Accelerated Training for Matrix-norm Regularization: A Boosting Approach

Xinhua Zhang[*], **Yaoliang Yu and Dale Schuurmans**
Department of Computing Science, University of Alberta, Edmonton AB T6G 2E8, Canada
{xinhua2,yaoliang,dale}@cs.ualberta.ca

Abstract

Sparse learning models typically combine a smooth loss with a nonsmooth penalty, such as trace norm. Although recent developments in sparse approximation have offered promising solution methods, current approaches either apply only to matrix-norm *constrained* problems or provide suboptimal convergence rates. In this paper, we propose a boosting method for *regularized* learning that guarantees ϵ accuracy within $O(1/\epsilon)$ iterations. Performance is further accelerated by interlacing boosting with fixed-rank local optimization—exploiting a simpler local objective than previous work. The proposed method yields state-of-the-art performance on large-scale problems. We also demonstrate an application to latent multiview learning for which we provide the first efficient weak-oracle.

1 Introduction

Our focus in this paper is on unsupervised learning problems such as matrix factorization or latent subspace identification. Automatically uncovering latent factors that reveal important structure in data is a longstanding goal of machine learning research. Such an analysis not only provides understanding, it can also facilitate subsequent data storage, retrieval and processing. We focus in particular on coding or dictionary learning problems, where one seeks to decompose a data matrix X into an approximate factorization $\hat{X} = UV$ that minimizes reconstruction error while satisfying other properties like low rank or sparsity in the factors. Since imposing a bound on the rank or number of non-zero elements generally makes the problem intractable, such constraints are usually replaced by carefully designed regularizers that promote low rank or sparse solutions [1–3].

Interestingly, for a variety of dictionary constraints and regularizers, the problem is equivalent to a matrix-norm regularized problem on the reconstruction matrix $\hat{X}$ [1, 4]. One intensively studied example is the trace norm, which corresponds to bounding the Euclidean norm of the code vectors in U while penalizing V via its ℓ_{21} norm. To solve trace norm regularized problems, variational methods that optimize over U and V only guarantee local optimality, while proximal gradient algorithms that operate on $\hat{X}$ [5, 6] can achieve an ϵ accurate (global) solutions in $O(1/\sqrt{\epsilon})$ iterations, but these require singular value thresholding [7] at each iteration, preventing application to large problems.

Recently, remarkable promise has been demonstrated for sparse approximation methods. [8] converts the trace norm problem into an optimization over positive semidefinite (PSD) matrices, then solves the problem via greedy sparse approximation [9, 10]. [11] further generalizes the algorithm from trace norm to gauge functions [12], dispensing with the PSD conversion. However, these schemes turn the regularization into a constraint. Despite their theoretical equivalence, many practical applications require the solution to the regularized problem, *e.g.* when nested in another problem.

In this paper, we optimize the regularized objective directly by reformulating the problem in the framework of ℓ_1 penalized boosting [13, 14], allowing it to be solved with a generalized procedure developed in Section 2. Each iteration of this procedure calls an oracle to find a weak hypothesis

[*] Xinhua Zhang is now at the National ICT Australia (NICTA), Machine Learning Group.

(typically a rank-one matrix) yielding the steepest local reduction of the (unregularized) loss. The associated weight is then determined by accounting for the ℓ_1 regularization. Our first key contribution is to establish that, when the loss is convex and smooth, the procedure finds an ϵ accurate solution within $O(1/\epsilon)$ iterations, and furthermore that the rate can be improved to $O(\log(1/\epsilon))$ when the loss is also strongly convex. To the best of our knowledge, these are the first $O(1/\epsilon)$ objective value rates that have been rigorously established for ℓ_1 *regularized* boosting. [15] considered a similar boosting approach, but required totally corrective updates. In addition, their rates characterize the diminishment of the gradient, and are $O(1/\epsilon^2)$ as opposed to $O(1/\epsilon)$ established here. [9–11, 16–18] establish similar rates, but only for the constrained version of the problem.

We also show in Section 3 how the empirical performance of ℓ_1 penalized boosting can be greatly improved by introducing an auxiliary rank-constrained local-optimization within each iteration. Interlacing rank constrained optimization with sparse updates has been shown effective in semi-definite programming [19–21]. [22] applied the idea to trace norm optimization by factoring the reconstruction matrix into two orthonormal matrices and a positive semi-definite matrix. Unfortunately, this strategy creates a very difficult constrained optimization problem, compelling [22] to resort to manifold techniques. Instead, we use a simpler variational representation of matrix norms that leads to a new local objective that is both *unconstrained* and *smooth*. This allows the application of much simpler and much more efficient solvers to greatly accelerate the overall optimization.

Underlying standard sparse approximation methods is an oracle that *efficiently* selects a weak hypothesis (using boosting terminology). Unfortunately these oracle problems are extremely challenging except in limited cases [3, 11]. Our next major contribution, in Section 4, is to formulate an efficient oracle for latent *multiview* factorization models [2, 4], based on a positive semi-definite relaxation that we prove incurs no gap.

Finally, we point out that our focus in this paper is on the optimization of convex problems that relax the "hard" rank constraint. We do *not* explicitly minimize the rank, which is different from [23].

Notation We use $\gamma_{\mathcal{K}}$ to denote the gauge induced by set $\mathcal{K}$; $\|\cdot\|^*$ to denote the dual norm of $\|\cdot\|$; and $\|\cdot\|_F$, $\|\cdot\|_{\text{tr}}$ and $\|\cdot\|_{\text{sp}}$ to denote the Frobenius norm, trace norm and spectral norm respectively. $\|X\|_{R,1}$ denotes the row-wise norm $\sum_i \|X_{i:}\|_R$, while $\langle X, Y\rangle := \text{tr}(X'Y)$ denotes the inner product. The notation $X \succcurlyeq \mathbf{0}$ will denote positive semi-definite; $X_{:i}$ and $X_{i:}$ stands for the i-th column and i-th row of matrix X; and $\text{diag}\,\{c_i\}$ denotes a diagonal matrix with the (i,i)-th entry c_i.

2 The Boosting Framework with ℓ_1 Regularization

Consider a coding problem where one is presented an $n \times m$ matrix Z, whose columns correspond to m training examples. Our goal is to learn an $n \times k$ dictionary matrix U, consisting of k basis vectors, and a $k \times m$ coefficient matrix V, such that UV approximates Z under some loss $L(UV)$. We suppress the dependence on the data Z throughout the paper. To remove the scaling invariance between U and V, it is customary to restrict the bases, *i.e.* columns of U, to the unit ball of some norm $\|\cdot\|_C$. Unfortunately, for a fixed k, this coding problem is known to be computationally tractable only for the squared loss. To retain tractability for a variety of convex losses, a popular and successful recent approach has been to avoid any "hard" constraint on the number of bases, *i.e.* k, and instead impose regularizers on the matrix V that encourage a low rank or sparse solution.

To be more specific, the following optimization problem lies at the heart of many sparse learning models [*e.g.* 1, 3, 4, 24]:

$$\min_{U:\|U_{:i}\|_C \le 1} \min_{\tilde{V}} L(U\tilde{V}) + \lambda \|\tilde{V}\|_{R,1}, \tag{1}$$

where $\lambda > 0$ specifies the tradeoff between loss and regularization. The $\|\cdot\|_R$ norm in the block R-1 norm provides the flexibility of promoting useful structures in the solution, *e.g.* ℓ_1 norm for sparse solutions, ℓ_2 norm for low rank solutions, and block structured norms for group sparsity. To solve (1), we first reparameterize the rows of $\tilde{V}$ by $\tilde{V}_{i:} = \sigma_i V_{i:}$, where $\sigma_i \ge 0$ and $\|V_{i:}\|_R \le 1$. Now (1) can be reformulated by introducing the reconstruction matrix $X := U\tilde{V}$:

$$(1) = \min_X L(X) + \lambda \min_{U,\tilde{V}:\|U_{:i}\|_C \le 1, U\tilde{V}=X} \|\tilde{V}\|_{R,1} = \min_X L(X) + \lambda \min_{\sigma,U,V:\sigma\ge 0, U\Sigma V = X} \sum_i \sigma_i, \tag{2}$$

where $\Sigma = \text{diag}\{\sigma_i\}$ and we omitted the norm constraints on U and V in the last minimization. (2) is illuminating in two respects. First it reveals that the regularizer essentially seeks a rank-one decomposition of the reconstruction matrix X, and penalizes the ℓ_1 norm of the combination coefficients as a proxy of the "rank". Secondly, the regularizer in (2) is now expressed precisely in

Algorithm 1: The vanilla boosting algorithm.

Require: The weak hypothesis set $\mathcal{A}$ in (3).

1: Set $X_0 = \mathbf{0}$, $s_0 = 0$.
2: **for** $k = 1, 2, \ldots$ **do**
3: $H_k \leftarrow \operatorname{argmin}_{H \in \mathcal{A}} \langle \nabla L(X_{k-1}), H \rangle$.
4: $(a_k, b_k) \leftarrow \operatorname{argmin}_{a \geq 0, b \geq 0} L(aX_{k-1} + bH_k) + \lambda(as_k + b)$.
5: $\sigma_i^{(k)} \leftarrow a_k \sigma_i^{(k-1)}$, $A_i^{(k)} \leftarrow A_i^{(k-1)}, \forall\, i < k$
$\sigma_k^{(k)} \leftarrow b_k$, $A_k^{(k)} \leftarrow H_k$.
6: $X_k \leftarrow \sum_{i=1}^k \sigma_i^{(k)} A_i^{(k)} = a_k X_{k-1} + b_k H_k$,
$s_k \leftarrow \sum_{i=1}^k \sigma_i^{(k)} = a_k s_{k-1} + b_k$.
7: **end for**

Algorithm 2: Boosting with local search.

Require: A set of weak hypotheses $\mathcal{A}$.

1: Set $X_0 = \mathbf{0}$, $U_0 = V_0 = \Lambda_0 = [\,]$, $s_0 = 0$.
2: **for** $k = 1, 2, \ldots$ **do**
3: $(\mathbf{u}_k, \mathbf{v}_k) \leftarrow \operatorname{argmin}_{\mathbf{u}\mathbf{v}' \in \mathcal{A}} \langle \nabla L(X_{k-1}), \mathbf{u}\mathbf{v}' \rangle$.
4: $(a_k, b_k) \leftarrow \operatorname{argmin}_{a \geq 0, b \geq 0} L(aX_{k-1} + b\,\mathbf{u}_k\mathbf{v}_k') + \lambda(as_k + b)$.
5: $U_{\text{init}} \leftarrow (\hat{U}_{k-1}\sqrt{a_k \Lambda_{k-1}}, \sqrt{b_k}\mathbf{u}_k)$,
$V_{\text{init}} \leftarrow (\sqrt{a_k \Lambda_{k-1}}\hat{V}_{k-1}, \sqrt{b_k}\mathbf{v}_k)'$.
6: Locally optimize $g(U, V)$ with initial value $(U_{\text{init}}, V_{\text{init}})$. Get a solution (U_k, V_k).
7: $X_k \leftarrow U_k V_k$, $\Lambda_k \leftarrow \operatorname{diag}\{\|U_{:i}\|_C \|V_{i:}\|_R\}$,
$s_k \leftarrow \frac{1}{2}\sum_{i=1}^k (\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2)$.
8: **end for**

the form of the gauge function $\gamma_{\mathcal{K}}$ induced by the convex hull $\mathcal{K}$ of the set[1]

$$\mathcal{A} = \{\mathbf{u}\mathbf{v}' : \|\mathbf{u}\|_C \leq 1, \|\mathbf{v}\|_R \leq 1\}. \tag{3}$$

Since $\mathcal{K}$ is convex and symmetric ($-\mathcal{K} = \mathcal{K}$), the gauge function $\gamma_{\mathcal{K}}$ is in fact a norm, hence the support function of $\mathcal{A}$ defines the dual norm $|\!|\!| \cdot |\!|\!|$ (see *e.g.* [25, Proposition V.3.2.1]):

$$|\!|\!|\Lambda|\!|\!| := \max_{X \in \mathcal{A}} \operatorname{tr}(X'\Lambda) = \max_{\mathbf{u},\mathbf{v}: \|\mathbf{u}\|_C \leq 1, \|\mathbf{v}\|_R \leq 1} \mathbf{u}'\Lambda\mathbf{v} = \max_{\mathbf{u}: \|\mathbf{u}\|_C \leq 1} \|\Lambda'\mathbf{u}\|_R^* = \max_{\mathbf{v}: \|\mathbf{v}\|_R \leq 1} \|\Lambda\mathbf{v}\|_C^*, \tag{4}$$

and the gauge function $\gamma_{\mathcal{K}}$ is simply its dual norm $|\!|\!| \cdot |\!|\!|^*$. For example, when $\|\cdot\|_R = \|\cdot\|_C = \|\cdot\|_2$, we have $|\!|\!| \cdot |\!|\!| = \|\cdot\|_{\text{sp}}$, so the regularizer (as the dual norm) becomes $\|\cdot\|_{\text{tr}}$. Another special case of this result was found in [4, Theorem 1], where again $\|\cdot\|_R = \|\cdot\|_2$ but $\|\cdot\|_C$ is more complicated than $\|\cdot\|_2$. Note that the original proofs in [1, 4] are somewhat involved. Moreover, this gauge function framework is flexible enough to subsume a number of structurally regularized problems [11, 12], and it is certainly possible to devise other $\|\cdot\|_R$ and $\|\cdot\|_C$ norms that would induce interesting matrix norms.

The gauge function framework also allows us to develop an efficient boosting algorithm for (2), by resorting to the following equivalent problem:

$$\{\sigma_i^*, A_i^*\} := \operatorname*{argmin}_{\sigma_i \geq 0, A_i \in \mathcal{A}} f(\{\sigma_i, A_i\}), \quad \text{where} \quad f(\{\sigma_i, A_i\}) := L\Big(\sum_i \sigma_i A_i\Big) + \lambda \sum_i \sigma_i. \tag{5}$$

The optimal solution X^* of (2) can be easily recovered as $\sum_i \sigma_i^* A_i^*$. Note that in the boosting terminology, $\mathcal{A}$ corresponds to the set of weak hypotheses.

2.1 The boosting algorithm

To solve (5) we propose the boosting strategy presented in Algorithm 1. At each iteration, a weak hypothesis H_k that yields the most rapid local decrease of the loss L is selected. Then H_k is combined with the previous ensemble by tuning its weights to optimize the regularized objective. Note that in Step 5 all the weak hypotheses selected in the previous steps are scaled by the *same* value.

As the ℓ_1 regularizer requires the sum of all the weights, we introduce a variable s_k that recursively updates this sum in Step 6. In addition, X_k is used only in Step 3 and 4, which do not require its explicit expansion in terms of elements of $\mathcal{A}$. Therefore this expansion of X_k does not need to be explicitly maintained and Step 5 is included only for conceptual clarity.

2.2 Rates of convergence

We prove the convergence rate of Algorithm 1, under the standard assumption:

Assumption 1 *L is bounded from below and has bounded sub-level sets. The problem (5) admits at least one minimizer. L is differentiable and satisfies the following inequality for all $\eta \in [0, 1]$ and A, B in the sub-level set of $f(\mathbf{0})$: $L((1-\eta)A + \eta B) \leq L(A) + \eta \langle B - A, \nabla L(A) \rangle + \frac{C_L \eta^2}{2}$. Here $C_L > 0$ is a finite constant that depends only on L.*

[1] Recall that the gauge function $\gamma_{\mathcal{K}}$ is defined as $\gamma_{\mathcal{K}}(X) := \inf\{\sum_i \sigma_i : \sum_i \sigma_i A_i = X,\ A_i \in \mathcal{K},\ \sigma_i \geq 0\}$.

Theorem 1 (Rates of convergence) *Under Assumption 1, Algorithm 1 finds an ϵ accurate solution to (5) in $O(1/\epsilon)$ steps. More precisely, denoting f^* as the minimum of (5), then*

$$f(\{\sigma_i^{(k)}, A_i^{(k)}\}) - f^* \leq \frac{4C_L}{k+2}. \tag{6}$$

The proof is given in Appendix A. Note that the rate is independent of the regularization constant λ.

In the proof we fix the variable a in Step 4 of Algorithm 1 to be simply $\frac{2}{k+2}$; it should be clear that setting a by line search will only accelerate the covergence. An even more aggressive scheme is the totally corrective update [15], which in Step 4 finds the weights for *all* $A_i^{(k)}$'s selected so far:

$$\min_{\sigma_i \geq 0} L\left(\sum_{i=1}^{k} \sigma_i A_i^{(k)}\right) + \lambda \sum_{i=1}^{k} \sigma_i. \tag{7}$$

But in this case we will have to explicitly maintain the expansion of X_t in terms of the $A_i^{(k)}$'s. For boosting without regularization, the $1/\epsilon$ rate of convergence is known to be optimal [26]. We conjecture that $1/\epsilon$ is also a lower bound for regularized boosting.

If the loss L is furthermore strongly convex, then the convergence rate can be tightened to linear.

Theorem 2 *Suppose Assumption 1 holds and L is furthermore strongly convex with modulus μ. Let $\{\sigma_i^*, A_i^*\}$ be a minimzer of (5) and denote $f^* := f(\{\sigma_i^*, A_i^*\})$, $s^* := \sum_i \sigma_i^*$. Then the totally corrective algorithm converges at least linearly. More precisely*

$$f(\{\sigma_i^{(k)}, A_i^{(k)}\}) - f^* \leq \left(1 - \min\left\{\frac{1}{2}, \frac{2\mu(s^*)^2}{m^2 C_L}\right\}\right)^k (f(\{0, \mathbf{0}\}) - f^*), \tag{8}$$

where m is the number of non-zeros in $\{\sigma_i^*\}$. (The proof is given in Appendix B.)

Extensions Our proof technique allows the regularizer to be generalized to the form $h(\gamma_{\mathcal{K}}(X))$, where h is a convex non-decreasing function over $[0, \infty)$. In (5), this replaces $\sum_i \sigma_i$ with $h(\sum_i \sigma_i)$. By taking $h(x)$ as an indicator $h(x) = \{0 \text{ if } x \leq 1; \infty \text{ otherwise}\}$, all of our rates can be straightforwardly translated into the constrained setting.

3 Local Optimization with Fixed Rank

In Algorithm 1, X_k is determined by searching in the conic hull of X_{k-1} and H_k.[2] Suppose there exists some auxiliary procedure that allows X_k to be further improved somehow into Y_k (*e.g.* by local greedy search), then the overall optimization can benefit from it. The only challenge, nevertheless, is how to restore the "context" from Y_k, especially the bases A_i and their weights σ_i.

In particular, suppose we have an auxiliary function g and the following procedure is feasible:

1. Initialization: given an ensemble $\{\sigma_i, A_i\}$, there exists a S such that $g(S) \leq f(\{\sigma_i, A_i\})$.

2. Local optimization: some (local) optimizer can find a T such that $g(T) \leq g(S)$.

3. Recovery: one can recover an ensemble $\{\beta_i, B_i : \beta_i \geq 0, B_i \in \mathcal{A}\}$ such that $g(T) \geq f(\{\beta_i, B_i\})$.

Then obviously the new ensemble $\{\beta_i, B_i\}$ improves upon $\{\sigma_i, A_i\}$. This local search scheme can be easily embedded into Algorithm 1 as follows. After Step 5, initialize S by $\{\sigma_i^{(k)}, A_i^{(k)}\}$. Perform local optimization and recover $\{\beta_i, B_i\}$. Then replace Step 6 by $X_k = \sum_i \beta_i B_i$ and $s_k = \sum_i \beta_i$. All rates of convergence will directly carry over. However, the major challenge here is the potentially expensive step of recovery because little assumption or constraint is made on T.

Fortunately, a careful examination of Algorithm 1 reveals that a complete recovery of $\{\beta_i, B_i\}$ is not required. Indeed, only two "sufficient statistics" are needed: X_k and s_k, and therefore it suffices to recover them only. Next we will show how this can be accomplished efficiently in (2) . Two simple propositions will play a key role. Both proofs can be found in Appendix C.

Proposition 1 *For the gauge $\gamma_{\mathcal{K}}$ induced by $\mathcal{K}$, the convex hull of $\mathcal{A}$ in (3), we have*

$$\gamma_{\mathcal{K}}(X) = \min_{U,V:UV=X} \frac{1}{2} \sum_i \left(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2 \right). \tag{9}$$

[2] This does *not* mean X_k is a minimizer of $L(X) + \lambda\gamma_{\mathcal{K}}(X)$ in that cone, because the bases are not optimized simultaneously. Incidently, this also shows why working with (5) turns out to be more convenient.

If $\|\cdot\|_R = \|\cdot\|_C = \|\cdot\|_2$, then $\gamma_{\mathcal{K}}$ becomes the trace norm (as we saw before), and $\sum_i (\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2)$ is simply $\|U\|_F^2 + \|V\|_F^2$. Then Proposition 1 is a well-known variational form of the trace norm [27]. This motivates us to choose the auxiliary function as

$$g(U, V) = L(UV) + \frac{\lambda}{2} \sum_i \left(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2 \right). \tag{10}$$

Proposition 2 *For any $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{k \times n}$, there exist $\sigma_i \geq 0$, $\mathbf{u}_i \in \mathbb{R}^m$, and $\mathbf{v}_i \in \mathbb{R}^n$ such that*

$$UV = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i', \quad \|\mathbf{u}_i\|_C \leq 1, \quad \|\mathbf{v}_i\|_R \leq 1, \quad \sum_{i=1}^k \sigma_i = \frac{1}{2} \sum_{i=1}^k \left(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2 \right). \tag{11}$$

Now we can specify concrete details for local optimization in the context of matrix norms:

1. Initialize: given $\{\sigma_i \geq 0, \mathbf{u}_i \mathbf{v}_i' \in \mathcal{A}\}_{i=1}^k$, set $(U_{\text{init}}, V_{\text{init}})$ to satisfy $g(U_{\text{init}}, V_{\text{init}}) = f(\{\sigma_i, \mathbf{u}_i \mathbf{v}_i'\})$:

$$U_{\text{init}} = (\sqrt{\sigma_1}\mathbf{u}_1, \ldots, \sqrt{\sigma_k}\mathbf{u}_k), \qquad \text{and} \qquad V_{\text{init}} = (\sqrt{\sigma_1}\mathbf{v}_1, \ldots, \sqrt{\sigma_k}\mathbf{v}_k)'. \tag{12}$$

2. Locally optimize $g(U, V)$ with initialization $(U_{\text{init}}, V_{\text{init}})$, to obtain a solution (U^*, V^*).

3. Recovery: use Proposition 2 to (conceptually) recover $\{\beta_i, \hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i\}$ from (U^*, V^*).

The key advantage of this procedure is that Proposition 2 allows X_k and s_k to be computed *directly* from (U^*, V^*), keeping the recovery completely implicit:

$$X_k = \sum_{i=1}^k \beta_i \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i' = U^* V^*, \quad \text{and} \quad s_k = \sum_{i=1}^k \sigma_i = \frac{1}{2} \sum_{i=1}^k \left(\|U_{:i}^*\|_C^2 + \|V_{i:}^*\|_R^2 \right). \tag{13}$$

In addition, Proposition 2 ensures that locally improving the solution does not incur an increment in the number of weak hypotheses. Using the same trick, the $(U_{\text{init}}, V_{\text{init}})$ in (12) for the $(k+1)$-th iteration can also be formulated in terms of (U^*, V^*). Different from the local optimization for trace norm in [21] which naturally works on the original objective, our scheme requires a nontrivial (variational) reformulation of the objective based on Propositions 1 and 2.

The final algorithm is summarized in Algorithm 2, where $\hat{U}$ and $\hat{V}$ in Step 5 denote the column-wise and row-wise normalized versions of U and V, respectively. Compared to the local optimization in [22], which is hampered by orthogonal and PSD constraints, our (local) objective in (10) is unconstrained and smooth for many instances of $\|\cdot\|_C$ and $\|\cdot\|_R$. This is plausible because no other constraints (besides the norm constraint), such as orthogonality, are imposed on U and V in Proposition 2. Thus the local optimization we face, albeit non-convex in general, is more amenable to efficient solvers such as L-BFGS.

Remark Consider if one performs totally corrective update as in (7). Then all of the coefficients and weak hypotheses from (U^*, V^*) have to be considered, which can be computationally expensive. For example, in the case of trace norm, this leads to a full SVD on $U^* V^*$. Although U^* and V^* usually have low rank, which can be exploited to ameliorate the complexity, it is clearly preferable to completely eliminate the recovery step, as in Algorithm 2.

4 Latent Generative Model with Multiple Views

Underlying most boosting algorithms is an oracle that identifies the steepest descent weak hypothesis (Step 3 of Algorithm 1). Approximate solutions often suffice [8, 9]. When $\|\cdot\|_R$ and $\|\cdot\|_C$ are both Euclidean norms, this oracle can be efficiently computed via the leading left and right singular vector pair. However, for most other interesting cases like low rank tensors, such an oracle is intractable [28]. In this section we discover that for an important problem of multiview learning, the oracle can be surprisingly solved in polynomial time, yielding an efficient computational strategy.

Multiview learning analyzes multi-modal data, such as heterogeneous descriptions of text, image and video, by exploiting the implicit conditional independence structure. In this case, beyond a single dictionary U and coefficient matrix V that model a single view $Z^{(1)}$, multiple dictionaries $U^{(k)}$ are needed to reconstruct multiple views $Z^{(k)}$, while keeping the latent representation V shared across all views. Formally the problem in multiview factorization is to optimize [2, 4]:

$$\min_{U^{(1)}: \|U_{:i}^{(1)}\|_C \leq 1} \cdots \min_{U^{(k)}: \|U_{:i}^{(k)}\|_C \leq 1} \min_V \sum_{t=1}^k L_t(U^{(t)} V) + \lambda \|V\|_{R,1}. \tag{14}$$

We can easily re-express the problem as an equivalent "single" view formulation (1) by stacking all $\{U^{(t)}\}$ into the rows of a big matrix U, with a new column norm $\|U_{:i}\|_C := \max_{t=1\dots k} \|U^{(t)}_{:i}\|_C$. Then the constraints on $U^{(t)}$ in (14) can be equivalently written as $\|U_{:i}\|_C \le 1$, and Algorithm 2 can be directly applied with two specializations. First the auxiliary function $g(U,V)$ in (10) becomes

$$g(U,V) = L(UV) + \frac{\lambda}{2}\sum_i \left(\left(\max_{t=1\dots k} \|U^{(t)}_{:i}\|_C\right)^2 + \|V_{i:}\|_R^2\right) = L(UV) + \frac{\lambda}{2}\sum_i \left(\max_{t=1\dots k} \|U^{(t)}_{:i}\|_C^2 + \|V_{i:}\|_R^2\right)$$

which can be locally optimized. The only challenge left is the oracle problem in (4), which takes the following form when all norms are Euclidean:

$$\max_{\|\mathbf{u}\|_C \le 1, \|\mathbf{v}\| \le 1} \mathbf{u}'\Lambda\mathbf{v} = \max_{\|\mathbf{u}\|_C \le 1} \|\Lambda'\mathbf{u}\|^2 = \max_{\mathbf{u}:\forall t, \|\mathbf{u}_t\| \le 1} \left\|\sum_t \Lambda'_t \mathbf{u}_t\right\|^2. \tag{15}$$

[4] considered the case where $k = 2$ and showed that exact solutions to (15) can be found efficiently. But their derivation does not seem to extend to $k > 2$. Fortunately there is still an interesting and tractable scenario. Consider multilabel classification with a small number of classes, and $U^{(1)}$ and $U^{(2)}$ are two views of features (*e.g.* image and text). Then each class label corresponds to a view and the corresponding u_t is univariate. Since there must be an optimal solution on the extreme points of the feasible region, we can enumerate $\{-1, 1\}$ for u_t ($t \ge 3$) and for each assignment solve a subproblem of the following form that instantiates (15) ($\mathbf{c}$ is a constant vector)

$$(QP) \quad \max_{\mathbf{u}_1, \mathbf{u}_2} \|\Lambda'_1\mathbf{u}_1 + \Lambda'_2\mathbf{u}_2 + \mathbf{c}\|^2, \ s.t. \ \|\mathbf{u}_1\| \le 1, \ \|\mathbf{u}_2\| \le 1. \tag{16}$$

Due to inhomogeneity, the technique in [4] is not applicable. Rewrite (16) in matrix form

$$(QP) \quad \min_{\mathbf{z}} \langle M_0, \mathbf{z}\mathbf{z}'\rangle \quad s.t. \quad \langle M_1, \mathbf{z}\mathbf{z}'\rangle \le 0 \quad \langle M_2, \mathbf{z}\mathbf{z}'\rangle \le 0 \quad \langle I_{00}, \mathbf{z}\mathbf{z}'\rangle = 1, \tag{17}$$

where $\mathbf{z} = \begin{pmatrix} r \\ \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}$, $M_0 = -\begin{pmatrix} 0 & \mathbf{c}'\Lambda'_1 & \mathbf{c}'\Lambda'_2 \\ \Lambda_1\mathbf{c} & \Lambda_1\Lambda'_1 & \Lambda_1\Lambda'_2 \\ \Lambda_2\mathbf{c} & \Lambda_2\Lambda'_1 & \Lambda_2\Lambda'_2 \end{pmatrix}$, $M_1 = \begin{pmatrix} -1 & & \\ & I & \\ & & \mathbf{0} \end{pmatrix}$, $M_2 = \begin{pmatrix} -1 & & \\ & \mathbf{0} & \\ & & I \end{pmatrix}$,

and I_{00} is a zero matrix with only the $(1,1)$-th entry being 1. Let $X = \mathbf{z}\mathbf{z}'$, a semi-definite programming relaxation for (QP) can be obtained by dropping the rank-one constraint:

$$(SP) \quad \min_X \langle M_0, X\rangle, \quad s.t. \quad \langle M_1, X\rangle \le 0, \quad \langle M_2, X\rangle \le 0, \quad \langle I_{00}, X\rangle = 1, \quad X \succeq \mathbf{0}. \tag{18}$$

Its dual problem, which is also the Lagrange dual of (QP), can be written as

$$(SD) \quad \max_{y_0, y_1, y_2} y_0, \quad s.t. \quad Z := M_0 - y_0 I_{00} + y_1 M_1 + y_2 M_2 \succeq \mathbf{0}, \quad y_1 \ge 0, \quad y_2 \ge 0. \tag{19}$$

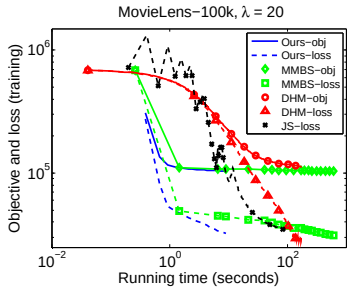
(SD) is a convex problem that can be solved efficiently by, *e.g.*, cutting plane methods. (SP) is also a convex semidefinite program (SDP) amenable for standard SDP solvers. However further recovering the solution to (QP) is *not* straightforward, because there may be a gap between the optimal values of (SP) and (QP). The gap is zero (*i.e.* strong duality between (QP) and (SD)) only if the rank-one constraint that (SP) dropped from (QP) is automatically satisfied, *i.e.* if (SP) has a rank-one optimal solution.

Fortunately, as one of our main results, we prove that strong duality always holds for the particular problem originating from (16). Our proof utilizes some recent development in optimization [29], and is relegated to Appendix D.

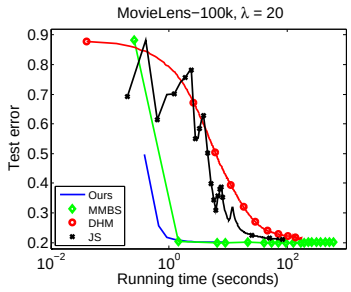
5 Experimental Results

We compared our Algorithm 2 with three state-of-the-art solvers for trace norm regularized objectives: MMBS[3] [22], DHM [15], and JS [8]. JS was proposed for solving the constrained problem: $\min_X L(X)$ s.t. $\|X\|_{\text{tr}} \le \eta$, which makes it hard to be compared with solvers for penalized problems: $\min_X L(X) + \lambda \|X\|_{\text{tr}}$. As a workaround, we first chose a λ, and found the optimal solution X^* for the penalized problem. Then we set $\eta = \|X^*\|_{\text{tr}}$ and finally solved the constrained problem by JS. In this case, it is only fair to compare how fast $L(X)$ (loss) is decreased by various solvers, rather than $L(X) + \lambda \|X\|_*$ (objective). DHM is sensitive to the estimate of the Lipschitz constant H of the gradient of L. We manually tuned H for a small value such that DHM still converges. Since the code for MMBS is specialized to matrix completion, it was used only in this comparison. Traditional solvers such as proximal methods [6] were not included because they are much slower.

[3] http://www.montefiore.ulg.ac.be/ mishra/softwares/traceNorm.html

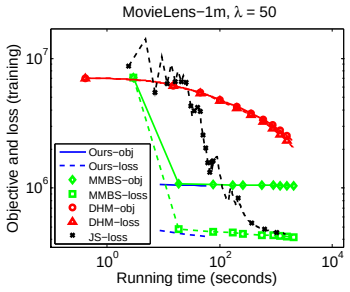


(a) Objective & loss vs time (loglog)

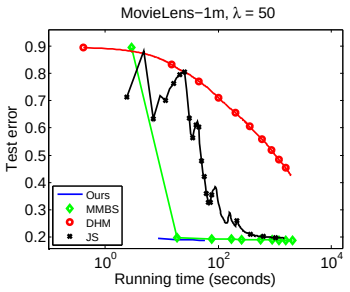


(b) Test NMAE vs time (semilogx)

Figure 1: MovieLens100k.

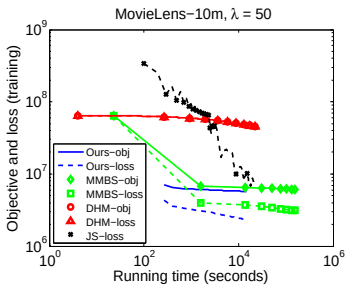


(a) Objective & loss vs time (loglog)

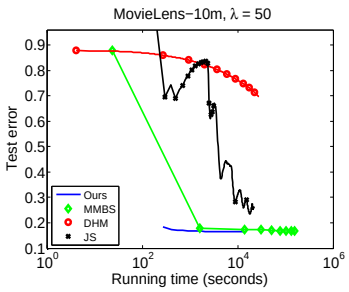


(b) Test NMAE vs time (semilogx)

Figure 2: MovieLens1M.



(a) Objective & loss vs time (loglog)



(b) Test NMAE vs time (semilogx)

Figure 3: MovieLens10M.

Comparison 1: Matrix completion We first compared all methods on a matrix completion problem, using the standard datasets MovieLens100k, MovieLens1M, and MovieLens10M [6, 8, 21], which are sized 943×1682, 6040×3706, and 69878×10677 respectively (#user $\times$ #movie). They contain 10^5, 10^6 and 10^7 movie ratings valued from 1 to 5, and the task is to predict the rating for a user on a movie. The training set was constructed by randomly selecting 50% ratings for each user, and the prediction is made on the rest 50% ratings. In Figure 1 to 3, we show how fast various algorithms drive down the training objective, training loss L (squared Euclidean distance), and the normalized mean absolute error (NMAE) on the test data [see, *e.g.*, 6, 8]. We tuned the λ to optimize the test NMAE.

From Figure 1(a), 2(a), 3(a), it is clear that it takes much less amount of CPU time for our method to reduce the objective value (solid line) and the loss L (dashed line). This implies that local search and partially corrective updates in our method are very effective. Not surprisingly MMBS is the closest to ours in terms of performance because it also adopts local optimization. However it is still slower because their local search is conducted on a *constrained* manifold. In contrast, our local search objective is entirely unconstrained and smooth, which we manage to solve efficiently by L-BFGS.[4]

JS, though applied indirectly, is faster than DHM in reducing the loss. We observed that DHM kept running coordinate descent with a constant step size, while the totally corrective update was rarely taken. We tried accelerating it by using a smaller value of the estimate of the Lipschitz constant H, but it leads to divergence after a rapid decrease of the objective for the first few iterations. A hybrid approach might be useful.

We also studied the evolution of the NMAE performance on the test data. For this we compared the matrix reconstruction after each iteration against the ground truth. As plotted in Figure 1(b), 2(b), 3(b), our approach achieves comparable (or better) NMAE in much less time than all other methods.

Comparison 2: multitask and multiclass learning Secondly, we tested on a multiclass classification problem with synthetic dataset. Following [15], we generated a dataset of $D = 250$ features and $C = 100$ classes. Each class c has 10 training examples and 10 test examples drawn independently and identically from a class-specific multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}_c, \Sigma_c)$. $\boldsymbol{\mu}_c \in \mathbb{R}^{250}$ has the last 200 coordinates being 0, and the top 50 coordinates were chosen uniformly random from $\{-1, 1\}$. The (i, j)-th element of Σ_c is $2^2(0.5)^{|i-j|}$. The task is to predict the class membership of a given example. We used the logistic loss for a model matrix $W \in \mathbb{R}^{D \times C}$. In particular, for each

[4] http://www.cs.ubc.ca/~pcarbo/lbfgsb-for-matlab.html

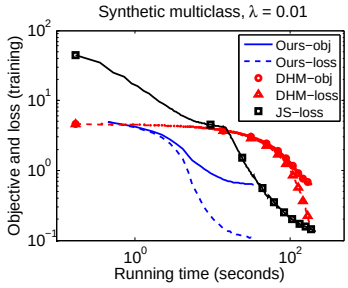
training example $\mathbf{x}_i$ with label $y_i \in \{1, .., C\}$, we defined an individual loss $L_i(W)$ as

$$L_i(W) = -\log p(y_i|\mathbf{x}_i; W),$$

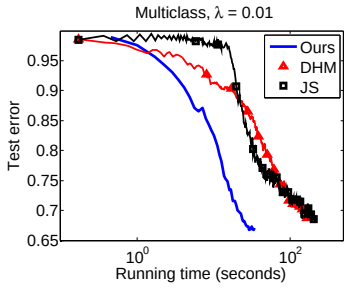
where for any class c,

$$p(c|\mathbf{x}_i; W) = Z_i^{-1} \exp(W'_{:c}\mathbf{x}_i),$$
$$Z_i = \sum_c \exp(W'_{:c}\mathbf{x}_i).$$

Then $L(W)$ is defined as the average of $L_i(W)$ over the whole training set. We found that $\lambda = 0.01$ yielded the lowest test classification error; the corresponding results are given in Figure 4. Clearly, the intermediate models output by our scheme achieve comparable (or better) training objective and test error in orders of magnitude less time than those generated by DHM and JS.

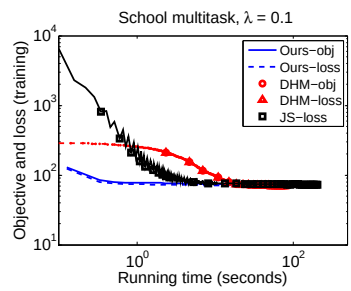


(a) Objective & loss vs time (loglog)

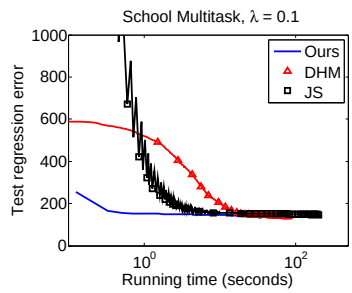


(b) Test error vs time (semilogx)

Figure 4: Multiclass classification with synthetic datset.



(a) Objective & loss vs time (loglog)



(b) Test error vs time (semilogx)

Figure 5: Multitask learning for school dataset.

We also applied the solvers to a multitask learning problem with the school dataset [24]. The task is to predict the score of 15362 students from 139 secondary schools based on a number of school-specific and student-specific attributes. Each school is considered as a task for which a predictor is learned. We used the first random split of training and testing data provided by [24] [5], and set λ so as to achieve the lowest test squared error. Again, as shown in Figure 5 our approach is much faster than DHM and JS in finding the optimal solution for training objective and test error. As the problem requires a large λ, the trace norm penalty is small, making the loss close to the objective.

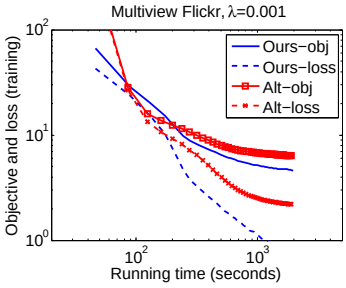


Figure 6: Multiview training.

Comparison 3: Multiview learning Finally we perform an initial test on our global optimization technique for learning latent models with multiple views. We used the Flickr dataset from NUS-WIDE [30]. Its first view is a 634 dimensional low-level feature, and the second view consists of 1000 dimensional tags. The class labels correspond to the type of animals and we randomly chose 5 types with 20 examples in each type. The task is to train the model in (14) with $\lambda = 10^{-3}$. We used squared loss for the first view, and logistic loss for the other views.

We compared our method with a local optimization approach to solving (14). The local method first fixes all $U^{(t)}$ and minimizes V, which is a convex problem that can be solved by FISTA [31]. Then it fixes V and optimizes $U^{(t)}$, which is again convex. We let Alt refer to the scheme that alternates these updates to convergence. From Figure 6 it is clear that Alt is trapped by a locally optimal solution, which is inferior to a globally optimal solution that our method is guaranteed to find. Our method also reduces both the objective and the loss slightly faster than Alt.

6 Conclusion and Outlook

We have proposed a new boosting algorithm for a wide range of matrix norm regularized problems. It is closely related to generalized conditional gradient method [32]. We established $O(1/\epsilon)$ rates of convergence, and showed its empirical advantage over state-of-the-art solvers on large scale problems. We also applied the method to a novel problem, latent multiview learning, for which we designed a new efficient oracle. We plan to study randomized boosting with ℓ_1 regularization [33–35], and to extend the framework to more general nonlinear regularization [3].

[5] http://ttic.uchicago.edu/~argyriou/code/mtl_feat/school_splits.tar

References

[1] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. arXiv:0812.1869v1, 2008.

[2] H. Lee, R. Raina, A. Teichman, and A. Ng. Exponential family sparse coding with application to self-taught learning. In *IJCAI*, 2009.

[3] D. Bradley and J. Bagnell. Convex coding. In *UAI*, 2009.

[4] X. Zhang, Y-L Yu, M. White, R. Huang, and D. Schuurmans. Convex sparse coding, subspace learning, and semi-supervised extensions. In *AAAI*, 2011.

[5] T. K. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

[6] K-C Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

[7] J-F Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[8] M. Jaggi and M. Sulovsky. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

[9] E. Hazan. Sparse approximate solutions to semidefinite programs. In *LATIN*, 2008.

[10] K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. In *SODA*, 2008.

[11] A. Tewari, P. Ravikumar, and I. S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *NIPS*, 2011.

[12] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. Technical report, 2012. http://arxiv.org/abs/1012.0621.

[13] Y. Bengio, N.L. Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. In *NIPS*, 2005.

[14] L. Mason, J. Baxter, P. L. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*, pages 221–246, Cambridge, MA, 2000. MIT Press.

[15] M. Dudik, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularizations. In *AISTATS*, 2012.

[16] S. Shalev-Shwartz, N. Srebro, and T. Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20:2807–2832, 2010.

[17] X. Yuan and S. Yan. Forward basis selection for sparse approximation over dictionary. In *AISTATS*, 2012.

[18] T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. Information Theory*, 49(3):682–691, 2003.

[19] S. Burer and R. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[20] M. Journee, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20:2327C–2351, 2010.

[21] S. Laue. A hybrid algorithm for convex semidefinite optimization. In *ICML*, 2012.

[22] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. Technical report, 2011. http://arxiv.org/abs/1112.2318.

[23] S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *ICML*, 2011.

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

[25] J-B Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms, I and II*, volume 305 and 306. Springer-Verlag, 1993.

[26] I. Mukherjee, C. Rudin, and R. Schapire. The rate of convergence of Adaboost. In *COLT*, 2011.

[27] N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, 2005.

[28] C. Hillar and L-H Lim. Most tensor problems are NP-hard. arXiv:0911.1393v3, 2012.

[29] W. Ai and S. Zhang. Strong duality for the CDT subproblem: A necessary and sufficient condition. *SIAM Journal on Optimization*, 19:1735–1756, 2009.

[30] T.S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.T. Zhang. A real-world web image database from national university of singapore. In *International Conference on Image and Video Retrieval*, 2009.

[31] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[32] K. Bredies, D. Lorenz, and P. Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42:173–193, 2009.

[33] A. Kleiner, A. Rahimi, and M. Jordan. Random conic pursuit for semidefinite programming. In *NIPS*, 2010.

[34] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[35] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS*, 2008.

Supplementary Material

A Proof of Theorem 1

In this section we prove the $O(1/\epsilon)$ convergence rate of the boosting Algorithm 1.

Theorem 1 (Rate of convergence) *Under Assumption 1, Algorithm 1 finds an ϵ accurate solution to (5) in $O(1/\epsilon)$ number of steps. More precisely, denoting f^* as the minimum of (5), then*

$$f(\{\sigma_i^{(k)}, A_i^{(k)}\}) - f^* \leq \frac{4C_L}{k+2}.$$

Proof: Denoting $s^* = \sum_i \sigma_i^*$, where recall that $\{A_i, \sigma_i\}$ is some optimal solution to (5). Our proof is based upon the following observation:

$$\begin{aligned} f^* &= \min_{A_i \in \mathcal{A}, \sigma_i \geq 0} L\left(\sum_i \sigma_i A_i\right) + \lambda \sum_i \sigma_i \\ &= \min_{Y \in s^* \mathcal{K}} L(Y) + \lambda s^*, \end{aligned} \tag{20}$$

where $\mathcal{K}$ is the convex hull of the set $\mathcal{A}$.

Let $s_k := \sum_i \sigma_i^{(k)}$. We prove Theorem 1 for a "weaker" version of Algorithm 1, where a_k is set to some constant $1 - \eta_k$. The following chain of inequalities consists the main part of our proof.

$$\begin{aligned} f(X_k) &= L(X_k) + \lambda s_k \\ \text{(Definition of } X_k, s_k) &= \min_{\rho \geq 0} L\left((1-\eta_k)X_{k-1} + \rho \eta_k H_k\right) + \lambda(1-\eta_k)s_{k-1} + \lambda \rho \eta_k \qquad (21) \\ &\leq L((1-\eta_k)X_{k-1} + \eta_k(s^* H_k)) + \lambda(1-\eta_k)s_{k-1} + \lambda s^* \eta_k \\ \text{(Assumption 1)} &\leq f(X_{k-1}) + \eta_k \left\langle s^* H_k - X_{k-1}, \nabla L(X_{k-1}) \right\rangle + \frac{C_L}{2}\eta_k^2 - \lambda \eta_k s_{k-1} + \lambda \eta_k s^* \qquad (22) \\ \text{(Definition of } H_k) &\leq \min_{Y \in s^* \cdot \mathcal{A}} f(X_{k-1}) + \eta_k \left\langle Y - X_{k-1}, \nabla L(X_{k-1}) \right\rangle + \frac{C_L}{2}\eta_k^2 - \lambda \eta_k s_{k-1} + \lambda \eta_k s^* \\ \text{(Linearity)} &\leq \min_{Y \in s^* \cdot \mathcal{K}} f(X_{k-1}) + \eta_k \left\langle Y - X_{k-1}, \nabla L(X_{k-1}) \right\rangle + \frac{C_L}{2}\eta_k^2 - \lambda \eta_k s_{k-1} + \lambda \eta_k s^* \qquad (23) \\ \text{(Convexity of } L) &\leq \min_{Y \in s^* \cdot \mathcal{K}} f(X_{k-1}) + \eta_k (L(Y) - L(X_{k-1})) + \frac{C_L}{2}\eta_k^2 - \lambda \eta_k s_{k-1} + \lambda \eta_k s^* \\ \text{(Rearrangement)} &= (1-\eta_k) f(X_{k-1}) + \eta_k \min_{Y \in s^* \cdot \mathcal{K}} \left(L(Y) + \lambda s^*\right) + \frac{C_L}{2}\eta_k^2 \\ \text{(Observation (20))} &= (1-\eta_k) f(X_{k-1}) + \eta_k f^* + \frac{C_L}{2}\eta_k^2, \end{aligned}$$

hence

$$f(X_k) - f^* \leq (1-\eta_k)(f(X_{k-1}) - f^*) + \frac{C_L}{2}\eta_k^2.$$

Setting $\eta_k = \frac{2}{k+2}$, and an easy induction argument establishes that

$$f(X_k) - f^* \leq \frac{4C_L}{k+2}.$$

■

The proof, although completely elementary, does harness several interesting ideas. Note first that in, say, the analysis of the ordinary gradient algorithm, one usually upper bounds the convex function L by its quadratic expansion

$$L(Y) \leq L(X) + \langle Y - X, \nabla L(X) \rangle + \frac{\hat{C}_L}{2}\|Y - X\|^2,$$

and then tries to minimize the quadratic upper bound; while in contrast, our analysis above takes perhaps a surprisingly loose step: upper bound L by the *linear* function

$$L(y) \leq L(x) + \langle Y - X, \nabla L(X) \rangle + \frac{C_L}{2}.$$

The (huge) gain, of course, is the possibility of inequality (23), which allows us to select the next update by optimizing over the (potentially much simpler) set $\mathcal{A}$, instead of the convex hull $\mathcal{K}$.

The next key ingredient in the proof is our observation (20), which is completely trivial, yet after combining it with the one dimensional line search over $\rho \geq 0$ (or b in Algorithm 1), Algorithm 1 behaves as if it knew the *unknown* but fixed constant s^*.

Some remarks regarding to Theorem 1 are in order.

- If the loss function L is only Lipchitz continuous, then one can apply the "smoothing" trick [36] to get $O(\frac{1}{\epsilon^2})$ convergence rate for algorithm 1.
- Our result heavily builds on previous work [37, 38], however, it seems that our treatment is slightly more general. For instance, the ℓ_1 norm regularizer $\sum_i \sigma_i$ can be readily replaced by $h(\sum_i \sigma_i)$, where $h : \mathbb{R}_+ \mapsto \mathbb{R}$ is some convex function. Essentially the same proof would still go through. Take h as the indicator of some convex set recovers most previous results, which all consider the constrained problem instead of the arguably more natural regularized problem[6].
- The line search step in Algorithm 1 need not be solved exactly. We can derive essentially the same rate as long as the error decays at the rate $O(\frac{1}{k})$.
- The step size $\eta_k = O(\frac{1}{k})$ is optimal, among constant ones, in the following sense. We usually prefer large step sizes since they often than not result in faster convergence; on the other hand, Algorithm 1 needs to be able to reset any σ_i to 0, which requires that the discount factor $\prod_{k=1}^{\infty}(1-\eta_k) = 0$. It is not hard to show that the latter condition is satisfied iff $\sum_{k=1}^{\infty} \eta_k = \infty$, hence the near optimality of the step size $O(\frac{1}{k})$.

B Proof of Theorem 2

In this section, under an additional assumption, we improve the convergence rate in Theorem 1 by considering the totally corrective algorithm in (7).

Recall that strong convexity (with modulus μ) of L implies that

$$L(Y) \geq L(X) + \langle Y - X, \nabla L(X) \rangle + \frac{\mu}{2}\|L - X\|^2. \tag{24}$$

Note that the constant μ depends on the choice of the norm $\|\cdot\|$. In the proof we fix the norm to be essentially ℓ_1.

Theorem 2 *Suppose Assumption 1 holds and L is furthermore strongly convex with modulus μ. Let $\{A_i^*, \sigma_i^*\}$ be a minimzer of (5) and denote $f^* := f(\{A_i^*, \sigma_i^*\})$, $s^* := \sum_i \sigma_i^*$. Then the totally corrective algorithm converges at least linearly. More precisely*

$$f(\{\sigma_i^{(k)}, A_i^{(k)}\}) - f^* \leq \left(1 - \min\left\{\frac{1}{2}, \frac{2\mu(s^*)^2}{m^2 C_L}\right\}\right)^k (f(\{0, \mathbf{0}\}) - f^*),$$

where m is the number of non-zeros in $\{\sigma_i^*\}$.

Our proof is essentially in the same spirit as that of [16, Theorem 2.8], see also [17, Theorem 2]. It is a pleasant surprise that the latter proof extends without much difficulty to the regularized problem considered here.

Proof: In the proof we will use $f(X_k)$ to denote $L(X_k) + \lambda \sum_i \sigma_i^{(k)}$ where $X_k := \sum_{i=1}^{k} \sigma_i^{(k)} A_i^{(k)}$.

[6] [39] proposed an algorithm similar as our totally corrective version in (7) for the regularized problem, but the rate proven there, $O(\frac{1}{\epsilon^2})$, is worse than the one presented in our Theorem 1.

Let us record the optimality condition in (7): $\forall\, \boldsymbol{\tau} \in \mathbb{R}^k_+$, the following holds

$$\sum_{i=1}^{k}(\langle A_i^{(k)}, \nabla L(X_k)\rangle + \lambda)(\tau_i - \sigma_i^{(k)}) \geq 0, \tag{25}$$

where $\sigma_i^{(k)}$ denotes the optimal solution in (7).

Take $0 \leq \eta \leq 1$ whose value will be optimized later. Let $s_k := \sum_{i=1}^{k} \sigma_i^{(k)}$. From Assumption 1 we have

$$\begin{aligned} f((1-\eta)X_k + \eta s^* H_{k+1}) &= L((1-\eta)X_k + \eta s^* H_{k+1}) + (1-\eta)\lambda s_k + \eta\lambda s^* \\ &\leq f(X_k) + \eta \left\langle s^* H_{k+1} - X_k, \nabla L(X_k)\right\rangle + \frac{C_L}{2}\eta^2 + \eta\lambda(s^* - s_k). \end{aligned} \tag{26}$$

We need to define two index sets I and J, where I contains the indexes of the elements in $\{A_i^*\}$ but not in $\{A_i^{(k)}\}$ while J contains the indexes of the elements in both $\{A_i^*\}$ and $\{A_i^{(k)}\}$. Note that we can assume that I is nonempty since otherwise the current totally corrective step will find an optimal solution.

Define $r = \sum_{i\in I} \sigma_i^*$, and by the definition of H_{k+1},

$$\begin{aligned} r\left\langle s^* H_{k+1}, \nabla L(X_k)\right\rangle &\leq \sum_{i\in I} s^*\sigma_i^* \left\langle A_i, \nabla L(X_k)\right\rangle \\ &= \sum_{i\in I}(s^*\sigma_i^* - (s^* - r)\sigma_i^{(k)}) \left\langle A_i, \nabla L(X_k)\right\rangle \\ &\leq \sum_{i\in J}(s^*\sigma_i^* - (s^* - r)\sigma_i^{(k)}) \left\langle A_i, \nabla L(X_k)\right\rangle + \lambda(s^* - r)(s^* - s_k) \\ &= s^*(\langle X^* - X_k, \nabla L(X_k)\rangle + \lambda(s^* - s_k)) - \lambda r(s^* - s_k) + r\left\langle X_k, \nabla L(X_k)\right\rangle \\ &\leq s^*(f^* - f(X_k) - \frac{\mu}{2}\|\sigma^* - \sigma^{(k)}\|_1^2) - \lambda r(s^* - s_k) + r\left\langle X_k, \nabla L(X_k)\right\rangle, \end{aligned} \tag{27}$$

where the last inequality follows from the strong convexity assumption, and the second inequality follows from the optimality of $\sigma^{(k)}$. Indeed, if $J - I = \emptyset$, then $s^* = r$, hence we in fact have an equality. Assume otherwise, then the inequality follows from the optimality condition (25).

Now apply (27) to (26), we get

$$\begin{aligned} f((1-\eta)X_k + \eta s^* H_{k+1}) &\leq f(X_k) + \eta\frac{r\left\langle s^* H_{k+1}, \nabla L(X_k)\right\rangle - r\left\langle X_k, \nabla L(X_k)\right\rangle}{r} + \frac{C_L}{2}\eta^2 + \eta\lambda(s^* - s_k) \\ &\leq f(X_k) - \eta\frac{s^*(f(X_k) - f^* + \frac{\mu}{2}\|\sigma^* - \sigma^{(k)}\|_1^2)}{r} + \frac{C_L}{2}\eta^2. \end{aligned}$$

Apparently $f(X_{k+1}) \leq \min_{\eta\in[0,1]} f((1-\eta)X_k + \eta s^* H_{k+1})$, hence

$$f(X_{k+1}) - f^* \leq f(X_k) - f^* - \eta\frac{s^*(f(X_k) - f^* + \frac{\mu}{2}\|\sigma^* - \sigma^{(k)}\|_1^2)}{r} + \frac{C_L}{2}\eta^2.$$

Minimizing η on the right-hand side yields

$$f(X_{k+1}) - f^* \leq f(X_k) - f^* - \min\left\{\frac{s^*\delta}{2r}, \frac{\delta^2(s^*)^2}{2r^2 C_L}\right\},$$

where $\delta := f(X_k) - f^* + \frac{\mu}{2}\|\sigma^* - \sigma^{(k)}\|_1^2 \geq 0$. It is easy to see that

$$\frac{s^*\delta}{2r} \geq \frac{1}{2}(f(X_k) - f^*).$$

On the other hand,

$$\begin{aligned}\frac{\delta^2(s^*)^2}{2r^2C_L} &\geq \frac{2\mu(f(X_k)-f^*)(s^*)^2\|\sigma^*-\sigma^{(k)}\|_1^2}{r^2C_L} \geq \frac{2\mu(f(X_k)-f^*)(s^*)^2\sum_{i\in I}(\sigma_i^*)^2}{C_L(\sum_{i\in I}\sigma_i^*)^2} \\ &\geq \frac{2\mu(f(X_k)-f^*)(s^*)^2}{C_L|I|^2} \geq \frac{2\mu(f(X_k)-f^*)(s^*)^2}{C_Lm^2} = \frac{2\mu(s^*)^2}{C_Lm^2}(f(X_k)-f^*),\end{aligned} \tag{28}$$

where recall that m is the number of nonzeros entries in $\{\sigma_i^*\}$.

Combining the above two estimates completes the proof:

$$f(X_{k+1})-f^* \leq \left(1-\min\left\{\frac{1}{2}, \frac{2\mu(s^*)^2}{C_Lm^2}\right\}\right)(f(X_k)-f^*).$$

■

C Proof of Proposition 1 and 2

Recall that $\mathcal{K}$ is the convex hull of $\mathcal{A}$.

Proposition 1 $\gamma_{\mathcal{K}}(X) = \min_{U,V:UV=X} \frac{1}{2}\sum_i(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2) = \min_{U,V:UV=X}\sum_i \|U_{:i}\|_C \|V_{i:}\|_R$.

Proof: This proof is similar in spirit to [40]. For any $UV = X$, we can write

$$X = \sum_i \|U_{:i}\|_C \|V_{i:}\|_R \frac{U_{:i}}{\|U_{:i}\|_C}\frac{V_{i:}}{\|V_{i:}\|_R}. \tag{29}$$

So by the definition of gauge function,

$$\gamma_{\mathcal{K}}(X) \leq \sum_i \|U_{:i}\|_C \|V_{i:}\|_R \leq \frac{1}{2}\sum_i\left(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2\right). \tag{30}$$

To attain equality, by the the definition of the gauge $\gamma_{\mathcal{K}}$, there exist σ_i, $\hat{U}$, and $\hat{V}$ which satisfy

$$\|\hat{U}_{:i}\|_C = \|\hat{V}_{i:}\|_R = 1, \qquad \sum_i \sigma_i\hat{U}_{:i}\hat{V}_{i:} = X, \qquad \gamma_{\mathcal{K}}(X) = \sum_i \sigma_i, \qquad \sigma_i \geq 0. \tag{31}$$

Then define $U_{:i} = \sqrt{\sigma_i}\hat{U}_{:i}$ and $V_{i:} = \sqrt{\sigma_i}\hat{V}_{i:}$. It is easy to verify that $UV = X$ and $\frac{1}{2}(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2) = \sum_i \|U_{:i}\|_C\|V_{i:}\|_R = \sum_i \sigma_i = \gamma_{\mathcal{K}}(X)$. ■

Proposition 2 *For any $U \in \mathbb{R}^{m\times k}, V \in \mathbb{R}^{k\times n}$, there exist $\alpha_i \geq 0$, $\|\boldsymbol{\alpha}\|_0 \leq k$ and $\mathbf{u}_i$, $\mathbf{v}_i$ such that*

$$UV = \sum_i \alpha_i\mathbf{u}_i\mathbf{v}_i', \quad \|\mathbf{u}_i\|_C \leq 1, \quad \|\mathbf{v}_i\|_R \leq 1, \quad \sum_i \alpha_i = \frac{1}{2}\sum_i(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2).$$

Proof: Denote $a_i = \|U_{:i}\|_C$ and $b_i = \|V_{i:}\|_R$. Then

$$UV = \sum_i a_ib_i\frac{U_{:i}}{a_i}\frac{V_{i:}}{b_i} = \sum_i \underbrace{\frac{1}{2}(a_i^2+b_i^2)}_{:=\alpha_i}\underbrace{\sqrt{\frac{a_ib_i}{\frac{1}{2}(a_i^2+b_i^2)}}\frac{U_{:i}}{a_i}}_{:=\mathbf{u}_i}\underbrace{\sqrt{\frac{a_ib_i}{\frac{1}{2}(a_i^2+b_i^2)}}\frac{V_{i:}}{b_i}}_{:=\mathbf{v}_i'}. \tag{32}$$

Clearly $\|\mathbf{u}_i\|_C \leq 1$, $\|\mathbf{v}_i\|_R \leq 1$, and $\sum_i \alpha_i = \frac{1}{2}\sum_i(\|U_{:i}\|_C^2 + \|V_{i:}\|_R^2)$. ■

D Proof of the strong duality

The goal of this note is to solve the following problem:

$$(QP)\ \max_{\mathbf{x},\mathbf{y}} \|A\mathbf{x}+B\mathbf{y}+\mathbf{c}\|,\ s.t.\ \|\mathbf{x}\|\le 1,\ \|\mathbf{y}\|\le 1. \tag{33}$$

Here $\mathbf{c}$ is a non-zero vector, and all the norms are Euclidean. Let $\mathbf{x}\in\mathbb{R}^m$, $\mathbf{y}\in\mathbb{R}^n$, $\mathbf{c}\in\mathbb{R}^t$, $A\in\mathbb{R}^{t\times m}$ and $B\in\mathbb{R}^{t\times n}$.

The problem is not convex in this form, because it is *maximizing* a positive semi-definite quadratic. To find a global solution, we first reformulate it. Define

$$\mathbf{z}=\begin{pmatrix} r\\ \mathbf{x}\\ \mathbf{y}\end{pmatrix},\quad \mathbf{b}=\begin{pmatrix} A'\mathbf{c}\\ B'\mathbf{c}\end{pmatrix} \tag{34}$$

$$Q=-\begin{pmatrix} A'A & A'B\\ B'A & B'B\end{pmatrix},\ M_0=\begin{pmatrix} 0 & -\mathbf{b}'\\ -\mathbf{b} & Q\end{pmatrix} \tag{35}$$

$$M_1=\begin{pmatrix} -1 & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times n}\\ \mathbf{0}_{m\times 1} & I_{m\times m} & \mathbf{0}_{m\times n}\\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times n}\end{pmatrix} \tag{36}$$

$$M_2=\begin{pmatrix} -1 & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times n}\\ \mathbf{0}_{m\times 1} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n}\\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times m} & I_{n\times n}\end{pmatrix}. \tag{37}$$

Then the problem (QP) can be rewritten as

$$\begin{aligned}(QP)\quad \max_{\mathbf{z}}\quad & \mathbf{z}'M_0\mathbf{z} && (38)\\ s.t.\quad & \mathbf{z}'M_1\mathbf{z}\le 0 && (39)\\ & \mathbf{z}'M_2\mathbf{z}\le 0 && (40)\\ & r^2=1. && (41)\end{aligned}$$

Denote the inner product between matrices X and Y as $X\bullet Y:=\operatorname{tr}X'Y$. Then we can further rewrite (QP) as:

$$\begin{aligned}(QP)\quad \min_{\mathbf{z}}\quad & M_0\bullet(\mathbf{z}\mathbf{z}') && (42)\\ s.t.\quad & M_1\bullet(\mathbf{z}\mathbf{z}')\le 0 && (43)\\ & M_2\bullet(\mathbf{z}\mathbf{z}')\le 0 && (44)\\ & I_{00}\bullet(\mathbf{z}\mathbf{z}')=1, && (45)\end{aligned}$$

where $I_{00}=\begin{pmatrix} 1 & \mathbf{0}_{1\times(m+n)}\\ \mathbf{0}_{(m+n)\times 1} & \mathbf{0}_{(m+n)\times(m+n)}\end{pmatrix}$. Then a natural SDP relaxation of (QP) is

$$\begin{aligned}(SP)\quad \min_{X}\quad & M_0\bullet X && (46)\\ s.t.\quad & M_1\bullet X\le 0 && (47)\\ & M_2\bullet X\le 0 && (48)\\ & I_{00}\bullet X=1, && (49)\\ & X\succeq\mathbf{0}. && (50)\end{aligned}$$

Note (SP) is a convex problem, but there may be a gap between the optimal values of (SP) and (QP) because (SP) dropped the rank-one constraint on X. The dual problem of (SP) is

$$\begin{aligned}(SD)\quad \max_{y_0,y_1,y_2}\quad & y_0 && (51)\\ s.t.\quad & Z:=M_0-y_0I_{00}+y_1M_1+y_2M_2\succeq\mathbf{0} && (52)\\ & y_1\ge 0,\ y_2\ge 0. && (53)\end{aligned}$$

With slight abuse of notation, we denote as QP, SP, and SD the optimal objective value of the respective problems. We may also write $QP(A,B,\mathbf{c})$ to make explicit their dependence on $(A,B,\mathbf{c})$.

Clearly $SP = SD$ since the Slater's condition is always met. However, $QP \geq SP$ because (SP) does not necessarily admit a rank-one optimal solution. The key conclusion of this note is to rule out this possibility, and show that

$$QP(A, B, \mathbf{c}) = SP(A, B, \mathbf{c}) \quad \text{for all } A, B, \mathbf{c}, \quad \text{(strong duality)}. \tag{54}$$

So there must be a rank-one optimal solution to (SP), based on which we can easily recover an optimal $\mathbf{z}$ for (QP).

Generalization Note the $\mathbf{b}$ in (34) is determined by A, B and $\mathbf{c}$ and does not have full freedom. In this note we will prove a stronger result by dropping this constraint and consider for general unconstrained $\mathbf{b}$. Accordingly, we will show a slightly more general relationship:

$$QP(A, B, \mathbf{b}) = SP(A, B, \mathbf{b}) \quad \text{for all } A, B, \mathbf{b}, \quad \text{(strong duality)}. \tag{55}$$

Besides proving (55), two computational issues need to be resolved. First, given the optimal $\{y_i\}$ for (SD), how to recover the optimal $(\mathbf{x}, \mathbf{y})$ for (QP). The details are given in Section D.1.3. Second, how to solve (SD). We propose using the cutting plane method. Note there are only three variables in (SD), and the only tricky part is the positive semi-definite constraint (52). For low dimensional convex optimization, it is quite easy to approximate this (nontrivial) constraint by cutting planes, which relies on the oracle: given an assignment of $\{y_i\}$, find a maximal violator of (52), *i.e.* $\operatorname{argmin}_{\mathbf{u}:\|\mathbf{u}\|=1} \mathbf{u}' Z \mathbf{u}$ (≤ 0). The solution is simply the eigenvector corresponding to the least algebraically eigenvalue.

Notation The set of all n-by-n symmetric matrices is denoted as $\mathcal{S}^{n\times n}$, and the set of all n-by-n positive semi-definite matrices is denoted as $\mathcal{S}_+^{n\times n}$. $\det(A)$ is the determinant of a matrix A. Denote the kernel (null space) of a linear map A as $\mathrm{Ker}(A)$, and the range of A as $\mathrm{Im}(A)$ (the span of the column space of A).

D.1 Strong Duality

This section proves the strong duality. Our idea is similar to [41]. We first define a set of Properties (called Property $\mathcal{I}$) over the optimal solutions of (SP) and (SD). Next we show that if Property $\mathcal{I}$ does not hold, then strong duality is guaranteed. Finally we show that in our case, Property $\mathcal{I}$ can never be met.

D.1.1 Property $\mathcal{I}$

Let $\hat{X}$ and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ be a pair of optimal solutions for (SP) and (SD), respectively. The KKT condition states

$$\hat{X}\hat{Z} = \mathbf{0} \tag{56}$$

$$\hat{y}_i M_i \bullet \hat{X} = 0, \qquad i \in \{1, 2\}. \tag{57}$$

We define a Property $\mathcal{I}$ in the same spirit as [41].

Definition 1 *We say the optimal pair $\hat{X}$ and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ has Property $\mathcal{I}$ if:*

1. $M_1 \bullet \hat{X} = 0$ *and* $M_2 \bullet \hat{X} = 0$.
2. $\mathrm{rank}(\hat{Z}) = m + n - 1$.
3. $\mathrm{rank}(\hat{X}) = 2$, *and P3: there is a rank-one decomposition of* $\hat{X}$, $\hat{X} = \mathbf{x}_1\mathbf{x}_1' + \mathbf{x}_2\mathbf{x}_2'$, *such that* $M_1 \bullet \mathbf{x}_i\mathbf{x}_i' = 0$ $(i = 1, 2)$, *and* $(M_2 \bullet \mathbf{x}_1\mathbf{x}_1')(M_2 \bullet \mathbf{x}_2\mathbf{x}_2') < 0$.

The concept of rank-one decomposition is available in subsection D.2. It is simple to symmetrize the item 3 of Property $\mathcal{I}$ (*i.e.* swap the role of M_1 and M_2), but this is not needed for our purposes. Our key result is to use the Property $\mathcal{I}$ to characterize the case of strong duality.

Theorem 3 *If (SP) and (SD) have a pair of optimal solution $\hat{X}$ and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ which do not satisfy Property $\mathcal{I}$, then strong duality holds,* i.e. *$SP = QP$ and (SP) has a rank-one optimal solution.*

Proof: Assume Property $\mathcal{I}$ does not hold, and we enumerate four exhaustive (but not mutually exclusive) possibilities.

Case 1: $M_1 \bullet \hat{X} \neq 0$ or $M_2 \bullet \hat{X} \neq 0$. Without loss of generality, suppose $M_2 \bullet \hat{X} \neq 0$. Then $\hat{y}_2 = 0$ by KKT condition (57). So when solving (SD), we can equivalently clamp y_2 to 0 and optimize only in y_0 and y_1. This corresponds to solving (SP) by ignoring the constraint (48). By [42], all extreme points of the new feasible region of (SP) has rank 1, and so (SP) must have an optimal solution with rank 1.

Case 2: $M_1 \bullet \hat{X} = M_2 \bullet \hat{X} = 0$ and $\text{rank}(\hat{X}) \neq 2$. Let $r = \text{rank}(\hat{X})$. Obviously $r > 0$ since $I_{00}\hat{X} = 1$. If $r = 1$, then (SP) already has a rank-one solution and (QP) is solved. So we only need to consider the case $r \geq 3$. By Proposition 6 with $\delta_1 = \delta_2 = 0$, there is a rank-one decomposition of $\hat{X}$ satisfying

$$\hat{X} = \mathbf{x}_1\mathbf{x}_1' + \mathbf{x}_2\mathbf{x}_2' + \ldots + \mathbf{x}_r\mathbf{x}_r' \tag{58}$$

$$M_1 \bullet \mathbf{x}_i\mathbf{x}_i' = 0, \quad for\ i = 1, \ldots, r \tag{59}$$

$$M_2 \bullet \mathbf{x}_i\mathbf{x}_i' = 0, \quad for\ i = 1, \ldots, r-2. \tag{60}$$

By Proposition 3, we have $Z(\mathbf{x}_1\mathbf{x}_1') = \mathbf{0}$. Let $\mathbf{x}_1 = (t_1, \mathbf{u}_1', \mathbf{v}_1')'$. Then

$$(59) \quad \Rightarrow \quad -s_1^2 + \|\mathbf{u}_1\|^2 = 0 \tag{61}$$

$$(60) \quad \Rightarrow \quad -s_1^2 + \|\mathbf{v}_1\|^2 = 0. \tag{62}$$

So if $s_1 = 0$ then $\mathbf{u}_1 = \mathbf{v}_1 = 0$, which means $\mathbf{x}_1 = \mathbf{0}$. Contradiction. So $s_1 \neq 0$ and we can easily see that $\hat{X}_1 := \mathbf{x}_1\mathbf{x}_1'/s_1^2$ satisfies the KKT conditions (56) and (57), together with $I_{00}\hat{X}_1 = 1$. Hence $\mathbf{x}_1\mathbf{x}_1'/s_1^2$ is a rank-one optimal solution to (SP) and $\mathbf{x}_1/s_1$ is an optimal solution to (QP).

Case 3: $M_1 \bullet \hat{X} = M_2 \bullet \hat{X} = 0$, $\text{rank}(\hat{X}) = 2$, but P3 does not hold. By Proposition 4, there must be a rank-one decomposition $\hat{X} = \mathbf{x}_1\mathbf{x}_1' + \mathbf{x}_2\mathbf{x}_2'$ such that

$$M_1 \bullet (\mathbf{x}_1\mathbf{x}_1') = M_1 \bullet (\mathbf{x}_2\mathbf{x}_2') = 0. \tag{63}$$

So the failure of P3 implies

$$M_2 \bullet \mathbf{x}_1\mathbf{x}_1' = M_2 \bullet \mathbf{x}_2\mathbf{x}_2' = 0, \tag{64}$$

because $M_2 \bullet \mathbf{x}_1\mathbf{x}_1' + M_2 \bullet \mathbf{x}_2\mathbf{x}_2' = M_2 \bullet \hat{X} = 0$. Using exactly the same argument as in Case 2, we conclude that s_1, the first element of $\mathbf{x}_1$, is non-zero, and $\mathbf{x}_1\mathbf{x}_1'/s_1^2$ is a rank-one optimal solution to (SP). Obviously, $\mathbf{x}_2\mathbf{x}_2'/s_2^2$ is also a rank-one solution to (SP), where s_2 is the first element of $\mathbf{x}_2$.

Case 4: $M_1 \bullet \hat{X} = M_2 \bullet \hat{X} = 0$, $\text{rank}(\hat{X}) = 2$, $M_1 \bullet (\mathbf{x}_1\mathbf{x}_1') = M_1 \bullet (\mathbf{x}_2\mathbf{x}_2') = 0$, $(M_2 \bullet \mathbf{x}_1\mathbf{x}_1')(M_2 \bullet \mathbf{x}_2\mathbf{x}_2') < 0$, and $\text{rank}(\hat{Z}) \neq m + n - 1$. By Sylvester's inequality,

$$\text{rank}(\hat{Z}) + \text{rank}(\hat{X}) - (m + n + 1) \leq \text{rank}(\hat{Z}\hat{X}). \tag{65}$$

Now $\text{rank}(\hat{X}) = 2$ and $\hat{Z}\hat{X} = \mathbf{0}$, so $\text{rank}(\hat{Z}) \leq m + n - 1$. Therefore in this particular case $\text{rank}(\hat{Z}) \leq m + n - 2$. So by 0.4.5(d) of [43],

$$\text{rank}(\hat{X} + \hat{Z}) \leq \text{rank}(\hat{X}) + \text{rank}(\hat{Z}) \tag{66}$$

$$\leq 2 + (m + n - 2) = m + n. \tag{67}$$

Thus there must be a $\mathbf{y} \neq \mathbf{0}$ such that $(\hat{X} + \hat{Z})\mathbf{y} = \mathbf{0}$, and

$$\mathbf{y}'\hat{X}\mathbf{y} + \mathbf{y}'\hat{Z}\mathbf{y} = \mathbf{y}'(\hat{X} + \hat{Z})\mathbf{y} = 0. \tag{68}$$

Since both $\hat{X}$ and $\hat{Z}$ are positive semi-definite, we conclude that $\mathbf{y} \in \text{Ker}(\hat{X}) \cap \text{Ker}(\hat{Z})$. Now define

$$X := \hat{X} + \mathbf{y}\mathbf{y}' = \mathbf{x}_1\mathbf{x}_1' + \mathbf{x}_2\mathbf{x}_2' + \mathbf{y}\mathbf{y}'. \tag{69}$$

Obviously $\text{rank}(X) = 3$ and $\hat{Z}X = 0$. Since

$$M_1 \bullet (\mathbf{x}_1\mathbf{x}_1') = M_1 \bullet (\mathbf{x}_2\mathbf{x}_2') = 0 \tag{70}$$

$$(M_2 \bullet \mathbf{x}_1\mathbf{x}_1')(M_2 \bullet \mathbf{x}_2\mathbf{x}_2') < 0, \tag{71}$$

so by Proposition 5 with $\delta_1 = \delta_2 = 0$, there must be an $\mathbf{x}$ such that X is rank-one decomposable at $\mathbf{x}$ and

$$M_1 \bullet \mathbf{x}\mathbf{x}' = 0, \quad M_2 \bullet \mathbf{x}\mathbf{x}' = 0. \tag{72}$$

Since $\hat{Z}X = \mathbf{0}$, Proposition 3 implies $\hat{Z}\mathbf{x} = \mathbf{0}$ and so $\hat{Z} \bullet \mathbf{x}\mathbf{x}' = 0$. Based on the satisfaction of the KKT conditions (56) and (57), we conclude that $\mathbf{x}\mathbf{x}'/s^2$ is a rank-one optimal solution to (SP), where s is the first element of $\mathbf{x}$. s must be non-zero because of (72) and the same argument as in Case 2. ■

D.1.2 Strong Duality

Let us denote $(A, B, \mathbf{b})$ collectively as $\Gamma := (A, B, \mathbf{b})$, and define a "Frobenius" norm on Γ as $\|\Gamma\|^2 := \|A\|_F^2 + \|B\|_F^2 + \|\mathbf{b}\|^2$. Ideally we wish to show that for any Γ, the Property $\mathcal{I}$ does not hold for some solutions to (SP) and (SD), hence strong duality holds (Theorem 3). However, this is hard. So we resort to the argument of ϵ-perturbation.

Before proceeding, we first make a very simple rewriting of (QP). Let $p = \max\{t, m, n\}$. By padding zeros if necessary, we can expand A and B into p-by-p dimensional matrices, and $\mathbf{c}$ into an p dimensional vector. Let $\mathbf{x}$ and $\mathbf{y}$ be p dimensional too. Obviously, the optimal values of (QP) and (SP) in this new problem are the same as those in the original problem, respectively. Therefore, henceforth we will only consider square matrices A and B. For notational convenience, we just call all t, m, and n as n.

Of key importance is the Danskin's theorem.

Lemma 1 (Danskin) *Suppose $f : Z \times \Omega \mapsto \mathbb{R}$ is a continuous function, where $Z \subseteq \mathbb{R}^n$ is a compact set and $\Omega \subseteq \mathbb{R}^m$ is an open set. For any $\mathbf{z}$, $\nabla_\omega f(\mathbf{z}, \omega)$ exists and is continuous. Then the marginal function*

$$\phi(\omega) := \max_{\mathbf{z} \in Z} f(\mathbf{z}, \omega) \tag{73}$$

is continuous.

Note that Danskin's theorem does not require convexity. Let the $\mathbf{z}$ in Lemma 1 correspond to $(\mathbf{x}', \mathbf{y}')'$ in (QP), ω to Γ, Z to $\{\mathbf{x} : \|\mathbf{x}\| \leq 1\} \times \{\mathbf{y} : \|\mathbf{y}\| \leq 1\}$, and Ω to the whole Euclidean space. Then Lemma 1 implies that $QP(\Gamma)$ is continuous in Γ. Similarly, $SP(\Gamma)$ is continuous.

The continuity at Γ means that for any $\epsilon > 0$, there exists $\delta > 0$, such that for all $\hat{\Gamma}$ in the δ neighborhood of Γ:

$$\mathcal{B}_\delta(\Gamma) := \left\{ \hat{\Gamma} : \left\| \hat{\Gamma} - \Gamma \right\| < \delta \right\}, \tag{74}$$

we have

$$\left| QP(\hat{\Gamma}) - QP(\Gamma) \right| < \epsilon, \tag{75}$$

$$\left| SP(\hat{\Gamma}) - SP(\Gamma) \right| < \epsilon. \tag{76}$$

Our key result will be the following theorem.

Theorem 4 *For any Γ and $\delta > 0$, there exists $\Gamma_\delta \in \mathcal{B}_\delta(\Gamma)$ such that strong duality holds at Γ_δ:*

$$QP(\Gamma_\delta) = SP(\Gamma_\delta). \tag{77}$$

Using Theorem 4, we can easily prove strong duality.

Corollary 1 *$QP(\Gamma) = SP(\Gamma)$ for all Γ.*

Proof: It suffices to show that for any $\epsilon > 0$,

$$|QP(\Gamma) - SP(\Gamma)| < 2\epsilon. \tag{78}$$

By continuity of QP and SP, there exists a $\delta > 0$, such that (75) and (76) hold for all $\hat{\Gamma} \in \mathcal{B}_\delta(\Gamma)$. By Theorem 4, there exists $\Gamma_\delta \in \mathcal{B}_\delta(\Gamma)$ such that (77) holds. Combining it with (75) and (76) (with $\hat{\Gamma} = \Gamma_\delta$), we obtain (78). ■

Finally we prove Theorem 4.

Proof: Clearly $\mathcal{B}_{\delta/2}(A, B, \mathbf{b})$ contains invertible matrices for any A, B, and $\delta > 0$. Arbitrarily pick two such matrices and call them A_δ and B_δ. By Theorem 3, to establish (77) it suffices to show that the corresponding (SP) and (SD) problems at (A_δ, B_δ) have a pair of optimal solutions $\hat{X}$ and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ which do not satisfy Property $\mathcal{I}$. We will focus on the second condition: $\mathrm{rank}(\hat{Z}) = 2n - 1$.

If $\mathrm{rank}(\hat{Z}) \neq 2n-1$, then by Theorem 3 strong duality holds at $\Gamma_\delta := (A_\delta, B_\delta, \mathbf{b})$. Otherwise suppose $\mathrm{rank}(\hat{Z}) = 2n - 1$. Noting (52), we have

$$\hat{Z} = M_0 - \hat{y}_0 I_{00} + \hat{y}_1 M_1 + \hat{y}_2 M_2 = \begin{pmatrix} -\hat{y}_0 - \hat{y}_1 - \hat{y}_2 & -\mathbf{b}' \\ -\mathbf{b} & R \end{pmatrix}, \tag{79}$$

$$\text{where} \quad R = \begin{pmatrix} \hat{y}_1 I - A_\delta' A_\delta & -A_\delta' B_\delta \\ -B_\delta' A_\delta & \hat{y}_2 I - B_\delta' B_\delta \end{pmatrix}. \tag{80}$$

Note that for any given y_1 and y_2, (SD) maximizes y_0 subject to $\hat{Z} \succeq \mathbf{0}$. By Proposition 7, we know that

$$2n - 1 = \mathrm{rank}(\hat{Z}) = \mathrm{rank}(R). \tag{81}$$

Denote $P = \hat{y}_1 I - A_\delta' A_\delta$ and $Q = \hat{y}_2 I - B_\delta' B_\delta$. Then by Proposition 8, we have $\mathrm{rank}(P) + \mathrm{rank}(Q) = 2n - 1$ or $2n$. Now we discuss three cases.

Case 1: $\mathrm{rank}(P) = n$ and $\mathrm{rank}(Q) = n-1$. By Schur complement, we have $Q \succeq B_\delta' A_\delta P^{-1} A_\delta' B_\delta$. So by Exercise 4.3.14 of [43],

$$\lambda_{\min}(Q) \geq \lambda_{\min}(B_\delta' A_\delta P^{-1} A_\delta' B_\delta), \tag{82}$$

where $\lambda_{\min}$ stands for the smallest eigenvalue. Since A_δ and B_δ are both invertible, $B_\delta' A_\delta P^{-1} A_\delta' B_\delta$ must be positive definite and its smallest eigenvalue is strictly positive. But $\mathrm{rank}(Q) = n - 1$, meaning the minimum eigenvalue of Q is 0. So contraction with (82).

Case 2: $\mathrm{rank}(P) = n - 1$ and $\mathrm{rank}(Q) = n$. Same argument as for Case 1.

Case 3: $\mathrm{rank}(P) = \mathrm{rank}(Q) = n$. Since $\mathrm{rank}(R) = 2n-1$, R must have an eigen-vector $\mathbf{u}_0$ whose corresponding eigen-value is 0. In fact $\mathbf{u}_0$ is unique up to negation. By Proposition 7, $\mathbf{b} \in \mathrm{Im}(R)$, so $\mathbf{b}'\mathbf{u}_0 = 0$. Now perturb the $\mathbf{b}$ in Z in the direction of $\mathbf{u}_0$:

$$\hat{Z}(t) = \begin{pmatrix} -\hat{y}_0(t) - \hat{y}_1(t) - \hat{y}_2(t) & -\mathbf{b}' - t\mathbf{u}_0' \\ -\mathbf{b} - t\mathbf{u}_0 & R(t) \end{pmatrix}, \quad t \in \mathbb{R}, \tag{83}$$

where $\hat{y}_i(t)$ are the optimal solutions for $SD(A_\delta, B_\delta, \mathbf{b} + t\mathbf{u}_0)$ and $R(t)$ uses $\hat{y}_i(t)$. Denote $P(t) = \hat{y}_1(t) I - A_\delta' A_\delta$ and $Q(t) = \hat{y}_2(t) I - B_\delta' B_\delta$. If there exists $t \in (-\delta/2, \delta/2)$ such that $\mathrm{rank}(\hat{Z}(t)) \neq 2n-1$, then $(A_\delta, B_\delta, \mathbf{b} + t\mathbf{u}_0)$ is the Γ_δ in Theorem 4. Otherwise, $\mathrm{rank}(\hat{Z}(t)) = 2n-1$ for all $|t| < \delta/2$ and by the same argument as in Case 1 and 2, we conclude that $\mathrm{rank}(P(t)) = \mathrm{rank}(Q(t)) = n$, $\forall\ t$. Since $\mathrm{rank}(R(t)) = \mathrm{rank}(\hat{Z}(t)) = 2n - 1$, $R(t)$ must have an eigen-vector $\mathbf{u}(t)$ whose corresponding eigen-value is 0. Clearly $\mathbf{u}(t)$ is unique up to the sign, and we can set $\mathbf{u}(0) = \mathbf{u}_0$. By Proposition 7, $\mathbf{b} + t\mathbf{u}_0$ must be in the range of $R(t)$. If we can show that $\mathbf{u}(t) = (1 + ct)\mathbf{u}_0 + \mathbf{o}(t)$ where $\lim_{t \to 0} \mathbf{o}(t)/t = \mathbf{0}$ and $c \in \mathbb{R}$ is independent of t, then

$$0 = (\mathbf{b} + t\mathbf{u}_0)'\mathbf{u}(t) = (\mathbf{b} + t\mathbf{u}_0)'((1 + ct)\mathbf{u}_0 + \mathbf{o}(t)) = t + ct^2 + \mathbf{b}'\mathbf{o}(t) + t\mathbf{u}_0'\mathbf{o}(t). \tag{84}$$

Dividing both sides by t and driving t to 0, we get $0 = 1 + 0 + 0 + 0$. Contradiction.

To show $\mathbf{u}(t) = (1 + ct)\mathbf{u}_0 + \mathbf{o}(t)$, we need to analyze the gradient of $\mathbf{u}(t)$ at $t = 0$. First we show $\hat{y}_i(t)$ is differentiable in t at $t = 0$ for $i = 1, 2$. Since $\mathrm{rank}(P(t)) = \mathrm{rank}(Q(t)) = n$ and

$\operatorname{rank}(R(t)) = 2n-1$, we have $0 = \det(R(t)) = \det(P(t)) \cdot \det(Q(t) - B_\delta' A_\delta P(t)^{-1} A_\delta' B_\delta)$. In conjunction with Schur complement, we get

$$\hat{y}_2(t) = \lambda_{\max}\left(B_\delta' B_\delta + B_\delta' A_\delta (\hat{y}_1(t) I - A_\delta' A_\delta)^{-1} A_\delta' B_\delta\right), \tag{85}$$

$$\hat{y}_1(t) = \lambda_{\max}\left(A_\delta' A_\delta + A_\delta' B_\delta (\hat{y}_2(t) I - B_\delta' B_\delta)^{-1} B_\delta' A_\delta\right). \tag{86}$$

So a larger $\hat{y}_1(t)$ implies a smaller $\hat{y}_2(t)$ and a smaller $\hat{y}_1(t)$ implies a larger $\hat{y}_2(t)$. By Proposition 7,

$$(\hat{y}_1(t), \hat{y}_2(t)) = \operatorname*{argmin}_{y_1, y_2} y_1 + y_2 + (\mathbf{b} + t\mathbf{u}_0)' \begin{pmatrix} y_1 I - A_\delta' A_\delta & -A_\delta' B_\delta \\ -B_\delta' A_\delta & y_2 I - B_\delta' B_\delta \end{pmatrix}^\dagger (\mathbf{b} + t\mathbf{u}_0). \tag{87}$$

In general, pseudo-inverse is not even continuous. However, since we know that $\operatorname{rank}(R(t)) = 2n-1$ (constant rank), so the pseudo-inverse is differentiable in $R(t)$ [44]. So $\hat{y}_1(t)$ and $\hat{y}_2(t)$ are differentiable in t at $t = 0$.

By Theorem 1 of [45], we know there exists a choice of the sign for $\mathbf{u}(t)$ which satisfies

$$\left.\frac{\partial \mathbf{u}(t)}{\partial t}\right|_{t=0} = \mathbf{u}_0 \sum_{ij} A_{ij} \left.\frac{\partial R_{ij}(t)}{\partial t}\right|_{t=0}, \quad \text{where} \quad A = -R(0)^\dagger \tag{88}$$

$$= \mathbf{u}_0 \left(\hat{y}_1'(0) \sum_{i=1}^{n} A_{ii} + \hat{y}_2'(0) \sum_{i=n+1}^{2n} A_{ii}\right). \tag{89}$$

Setting $c := \hat{y}_1'(0) \sum_{i=1}^{n} A_{ii} + \hat{y}_2'(0) \sum_{i=n+1}^{2n} A_{ii}$ yields $\mathbf{u}(t) = (1+ct)\mathbf{u}_0 + \mathbf{o}(t)$. ■

D.1.3 Recovering the optimal solution

With the guarantee of strong duality, an algorithm is needed to recover a rank-one optimal solution to (SP) when given an optimal dual solution $\hat{Z}$ to (SD). By the KKT condition, all we need is two vectors $\mathbf{x}$ and $\mathbf{y}$ satisfying:

$$\mathbf{z}' \hat{Z} \mathbf{z} = 0, \quad \|\mathbf{x}\| \le 1, \quad \|\mathbf{y}\| \le 1, \tag{90}$$

where $\mathbf{z} = (1, \mathbf{x}', \mathbf{y}')'$. Note this is a necessary and sufficient condition for optimal $\mathbf{x}$ and $\mathbf{y}$. Since $\hat{Z}$ is positive semi-definite, $\mathbf{z}$ must be in the null space of $\hat{Z}$. Suppose $\operatorname{Ker}(\hat{Z})$ is spanned by $(\mathbf{g}_1, \ldots, \mathbf{g}_k)$. Let

$$G = (\mathbf{g}_1, \ldots, \mathbf{g}_k) = \begin{pmatrix} G_0 \\ G_X \\ G_Y \end{pmatrix}. \tag{91}$$

Then it suffices to find $\boldsymbol{\alpha} \in \mathbb{R}^k$ such that $|G_0 \boldsymbol{\alpha}| = 1$, $\|G_X \boldsymbol{\alpha}\| = 1$, and $\|G_Y\| \, \boldsymbol{\alpha} = 1$. To this end, we only need to find $\boldsymbol{\alpha}$ satisfying

$$\boldsymbol{\alpha}'(G_X' G_X - G_0' G_0)\boldsymbol{\alpha} = 0 \tag{92}$$

$$\boldsymbol{\alpha}'(G_Y' G_Y - G_0' G_0)\boldsymbol{\alpha} = 0 \tag{93}$$

$$G_0 \boldsymbol{\alpha} \neq 0, \tag{94}$$

and then scale it properly. In the sequel, we will first find $\boldsymbol{\alpha}$ which satisfies the first two conditions and then show how to satisfy the last one. Denote $\tilde{S} = G_X' G_X - G_0' G_0$ and $\tilde{T} = G_Y' G_Y - G_0' G_0$. Let their algebraically smallest eigenvalues be s_X and s_Y, and define $s = 1 - \min(s_X, s_Y)$. Then $S := \tilde{S} + sI$ and $T := \tilde{T} + sI$ must be positive definite, and $\boldsymbol{\alpha}$ only needs to satisfy

$$\boldsymbol{\alpha}' S \boldsymbol{\alpha} = s \boldsymbol{\alpha}' \boldsymbol{\alpha} \tag{95}$$

$$\boldsymbol{\alpha}' T \boldsymbol{\alpha} = s \boldsymbol{\alpha}' \boldsymbol{\alpha} \tag{96}$$

$$G_0 \boldsymbol{\alpha} \neq 0. \tag{97}$$

Denote $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}/\left\|\boldsymbol{\alpha}\right\|$, then it is equivalent to

$$\hat{\boldsymbol{\alpha}}' S \hat{\boldsymbol{\alpha}} = s \tag{98}$$

$$\hat{\boldsymbol{\alpha}}' T \hat{\boldsymbol{\alpha}} = s \tag{99}$$

$$G_0 \hat{\boldsymbol{\alpha}} \neq 0. \tag{100}$$

Because both S and T are positive semidefinite, by [43, Corollary 4.6.12] there exists a nonsingular matrix R such that $RSR' = I$ and RTR' is real diagonal. In fact this R can be constructed analytically. Let S have eigen-decomposition $S = UDU'$ where D is diagonal. Denote $H = U\sqrt{D}U'$ and let HTH have eigen-decomposition $HTH = V\Lambda V'$. Then R can be simply chose as $R = V'H^{-1} = VUD^{-1/2}U'$. Let RTR' be $\operatorname{diag}\{\sigma_i\}$. Denote $\boldsymbol{\beta} = R\hat{\boldsymbol{\alpha}}$, then $\boldsymbol{\beta}$ only needs to satisfy

$$\boldsymbol{\beta}'\boldsymbol{\beta} = s \tag{101}$$

$$\boldsymbol{\beta}'\Sigma\boldsymbol{\beta} = s \tag{102}$$

$$G_0 R^{-1}\boldsymbol{\beta} \neq 0. \tag{103}$$

It is easy to find a $\boldsymbol{\beta}$ which satisfies the first two constraints, because it is guaranteed that there exists a $\boldsymbol{\beta}$ which satisfies all the three conditions. Once we get such a $\boldsymbol{\beta}$, suppose $G_0 R^{-1}\boldsymbol{\beta} = 0$. Then we can flip the sign of one of its nonzero components. If its product with $G_0 R^{-1}$ is still 0, then it means the corresponding entry in $G_0 R^{-1}$ is 0. But $G_0 R^{-1}$ cannot be straight 0 because that would imply G_0 is a zero vector which violates the assumption that G is the basis of $\operatorname{Ker}(\hat{Z})$. Therefore we can always find a $\boldsymbol{\beta}$ which satisfies (101) to (103).

D.2 Preliminaries in Matrix Analysis

D.2.1 Matrix Rank-one decomposition

Let X be a n-by-n positive semi-definite matrix with $\operatorname{rank}(X) = r$. Then a set of r vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ in $\mathbb{R}^n$ is called a rank-one decomposition of X if $X = \sum_{i=1}^r \mathbf{x}_i\mathbf{x}_i'$.

It is noteworthy that the $\mathbf{x}_i$'s are not necessarily orthogonal to each other ($\mathbf{x}_i'\mathbf{x}_j = 0$ for $i \neq j$), but they must be linearly independent. This leads to the following useful result.

Proposition 3 *Suppose $ZX = \mathbf{0}$ and $\{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ is a rank-one decomposition of X. Then $Z\mathbf{x}_i = \mathbf{0}$, $\forall\, i$.*

Proof: Denote $\mathbf{y}_i := Z\mathbf{x}_i$. Suppose otherwise $\mathbf{y}_1 \neq \mathbf{0}$. Since $ZX = \mathbf{0}$, we have

$$\mathbf{0} = X'Z'\mathbf{y}_1 = \sum_{i=1}^r \mathbf{x}_i\mathbf{x}_i'Z'\mathbf{y}_1 = \sum_{i=1}^r (\mathbf{y}_i'\mathbf{y}_1)\mathbf{x}_i. \tag{104}$$

Since $\mathbf{y}_1 \neq \mathbf{0}$, this violates the linear independence of $\mathbf{x}_1, \ldots, \mathbf{x}_r$. ■

X is called rank-one decomposable at a vector $\mathbf{x}_1$ if there exist other $r-1$ vectors $\mathbf{x}_2, \ldots, \mathbf{x}_r$ such that $X = \sum_{i=1}^r \mathbf{x}_i\mathbf{x}_i'$.

The following three theorems play an important role in our proof.

Proposition 4 (Corollary 4 of [46]) *Suppose $X \in \mathcal{S}_+^{n\times n}$ with $\operatorname{rank}(X) = r$. $Z \in \mathcal{S}^{n\times n}$ and $Z \bullet X \geq 0$. Then there must be a rank-one decomposition of $X = \mathbf{x}_1\mathbf{x}_1' + \ldots + \mathbf{x}_r\mathbf{x}_r'$ such that $Z \bullet (\mathbf{x}_i\mathbf{x}_i') = Z \bullet X/r$ for all $i = 1, \ldots, r$.*

Proposition 5 (Lemma 3.3 of [41]) *Suppose $X \in \mathcal{S}_+^{n\times n}$ with rank $r \geq 3$. $A_1, A_2 \in \mathcal{S}^{n\times n}$. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ be a rank-one decomposition of X. If*

$$A_1 \bullet \mathbf{x}_1\mathbf{x}_1' = A_1 \bullet \mathbf{x}_2\mathbf{x}_2' = \delta_1 \tag{105}$$

$$(A_2 \bullet \mathbf{x}_1\mathbf{x}_1' - \delta_2)(A_2 \bullet \mathbf{x}_2\mathbf{x}_2' - \delta_2) < 0, \tag{106}$$

then there is a vector $\mathbf{y} \in \mathbb{R}^n$ such that X is rank-one decomposable at $\mathbf{y}$ and

$$A_1 \bullet \mathbf{y}\mathbf{y}' = \delta_1, \qquad A_2 \bullet \mathbf{y}\mathbf{y}' = \delta_2. \tag{107}$$

Proposition 6 (Theorem 3.4 of [41]) *Suppose $X \in \mathcal{S}_+^{n\times n}$ with rank $r \geq 3$. $A_1, A_2 \in \mathcal{S}^{n\times n}$. If*

$$A_1 \bullet X = \delta_1, \qquad A_2 \bullet X = \delta_2, \tag{108}$$

then X has a rank-one decomposition $\{\mathbf{x}_1, \ldots, \mathbf{x}_r\}$ such that

$$A_1 \bullet \mathbf{x}_i\mathbf{x}_i' = \delta_1/r \quad for\ i = 1, \ldots, r, \tag{109}$$
$$A_2 \bullet \mathbf{x}_i\mathbf{x}_i' = \delta_2/r \quad for\ i = 1, \ldots, r-2. \tag{110}$$

D.2.2 Bounding the rank of block matrices

Proposition 7 *Let $X \in \mathcal{S}_+^{n\times n}$ and $\mathbf{b} \in im(X)$. Define*

$$Y(c) = \begin{pmatrix} c & \mathbf{b}' \\ \mathbf{b} & X \end{pmatrix}, \qquad c \in \mathbb{R}. \tag{111}$$

Suppose $Y(c) \succeq \mathbf{0}$ and $\mathrm{rank}(X) = r$. Then

$$\mathrm{rank}(Y(c)) \in \{r, r+1\}. \tag{112}$$

Furthermore, if c^* is the minimum value such that $Y(c) \succeq \mathbf{0}$:

$$c^* = \operatorname*{arginf}_{c: Y(c)\succeq \mathbf{0}} c, \tag{113}$$

then we have $\mathrm{rank}(Y(c^*)) = r$.

Finally, if $\mathbf{b} \notin im(X)$, then $Y(c) \succeq \mathbf{0}$ cannot hold for any $c \in \mathbb{R}$.

Proof: Since adding rows and columns to a matrix will not decrease its rank, so obviously $\mathrm{rank}(Y(c)) \geq \mathrm{rank}(X) = r$. To show $\mathrm{rank}(Y(c)) \leq r+1$, let the eigenvalues of X and $Y(c)$ be $\lambda_1, \ldots, \lambda_n$ and $\hat{\lambda}_1, \ldots, \hat{\lambda}_{n+1}$, both in increasing order. Then by Theorem 4.3.8 of [43], we have

$$\hat{\lambda}_1 \leq \lambda_1 \leq \hat{\lambda}_2 \leq \lambda_2 \leq \ldots \hat{\lambda}_n \leq \lambda_n \leq \hat{\lambda}_{n+1}. \tag{114}$$

Since $\mathrm{rank}(X) = r$ and $X \in \mathcal{S}_+^{n\times n}$, so $\lambda_1 = \ldots = \lambda_{n-r} = 0$. As $Y(c) \succeq \mathbf{0}$, we have

$$0 \leq \hat{\lambda}_1 \leq \ldots \leq \hat{\lambda}_{n-r} \leq 0. \tag{115}$$

Therefore $\mathrm{rank}(Y(c)) \leq (n+1) - (n-r) = r+1$.

As for the second part, we can actually compute c^* explicitly. $Y(c) \succeq \mathbf{0}$ if and only if $(\alpha, \mathbf{u}')Y(c)(\alpha, \mathbf{u}')' \geq 0$ for all $\alpha \in \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^n$, *i.e.*

$$c\alpha^2 + 2\alpha\mathbf{b}'\mathbf{u} + \mathbf{u}'X\mathbf{u} \geq 0, \quad \forall\, \alpha \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^n. \tag{116}$$

If $\alpha = 0$, this must hold true since $X \succeq \mathbf{0}$. Otherwise,

$$c^* = \max_{\alpha\neq 0, \mathbf{u}} \frac{-\mathbf{u}'X\mathbf{u} - 2\alpha\mathbf{b}'\mathbf{u}}{\alpha^2} \tag{117}$$
$$= \max_{\mathbf{z}} -\mathbf{z}'X\mathbf{z} - 2\mathbf{b}'\mathbf{z} \tag{118}$$
$$= \begin{cases} \mathbf{b}'X^\dagger\mathbf{b} & \text{if } \mathbf{b} \in im(X) \\ \infty & \text{if } \mathbf{b} \notin im(X) \end{cases}, \tag{119}$$

where $X^\dagger$ is the pseudo-inverse of X. To prove $\mathrm{rank}(Y(c^*)) = r$, it suffices to show that $\mathrm{Ker}(Y(c^*)) = n - r + 1$. Towards this end first note

$$Y(c^*)\begin{pmatrix} -1 \\ X^\dagger\mathbf{b} \end{pmatrix} = \begin{pmatrix} \mathbf{b}'X^\dagger\mathbf{b} & \mathbf{b}' \\ \mathbf{b} & X \end{pmatrix}\begin{pmatrix} -1 \\ X^\dagger\mathbf{b} \end{pmatrix} \tag{120}$$
$$= \begin{pmatrix} 0 \\ -\mathbf{b} + XX^\dagger\mathbf{b} \end{pmatrix} = \mathbf{0}. \tag{121}$$

where the last step also used $\mathbf{b} \in im(X)$. Hence $\begin{pmatrix} -1 \\ X^\dagger\mathbf{b} \end{pmatrix} \in \mathrm{Ker}(Y(c^*))$.

Furthermore, $\mathrm{rank}(X) = r$ implies there are $n - r$ linearly independent vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{n-r} \in \mathrm{Ker}(X)$. As $\mathbf{b} \in im(X)$, so $\mathbf{b}'\mathbf{u}_i = 0$ for all i. Therefore

$$Y(c^*) \begin{pmatrix} 0 \\ \mathbf{u}_i \end{pmatrix} = \begin{pmatrix} c^* & \mathbf{b}' \\ \mathbf{b} & X \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{u}_i \end{pmatrix} = \begin{pmatrix} \mathbf{b}'\mathbf{u}_i \\ X\mathbf{u}_i \end{pmatrix} = \mathbf{0}. \tag{122}$$

Clearly $\begin{pmatrix} -1 \\ X^\dagger \mathbf{b} \end{pmatrix}, \begin{pmatrix} 0 \\ \mathbf{u}_1 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ \mathbf{u}_{n-r} \end{pmatrix}$ are linearly independent, so $\mathrm{Ker}(Y(c^*)) \geq n-r+1$, *i.e.* $\mathrm{rank}(Y(c^*)) \leq r$.

Finally, it is obvious from (119) that no $c \in \mathbb{R}$ makes $Y(c) \succeq \mathbf{0}$ if $\mathbf{b} \notin im(X)$. ■

Proposition 8 *Let P, Q, R be n-by-n matrices, and*

$$Z = \begin{pmatrix} P & R \\ R' & Q \end{pmatrix}. \tag{123}$$

Suppose $Z \succeq \mathbf{0}$ and $\mathrm{rank}(Z) = 2n - 1$. Denote $r = \mathrm{rank}(P)$ and $s = \mathrm{rank}(Q)$. Then $r + s \in \{2n - 1, 2n\}$.

Proof: Let $\mathrm{Ker}(P)$ be spanned by $\mathbf{u}_1, \ldots, \mathbf{u}_{n-r}$, and $\mathrm{Ker}(Q)$ be spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_{n-s}$. Denote $\hat{\mathbf{u}}_i = \begin{pmatrix} \mathbf{u}_i \\ \mathbf{0} \end{pmatrix}$ and $\hat{\mathbf{v}}_i = \begin{pmatrix} \mathbf{0} \\ \mathbf{v}_i \end{pmatrix}$. Then

$$\hat{\mathbf{u}}_i' Z \hat{\mathbf{u}}_i = \mathbf{u}_i' P \mathbf{u}_i' = 0. \tag{124}$$

Since $Z \succeq \mathbf{0}$, so $\hat{\mathbf{u}}_i \in \mathrm{Ker}(Z)$. Similarly $\hat{\mathbf{v}}_i \in \mathrm{Ker}(Z)$. Clearly $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{n-r}, \hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_{n-s}$ are linearly independent, therefore

$$2n - 1 = \mathrm{rank}(Z) \leq 2n - (n - r) - (n - s) = r + s. \tag{125}$$

So $r + s \in \{2n - 1, 2n\}$. ■

Auxiliary References

[36] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[37] K. L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. In *SODA*, 2008.

[38] T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. Information Theory*, 49(3):682–691, 2003.

[39] M. Dudik, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularizations. In *AISTATS*, 2012.

[40] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. arXiv:0812.1869v1, 2008.

[41] W. Ai and S. Zhang. Strong duality for the CDT subproblem: A necessary and sufficient condition. *SIAM Journal on Optimization*, 19:1735–1756, 2009.

[42] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358, 1998.

[43] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[44] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.

[45] J. R. Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(2): 179–191, 1985.

[46] J. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28:246–267, 2003.