Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. 193

Sparse nonnegative matrix approximation: new formulations and algorithms

Rashish Tandon[1] and Suvrit Sra[1]

Sep. 13, 2010

This Technical Report has been approved by:

Director at MPIK

Postdoc at MPIK

Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. 193

Sparse nonnegative matrix approximation: new formulations and algorithms

Rashish Tandon[1] and Suvrit Sra[1]

Sep. 13, 2010

[1] MPI für biologische Kybernetik

Sparse nonnegative matrix approximation: new formulations and algorithms

Rashish Tandon and Suvrit Sra

Abstract. We introduce several new formulations for sparse nonnegative matrix approximation. Subsequently, we solve these formulations by developing generic algorithms. Further, to help selecting a particular sparse formulation, we briefly discuss the interpretation of each formulation. Finally, preliminary experiments are presented to illustrate the behavior of our formulations and algorithms.

1 Introduction

A wide variety of applications regularly deal with inherently nonnegative data. Not surprisingly, such data often submit to modern data-analytic techniques such as nonnegative matrix approximation (NMA).[1] But while using NMA to analyze the data, a frequent requirement is to either incorporate prior knowledge, or to obtain models that capture additional structure such as sparsity. Indeed, sparsity has become the mainstay among data analysis requirements owing to important practical benefits such as: (i) simpler, more robust models; (ii) countering of overfitting; (iii) discovery of the most-relevant features; and (iv) potentially better modeling of prior information. Based on these benefits, as well as other motivations, sparsity continues to enjoy great interest in various models and applications [6, 20, 24, 26].

However, despite the ever growing importance of sparsity and its widespread use in numerous applications, sparsity constrained NMA has received comparatively little attention [10, 12]. We make up for this deficiency and present two new formulations for sparse NMA, each of which can be further specialized to yield a series of different sparsity constrained or penalized NMA problems. Our formulations are richer than the previously considered sparse NMA models, and therefore offer greater modeling power. Moreover, our formulations are computationally feasible and for solving them we derive generic algorithms based on traditional multiplicative updates as well as proximal-splitting ideas.

To put our work in context we briefly summarize related work below, before proceeding to the technical details.

1.1 Related work

The amount of related work on NMA and its applications has grown too rapidly; the number of references is thus too large to summarize here. For a broad listing of references and survey type summaries we refer the reader to [2, 22]. Below we summarize only the most directly related work.

The idea of sparsity constraints on NMA is natural, but became popular largely after the 2004 paper of Hoyer [10] who introduced a new measure of sparsity based on the ratio of the ℓ_1-norm of a vector to the ℓ_2-norm. This measure allows the user to intuitively tune the sparsity, and it was also used later by Heiler and Schnörr [9], who developed a cone-programming algorithm to solve the associated NMA formulation. While theoretically appealing, this second-order cone-programming formulation can be needlessly expensive. Our sparsity formulations are richer and the associated optimization algorithms simpler, and therefore potentially much faster.

Other authors have considered enforcing sparsity using ℓ_1-norm (trivial choice; no definite reference possible), or even squared ℓ_1-norm penalization [12]. However, to our knowledge, no author has considered sparsity measures for NMA, in the generality treated in this paper. Our sparsity measures are based on the recently popular *mixed-norms*, for which a sizable literature has grown, e.g., [11, 15, 17, 29]; we note, however, that in addition, we also permit mixed-quasinorms.

On the algorithmic front, most of the attention to sparsity based problems has been limited to convex problems, as opposed to this paper where the central problem is non-convex. But the work on convex sparse problems remains highly relevant, because the NMA subproblems that we study, are convex. Thus, we can leverage on the powerful

[1] NMA is better known as nonnegative matrix factorization, or just NMF; we prefer using the word 'approximation' because usually an exact factorization does not exist.

machinery developed in convex-analysis for tackling sparsity penalized subproblems. Amongst other algorithmic approaches, the most directly are two recent papers: [8, 17]. Both these papers present incremental algorithms that can compute matrix approximations; the methods of Mairal et al. [17] can be easily modified to tackle sparsity if needed. Our approach is different, since we focus on batch methods, as well are more general sparsity constraints.

2 NMA: Background

The basic NMA problem is formulated as follows. Let $\boldsymbol{A} \in \mathbb{R}_+^{m\times n}$ be an input matrix. NMA aims to obtain a set of nonnegative "basis" vectors $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k]$ that may be nonnegatively (conically) combined to well-approximate $\boldsymbol{A}$. In symbols, NMA seeks nonnegative $\boldsymbol{b}_j$ and $\boldsymbol{c}_{jk}$ such that

$$\boldsymbol{a}_i \approx \sum_{j=1}^{k} \boldsymbol{b}_j c_{jk}, \qquad c_{jk} \geq 0, \tag{2.1}$$

which using matrix notation may be rewritten as

$$\boldsymbol{A} \approx \boldsymbol{BC}, \qquad \text{where} \quad \boldsymbol{B} \in \mathbb{R}_+^{m\times k}, \ \boldsymbol{C} \in \mathbb{R}_+^{k\times n}. \tag{2.2}$$

Beginning from the formulations (2.1) and (2.2) we may now consider several different alternatives, each of which augments the basic approximation to incorporate other desirable characteristics. For example, suppose that we want the "basis" vectors $\boldsymbol{b}_j$ to be sparse, or perhaps we are using NMA for overcomplete dictionary learning (e.g. [17]), and we wish to have sparse combinations, i.e., each column $\boldsymbol{c}_i$ of $\boldsymbol{C}$ is required to be sparse. Beyond sparsity other constraints may also be imposed on $\boldsymbol{B}$ and $\boldsymbol{C}$, leading to different problems such as clustering and co-clustering—see the discussion in [22, Chapter 5] and [25, Chapter 8] for more examples.

Computing the approximation. To actually compute an approximation of the form (2.2), we need some quality measure that judges how well $\boldsymbol{BC}$ approximates $\boldsymbol{A}$. To that end, we use the *divergence* $D : \mathbb{R}_+^{m\times k} \times \mathbb{R}_+^{k\times n} \to \mathbb{R}_+$, which satisfies the following properties:

1. it is continuously differentiable (at least once) in both variables;
2. it is individually convex in $\boldsymbol{B}$ and $\boldsymbol{C}$;
3. it equals 0, if and only if $\boldsymbol{A} = \boldsymbol{BC}$.

Notice that D is a function of $\boldsymbol{B}$ and $\boldsymbol{C}$, but is parametrized by $\boldsymbol{A}$, so we choose to write $D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C})$. Now the NMA problem may be formulated as

$$\min \quad D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}) \qquad \boldsymbol{B} \geq 0, \quad \boldsymbol{C} \geq 0. \tag{2.3}$$

The most common special instance of (2.3) is given the *least-squares* NMA problem:

$$\min \quad D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}) = \tfrac{1}{2}\|\boldsymbol{A} - \boldsymbol{BC}\|_{\mathrm{F}}^2, \qquad \boldsymbol{B} \geq 0, \quad \boldsymbol{C} \geq 0. \tag{2.4}$$

Many other distance measures have also been used, including those that do not satisfy the convexity requirement—see [5], or the recent book by Cichocki et al. [3]. We restrict our attention to distance measures that satisfy the two basic properties described above, as these properties are crucial to obtaining efficient algorithms. Furthermore, we will motivate most of our discussion by using (2.4) as the quality measure, though most of the results will actually also hold for the general case.

Sparsity. The additional requirements on $\boldsymbol{B}$ and $\boldsymbol{C}$ may be (approximately) enforced by introducing additional constraints or penalty functions. Our focus in this paper is on modeling various forms of sparsity requirements that either $\boldsymbol{B}$ or $\boldsymbol{C}$ should satisfy. We assume that this sparsity requirement is encoded by a function $f(\boldsymbol{B})$ or $g(\boldsymbol{C})$ so that instead of the basic problem (2.3) we consider the more general formulation

$$\begin{aligned} &\min \quad D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}) \\ &f(\boldsymbol{B}) \leq \gamma, \quad g(\boldsymbol{C}) \leq \delta, \qquad \boldsymbol{B} \geq 0, \quad \boldsymbol{C} \geq 0. \end{aligned} \tag{2.5}$$

We also consider the penalized version of (2.5), where one or both of the sparsity constraints appear as penalizers to the objective function. But for simplicity of exposition, without loss of generality we henceforth that only $\boldsymbol{B}$ is required to fulfill sparsity. Our arguments hold with trivial modifications for $\boldsymbol{C}$ also, so we may safely omit them.

The most common sparsity measure is the "counting"-norm, i.e., $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_0$, which counts the number of nonzeros in (support of) matrix $\boldsymbol{B}$. This particular $f(\boldsymbol{B})$ is *not* a norm, and using it as a sparsity measure usually results in a combinatorial optimization problem. Therefore, for practicality we prefer the nearest convex relaxation: $\|\boldsymbol{B}\|_1$ instead, which also favors sparsity. There also exist several non-convex functions that approximate $\|\boldsymbol{B}\|_0$, but we do not study them in this paper.

Beyond mere vector norms, recently there has been a lot of interest in *group-sparsity*, where instead of computing sparsity of individual vectors (or matrices), one computes a joint sparsity measure across a group of (possibly overlapping) vectors [29]. This type of sparsity will form a central part of our new formulations, but before we formally describe it, let us briefly look at some related work.

3 Sparse NMA

Now we are ready to introduce our new sparse nonnegative matrix approximation (SNMA) formulations. But first we need to define some notation.

Definition 1 (Mixed-norm: Vectors)**.** Let $\boldsymbol{w} \in \mathbb{R}^d$ be split[2] into the set $\{\boldsymbol{w}_t : \boldsymbol{w}_t \in \mathbb{R}^{d_t},\ 1 \le t \le n\}$ of (column) vectors. We define the mixed $\ell_{p,q}$-norm (read as p norm of q norms) ($1 \le p, q \le \infty$) for $\boldsymbol{w}$ as

$$\|\boldsymbol{w}\|_{p,q} = \left\| \left[\|\boldsymbol{w}_1\|_q; \|\boldsymbol{w}_2\|_q; \ldots, \|\boldsymbol{w}_n\|_q\right] \right\|_p. \tag{3.1}$$

That is, we compute the ℓ_p-norm of the vector formed by taking ℓ_q-norms of the subvectors $\boldsymbol{w}_t$ ($1 \le t \le n$). We may similarly define *mixed-quasi-norms*, where p or $q = 0$ are allowed.

We alert the reader that some authors (e.g., [1]) use p, q in the order opposite to ours. Another point worth noting is that the definition (3.1) can be further generalized, e.g., if we take the ℓ_{q_t}-norm $\|\boldsymbol{w}_t\|_{q_t}$ of $\boldsymbol{w}_t$. This generalization comes up for example, while studying L_p-nested symmetric distributions [21]. Finally, even more generality is gained when the subvectors $\boldsymbol{w}_t$ are not required to be disjoint [29]. However, for simplicity, we do not focus on the associated overlapping mixed-norms in this paper.

Recall that without loss of generality we may discuss sparsity for $\boldsymbol{B}$ alone; the matrix $\boldsymbol{C}$ can be treated similarly if desired. Assume that the sparsity requirement on matrix $\boldsymbol{B}$ is encoded by a function $f : \mathbb{R}_+^{m\times k} \to \mathbb{R}_+$. We consider both *penalized* and *constrained* versions of SNMA:

Penalized SNMA

Given a matrix $\boldsymbol{A}_{m\times n}$, a natural number $k \le \min(m,n)$, and a parameter $\lambda > 0$, find matrices $\boldsymbol{B}_{m\times k}$ and $\boldsymbol{C}_{k\times n}$ that solve

$$\begin{aligned} \text{minimize} \quad & D(\boldsymbol{A};\ \boldsymbol{B},\boldsymbol{C}) + \lambda f(\boldsymbol{B}) \\ \text{s.t.} \quad & \boldsymbol{B},\ \boldsymbol{C} \ge 0. \end{aligned} \tag{3.2}$$

Constrained SNMA

Given a matrix $\boldsymbol{A}_{m\times n}$, a natural number $k \le \min(m,n)$, and a parameter $\gamma > 0$, find matrices $\boldsymbol{B}_{m\times k}$ and $\boldsymbol{C}_{k\times n}$ that solve

$$\begin{aligned} \text{minimize} \quad & D(\boldsymbol{A};\ \boldsymbol{B},\boldsymbol{C}) \\ \text{subject to} \quad & \boldsymbol{B},\ \boldsymbol{C} \ge 0, \quad f(\boldsymbol{B}) \le \gamma. \end{aligned} \tag{3.3}$$

We specialize (3.2) and (3.3) below by describing several practical choices; our objective is to illustrate how different 'kinds' of sparsity can be modeled by different choices of norms for $f(\boldsymbol{B})$. Finally, we also look at a possibility for $f(\boldsymbol{B})$ which seeks to promote sparsity in $\boldsymbol{B}$ while maximizing diversity across its columns.

3.1 Overall sparsity

The function

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,0} = \sum\nolimits_i \|\boldsymbol{b}^i\|_0,$$

[2] For computational efficiency we restrict our attention where the splitting *partitions* the vector $\boldsymbol{w}$ into subvectors; though generally overlapping groups may also be considered.

promotes overall sparsity in $\boldsymbol{B}$, since it exactly counts the total number of non-zero elements in $\boldsymbol{B}$. Notice that $\|\boldsymbol{B}\|_{1,0} = \|\boldsymbol{B}^T\|_{1,0}$, which tells us that this function does not distinguish between rows and columns.

A relaxation to the $\ell_{1,0}$-quasi-norm is the $\ell_{1,1}$-norm:

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,1} = \sum\nolimits_{ij} |b_{ij}|.$$

This is the familiar (vector) ℓ_1-norm, and is known to induce sparsity by shrinking small elements of $\boldsymbol{B}$ toward 0. The extent of sparsity attained is, however, harder to control, though because it is a separable, convex function, the optimization burden is less demanding. Here too, notice that $\|\boldsymbol{B}\|_{1,1} = \|\boldsymbol{B}^T\|_{1,1}$, which means that this norm promotes overall sparsity without distinguishing between rows and columns.

Another possible choice for promoting overall sparsity in $\boldsymbol{B}$ is the $\ell_{1,2}$-norm

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,2} = \sum\nolimits_{i} \|\boldsymbol{b}^i\|_2,$$

where $\boldsymbol{b}^i$ is the i-th row of $\boldsymbol{B}$. Observe that this norm favors making the ℓ_2-norm of each row as small as possible, rather than merely focusing on individual elements. Thus, even though this choice does promote overall sparsity, there is a relatively greater stress on rows. Note that in this case, $\|\boldsymbol{B}\|_{1,2} \neq \|\boldsymbol{B}^T\|_{1,2}$ in general.

3.2 Group sparsity: Maximum Row-wise Budget

The penalty function

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,0} = \max_{1 \leq i \leq m} \left\{ \|\boldsymbol{b}^i\|_0 \right\},$$

counts the maximum number of non-zero elements over all the rows. Thus, with this choice of $f(\boldsymbol{B})$, we can impose an upper bound on the maximum number of non-zero elements that any row of $\boldsymbol{B}$ is allowed to have. If needed, one could also subject the columns of $\boldsymbol{B}$ to a similar constraint by using $f(\boldsymbol{B}) = \|\boldsymbol{B}^T\|_{\infty,0}$. The great benefit of this function is its explicit ability to enforce a "sparsity-budget" on either the rows or columns of $\boldsymbol{B}$.

One disadvantage of the above choice of $f(\boldsymbol{B})$ is its non-convexity. We can instead use the convex relaxation

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,1} = \max_{1 \leq i \leq m} \left\{ \|\boldsymbol{b}^i\|_1 \right\},$$

which is a natural relaxation to the $\ell_{\infty,0}$-quasi-norm. This choice also imposes a maximum budget on the rows (or columns), but now in terms of the ℓ_1-norm.

3.3 Group sparsity: Zeroing out insignificant rows

Observe that $\|\boldsymbol{B}\|_{0,0} = \|\boldsymbol{B}\|_{0,1} = \|\boldsymbol{B}\|_{0,\infty}$ counts the total number of *non-zero rows* in $\boldsymbol{B}$. Therefore, the penalty function $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{0,0}$ would have a tendency to favor setting the less important rows in $\boldsymbol{B}$ to zero. When the rows of $\boldsymbol{B}$ correspond to features, this choice of $f(\boldsymbol{B})$ allows performing *feature selection* by eliminating less significant features from the matrix $\boldsymbol{B}$.

Since the above choice of $f(\boldsymbol{B})$ is non-convex, we might prefer to use

$$f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,\infty} = \sum\nolimits_{i} \|\boldsymbol{b}^i\|_\infty$$

instead, which may be viewed as a convex relaxation to the $\ell_{0,0}$-quasinorm. This choice of f favors reducing ℓ_∞-norms of the rows of $\boldsymbol{B}$; if for some row $\boldsymbol{b}^i$, the norm $\|\boldsymbol{b}^i\|_\infty$ is set to 0, then the entire row must be zero, thereby achieving the aforementioned feature selection effect.

3.4 Sparsity with Diversity

Sparsity with diversity seeks columns of $\boldsymbol{B}$ that have non-zero elements at different locations in each column. For example, $\boldsymbol{B}$ with orthogonal columns is maximally diverse, though not necessarily with sparse columns.

Two simple functions to enforce diversity could be

$$f(\boldsymbol{B}) = \|\boldsymbol{B}^T\boldsymbol{B} - \mathrm{Diag}(\boldsymbol{B}^T\boldsymbol{B})\|_{\mathrm{F}}^2, \quad \text{or} \quad f(\boldsymbol{B}) = \|\boldsymbol{B}^T\boldsymbol{B} - \boldsymbol{I}\|_{\mathrm{F}}^2.$$

An advantage of using the latter as compared to the former is that the former permits solutions where the inner product of a columns with itself can be close to zero, while the latter favors where this inner product is close to one. Both these diversity promoting functions can be combined with any of the abovementioned sparsity measures to jointly enforce sparsity with diversity. We note, however, that using these diversity measures seems to be harder for the constrained formulation (3.3), so we shall only look at solving (3.2) with these measures.

4 Sparse NMA: Algorithms

The NMA problem is a difficult non-convex optimization problem. Indeed, Vavasis [28] showed that it is NP-Hard to find the global optimum for NMA, and so most algorithms can at best hope to guarantee locally optimum solutions. Our Sparse NMA formulations that utilize a true norm (like $\|\boldsymbol{B}\|_{1,1}, \|\boldsymbol{B}\|_{1,2}, \|\boldsymbol{B}\|_{1,\infty}, \|\boldsymbol{B}\|_{\infty,1}$) can be reduced to the standard NMA problem and vice-versa. Thus, our SNMA formulations are essentially equivalent to the basic NMA problem. The NMA problem can also be reduced to our SNMA formulations that use quasi-norms (like $\|\boldsymbol{B}\|_{1,0}, \|\boldsymbol{B}\|_{0,0}, \|\boldsymbol{B}\|_{\infty,0}$), showing that these problems are potentially harder than basic NMA. In either case, it follows that it is NP-Hard to find a global optimum for our SNMA formulations as well.

Although non-convex, and NP-Hard, the SNMA problems do have nice structure which allows the use of *alternating-descent* procedures. Here, one iteratively minimizes (or merely descends) with respect to one of the matrices $\boldsymbol{B}$ or $\boldsymbol{C}$, at a time. Such an alternating approach is appealing because upon fixing one of the matrices, the optimization problems becomes convex in the other.

Formally, if $\boldsymbol{B}_t$ and $\boldsymbol{C}_t$ are the matrices at the t-th iteration, then we can obtain $\boldsymbol{B}_{t+1}$ and $\boldsymbol{C}_{t+1}$ as follows:

$$\boldsymbol{B}_{t+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{B}\geq 0} \quad D(\boldsymbol{A};\boldsymbol{B},\boldsymbol{C}_t) + \lambda f(\boldsymbol{B}), \qquad \text{for (3.2)} \tag{4.1}$$

$$\boldsymbol{B}_{t+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{B}\geq 0, f(\boldsymbol{B})\leq\gamma} \quad D(\boldsymbol{A};\boldsymbol{B},\boldsymbol{C}_t), \qquad \text{for (3.3)} \tag{4.2}$$

$$\boldsymbol{C}_{t+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{C}\geq 0} \quad D(\boldsymbol{A};\boldsymbol{B}_{t+1},\boldsymbol{C}). \tag{4.3}$$

Thanks to the *argmin*, descent is guaranteed, i.e.,

$$D(\boldsymbol{A};\boldsymbol{B}_t,\boldsymbol{C}_t) + \lambda f(\boldsymbol{B}_t) \geq D(\boldsymbol{A};\boldsymbol{B}_{t+1},\boldsymbol{C}_{t+1}) + \lambda f(\boldsymbol{B}_{t+1}), \qquad \text{for (3.2)} \tag{4.4}$$

$$D(\boldsymbol{A};\boldsymbol{B}_t,\boldsymbol{C}_t) \geq D(\boldsymbol{A};\boldsymbol{B}_{t+1},\boldsymbol{C}_{t+1}), \qquad \text{for (3.3)} \tag{4.5}$$

Under appropriate assumptions one can then show (e.g., using [7]) that the above procedure converges to a local optimum for the overall problem.

However, in general exact minimization in the subproblems might be either overkill, or perhaps complicated and computationally expensive. In such a case it is more useful to replace the argmin which merely a guaranteed descent. Although doing so destroys the theoretical guarantees of convergence to a local optimum, empirically the approach may still work well. In principle, algorithms performing mere descent can be augmented via appropriate line-searches (that guarantee sufficient descent) to ensure convergence to a stationary point.

Let us now look at methods that can be used to solve the *nonsmooth* optimization problems (4.1) and (4.2). We prefer particularly simple algorithms, which can also be stopped before reaching the minimum, in case a desired level of descent has been achieved.

4.1 Forward-Backward Splitting Algorithms

Recall that Forward-Backward Splitting (FBS) algorithms [4] are iterative procedures for solving the following optimization problem:

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}), \tag{4.6}$$

where $f_1(\boldsymbol{x})$ is a smooth, convex function, while f_2 is convex, but not necessarily smooth. To solve (4.6), FBS algorithms depend on proximity operators, which are defined as:

Definition 2. Let $f_2 : \mathbb{R}^n \to \mathbb{R}$ be a lower semi-continuous convex function. The *proximity-operator* for f_2 is denoted by $\operatorname{prox}_{f_2}$, and its application at a point $\boldsymbol{y}$ is defined by

$$\operatorname{prox}_{f_2}(\boldsymbol{y}) \quad \equiv \quad \operatorname*{argmin}_{\boldsymbol{x}\in\operatorname{dom} f_2} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + f_2(\boldsymbol{x}). \tag{4.7}$$

It can be shown that if $f_1(\boldsymbol{x})$ is a convex differentiable function with a β-Lipschitz continuous gradient ∇f_1, and $f_2(\boldsymbol{x})$ is a convex lower semi-continuous function, then (4.6) admits at least one solution; this solution is characterized by the following fixed point equation (for any $\alpha > 0$) (see [4] for details):

$$\boldsymbol{x} \leftarrow \operatorname{prox}_{\alpha f_2}(\boldsymbol{x} - \alpha\nabla f_1(\boldsymbol{x}))$$

This fixed-point equation suggests the following *forward-backward splitting* procedure

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \theta_t\big(\mathrm{prox}_{\alpha_t f_2}\big(\boldsymbol{x}_t - \alpha_t \nabla f_1(\boldsymbol{x}_t)\big) - \boldsymbol{x}_t\big), \tag{4.8}$$

where $\alpha_t \in [\epsilon, 2/\beta - \epsilon]$ and $\theta_t \in [\epsilon, 1]$ for a fixed $\epsilon \in (0, \min\{1, 1/\beta\})$. It can be shown (see [4]) that the sequence $\boldsymbol{x}_t$ generated via (4.8) converges to a solution of (4.6). A common implementation choice for (4.8) is $\theta_t = 1$, for which the iteration simplifies to

$$\boldsymbol{x}_{t+1} = \mathrm{prox}_{\alpha_t f_2}\big(\boldsymbol{x}_t - \alpha_t \nabla f_1(\boldsymbol{x}_t)\big). \tag{4.9}$$

We use iteration (4.9) to minimize over $\boldsymbol{B}$, noting that a similar choice could also be made for $\boldsymbol{C}$. Moreover, to save time, we need not run (4.9) to convergence, and may accept an intermediate iterate satisfying descent.

4.2 Details for (3.2)

For the penalized formulation (3.2), descending along $\boldsymbol{B}$ requires solving

$$\underset{\boldsymbol{B}\geq 0}{\mathrm{argmin}} \quad D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}_t) + \lambda f(\boldsymbol{B}). \tag{4.10}$$

First, define the indicator function for a set $\mathcal{B}$

$$\mathbb{1}_{\mathcal{B}}(\boldsymbol{B}) = \begin{cases} 0, & \boldsymbol{B} \in \mathcal{B} \\ \infty, & \text{otherwise.} \end{cases} \tag{4.11}$$

Now, with $\mathcal{B} = \mathbb{R}_+^{m\times k}$ (i.e., the nonnegativity constraints $\boldsymbol{B} \geq 0$), we can rewrite (4.10) as,

$$\underset{\boldsymbol{B}}{\mathrm{argmin}} \quad \underbrace{D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}_t)}_{f_1} + \underbrace{\lambda f(\boldsymbol{B}) + \mathbb{1}_{\geq 0}(\boldsymbol{B})}_{f_2}. \tag{4.12}$$

Now we may invoke (4.9) to solve (4.12), using the indicated choices for f_1 and f_2. Notice that $f_2 = \lambda f + \mathbb{1}_{\geq 0}$ is a sum of two functions. Normally, computing the proximity operator for a sum of two functions is not easy. However, below we show how one can compute prox_{f_2} efficiently. Another point worth mentioning is that to invoke (4.9), the divergence $f_1(\boldsymbol{B}) = D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}_t)$ must be a convex, differentiable function with a β-Lipschitz continuous, while f_2 must a convex lower semi-continuous function. However, not all choices of f that we have previously mentioned lead to a lower semi-continuous f_2. But we do not exclude such f, because we can compute the corresponding proximity operators efficiently.

4.3 Details for (3.3)

For the constrained formulation (3.3), descending along $\boldsymbol{B}$ requires solving

$$\underset{\boldsymbol{B}\geq 0, f(\boldsymbol{B})\leq \gamma}{\mathrm{argmin}} \quad D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}_t). \tag{4.13}$$

We move both constraints into the objective, and obtain the problem

$$\underset{\boldsymbol{B}}{\mathrm{argmin}}\ \underbrace{D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C}_t)}_{f_1} + \underbrace{\mathbb{1}_{\geq 0}(\boldsymbol{B}) + \mathbb{1}_{f\leq \gamma}(\boldsymbol{B})}_{f_2}. \tag{4.14}$$

Here, we need to compute $\mathrm{prox}_{f_2} = \mathrm{prox}_{\mathbb{1}_{\geq 0} + \mathbb{1}_{f\leq\gamma}}$, where as alluded to above, not all choices of f are convex and lower semi-continuous, although the proximity operator can still be computed efficiently.

Since we solve both (4.10) and (4.13) via the FBS iteration (4.9), we need to efficiently compute the associated proximity operators. We discuss this computation below.

4.4 Computing the proximity operators

Computing prox_{f_2} requires solving one of the following optimization problems:

$$\begin{aligned} &\underset{\boldsymbol{B}\geq 0}{\mathrm{argmin}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_2^2 + \lambda f(\boldsymbol{B}) \\ &\underset{\boldsymbol{B}\geq 0}{\mathrm{argmin}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_2^2, \quad \text{s.t.} \quad f(\boldsymbol{B}) \leq \gamma. \end{aligned}$$

To simplify solving these optimization problems, we derive two simple lemmas below.

Lemma 3 (Signs)**.** *Let $f : \mathbb{R}^{m\times k} \to \mathbb{R}$ be an elementwise even function, $\boldsymbol{B}_1^* = \operatorname{prox}_{\lambda f(\boldsymbol{B})}(\boldsymbol{B}')$, and $\boldsymbol{B}_2^* = \operatorname{prox} \mathbb{1}_{f\leq\gamma}(\boldsymbol{B}')$. Then, we must have*

$$\left.\begin{array}{rcl} \operatorname{sgn}(\boldsymbol{B}_1^*(i,j)) & = & \operatorname{sgn}(\boldsymbol{B}'(i,j)) \text{ or } 0, \\ \operatorname{sgn}(\boldsymbol{B}_2^*(i,j)) & = & \operatorname{sgn}(\boldsymbol{B}'(i,j)) \text{ or } 0, \end{array}\right\}, \quad i = 1,\ldots,m, \text{and } j = 1,\ldots,k. \tag{4.15}$$

Proof. Since $\boldsymbol{B}_1^* = \operatorname{prox}_{\lambda f}(\boldsymbol{B}')$, it is the optimal solution to

$$\operatorname*{argmin}_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_2^2 + \lambda f(\boldsymbol{B}).$$

To prove the lemma, we show that for all i, j, the inequality $\boldsymbol{B}_1^*(i,j)\boldsymbol{B}'(i,j) \geq 0$ holds. Thus, suppose by way of contradiction that there exist p, q such that $\boldsymbol{B}_1^*(p,q)\boldsymbol{B}'(p,q) < 0$. Then, we can construct $\hat{\boldsymbol{B}}$ such that,

$$\hat{\boldsymbol{B}}(i,j) = \begin{cases} \boldsymbol{B}_1^*(i,j) & \text{for } (i,j) \neq (p,q), \\ -\boldsymbol{B}_1^*(p,q) & \text{otherwise.} \end{cases} \tag{4.16}$$

Since we assumed f to be elementwise even, it holds that $f(\boldsymbol{B}_1^*) = f(\hat{\boldsymbol{B}})$. Now, consider the difference

$$\begin{aligned} &\tfrac{1}{2}\|\boldsymbol{B}_1^* - \boldsymbol{B}'\|_2^2 + \lambda f(\boldsymbol{B}_1^*) - \tfrac{1}{2}\|\hat{\boldsymbol{B}} - \boldsymbol{B}'\|_2^2 - \lambda f(\hat{\boldsymbol{B}}) \\ &\qquad = \tfrac{1}{2}(\boldsymbol{B}_1^*(p,q) - \boldsymbol{B}'(p,q))^2 - \tfrac{1}{2}(\boldsymbol{B}_1^*(p,q) + \boldsymbol{B}'(p,q))^2 \\ &\qquad = -2(\boldsymbol{B}_1^*(p,q)\boldsymbol{B}'(p,q)) > 0, \end{aligned}$$

which contradicts the optimality of $\boldsymbol{B}_1^*$. Similarly, one can also prove the inequality $\boldsymbol{B}_2^*(i,j)\boldsymbol{B}'(i,j) \geq 0$. □

Lemma 4. *Let $f : \mathbb{R}^{m\times k} \to \mathbb{R}$ be an elementwise function for which*

$$f(\boldsymbol{B}_1,\ldots,\underbrace{0}_{i^{th}pos},\ldots,\boldsymbol{B}_{mn}) \leq f(\boldsymbol{B}_1,\ldots,\boldsymbol{B}_i,\ldots,\boldsymbol{B}_{mn}), \quad i \in [mn], \boldsymbol{B}_i \in \mathbb{R}_+, \text{ and } \boldsymbol{B}_j \in \mathbb{R} \text{ for } j \neq i.$$

Then for $\boldsymbol{B}_1^* = \operatorname{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(\boldsymbol{B}')$ and $\boldsymbol{B}_2^* = \operatorname{prox}_{\mathbb{1}_{\geq 0}+\mathbb{1}_{f\leq\gamma}(\boldsymbol{B})}(\boldsymbol{B}')$, we must have:

$$\left.\begin{array}{rcl} \operatorname{sgn}(\boldsymbol{B}_1^*(p,q)) & = & 0 \quad \textit{if} \quad \operatorname{sgn}(\boldsymbol{B}'(p,q)) = -1 \quad \textit{or } 0, \\ \operatorname{sgn}(\boldsymbol{B}_2^*(p,q)) & = & 0 \quad \textit{if} \quad \operatorname{sgn}(\boldsymbol{B}'(p,q)) = -1 \quad \textit{or } 0, \end{array}\right\} \quad p \in [m], q \in [k].$$

Proof. Since $\boldsymbol{B}_1^* = \operatorname{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(\boldsymbol{B}')$, it is the optimal solution to

$$\operatorname*{argmin}_{\boldsymbol{B}\geq 0} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_2^2 + \lambda f(\boldsymbol{B}).$$

We again proceed by contradiction; thus, let there be some p, q such that $\boldsymbol{B}'(p,q) \leq 0$, and $\boldsymbol{B}_1^*(p,q) > 0$. Construct $\hat{\boldsymbol{B}}$ such that,

$$\hat{\boldsymbol{B}}(i,j) = \begin{cases} \boldsymbol{B}_1^*(i,j) & \text{if } (i,j) \neq (p,q), \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\hat{\boldsymbol{B}} \geq 0$ and $f(\boldsymbol{B}_1^*) \geq f(\bar{\boldsymbol{B}})$. Now, consider the difference

$$\begin{aligned} &\tfrac{1}{2}\|\boldsymbol{B}_1^* - \boldsymbol{B}'\|_2^2 + \lambda f(\boldsymbol{B}_1^*) - \tfrac{1}{2}\|\hat{\boldsymbol{B}} - \boldsymbol{B}'\|_2^2 - \lambda f(\hat{\boldsymbol{B}}) \\ &\qquad = \tfrac{1}{2}\left(\boldsymbol{B}_1^*(p,q) - \boldsymbol{B}'(p,q)\right)^2 - \tfrac{1}{2}\left(\boldsymbol{B}'(p,q)\right)^2 + \lambda(f(\boldsymbol{B}_1^*) - f(\hat{-}\boldsymbol{B})) \\ &\qquad = \tfrac{1}{2}\left(\boldsymbol{B}_1^*(p,q)\right)^2 - (\boldsymbol{B}_1^*(p,q)\boldsymbol{B}'(p,q)) + \lambda(f(\boldsymbol{B}_1^*) - f(\hat{\boldsymbol{B}})) > 0, \end{aligned}$$

which contradicts the optimality of $\boldsymbol{B}_1^*$. The proof for $\boldsymbol{B}_2^*$ follows similarly. □

Corollary 5. *Let f be as in Lemma 4, and let $P_+(\boldsymbol{X})$ denote projection onto the nonnegative orthant. Then,*

$$\operatorname{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(\boldsymbol{B}') = \operatorname{prox}_{\lambda f}(P_+(\boldsymbol{B}')), \tag{4.17}$$

$$\operatorname{prox}_{\mathbb{1}_{\geq 0}+\mathbb{1}_{f\leq\gamma}}(\boldsymbol{B}') = \operatorname{prox}_{\mathbb{1}_{f\leq\gamma}}(P_+(\boldsymbol{B}')). \tag{4.18}$$

Proof. By Lemma 4, we see that replacing the negative entries of $\boldsymbol{B}'$ by 0, does not affect $\mathrm{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(\boldsymbol{B}')$. Thus,

$$\mathrm{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(\boldsymbol{B}') = \mathrm{prox}_{\lambda f+\mathbb{1}_{\geq 0}}(P_+(\boldsymbol{B}')).$$

Now, since $P_+(\boldsymbol{B}') \geq 0$, by Lemma 3, it follows that $\mathrm{prox}_{\lambda f}(P_+(\boldsymbol{B}'))$ must have all non-negative entries. Thus, enforcing $\boldsymbol{B} \geq 0$ does not change the solution, and (4.17) follows; a similar argument holds for (4.18). □

All the norms and quasi-norms (as choices for $f(\boldsymbol{B})$) introduced above satisfy the conditions set forth in Corollary 5. Thus, we can compute the corresponding proximity operators with the constraint $\boldsymbol{B} \geq 0$, by merely replacing $\boldsymbol{B}'$ by $P_+(\boldsymbol{B}')$. Let us look at details of specific choices for $f(\boldsymbol{B})$ below.

4.4.1 Proximity and projection for $\ell_{1,0}$-quasinorm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,0}$. Computing $\mathrm{prox}_{\lambda f}(\boldsymbol{B}')$ requires solving

$$\underset{\boldsymbol{B}}{\mathrm{argmin}} \quad \tfrac{1}{2}\|\boldsymbol{B}-\boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{1,0}.$$

This problem has the following simple closed-form solution:

$$\boldsymbol{B}^*(i,j) = \begin{cases} \boldsymbol{B}'(i,j), & \text{if } |\boldsymbol{B}'(i,j)| > \sqrt{2\lambda}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

Thus, for an $m \times k$ matrix $\boldsymbol{B}'$, the proximity operator can be computed in $O(mk)$, i.e., linear time.

The proximity (actually, merely projection) $\mathrm{prox}_{\mathbb{1}_{f\leq\gamma}}(\boldsymbol{B}')$ requires solving

$$\underset{\|\boldsymbol{B}\|_{1,0}\leq\gamma}{\mathrm{argmin}} \quad \tfrac{1}{2}\|\boldsymbol{B}-\boldsymbol{B}'\|_{\mathrm{F}}^2.$$

This problem can be solved by simply picking the top $\lfloor\gamma\rfloor$ elements from $\boldsymbol{B}'$, and using them as the corresponding entries in $\boldsymbol{B}$; the remainder of the elements of $\boldsymbol{B}$ are set to 0. Using an order-statistic algorithm (median of medians procedure), this solution can be computed in $O(mk)$ time.

Remark: Note that $\|\boldsymbol{B}\|_{1,0}$ is not a convex function and $\{\boldsymbol{B}|\|\boldsymbol{B}\|_{1,0} \leq \gamma\}$ is not a convex set. Hence, convergence guarantees of a FBS procedure based on using the corresponding proximity operator do not hold. However, convergence guarantees hold for the relaxed versions of $\|\boldsymbol{B}\|_{1,0}$, namely the norms: $\|\boldsymbol{B}\|_{1,1}$ and $\|\boldsymbol{B}\|_{1,2}$.

4.4.2 Proximity and projection for $\ell_{1,1}$-norm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,1}$. Here we must solve

$$\underset{\boldsymbol{B}}{\mathrm{argmin}} \quad \|\boldsymbol{B}-\boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{1,1}.$$

The solution to this problem is well-known, and is called *soft-thresholding*:

$$\boldsymbol{B}^*(i,j) = \mathrm{sgn}(\boldsymbol{B}'(i,j))\max(|\boldsymbol{B}'(i,j)| - \lambda, 0). \tag{4.20}$$

This operation can obviously be computed in $O(mk)$ time.

The corresponding projection problem, i.e., $\mathrm{prox}_{\mathbb{1}_{f\leq\gamma}}(\boldsymbol{B}')$ requires solving

$$\min_{\|\boldsymbol{B}\|_{1,1}\leq\gamma} \quad \tfrac{1}{2}\|\boldsymbol{B}-\boldsymbol{B}'\|_{\mathrm{F}}^2.$$

This projection is also very well-known, e.g., [13, 16, 18], and can be solved in linear time.

4.4.3 Proximity and projection for $\ell_{1,2}$-norm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,2}$. Here we must solve

$$\min_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B}-\boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{1,2}.$$

Using separability of $\|\cdot\|_{\mathrm{F}}^2$ and $\|\cdot\|_{1,2}$, the above problems can be broken into m independent optimization problems of the form:

$$\min_{\boldsymbol{b}} \quad \|\boldsymbol{b}-\boldsymbol{b}'\|_2^2 + \lambda\|\boldsymbol{b}\|_2,$$

which has the well-known closed-form solution

$$\boldsymbol{b}^* = \max\left\{1 - \lambda/\|\boldsymbol{b}'\|_2, 0\right\} \boldsymbol{b}'. \tag{4.21}$$

Thus, the $\ell_{1,2}$-norm proximity requires only linear $O(mk)$ time.

The corresponding projection problem is

$$\min_{\|\boldsymbol{B}\|_{1,2} \leq \gamma} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2,$$

which can be solved in linear time using an algorithm of van den Berg et al. [27].

4.4.4 Proximity and projection for $\ell_{0,0}$-quasinorm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{0,0}$. Here we must solve

$$\min_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{0,0}.$$

One can obtain the optimal solution by noting that the i-th row $\boldsymbol{b}_i^*$, of the optimal solution $\boldsymbol{B}^*$ is given by

$$\boldsymbol{b}_i^* = \begin{cases} \boldsymbol{b}_i', & \text{if } \|\boldsymbol{b}_i'\|_2 > \sqrt{2\lambda}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

Thus this proximity operator too can be computed in linear time.

The corresponding projection requires solving

$$\min_{\|\boldsymbol{B}\|_{0,0} \leq \gamma} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2.$$

The optimal solution is comprised of the top $\lfloor \gamma \rfloor$ rows (ordered by $\|.\|_2$) of $\boldsymbol{B}'$ copied over into $\boldsymbol{B}$; the other rows of $\boldsymbol{B}$ are set to zero. Again, using a linear-time median of medians algorithm, the top rows can be determined in linear time.

Remark: Due to non-convexity of $\|\boldsymbol{B}\|_{0,0}$, the associated convergence guarantees for FBS schemes do not hold. Convergence does hold, however, with the convex relaxation $\|\boldsymbol{B}\|_{1,\infty}$.

4.4.5 Proximity and projection for $\ell_{1,\infty}$-norm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,\infty}$. Here we must solve

$$\min_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{1,\infty}.$$

This problem can be again decomposed into m independent optimization problems of the form

$$\min_{\boldsymbol{b}} \quad \tfrac{1}{2}\|\boldsymbol{b} - \boldsymbol{b}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{b}\|_{\infty}.$$

Recalling that $\|\boldsymbol{b}\|_1$ is the dual-norm to $\|\boldsymbol{b}\|_\infty$, we can express the optimal solution to the above problem as

$$\boldsymbol{b}^* = \boldsymbol{b}' - \mathrm{prox}_{\mathbb{1}_{\|\cdot\|_1 \leq \lambda}}(\boldsymbol{b}'),$$

which requires merely solving an ℓ_1-norm projection, which can be done in linear time as previously described.

The corresponding projection task is

$$\min_{\|\boldsymbol{B}\|_{1,\infty} \leq \gamma} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2.$$

Quattoni et al. [19] provided an algorithm for performing this projection in $O(mk \log k)$ time; though empirically the techniques described in [23] result in much faster computation.

4.4.6 Proximity and projection for $\ell_{\infty,0}$-quasinorm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,0}$. Here we must solve

$$\min_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{\infty,0}.$$

Let $\boldsymbol{b}_i^{'l}$ represent the l-largest element in the i-th row, $\boldsymbol{b}_i'$ of $\boldsymbol{B}'$. To solve the above problem, we need to find the appropriate rank p, such that

$$\sum_{i=1}^{m}(\boldsymbol{b}_i^{'p})^2 \geq 2\lambda, \quad \text{and} \quad \sum_{i=1}^{m}(\boldsymbol{b}_i^{'p+1})^2 < 2\lambda.$$

Once such a p has been found, the i^{th} row of the optimal solution is obtained by

$$\boldsymbol{b}_i^*(j) = \begin{cases} \boldsymbol{b}_i'(j), & \text{if } \boldsymbol{b}_i'(j) \geq \boldsymbol{b}_i^{'p}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.23}$$

If the rows of the matrix $\boldsymbol{B}'$ are sorted, then the required p can be found in linear time. Thus, sorting dominates the overall runtime, implying an $O(mk \log k)$ runtime.

The corresponding projection task is

$$\min_{\|\boldsymbol{B}\|_{\infty,0}\leq\gamma} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2.$$

This computation decomposes into m independent optimization problems of the form

$$\min_{\|\boldsymbol{b}\|_0\leq\gamma} \quad \tfrac{1}{2}\|\boldsymbol{b} - \boldsymbol{b}'\|_2^2.$$

Each of these problems can be solved easily; the optimal solution is obtained by picking the top $\lfloor\gamma\rfloor$ elements from $\boldsymbol{b}'$, and setting the rest to zero. As before, a median of medians algorithm allows doing this in linear time, thus the overall runtime is $O(mk)$.

Remark: Since $\|\boldsymbol{B}\|_{\infty,0}$ is non-convex, it may be desirable to replace it with its convex relaxation, $\|\boldsymbol{B}\|_{\infty,1}$.

4.4.7 Proximity and projection for $\ell_{\infty,1}$-norm

Let $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,1}$. Here we must solve

$$\min_{\boldsymbol{B}} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{\infty,1}.$$

Since the $\ell_{\infty,1}$-norm is dual to the $\ell_{1,\infty}$-norm, the optimal solution $\boldsymbol{B}^*$ to the above problem can be expressed as

$$\boldsymbol{B}^* = \boldsymbol{B}' - \mathrm{prox}_{\mathbb{1}_{\|\boldsymbol{B}\|_{1,\infty}\leq\lambda}}(\boldsymbol{B}).$$

We already described how to efficiently compute the $\ell_{1,\infty}$-norm projection involved above.

The corresponding projection problem is

$$\min_{\|\boldsymbol{B}\|_{\infty,1}\leq\gamma} \quad \tfrac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}'\|_{\mathrm{F}}^2,$$

which can be decomposed into m independent ℓ_1-norm ball projection problems, each solvable in linear time. Thus, the overall projection can be computed in $O(mk)$ time.

4.5 Alternative: Multiplicative Update Algorithms

A popular alternative for solving NMA subproblems are multiplicative update algorithms. These algorithms ensure descent for the alternating sub-problems over $\boldsymbol{B}$ or $\boldsymbol{C}$; but they do not obtain convergence to the minimum of each separate alternate minimization, as they usually perform only one descent iteration at a time.

When the regularizer $f(\boldsymbol{B}) = 0$, several multiplicative algorithms are known depending on the choice of the divergence measure $D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C})$; the most well-known amongst these are the updates of Lee and Seung [14].

For $D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C})$ being the least squares distance[3], we provide two new update rules that guarantee descent on our penalized formulation (3.2), when $f(\boldsymbol{B})$ is either $\|\boldsymbol{B}\|_{1,1}$ or $\|\boldsymbol{B}\|_{1,2}$. Since we penalize only $\boldsymbol{B}$, the updates for $\boldsymbol{C}$ remain the usual ones.

[3]Although we do not explicitly state the update rules, yet it is fairly straightforward to derive update rules even when $D(\boldsymbol{A}; \boldsymbol{B}, \boldsymbol{C})$ is a Bregman Divergence

4.5.1 Overall sparsity: $\ell_{1,1}$-norm

Here $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,1}$. The subproblem to be considered is

$$\boldsymbol{B}_{t+1} = \underset{\boldsymbol{B}\geq 0}{\operatorname{argmin}} \quad \tfrac{1}{2}\|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{C}_t\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{B}\|_{1,1}. \tag{4.24}$$

We shall show that the following update rule

$$\boldsymbol{B}_{t+1}(i,j) \leftarrow \max\left(0, \boldsymbol{B}_t(i,j)\frac{[\boldsymbol{A}\boldsymbol{C}_t^T]_{ij} - \lambda}{[\boldsymbol{B}_t\boldsymbol{C}_t\boldsymbol{C}_t^T]_{ij}}\right), \tag{4.25}$$

ensures descent on the objective function in (4.24). Note that update (4.25) respects the non-negativity of $\boldsymbol{B}$.

Since (4.24) decomposes over rows of $\boldsymbol{B}$, it suffices to derive the updates for an arbitrary row $\boldsymbol{b}$, i.e., by ensuring descent for

$$\min_{\boldsymbol{b}\geq 0} \quad \tfrac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\boldsymbol{C}\|_2^2 + \lambda\|\boldsymbol{b}\|_1. \tag{4.26}$$

The following proposition derives the update for (4.26).

Proposition 6. *The function $h(\boldsymbol{b}) = \frac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\boldsymbol{C}\|_2^2 + \lambda\|\boldsymbol{b}\|_1$ is non-increasing under the update rule*

$$\boldsymbol{b}(p) \leftarrow \max\Big(0, \boldsymbol{b}(p)\frac{[\boldsymbol{a}\boldsymbol{C}^T]_p - \lambda}{[\boldsymbol{b}\boldsymbol{C}\boldsymbol{C}^T]_p}\Big), \quad p \in [k].$$

Proof. We prove the proposition by illustrating an appropriate auxiliary function technique for h. That is, first we construct a function $g(\boldsymbol{b}, \boldsymbol{b}')$ such that $h(\boldsymbol{b}) \leq g(\boldsymbol{b}, \boldsymbol{b}')$, and $h(\boldsymbol{b}) = g(\boldsymbol{b}, \boldsymbol{b})$; then, we show that the update is obtained by solving

$$\boldsymbol{b}^{t+1} \leftarrow \underset{\boldsymbol{b}}{\operatorname{argmin}} \quad g(\boldsymbol{b}, \boldsymbol{b}^t). \tag{4.27}$$

Following the above steps guarantees descent, as $h(\boldsymbol{b}^t) = g(\boldsymbol{b}^t, \boldsymbol{b}^t) \geq g(\boldsymbol{b}^{t+1}, \boldsymbol{b}^t) \geq h(\boldsymbol{b}^{t+1})$.
Now consider,

$$\begin{aligned} h(\boldsymbol{b}) &= \tfrac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\boldsymbol{C}\|_2^2 + \lambda\|\boldsymbol{b}\|_1, \\ &\leq \tfrac{1}{2}\|\boldsymbol{a}\|_2^2 - \sum\nolimits_i \boldsymbol{a}_i \sum\nolimits_j \boldsymbol{b}_j\boldsymbol{c}_{ij} + \tfrac{1}{2}\sum\nolimits_{ij}\frac{(\boldsymbol{b}_j\boldsymbol{c}_{ij})^2}{\lambda_{ij}} + \lambda\sum\nolimits_j |\boldsymbol{b}_j| = g(\boldsymbol{b}, \boldsymbol{b}'), \end{aligned} \tag{4.28}$$

where $\lambda_{ij} = \boldsymbol{b}_j'\boldsymbol{c}_{ij}/\boldsymbol{b}'\boldsymbol{c}_i$, $\boldsymbol{c}_i$ is the i^{th} column of $\boldsymbol{C}$ and $\boldsymbol{c}_{ij}$ is its j^{th} term , $\boldsymbol{b}_j$ is the j^{th} term of the vector $\boldsymbol{b}$. Observe that $g(\boldsymbol{b}, \boldsymbol{b}) = h(\boldsymbol{b})$ holds too; thus g is a valid auxiliary function.

Now, differentiate g to obtain

$$\begin{aligned} \frac{\partial g}{\partial \boldsymbol{b}_p} &= -\sum\nolimits_i \boldsymbol{a}_i\boldsymbol{c}_{ip} + \sum\nolimits_{ij}\frac{\boldsymbol{b}_p\boldsymbol{c}_{ip}\boldsymbol{b}'\boldsymbol{c}_i}{\boldsymbol{b}_p'} + \lambda\frac{\boldsymbol{b}_p}{|\boldsymbol{b}_p|} \\ &= -[\boldsymbol{a}\boldsymbol{C}^T]_p + \frac{\boldsymbol{b}_p}{\boldsymbol{b}_p'}[\boldsymbol{b}'\boldsymbol{C}\boldsymbol{C}^T]_p + \lambda\frac{\boldsymbol{b}_p}{|\boldsymbol{b}_p|}. \end{aligned} \tag{4.29}$$

We must now solve $\frac{\partial g}{\partial \boldsymbol{b}_p} = 0$. Here we have the following two cases:

1. $[\boldsymbol{a}\boldsymbol{C}^T]_p > \lambda$: A positive root exists, and is given by $\boldsymbol{b}_p = \boldsymbol{b}_p'\left(\frac{[\boldsymbol{a}\boldsymbol{C}^T]_p - \lambda}{[\boldsymbol{b}'\boldsymbol{C}\boldsymbol{C}^T]_p}\right)$. No other roots exist.

2. $[\boldsymbol{a}\boldsymbol{C}^T]_p < \lambda$: No roots exist. However, for $\boldsymbol{b}_p > 0$ we see that $\frac{\partial g}{\partial \boldsymbol{b}_p} > 0$. Thus, decreasing $\boldsymbol{b}_p$ at any positive value will lead to a decrease in g. Also, for $\boldsymbol{b}_p < 0$ we see that $\frac{\partial g}{\partial \boldsymbol{b}_p} < 0$. So, increasing $\boldsymbol{b}_p$ at any of its negative values will lead to a decrease in g. So, setting $\boldsymbol{b}_p = 0$ will lead to the smallest value of g.

Thus, $\boldsymbol{b}(p) = \max\Big(0, \boldsymbol{b}(p)\frac{[\boldsymbol{a}\boldsymbol{C}^T]_p - \lambda}{[\boldsymbol{b}\boldsymbol{C}\boldsymbol{C}^T]_p}\Big)$ corresponds to (4.27), completing the proof. □

4.5.2 Group-sparsity: $\ell_{1,2}$-norm

For $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,2}$, we shall show that the following update

$$\boldsymbol{B}_{t+1}(i,j) = \boldsymbol{B}_t(i,j)\left(\frac{[\boldsymbol{A}\boldsymbol{C}_t^T]_{ij}}{[\boldsymbol{B}_t\boldsymbol{C}\boldsymbol{C}^T]_{ij} + \lambda[\boldsymbol{B}_t]_{ij}\|\boldsymbol{b}_i\|_2^{-1}}\right),$$

ensures descent. As before, we consider the matrix $\boldsymbol{B}$ row-wise; the associated claim is formalized below.

Proposition 7. *The objective function $h(\boldsymbol{b}) = \frac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\boldsymbol{C}\|_2^2 + \lambda\|\boldsymbol{b}\|_2$ is non-increasing under the update rule*

$$\boldsymbol{b}(p) = \boldsymbol{b}(p)\Big(\frac{[\boldsymbol{a}\boldsymbol{C}^T]_p}{[\boldsymbol{b}\boldsymbol{C}\boldsymbol{C}^T]_p + \lambda\boldsymbol{b}(p)\|\boldsymbol{b}\|_2^{-1}}\Big).$$

Proof. Consider the inequality

$$\begin{aligned} h(\boldsymbol{b}) &= \tfrac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\boldsymbol{C}\|_2^2 + \lambda\|\boldsymbol{b}\|_2 \\ &\leq \tfrac{1}{2}\|\boldsymbol{a}\|_2^2 - \sum\nolimits_i \boldsymbol{a}_i\Big(\sum\nolimits_j \boldsymbol{b}_j\boldsymbol{c}_{ij}\Big) + \tfrac{1}{2}\sum_{ij}\frac{(\boldsymbol{b}_j\boldsymbol{c}_{ij})^2}{\lambda_{ij}} + \lambda\sqrt{\sum\nolimits_j \boldsymbol{b}_j^2}, \end{aligned} \tag{4.30}$$

where $\lambda_{ij} = \frac{\boldsymbol{b}_j'\boldsymbol{c}_{ij}}{\boldsymbol{b}'\boldsymbol{c}_i}$ as before. We now bound $\|\boldsymbol{b}\|_2$ by invoking the inequality $\sqrt{x} \leq \frac{\sqrt{y}}{2} + \frac{x}{2\sqrt{y}}$, whereby

$$f(\boldsymbol{b}) \leq \tfrac{1}{2}\|\boldsymbol{a}\|_2^2 - -\sum\nolimits_i \boldsymbol{a}_i\Big(\sum\nolimits_j \boldsymbol{b}_j\boldsymbol{c}_{ij}\Big) + \tfrac{1}{2}\sum_{ij}\frac{(\boldsymbol{b}_j\boldsymbol{c}_{ij})^2}{\lambda_{ij}} + \left(\tfrac{1}{2}\|\boldsymbol{b}'\|_2 + \frac{1}{2\|\boldsymbol{b}'\|_2}\sum\nolimits_j \boldsymbol{b}_j^2\right) = g(\boldsymbol{b}, \boldsymbol{b}'). \tag{4.31}$$

Additionally, one can easily verify that $g(\boldsymbol{b}, \boldsymbol{b}) = f(\boldsymbol{b})$, thus establishing g to be a valid auxiliary function.

Finally, to obtain the update, we first compute the derivative

$$\begin{aligned} \frac{\partial g}{\partial \boldsymbol{b}_p} &= -\sum\nolimits_i \boldsymbol{a}_i\boldsymbol{c}_{ip} + \sum_{ij}\frac{\boldsymbol{b}_p\boldsymbol{c}_{ip}\boldsymbol{b}'\boldsymbol{c}_i}{\boldsymbol{b}_p'} + \lambda\frac{\boldsymbol{b}_p}{\|\boldsymbol{b}'\|_2} \\ &= -[\boldsymbol{a}\boldsymbol{C}^T]_p + \frac{\boldsymbol{b}_p}{\boldsymbol{b}_p'}[\boldsymbol{b}'\boldsymbol{C}\boldsymbol{C}^T]_p + \lambda\frac{\boldsymbol{b}_p}{\|\boldsymbol{b}'\|_2}. \end{aligned}$$

Then, we solve $\frac{\partial g}{\partial \boldsymbol{b}_p} = 0$; this solution immediately yields the desired update. □

5 Extension to Multifactor Matrix Approximation

We shall digress here from our main problem and see that the methods described for solving our problem can be applied to the more general setting of Multifactor Matrix Approximation, wherein a matrix $\boldsymbol{A}$ may be required to be decomposed as a product of more than 2 matrices.

As an example, consider 3-factor decomposition with non-negativity constraints. Here we wish to decompose $\boldsymbol{A}$ approximately into 3 matrices, $\boldsymbol{R}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, with appropriately chosen dimensions, by solving

$$\min_{\boldsymbol{R}\geq 0, \boldsymbol{B}\geq 0, \boldsymbol{C}\geq 0} \quad D(\boldsymbol{A}; \boldsymbol{R}, \boldsymbol{B}, \boldsymbol{C}). \tag{5.1}$$

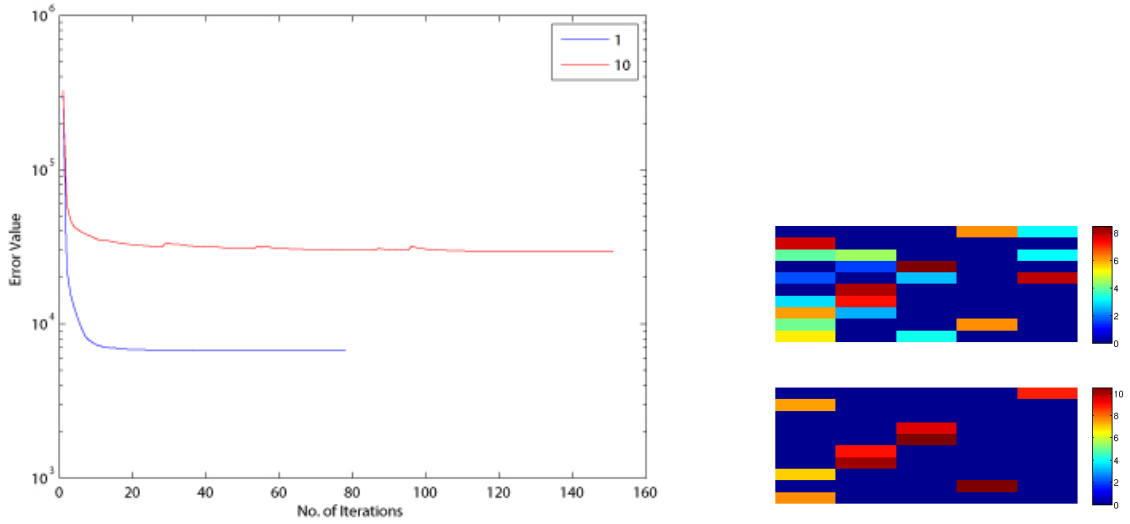
Problem (5.1) can also be solved using the obvious Alternating Minimization approach:

$$\begin{aligned} \boldsymbol{R}_{t+1} &\leftarrow \operatorname*{argmin}_{\boldsymbol{R}\geq 0} \quad D(\boldsymbol{A}; \boldsymbol{R}, \boldsymbol{B}_t, \boldsymbol{C}_t), \\ \boldsymbol{B}_{t+1} &\leftarrow \operatorname*{argmin}_{\boldsymbol{B}\geq 0} \quad D(\boldsymbol{A}; \boldsymbol{R}_{t+1}, \boldsymbol{B}, \boldsymbol{C}_t), \\ \boldsymbol{C}_{t+1} &\leftarrow \operatorname*{argmin}_{\boldsymbol{C}\geq 0} \quad D(\boldsymbol{A}; \boldsymbol{R}_{t+1}, \boldsymbol{B}_{t+1}, \boldsymbol{C}). \end{aligned}$$

From this iteration, it is evident that any sparsity constraints imposed on $\boldsymbol{R}$, $\boldsymbol{B}$ or $\boldsymbol{C}$, may be easily incorporated by suitably modifying the corresponding minimization subproblem. Thus, our previous derivations extend to handle sparsity constrained multifactor approximation too.

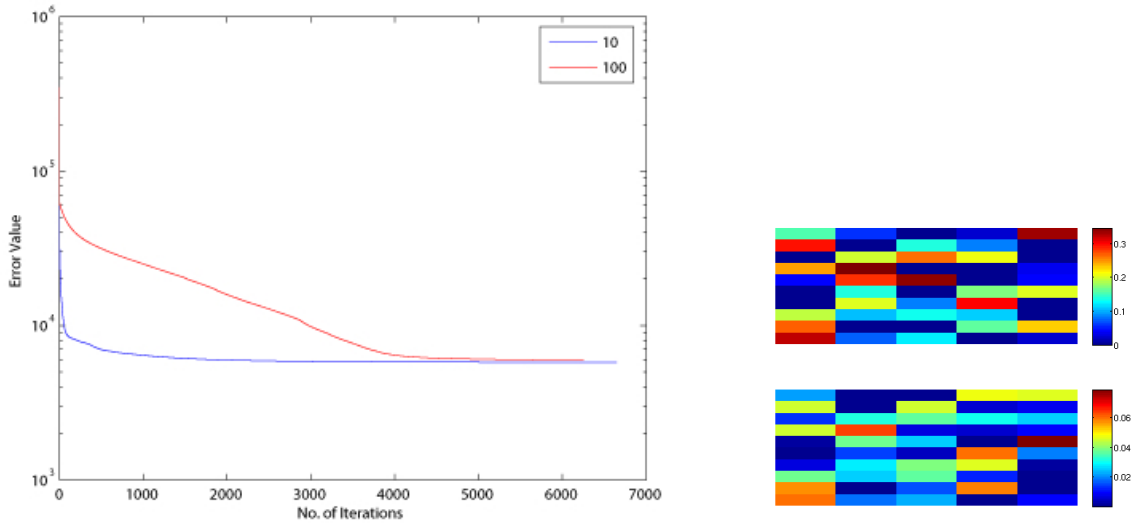
6 Preliminary experiments

We took a random 10×10 matrix $\boldsymbol{A}$, with values in the range of $0 - 100$, and ran our Sparse Formulations (only **Formulation 1**) on this matrix, for a random intial choice of $\boldsymbol{B}$ and $\boldsymbol{C}$. The following figures illustrate the descent in the objective function, for different formulations.



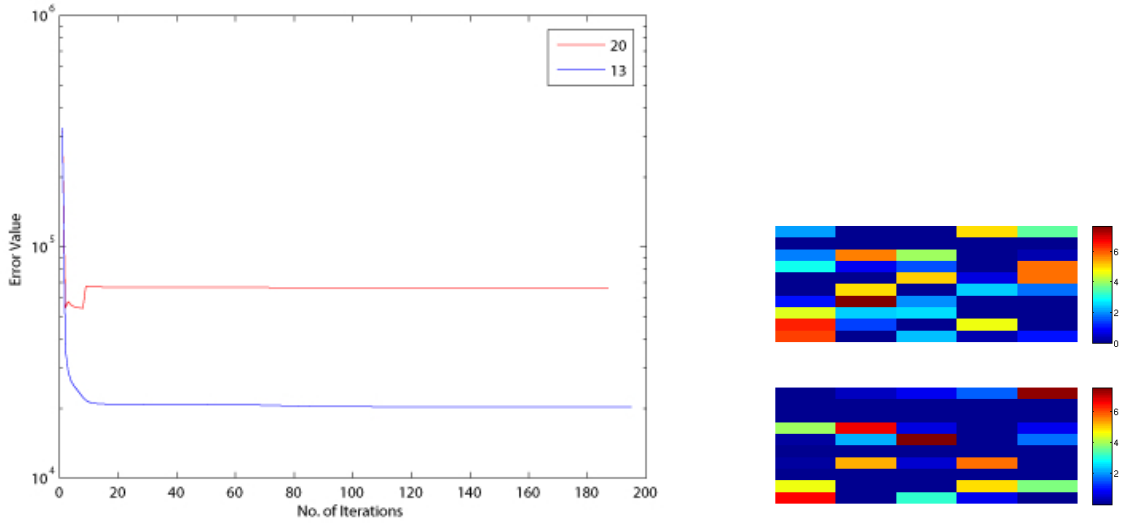
(a) Descent Graph (b) Corresponding $\boldsymbol{B}$'s

Figure 1: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,0}$, with $\lambda = 1$ and 10. The number of non-zeros in the matrix $\boldsymbol{B}$ obtained finally were — 20 for $\lambda = 1$ and 9 for $\lambda = 10$. The increase in error on increasing sparsity is evident here



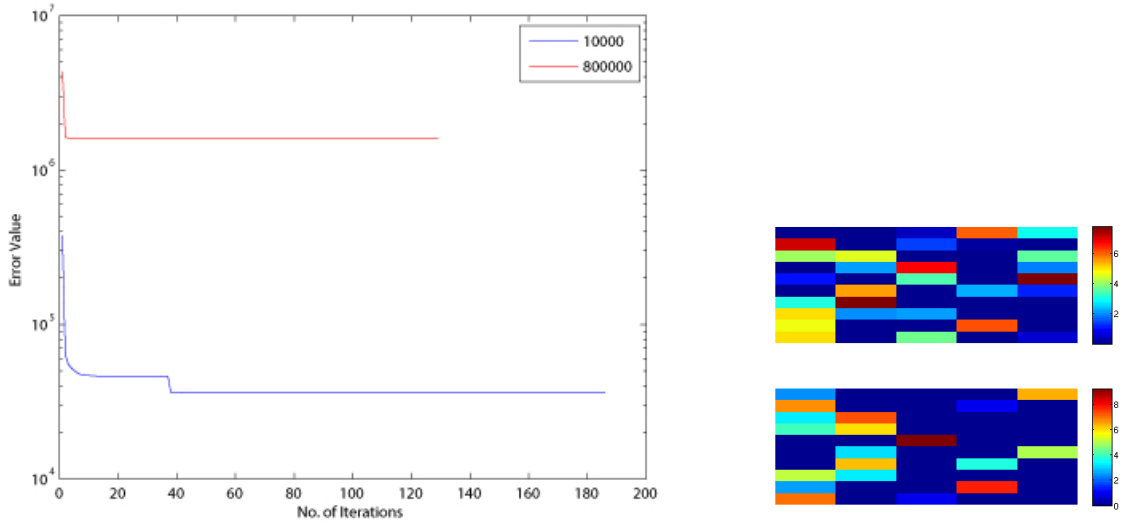
(a) Descent Graph (b) Corresponding $\boldsymbol{B}$'s

Figure 2: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,1}$, with $\lambda = 10$ and 100. The number of non-zeros in the matrix $\boldsymbol{B}$ obtained finally were almost the same in both cases — 36 for $\lambda = 10$ and 42 for $\lambda = 100$. The reason for this ambiguity is that since $\|\frac{\boldsymbol{B}}{\alpha}\|_{1,1} = \frac{1}{\alpha}\|\boldsymbol{B}\|_{1,1}$, the (1,1) norm can be made arbitrarily small by dividing $\boldsymbol{B}$ by an appropriate α. The objective function can be kept the same by multiplying $\boldsymbol{C}$ with the same α. So, increasing λ here seems to result in every element becoming comparitively smaller. One way around this could be to threshold the value of the elements to 0 below a particular point, and then rescale $\boldsymbol{B}$ and $\boldsymbol{C}$ in the end. Another possible way out could be to impose some reasonable sparsity constraints on $\boldsymbol{C}$ as well.



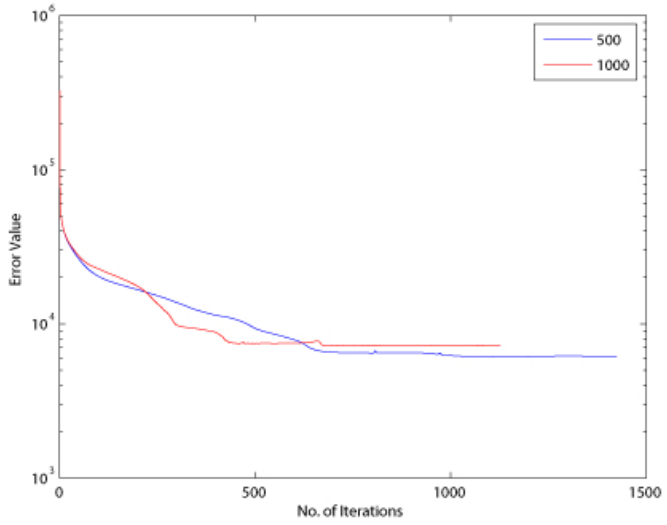
(a) Descent Graph (b) Corresponding $\boldsymbol{B}$'s

Figure 3: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{0,0}$, with $\lambda = 13$ and 20. The number of zero rows in the matrix $\boldsymbol{B}$ obtained finally was — 1 for $\lambda = 13$ and 4 for $\lambda = 20$, with the total number of zero elements being 20 and 28 respectively.

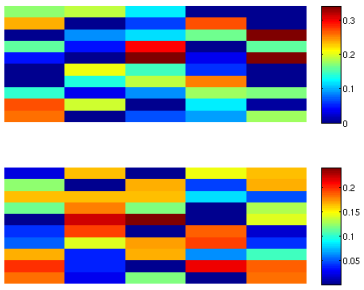


(a) Descent Graph (b) Corresponding $\boldsymbol{B}$'s

Figure 5: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,0}$, with $\lambda = 10000$ and 800000. Every row of $\boldsymbol{B}$ had exactly 2 zeros for $\lambda = 10000$ and exactly 3 zeros for $\lambda = 800000$.

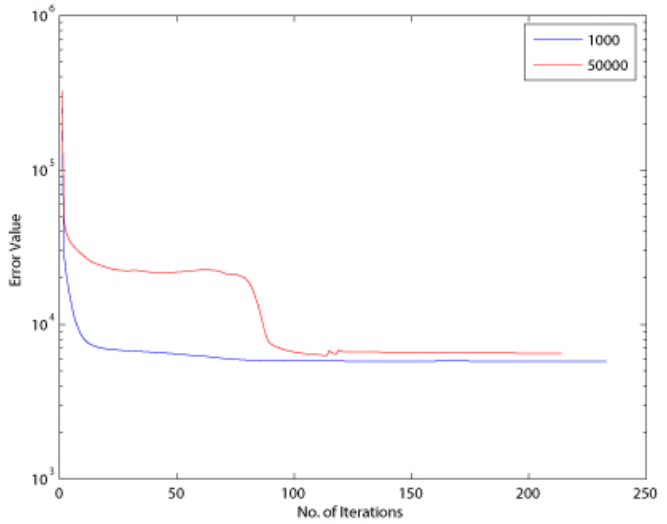


(a) Descent Graph

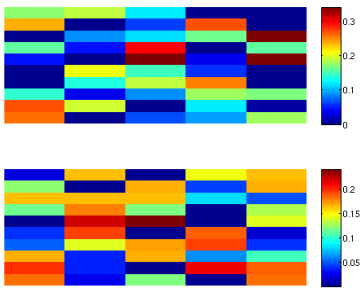


(b) Corresponding $\boldsymbol{B}$'s

Figure 4: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{1,\infty}$, with $\lambda = 500$ and 1000. No complete zero rows were obtained in the matrix $\boldsymbol{B}$ here, but for $\lambda = 1000$, more number of rows had all their elements close to 0 than for $\lambda = 500$. Again, a thresholding scheme might be useful here since we have $\|\frac{\boldsymbol{B}}{\alpha}\|_{1,\infty} = \frac{1}{\alpha}\|\boldsymbol{B}\|_{1,\infty}$

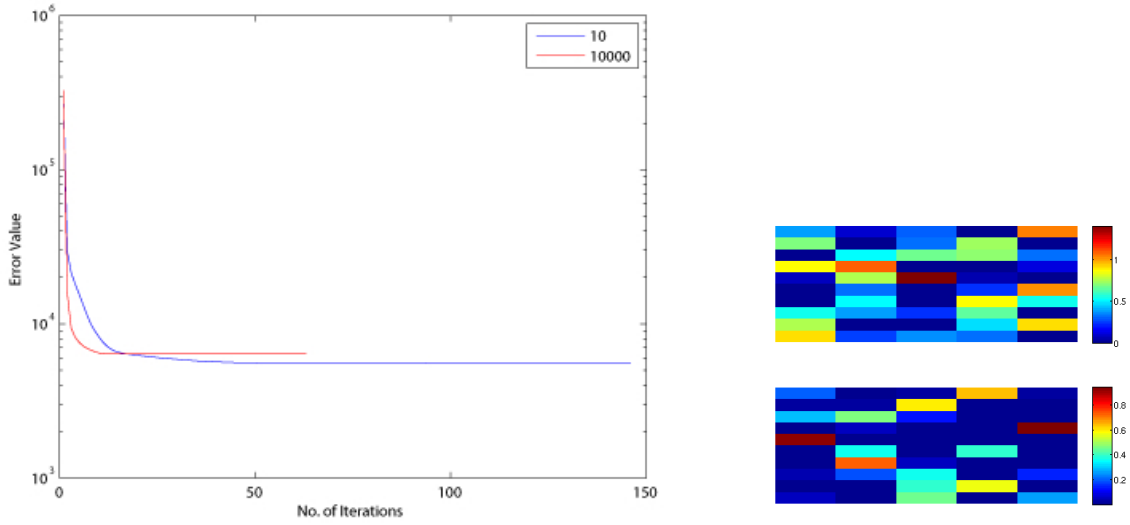


(a) Descent Graph



(b) Corresponding $\boldsymbol{B}$'s

Figure 6: $f(\boldsymbol{B}) = \|\boldsymbol{B}\|_{\infty,1}$, with $\lambda = 1000$ and 50000. In $\boldsymbol{B}$, more number of elements per row were closer to 0 for $\lambda = 50000$ than for $\lambda = 1000$. Imposing a thresholding scheme or sparsity constraints on $\boldsymbol{C}$ will be meaningful here as well since $\|\frac{\boldsymbol{B}}{\alpha}\|_{\infty,1} = \frac{1}{\alpha}\|\boldsymbol{B}\|_{\infty,1}$



(a) Descent Graph

(b) Corresponding $\boldsymbol{B}$'s

Figure 7: $f(\boldsymbol{B}) = \frac{1}{4}\|\boldsymbol{B}^T\boldsymbol{B} - I\|_F^2$, with $\lambda = 10$ and 10000. $f(\boldsymbol{B})$ was 12.5941 for $\lambda = 10$ and 0.1016 for $\lambda = 10000$. Qualitatively as well, $\boldsymbol{B}$ obtained for $\lambda = 10000$ seemed more diverse than that obtained for $\lambda = 10$.

6.1 On a Real Dataset

We have tested out our Sparse NMA formulations on the ORL face image datasets (***http://www.uk.research.att.com/facedatabase.html***), also used by Hoyer for testing sparsity on NMF[10]. The preliminary results are described below. However, more extensive experiments, requiring a careful tuning of the parameters involved, is still to be done.

7 Discussion

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Mach. Learn.*, 73(3):243–272, 2008. 3

[2] M. Berry, M. Browne, A. Langville, P. Pauca, and R. J. Plemmons. Algorithms and applications for approximation nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52:155–173, 2007. 1

[3] A. Cichocki, R. Zdunek, A. H. Phan, and S. ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, Ltd., 2009. 2

[4] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. *arXiv:*, (0912.3522), 2010. 5, 6

[5] I. S. Dhillon and S. Sra. Generalized Nonnegative Matrix Approximations with Bregman Divergences. In *NIPS 18*, Vancouver, Canada, 2006. 2

[6] D. L. Donoho. Compressed sensing. *IEEE Tran. Inf. Theory*, 52(4):1289–1306, 2006. 1

[7] L. Grippo and M. Sciandrone. On The Convergence of The Block Nonlinear Gauss-Seidel Method under Convex Constraints. *Operations Research Letters*, 26:127–136, 2000. 5

[8] S. Harmeling, M. Hirsch, S. Sra, and B. Schölkopf. Online Blind Deconvolution for Astronomy. In *IEEE Int. Conf. on Computational Photography*, Apr. 2009. 2

[9] M. Heiler and C. Schnörr. Learning Sparse Representations by Non-Negative Matrix Factorization and Sequential Cone Programming. *JMLR*, 7:1385–1407, 2006. 1

[10] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. 1, 16

[11] D. Kim, S. Sra, and I. S. Dhillon. A scalable trust-region algorithm with application to mixed-norm regression. In *Int. Conf. Machine Learning (ICML)*, 2010. 1

[12] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007. 1

[13] K. Kiwiel. On linear-time algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications*, 134:549–554, 2007. 8

[14] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In *NIPS*, pages 556–562, 2000. 10

[15] H. Liu, M. Palatucci, and J. Zhang. Blockwise Coordinate Descent Procedures for the Multi-task Lasso, with Applications to Neural Semantic Basis Discovery. In *ICML*, Jun. 2009. 1

[16] N. Maculan and G. G. de Paula. A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$. *Operations Research Letters*, 8(4):219 – 222, 1989. 8

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *JMLR*, 11:10–60, 2010. 1, 2

[18] M. Patriksson. A survey on a classic core problem in operations research. Technical Report 2005:33, Chalmers University of Technology and Göteborg University, Oct. 2005. 8

[19] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for $\ell_{1,\infty}$ regularization. In *ICML*, page 108, 2009. 9

[20] I. Rish and G. Grabarnik. Sparse modeling: ICML 2010 tutorial. Online, Jun. 2010. 1

[21] F. Sinz, E. Simoncelli, and M. Bethge. Hierarchical modeling of local image features through lp-nested symmetric distributions. In *NIPS*, 2009. 3

[22] S. Sra. *Matrix Nearness Problems in Data Mining*. PhD thesis, Univ. of Texas at Austin, Aug. 2007. 1, 2

[23] **S. Sra**. Generalized proximity and projection with norms and mixed-norms. Technical Report 192, Max Planck Institute for Biological Cybernetics, May 2010. 9

[24] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. 1

[25] J. A. Tropp. *Topics in Sparse Approximation*. PhD thesis, Univ. of Texas at Austin, 2004. 2

[26] R. University. Compressive sensing resources. *http://dsp.rice.edu/cs*, 2010. 1

[27] E. van den Berg, M. Schmidt, M. Friedlander, and K. Murphy. Group Sparsity via Linear-Time Projection. *Technical Report, Department of Computer Science, University of British Columbia*, 2008. 9

[28] S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3): 1364–1377, 2009. 5

[29] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, 37(6A):3468–3497, 2009. 1, 3