)$ is also an optimal solution to the non-convex problem $\min_{U,V,Q} f(U, V, Q)$. These properties lead to the following result.

**Theorem 1.** Given a function $\ell(Y, X, Q)$ that is jointly convex in $(X, Q)$ and once differentiable w.r.t. $X$; a rank-1 regularizer $\theta$ that satisfies the conditions in Definition 2; and a constant $\lambda > 0$, local minima $(\hat{U}, \hat{V}, \hat{Q})$ of $f(U, V, Q)$ in (19) are globally optimal if $(U_i, V_i) = (0, 0)$ for some $i \in [r]$. Moreover, $(\hat{X}, \hat{Q}) = (UV^T, \hat{Q})$ is a global minima of $F(X, Q)$ and $\hat{U}V^T$ is an optimal factorization of $\hat{X}$.

**Proof.** Since $F(X, Q)$ provides a global lower bound for $f(U, V, Q)$, the result follows from the fact that local minima
of $f(U, V, Q)$ that satisfy the conditions of the theorem also satisfy the conditions for global optimality of $F(X, Q)$. More specifically, because $F(X, Q)$ is a convex function, we have that $(\hat{X}, \hat{Q})$ is a global minimum of $F(X, Q)$ iff

$$-\frac{1}{\Delta} \nabla_X \ell(Y, \hat{X}, \hat{Q}) \in \partial \Omega_\theta (X) \quad \text{and} \quad 0 \in \partial Q \ell(Y, \hat{X}, \hat{Q}). \quad (21)$$

Since $(\hat{U}, \hat{V}, \hat{Q})$ is a local minimum of $f(U, V, Q)$, it is necessary that $0 \in \partial Q \ell(Y, \hat{U}^T V, \hat{Q})$. Moreover, from the characterization of the subgradient of $\Omega_\theta (X)$ given in Proposition 1, we also have that $-\frac{1}{\Delta} \nabla_X \ell(Y, \hat{X}, \hat{Q}) \in \partial \Omega_\theta (X)$ will be true for $X = \hat{U}^T V$ iff the following conditions are satisfied

$$u^T (\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q})) v \leq \theta(u, v) \quad \forall (u, v) \quad (22)$$

$$\sum_{i=1}^r \hat{U}_i^T (\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q})) \hat{V}_i = \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i). \quad (23)$$

To show (22), recall that the local minimum $(\hat{U}, \hat{V}, \hat{Q})$ is such that one column pair of $(\hat{U}, \hat{V})$ is 0. Assume without loss of generality that the final column pair of $(\hat{U}, \hat{V})$ is 0 and let $U_\epsilon = [\hat{U}_1, \ldots, \hat{U}_{i-1}, e^{1/2} u]$ and $V_\epsilon = [\hat{V}_1, \ldots, \hat{V}_{r-1}, e^{1/2} v]$ for some $\epsilon > 0$ and arbitrary $(u, v)$. Then, due to the fact that $(\hat{U}, \hat{V}, \hat{Q})$ is a local minimum, we have that for all $(u, v)$ there exists $\delta > 0$ such that for all $\epsilon \in (0, \delta)$ we have

$$\ell(Y, U_\epsilon V^T_\epsilon \hat{Q}) + \lambda \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) + \lambda \theta(\epsilon^{1/2} u, \epsilon^{1/2} v) = \ell(Y, \hat{U}^T V, \hat{Q}) + \lambda \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) + \lambda \theta(u, v) \quad \forall (u, v) \quad (24)$$

$$\ell(Y, \hat{U}^T V + \epsilon u u^T, \hat{Q}) + \lambda \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) + \epsilon \lambda \theta(u, v) \geq \lambda \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) + \lambda \theta(u, v) \quad (25)$$

$$\ell(Y, \hat{U}^T V, \hat{Q}) + \lambda \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) \quad (26)$$

where the equivalence between (24) and (25) follows from the positive homogeneity of $\theta$. Rearranging terms, we have

$$\frac{1}{\Delta} [\ell(Y, \hat{U}^T V + \epsilon u u^T, \hat{Q}) - \ell(Y, \hat{U}^T V, \hat{Q})] \leq \theta(u, v). \quad (27)$$

Since $\ell(Y, X, Q)$ is differentiable w.r.t. $X$, and after taking the limit $\epsilon \to 0$, we obtain $\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q}) u u^T \leq \theta(u, v)$ for any $(u, v)$ vector pair, showing (22).

To show (23), let $U_{1\pm} = (1 \pm \epsilon)^{1/2} \hat{U}$ and $V_{1\pm} = (1 \pm \epsilon)^{1/2} \hat{V}$ for some $\epsilon > 0$. Since $(\hat{U}, \hat{V}, \hat{Q})$ is a local minimum, there exists $\delta > 0$ such that for all $\epsilon \in (0, \delta)$ we have

$$\ell(Y, U_{1\pm} V_{1\pm}^T, \hat{Q}) + \lambda \sum_{i=1}^r \theta((1 \pm \epsilon)^{1/2} \hat{U}_i, (1 \pm \epsilon)^{1/2} \hat{V}_i) = \ell(Y, (1 \pm \epsilon)^{1/2} \hat{U}^T V, \hat{Q}) + \lambda ((1 \pm \epsilon)^{1/2} \hat{U}) \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) \quad (28)$$

Rearranging terms gives

$$\frac{1}{\Delta} [\ell(Y, (1 \pm \epsilon) \hat{U}^T V, \hat{Q}) - \ell(Y, \hat{U}^T V, \hat{Q})] \leq \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i),$$

and taking the limit w.r.t. $\epsilon \to 0$ gives

$$\sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i) \leq \frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q}) \hat{U}^T \leq \sum_{i=1}^r \theta(\hat{U}_i, \hat{V}_i),$$

showing (23). The last two statements of the result follow from the discussion before the theorem, together with Proposition 1 part (8).

Note that the above proof provides sufficient conditions to guarantee the global optimality of local minima with specific properties, but in addition it also proves the following sufficient conditions for global optimality of any point.

**Corollary 1.** Given a function $\ell(Y, X, Q)$ that is jointly convex in $(X, Q)$ and once differentiable w.r.t. $X$; a rank-1 regularizer $\theta$ that satisfies the conditions in Definition 2; and a constant $\lambda > 0$, a point $(\hat{U}, \hat{V}, \hat{Q})$ is a global minimum of $f(U, V, Q)$ in (19) if it satisfies the conditions

1) $0 \in \partial Q \ell(Y, \hat{U}^T V, \hat{Q})$
2) $\hat{U}^T (\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q})) \hat{V}_i = \theta(\hat{U}_i, \hat{V}_i) \forall i \in [r]$
3) $u^T (\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q})) v \leq \theta(u, v) \forall (u, v)$.

Condition 1 is easy to verify, as one can hold $(U, V)$ constant and solve a convex optimization problem for $Q$. Likewise, condition 2 is simple to test (note that the condition is equivalent to (23) due to the bound in (22)), and if a $(\hat{U}_i, \hat{V}_i)$ pair exists which does not satisfy the equality, then one can decrease the objective function by scaling $(\hat{U}_i, \hat{V}_i)$ by a non-negative constant. Further, for many problems, it is possible to show that points that satisfy first-order optimality will satisfy conditions 1 and 2, such as in the following result.

**Proposition 2.** Given a function $\ell(Y, X, Q)$ that is jointly convex in $(X, Q)$ and once differentiable w.r.t. $X$; a constant $\lambda > 0$; and two gauge functions $(\sigma_u(u), \sigma_v(v))$, then for $\theta(u, v) = \sigma_u(u) \sigma_v(v)$ or $\theta(u, v) = \frac{1}{2} (\sigma_u(u)^2 + \sigma_v(v)^2)$, any first-order optimal point $(\hat{U}, \hat{V}, \hat{Q})$ of $f(U, V, Q)$ in (19) satisfies conditions 1-2 of Corollary 1.

**Proof.** See Appendix C.2.

As many optimization algorithms can guarantee convergence to first-order optimal points, from the above result and discussion it is clear that the primary challenge in verifying if a given point is globally optimal is to test if condition 3 of Corollary 1 holds true. This is known as the polar problem and is discussed in detail next.

### 4.2 Ensuring Local Minimality via the Polar Problem

Note that, because the optimization problem in (19) is non-convex, first-order optimality is not sufficient to guarantee a local minimum. Thus, to apply the results from Section 4.1 in practice one needs to verify that condition 3 from Corollary 1 is satisfied. This problem is known as the polar problem and generalizes the concept of a dual norm. In particular, the proof of Proposition 1 shows that the polar function $\Omega_\theta^*(Z) \equiv \sup_{X} \langle Z, X \rangle$ s.t. $\Omega_\theta(X) \leq 1$ for a given matrix factorization regularizer $\Omega_\theta(X)$ can be computed as

$$\Omega_\theta^*(Z) = \sup_{u, v} u^T Z v \quad \text{s.t.} \quad \theta(u, v) \leq 1. \quad (29)$$

Therefore, condition 3 of Corollary 1 is equivalent to $\Omega_\theta^*(\frac{1}{\Delta} \nabla_X \ell(Y, \hat{U}^T V, \hat{Q})) \leq 1$. Note that the difficulty of solving the polar problem heavily depends on the particular choice of the $\theta$ function. For example for $\theta(u, v) = \|u\|_1 \|v\|_1$ the polar problem reduces to simply finding the largest entry of $Z$ in absolute value, while for $\theta(u, v) = \|u\|_\infty \|v\|_\infty$ solving the polar problem is known to be NP-hard [21].
While for general $\theta(u, v)$ functions it is not necessarily known how to efficiently solve the polar problem, given a point $(\tilde{U}, \tilde{V}, \tilde{Q})$ that satisfies conditions 1 and 2 of Corollary 1, the value of the polar problem solution at a given point and how closely the polar problem can be approximated provide a bound on how far a particular point is from being globally optimal. The bound is based on the following result.

**Proposition 3.** Given a function $\ell(Y, X, Q)$ that is lower-semicontinuous, jointly convex in $(X, Q)$, and once differentiable w.r.t. $X$; a rank-1 regularizer $\theta$ that satisfies the conditions in Definition 2; and a constant $\lambda > 0$, for any point $(\tilde{U}, \tilde{V}, \tilde{Q})$ that satisfies conditions 1 and 2 of Corollary 1, we have the following bound

$$f(\tilde{U}, \tilde{V}, \tilde{Q}) - F(X, Q) \leq \lambda \Omega_\theta(\tilde{X})[\Omega_\theta(\frac{1}{\lambda} \nabla_X \ell(Y, \tilde{U} \tilde{V}^T, \tilde{Q})) - 1] - \frac{m_X}{2} \|\tilde{U} \tilde{V}^T - \tilde{X}\|^2_F - \frac{m_Q}{2} \|\tilde{Q} - \tilde{Q}\|^2_F,$$

where $(\tilde{X}, \tilde{Q})$ denotes a global minimizer of $F(X, Q)$ in (18), and $m_X \geq 0$ and $m_Q \geq 0$ denote the constants of strong-convexity of $\ell$ w.r.t. $X$ and $Q$, respectively (note that both $m$ constants can be 0 if $\ell$ is not strongly convex).

**Proof.** See Appendix C.3. 

There are a few interpretations one can draw from the above proposition. First, if $\tilde{X} = 0$ is a solution to $\min_{X,Q} F(X, Q)$, then the only $(U, V)$ pair that will satisfy conditions 1 and 2 of Corollary 1 is a global optimum (and if $m_X > 0$ then the solution is the unique solution $UV^T = 0$). Second, for $\tilde{X} \neq 0$ recall that $\Omega_\theta(\tilde{X}) > 0$ and $\Omega_\theta(\frac{1}{\lambda} \nabla_X \ell(Y, \tilde{U} \tilde{V}^T, \tilde{Q})) \geq 1$, since condition 2 of Corollary 1 is satisfied then the polar is clearly at least equal to 1 by definition of the polar. If the polar is truly greater than 1, then the $[\Omega_\theta(\frac{1}{\lambda} \nabla_X \ell(Y, \tilde{U} \tilde{V}^T, \tilde{Q})) - 1]$ in the above proposition effectively measures the error between the true value of the polar and our lower-bound estimate of the polar. Further, the maximum difference between a first-order optimal point and the global minimum is upper bounded by the value of the polar at that point, and if the loss function $\ell$ is strongly convex, the error in the objective function is decreased further. As a result, if one can guarantee solutions to the polar problem to within a given error level or provide an upper-bound on the polar problem, one can also guarantee solutions that are within a given error level of the global optimum.

A final interpretation of Proposition 3 is to note that condition 3 of Corollary 1 (and the need for an all-zero column in Theorem 1) is essentially a check that the size of the representation (i.e., the number of columns $r$ in $U$ and $V$) is sufficiently large to represent the global optimum. If, instead, we find a local minimum with a smaller representation than the global optimum, $r < \hat{r}$, where $\hat{r}$ denotes the number of columns in the global optimum, then the value of the $[\Omega_\theta(\frac{1}{\lambda} \nabla_X \ell(Y, \tilde{U} \tilde{V}^T, \tilde{Q})) - 1]$ bounds how far from the global minimum we are by using a more compact representation (i.e., using only $r$ instead of $\hat{r}$ columns).

As a concrete example, consider the case where $\theta(u, v) = \|u\|_2 \|v\|_2$. Recall that this choice of $\theta$ gives the nuclear norm, $\Omega_\theta(X) = \|X\|_*$, whose polar function is given by

$$\Omega_\theta(X) = \|X\|_*,$$

where $\sigma_{\max}(Z)$ denotes the largest singular value of $Z$ (and thus $\|\cdot\|_*$ is the spectral norm). In this case, given any first-order optimal point $(\tilde{U}, \tilde{V}, \tilde{Q})$, then Proposition 3 guarantees that the distance of the current point from the global minimum is bounded by $\lambda \|\tilde{X}\|_* |\sigma_{\max}(\frac{1}{\lambda U V^T}) - 1|$. If $(\tilde{U}, \tilde{V}, \tilde{Q})$ is a global minimizer, then the largest singular value term will be equal to 1 (and hence the bound is 0), while if the largest singular value term is greater than 1 this indicates that $(U, V)$ do not have sufficiently many columns to represent the global optimum (or some of the columns are redundant), and the size of the representation should be increased. Further, appending the largest singular vector pair $(u, v)$ to the factorization $U \leftarrow [U \tau u]$ and $V \leftarrow [V \tau v]$ (as this is the vector pair that achieves the supremum of the polar function) will be guaranteed to reduce the objective function for some step size $\tau > 0$, as Theorem 1 showed this will be a descent direction if the polar is larger than 1.

### 4.3 Reachability of Global Minima via Local Descent

Building on the above results, which provide sufficient conditions to guarantee global optimality, we now describe a generic meta-algorithm that is guaranteed to reach the global minimum of problem (19) via local descent, provided the number of columns in $(U, V)$ is allowed to be variable and adapted to the data via regularization. In particular, recall that the condition of Theorem 1 that one column of $(U, V)$ must be entirely 0 is essentially a check that $r$ is sufficiently large to represent the optimal factorization of a global optimum, $\tilde{X}$, of (18). If the condition is not satisfied, then the size of the factorization can be increased by appending a column of all zeros without changing the value of the objective and then proceeding with local descent. This suggests a meta-algorithm of alternating between local descent on (19), followed by incrementing $r$ by appending a column of all zeros. Algorithm 1 outlines the steps of this approach and adds an additional step to remove redundant columns of $(U, V)$ from the factorization (without changing the value of the objective function), and the following result formally proves that the meta-algorithm will find a global minimizer of (18) and (19), with the number of columns in $(U, V)$ never growing larger than the dimensionality of $X$.

**Theorem 2.** Given a function $\ell(Y, X, Q)$ that is jointly convex in $(X, Q)$ and once differentiable w.r.t. $X$; a rank-1 regularizer $\theta$ that satisfies the conditions in Definition 2; and a constant $\lambda > 0$, then Algorithm 1 will monotonically decrease the value of (19) and terminate at a global minimum of (18) and (19) with $r \leq \max\{DN + 1, r_{\text{init}}\}$.

**Proof.** See Appendix C.4. 

Additionally, note that the proof of Theorem 2 provides a formal proof that the error surface of (19) does not contain any local minima which require one to increase the value of the objective function to escape from, provided the number...
of columns are initialized to be sufficiently large as formalized in the following immediate corollary.

**Corollary 2.** Under the conditions in Theorem 2, if \( r > D N \) then from any \((U_0, V_0, Q_0)\) such that \( f(U_0, V_0, Q_0) < \infty \) there must exist a non-increasing path from \((U_0, V_0, Q_0)\) to a global minimizer of \( f(U, V, Q) \).

Finally, we pause to caution that the above results apply to local descent, not necessarily first-order descent (e.g., gradient descent or the algorithm we describe below), and many first-order descent algorithms can typically only guarantee convergence to a critical point but not necessarily a local minimum. As a result the main computational challenge is to find a local descent direction from a critical point, which can be accomplished by finding a \((u, v)\) pair such that \( \theta(u, v) \leq 1 \) and \( u^T \left( \frac{\lambda}{2} \nabla_X \ell(Y, UV^T, Q) \right) v > 1 \). However, to find such a pair (if it exists) in the worst case we would need to be able to solve the polar problem to guarantee global optimality, which can be challenging as discussed above, but note that from Proposition 3, as it becomes harder to find \((u, v)\) pairs that can be used to decrease the objective function (i.e., the true value of the polar function at a critical point moves closer to 1) we are also guaranteed to be closer to the global minimum. Additionally, another practical challenge is that even if a particular local descent algorithm guarantees convergence to a critical point, due to numerics one may never be exactly at a critical point. Nevertheless, the polar problem still provides a bound on the distance to the global optimum and a guaranteed \((u, v)\) descent direction for an arbitrary point.

**Algorithm 1 (Meta-Algorithm)**

```plaintext
input Initialization for variables, \((U_{\text{init}}, V_{\text{init}}, Q_{\text{init}})\) while Not Converged do 
  Local descent to arrive at a local minimum \((\tilde{U}, \tilde{V}, \tilde{Q})\).
  if \((U_i, V_i) = (0, 0)\) for some \( i \in [r] \) then 
    At global optimum, return.
  else if \( \exists \beta \in \mathbb{R}^r \) such that \( \sum_{i=1}^r \beta_i \tilde{U}_i \tilde{V}_i = 0 \) then 
    Scale \( \beta \) so that \( \min_i \beta_i = -1 \).
    Set \((U_i, V_i) \leftarrow ((1 + \beta_i)^{1/2} \tilde{U}_i, (1 + \beta_i)^{1/2} \tilde{V}_i)\).
    \( \text{// We now have an all-zero column in } (U, V). \)
  else 
    Append an all zero column to \((U, V)\).
    \((U, V) \leftarrow ((\tilde{U}, 0), (\tilde{V}, 0)).\)
  end if
end while
```

## 5 Minimization Algorithm

In the previous section we described and analyzed a generic meta-algorithm that can be used to solve the family of problems we consider. However, while the meta-algorithm is useful from an analysis perspective, it still needs to be paired with a computationally practical local descent algorithm to be of use. Here we describe a specific and easily implementable algorithm that applies to a wide range of the problems discussed above that allows one to be guaranteed convergence to a critical point of the non-convex objective function (19).

Before we begin the discussion of the algorithm, note that in addition to the conditions included in Theorem 1, the particular method we present here assumes that the gradients of the loss function \( \ell(Y, UV^T, Q) \) w.r.t. \( U \) and \( w.r.t. V \) (denoted as \( \nabla_U \ell(Y, UV^T, Q) \) and \( \nabla_V \ell(Y, UV^T, Q) \), respectively) are Lipschitz continuous (i.e. the gradient w.r.t. \( U \) is Lipschitz continuous for any fixed value of \( V \) and vice versa). For a variable \( Z \), \( L_Z^2 \) will be used to denote the Lipschitz constant of the gradient w.r.t. \( Z \) with the other variables held fixed at iteration \( k \). Additionally, we will assume that \( \theta(u, v) \) is convex in \( u \) if \( v \) is held fixed and vice versa (but not necessarily jointly convex in \( u \) and \( v \)).

Under these assumptions on \( \ell \) and \( \theta \), the bilinear structure of our objective function (19) gives convex subproblems if we update \( U \) or \( V \) independently while holding the other fixed, making an alternating minimization strategy efficient and easy to implement. Further, we assume that \( \ell(Y, UV^T, Q) = \hat{\ell}(Y, UV^T, Q) + H(Q) \) where \( \hat{\ell}(Y, UV^T, Q) \) is a convex, once differentiable function of \( Q \) with Lipschitz continuous gradient and \( H(Q) \) is convex but possibly non-differentiable.\(^2\)

The updates to our variables are made using accelerated proximal-linear steps similar to the FISTA algorithm, which entails solving a proximal operator of an extrapolated gradient step to update each variable [22], [23]. The general structure of the alternating updates we use is given in Algorithm 2, and the key point is that to update either \( U \), \( V \), or \( Q \) the primary computational burden lies in calculating the gradient of the loss function and then calculating a proximal operator. The structure of the non-differentiable term in (19) allows the proximal operators for \( U \) and \( V \) to be separated into columns, greatly reducing the complexity of calculating the proximal operator and offering the potential for substantial parallelization.

### 5.1 Proximal Operators of Structured Factors

Recall from the introductory discussion that one means to induce general structure in the factorized matrices is to regularize the columns of a factorized matrix with an \( l_2 \) norm, to limit the rank of the solution, plus a general gauge function, to induce specific structure in the factors. For example, potential forms of the rank-1 regularizers could be of the form \( \theta(u, v) = \|u\|_2 \|v\|_2 + \gamma_u \sigma_u(u) \sigma_v(v) \) or \( \theta(u, v) = (\|u\|_2 + \gamma_u \sigma_u(u))(\|v\|_2 + \gamma_v \sigma_v(v)) \), where the \( \sigma_u \) and \( \sigma_v \) gauge functions are chosen to encourage specific properties in \( U \) and \( V \), respectively. In this case, to apply Algorithm 2 we need a way to solve the proximal operator of the \( l_2 \) norm plus a general gauge function. While the proximal operator of the \( l_2 \) norm is simple to calculate, even if the proximal operator of the gauge function is known, in general the proximal operator of the sum of two functions is not necessarily easy to compute or related to the proximal

\(^2\) Note that the assumption that there is a component of the objective that is differentiable w.r.t. \( Q \), \( \hat{\ell} \), is only needed to use the particular update strategy we describe here. In general one could also optimize objective functions that are non-differentiable w.r.t. \( Q \) but which do need to be convex w.r.t. \( Q \) by doing a full minimization w.r.t. \( Q \) at each iteration instead of just a proximal gradient update. See [22] for details.
operators of the individual functions. Fortunately, the following result shows that for the sum of the l2 norm plus a general gauge function, the proximal operator can be solved by sequentially calculating the two proximal operators.

**Proposition 4.** Let \( \sigma_C \) be any gauge function. The proximal operator of \( \psi(x) = \lambda_C(x) + \lambda_2 \|x\|_2 \) is the composition of the proximal operator of the l2 norm and the proximal operator of \( \sigma_C \), i.e., \( \text{prox}_\psi(y) = \text{prox}_{\lambda_2 \|\cdot\|_2} \circ \text{prox}_\lambda_C(y) \).

**Proof.** See Appendix C.5. \( \blacksquare \)

Combining these results with Theorem 1 and our previously discussed points, we now have a potential strategy to search for structured low-rank matrix factorizations as we can guarantee global optimality if we can find a local minimum with an all-zero column in \( U \) and \( V \) (or equivalently a critical point that also satisfies condition 3 of Corollary 1), and the above proposition provides a means to efficiently solve proximal operator problems that one typically encounters in structured factorization formulations. However, recall the critical caveats to note about the optimization problem; namely that first order descent methods as we have presented here are only guaranteed to converge to a critical point, not necessarily a local minimum, and solving the polar problem to guarantee global optimality can be quite challenging for certain classes of problems.

## 6 Applications in Image Processing

In the next two sections, we will explore applications of the proposed structured matrix factorization framework to two image processing problems: spatiotemporal segmentation of neural calcium imaging data and hyperspectral compressed recovery. Such problems are well modeled by low-rank linear models with square loss functions under the assumption that the spatial component of the data has low total variation (TV) and is optionally sparse in the row and/or column space. Specifically, we will consider the following objective

\[
\min_{U,V,Q} \frac{1}{2} \| Y - A(UV^T) - B(Q) \|^2_F + \lambda \sum_i \| U_i \|_u \| V_i \|_v \quad (32)
\]

(possibly s.t. \( U \geq 0, V \geq 0 \))

where \( A(\cdot) \) and \( B(\cdot) \) are linear operators, and the \( \| \cdot \|_u \) and \( \| \cdot \|_v \) norms have the form given by

\[
\begin{align*}
\| x \|_u &= \nu_{u_1} \| x \|_1 + \nu_{u_{TV}} \| x \|_{TV} + \nu_{u_2} \| x \|_2 \\
\| x \|_v &= \nu_{v_1} \| x \|_1 + \nu_{v_{TV}} \| x \|_{TV} + \nu_{v_2} \| x \|_2
\end{align*}
\]

for non-negative scalars \( \nu \). Here, the \( \ell_1 \) and TV terms allow the incorporation of sparsity or spatial coherence, respectively, while the \( \ell_2 \) term limits the rank of the solution though the connection with the variational form of the nuclear norm. Recall that the anisotropic TV of \( x \) is defined as [24]

\[
\| x \|_{TV} \equiv \sum_{i \in N_i} \sum_{j \in N_i} |x_i - x_j|,
\]

where \( N_i \) is the set of pixels in the neighborhood of pixel \( i \).

Further, note that this objective function exactly fits within the proposed framework as we can define a rank-1 regularizer \( \theta(u, v) = \| u \|_u \| v \|_v \), and optionally add indicator functions on \( u \) and/or \( v \) to enforce non-negativity constraints. Additionally, the proximal operators for the proposed regularization function can be easily evaluated as we show in Appendix B.

### 6.1 Neural Calcium Imaging Segmentation

This section demonstrates the applicability of the proposed structured matrix factorization framework to the problem of segmenting calcium imaging data. Calcium imaging is a microscopy technique in neuroscience that records fluorescent images from neurons that have been loaded with either synthetic or genetically encoded fluorescent calcium indicator molecules. When a neuron fires an electrical action potential (or spike), calcium enters the cell and binds to the fluorescent calcium indicator molecules, changing the fluorescence properties of the molecule. By recording movies of
the calcium-induced fluorescent dynamics it is possible to infer the spiking activity from large populations of neurons with single neuron resolution [25].

Specifically, let \( y \in \mathbb{R}^t \) be the fluorescence time series from a single neuron (normalized by the baseline fluorescence) during \( t \) imaging frames. We can infer the neuron’s spiking activity \( x \in \mathbb{R}^t \) (each entry of \( x \) is monotonically related to the number of action potentials of the neuron during that imaging frame) by solving a Lasso like problem:

\[
\hat{x} = \arg \min_{x \geq 0} \frac{1}{2} \| y - Dx \|_2^2 + \lambda \| x \|_1,
\]

where \( D \in \mathbb{R}^{t \times t} \) is a matrix that applies a convolution with a known decaying exponential to model the change in fluorescence resulting from a neural action potential [26]. However, the above model only applies to a single fluorescence time series extracted from a manually segmented data volume, and a significant challenge in neural calcium imaging is that the data can have a significant noise level, making manual segmentation extremely challenging. Additionally, it is also it is possible for two neurons to overlap in the spatial domain if the focal plane of the microscope is thicker than the size of the distinct neural structures in the data, making simultaneous spatiotemporal segmentation necessary. A possible strategy to address these issues would be to extend (36) to estimate spiking activity for the whole data volume via the objective

\[
\hat{X} = \arg \min_{X \geq 0} \frac{1}{2} \| Y - DX \|_F^2 + \lambda \| X \|_1,
\]

where now each column of \( Y \in \mathbb{R}^{t \times p} \) contains the fluorescent time series for a single pixel, the corresponding column of \( \hat{X} \in \mathbb{R}^{t \times p} \) contains the estimated spiking activity for that pixel, and \( \| \cdot \|_1 \) denotes the \( \ell_1 \) norm. However, due to the significant noise often present in the actual data, and the fact that each column of \( \hat{X} \) is estimated with data from a single pixel in the dataset without the benefit of averaging over a spatial area, solving (37) directly typically gives poor results. To address this issue, [27] exploits the knowledge that if two pixels are from the same neural structure they should have identical spiking activities. Therefore, the matrix \( X \) can be well approximated as \( X \approx UV^T = \sum_{i=1}^p U_iV_i^T \), where \( U_i \in \mathbb{R}^t \) is the spiking activity of one neural structure, \( V_i \in \mathbb{R}^t \) is a shape image of the same neural structure, and \( r \ll p \) is the number of neural structures in the data. This suggests adding a low-rank regularization term to (37), which leads to [27]

\[
\hat{X} = \arg \min_{X \geq 0} \frac{1}{2} \| Y - DX \|_F^2 + \lambda \| X \|_1 + \lambda_2 \| X \|_*.
\]

Given \( \hat{X} \), one can estimate the temporal and spatial features \((U, V)\) by applying non-negative matrix factorization to \( \hat{X} \).

While (38) provides a nice model of spiking activity within a dataset, recall from the introduction that solving a factorization problem in the product space (i.e., solving for \( X \)) is somewhat unsatisfactory as it does not provide us with the desired factors. For example, the number of calcium transients from each neuron should be small, hence each column of \( U \) should be sparse. Likewise, each neuron should occupy a small fraction of the image, hence each column of \( V \) should also be sparse. Notably, it can be shown that problem (32) is equivalent to a standard Lasso estimation when both \( U \) and \( V \) are regularized by the \( \ell_1 \) norm [11], while combined \( \ell_1, \ell_2 \) norms of the form (33) and (34) with \( \nu_{\text{TV}} = 0 \) promote solutions that are simultaneously sparse and low rank. Thus, the projective tensor norm (7) can generalize the two prior methods for calcium image processing by providing regularizations that are sparse or simultaneously sparse and low-rank, while also having the advantage of solving for the desired factors directly. Further, by working in the factorized space we can also model additional known structure in the factors. In particular, we extend formulations (37) and (38) by noting that neighboring pixels are likely to be from the same neural structure and thus have identical spiking activity, implying low TV in the spatial domain. We demonstrate the flexible nature of our formulation (32) by using it to process calcium image data with regularizations that are either sparse, simultaneously sparse and low-rank, or simultaneously sparse, low-rank, and with low TV. Additionally, by optimizing (32) to simultaneously estimate temporal spiking activity \( U \) and neuron shape \( V \), with \( A(UV^T) = DUV^T \), we inherently find spatial and temporal features in the data (which are largely non-negative even though we do not explicitly constrain them to be) directly from our optimization without the need for an additional matrix factorization step. Finally, note that the \( B(Q) \) term can be used to fit the background intensity of the pixels by taking \( B(Q) = 1Q^T \) for a vector \( Q \in \mathbb{R}^p \) and if the data exhibits temporal variations in pixel intensities not due to calcium activity, such as from slow movements of the sample or photo-bleaching, this can also be modeled via an appropriate choice of the \( B \) operator. For the experiments presented here the data has been normalized by background intensity, so the \( B(Q) \) term is not used.

**Simulation Data.** We first tested our algorithm on a simulated phantom dataset which was constructed with 19 non-overlapping spatial regions (see Figure 1, left panel) and 5 randomly timed action potentials and corresponding calcium dynamics per region. The phantom was 200 frames of 120x125 images, and the decaying exponentials in \( D \) had a time constant of 1.333 sec with a simulated sampling rate of 10 Hz. Gaussian white noise was added to the modeled calcium signal to produce an SNR of approximately -16dB.

Using this phantom, we used Algorithm 2 to solve the formulation given in (32) with different \( \nu \) parameters for the norms in (33) and (34). In particular, we used just sparse and low-rank regularization by taking \([\nu_{u_1}, \nu_{v_{TV}}, \nu_{u_2}] = [\nu_{u_1}, \nu_{v_{TV}}, \nu_{u_2}] = [1,0,1] \) and \( \lambda = 1.5\sigma \), where \( \sigma \) denotes the standard deviation of the Gaussian noise. Then, to demonstrate the benefit of adding TV regularization in the spatial domain, we used an 8-connected lattice for the TV graph,\(^3\) \([\nu_{u_1}, \nu_{u_{TV}}, \nu_{u_2}] = [1,0,1], \nu_{v_{TV}}, [1,1,1] \) and \( \lambda = 0.4\sigma \). For the sparse + low-rank condition \( U \) was initialized to be the identity matrix. For the experiments that include TV regularization, we again conducted experiments with \( U \) initialized to be the identity, and to study the effects of different initializations, we additionally also performed

\(^3\) The regularization parameters were roughly tuned by hand to produce the best qualitative results for the two experimental conditions (i.e., sparse + low-rank w/wo TV)
More than 10% were merged (note that this step is largely only necessary to combine duplicate components – see footnote 4 – and the results are very insensitive to the choice of the percentage of overlap as any duplicated components had almost identical non-zero supports). Despite the very high noise level, adding SLRTV regularization recovers the true spatial and temporal components with very high accuracy and faithfully reconstructs the true calcium signal. Further, this performance is robust to the choice of initialization, as initializing \( U \) with either the identity or random values still faithfully recovers true spatial and temporal components. Additionally, despite the very different initializations, the relative error between the two final objective values (given as \( |\text{obj}_1 - \text{obj}_2|/\min\{\text{obj}_1, \text{obj}_2\} \), where \( \text{obj}_1 \) and \( \text{obj}_2 \) denote the final objective values for the 2 different initializations) was only \( 3.8833 \times 10^{-5} \).

Note that the differences in the specific components shown in Figure 3 between the two initializations of \( U \) is due to the fact that the structure of the objective function (32) allows for components to be duplicated without changing the value of the objective function. For example, if \( U = [U_1 \ U_2] \) and \( V = [V_1 \ V_2] \), then \( \tilde{U} = [U_1 \ 0.2U_2 \ 0.8U_2] \) and \( \tilde{V} = [V_1 \ V_2 \ V_2] \) will give identical objective function values.

Figure 1. Recovered spatial segmentations from phantom dataset. \textit{Left:} True spatial labels. \textit{Middle:} Spatial labels recovered with sparse + low-rank + TV regularization, with \( U \) initialized as an identity matrix. \textit{Right:} Same as the middle panel but with \( U \) initialized as 50 columns of random values uniformly distributed in \([0, 1]\).

Figure 2. Example reconstructed calcium signal from phantom dataset. The two rows correspond to different example image frames. \textit{From left to right:} Raw data. True calcium signal. Reconstruction with sparse + low-rank regularization. Reconstruction with sparse + low-rank + TV regularization with \( U \) initialized as the identity. Reconstruction with sparse + low-rank + TV regularization with \( U \) initialized as 50 columns of random values uniformly distributed in \([0, 1]\).

Figure 3. Example recovered spatial components from phantom dataset. \textit{Top Left:} First 9 most significant spatial components recovered via Principal Component Analysis (PCA). \textit{Bottom Left:} First 9 most significant spatial components recovered with sparse and low-rank regularization. \textit{Top Right:} First 9 most significant spatial components recovered using sparse, low-rank, and TV regularization, with \( U \) initialized as the identity. \textit{Bottom Right:} Same as the top right panel but with \( U \) initialized as 50 columns of random values uniformly distributed in \([0, 1]\).
expected, the spatial features recovered by sparse regularization alone are highly noisy (Fig. 5, row 2). Adding low-rank regularization improves the recovered spatial features, but the features are still highly pixelated and contain numerous pixels outside of the desired regions (Fig. 5, row 3). Finally, by incorporating TV regularization, our method produces coherent spatial features which are highly similar to the desired manual labellings (Fig. 5, rows 1 and 4), noting again that these features are found directly from the alternating minimization of (32) without the need to solve a secondary matrix factorization. For comparison purposes, the top 5 spatial components recovered via PCA along with example image frames reconstructed using the top 20 principal components are shown in Figure 6. Note that while the PCA spatial components have a rough correspondence to the neural structures in the data, a significant amount of post-processing would be required to recover the segmentation of a specific neural structure from the PCA representation. Likewise, the example image frames recovered via PCA still contain a very large amount of noise.

In the application of our structured matrix factorization algorithm to the in vivo dataset, U was initialized to be 100 uniformly sampled columns from a $599 \times 599$ identity matrix, and V was initialized as $V = 0$, demonstrating the potential to reduce the problem size and achieve good results despite a very trivial initialization. Similar to the phantom experiments, choosing U to be initialized as random variables in $[0,1]$ produced nearly identical results (not shown). The regularization parameters were tuned manually to produce good qualitative performance for each regularization condition, and the specific values of the parameters are given in Table 2.

We conclude by noting that while TV regularization can
improve performance for a segmentation task, it also can cause a dilative effect when reconstructing the estimated calcium signal (e.g., distorting the size of the thin dendritic processes in the left two columns of the example frames in Figure 5). As a result, in a denoising task it might instead be desirable to only impose sparse and low-rank regularization. The fact that we can easily and efficiently adapt our model to account for many different features of the data depending on the desired task highlights the flexible nature and unifying framework of our proposed formulation (32).

6.2 Hyperspectral Compressed Recovery

The second application we considered is recovering a hyperspectral image volume from a set of compressed measurements. Hyperspectral imaging (HSI) is similar to regular digital photography, but instead of recording the intensities of light at just 3 wavelengths (red, green, blue) as in a typical camera, HSI records images for a very large number of wavelengths (typically hundreds or more).

Due to the nature of hyperspectral imaging the data often display a low-rank structure. For example, consider hyperspectral images taken during aerial reconnaissance. If one was given the spectral signatures of various materials in the hyperspectral image volume (trees, roads, buildings, dirt, etc.), as well as the spatial distributions of those materials, then one could construct a matrix \( X \in \mathbb{R}^{N \times P} \), whose \( i \)th column, \( U_i \), contains the spectral signature of the \( i \)th material (recorded at \( t \) wavelengths) along with a matrix \( V \in \mathbb{R}^{P \times T} \) which contains the spatial distribution of the \( i \)th material in its \( s \)th column \( V_s \) (where \( p \) denotes the number of pixels in the image). Then, \( r \) corresponds to the number of materials present in the given HSI volume, and since typically \( r \ll \min(t, p) \) the overall HSI volume can be closely approximated by the low-rank factorization \( Y \approx UV^T \). This fact, combined with the large data sizes typically encountered in HSI applications, has led to a large interest in developing compressed sampling and recovery techniques to compactly collect and reconstruct HSI datasets.

In addition, an HSI volume also displays significant structure in the spatial domain because neighboring pixels are highly likely to correspond to the same material [30]. This combination of low-rank structure along with strong correlation between neighboring pixels in the spatial domain of an HSI dataset led the authors of [31] to propose a combined nuclear norm and TV regularization (NucTV) method to reconstruct HSI volumes from compressed measurements with the form

\[
\min_X \|X\|_* + \lambda \sum_{i=1}^t \|X^i\|_{TV} \quad \text{s.t.} \quad \|Y - AX\|_F^2 \leq \epsilon.
\]

Here \( X \in \mathbb{R}^{N \times P} \) and \( p \) pixels, \( X^i \) denotes the \( i \)th row of \( X \) (or the \( i \)th spectral band), \( Y \in \mathbb{R}^{T \times M} \) contains the observed samples (compressed at a subsampling ratio of \( m/p \)), and \( A(\cdot) \) denotes the compressed sampling operator. To solve (39), [31] implemented a proximal gradient method, which required solving a TV proximal operator for every spectral slice of the data volume in addition to solving the proximal operator of the nuclear norm (singular value thresholding) at every iteration of the algorithm [32]. For the large data volumes typically encountered in HSI, this can require significant computation per iteration.

Here we demonstrate the use of our structured matrix factorization method to perform hyperspectral compressed recovery by optimizing (32), where \( A(\cdot) \) is a compressive sampling function that applies a random-phase spatial convolution at each wavelength [31], [33]. \( U \) contains estimated spectral features, and \( V \) contains estimated spatial abundance features.\(^5\) Compressed recovery experiments were performed on the dataset from [31]\(^6\) at various subsampling ratios and with different levels of sampling noise. We limited the number of columns of \( U \) and \( V \) to 15 (the dataset has 256 \( \times \) 256 pixels and 180 spectral bands), initialized one randomly selected pixel per column of \( V \) to one and all others to zero, and initialized \( U = 0 \).

Figure 7 in Appendix D shows examples of the recovered images at one wavelength (spectral band \( i = 50 \)) for various subsampling ratios and sampling noise levels and Table 3 shows the reconstruction recovery rates \( \|X_{true} - UV^T\|_F / \|X_{true}\|_F \), where \( X_{true} \) denotes the true hyperspectral image volume. We note that even though we optimized over a highly reduced set of variables \( (256 \times 256 \times 15 + 180 \times 15)/256 \times 256 \times 180 \approx 8.4\% \) with very trivial initializations, we were able to achieve reconstruction error rates equivalent to or better than those in [31].\(^\text{7}\) Additionally, by solving the reconstruction in a factorized form, our method offers the potential to perform blind hyperspectral unmixing directly from the compressed samples without ever needing to reconstruct the full dataset, an application extension we leave for future work.

7 Conclusions

We have proposed a highly flexible approach to structured matrix factorization, which allows specific structure to be promoted directly on the factors. While our proposed formulation is not jointly convex in all of the variables, we have shown that under certain criteria a local minimum of

\(^5\) For HSI experiments, we set \( \nu U = \nu V = 0 \) in (33) and (34).

\(^6\) The data used are a subset of the publicly available AVARIS Moffet Field dataset. We made an effort to match the specific spatial area and spectral bands of the data for our experiments to that used in [31] but note that slightly different data may have been used in our study.

\(^7\) The entries for NucTV in Table 3 were adapted from [31, Fig. 1].
the factorization is sufficient to find a global minimum of the product, offering the potential to solve the factorization using a highly reduced set of variables.

Acknowledgments: Work supported by NSF grants 1447822, 1618485, 1618637 and IARPA DIVA D17PC00345.

REFERENCES


