

# Approximating semidefinite packing programs \*

G. Iyengar † D. J. Phillips ‡ and C. Stein§

## Abstract

In this paper we define *semidefinite packing programs* and describe an algorithm to approximately solve these problems. Semidefinite packing programs arise in many applications such as semidefinite programming relaxations for combinatorial optimization problems, sparse principal component analysis, and sparse variance unfolding technique for dimension reduction. Our algorithm exploits the structural similarity between semidefinite packing programs and linear packing programs.

## 1 Introduction

In this paper we are concerned with solving optimization problems of the form

$$\begin{aligned} \max \quad & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} \quad & g_i(\mathbf{X}) \leq b_i, \quad i = 1, \dots, m, \\ & \mathbf{X} \succeq \mathbf{0}, \end{aligned} \tag{1}$$

where  $\mathbf{C} \in \mathbb{R}^{n \times n}$  is a symmetric, positive semidefinite matrix,  $\mathbf{X} \in \mathbb{R}^{n \times n}$  is the decision variable, and  $g_i(\mathbf{X})$  are *packing functions*. The class of packing functions is formally defined in Definition 1, and includes as special cases:  $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle$ ,  $\mathbf{A} \succeq \mathbf{0}$ ,  $g(\mathbf{X}) = \left( \sum_{i=1}^k (\langle \mathbf{A}_i, \mathbf{X} \rangle)^2 \right)^{\frac{1}{2}}$ , and  $g(\mathbf{X}) = \sum_{i,j=1}^n |X_{ij}|$ . The *Frobenius inner product* between symmetric matrices  $\mathbf{A}$ , and  $\mathbf{B}$  is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}(\mathbf{A}^\top \mathbf{B}) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}. \tag{2}$$

The constraint  $\mathbf{X} \succeq \mathbf{0}$  indicates that the matrix  $\mathbf{X}$  is *positive semidefinite*, i.e.,  $\mathbf{X}$  is symmetric and has nonnegative eigenvalues, or, equivalently,  $\mathbf{v}^\top \mathbf{X} \mathbf{v} \geq 0$  for all  $\mathbf{v} \in \mathbb{R}^n$ . We refer to semidefinite optimization problems of the form (1) as *packing SDPs*. Packing SDPs arise naturally in many applications, including semidefinite programming relaxations for combinatorial optimization problems, sparse principal component analysis and sparse variance unfolding technique for dimension reduction. (See Section 2.1 for a detailed discussion of optimization problems that can be reformulated as packing SDPs.)

Our contributions in this paper are as follows.

- (a) In Section 2.1 we show that a large class of optimization problems can be recast as semidefinite packing problems. Our solution algorithm treats these packing SDPs in a unified manner, and for some important classes of instances, the performance of our solution algorithm is significantly superior to any other previously known methods.

---

\*A preliminary version of this paper appeared in the proceedings of the Eleventh Conference on Integer Programming and Combinatorial Optimization [14].

†Department of IEOR, Columbia University, New York, NY. [garud@ieor.columbia.edu](mailto:garud@ieor.columbia.edu). Supported in part by NSF grants CCR-00-09972, DMS-01-04282 and ONR grant N000140310514

‡Mathematics Department, The College of William & Mary, [phillips@math.wm.edu](mailto:phillips@math.wm.edu). Supported in part by NSF grants CCF-0728733, DMS-0703532, DGE-0086390 and DMI-9970063.

§Department of IEOR, Columbia University, New York, NY. [cliff@ieor.columbia.edu](mailto:cliff@ieor.columbia.edu). Supported in part by NSF grants CCF-0728733 and DMI-9970063.

- (b) We develop a technique for solving a packing SDP to an absolute error  $\epsilon$ . We first solve the Lagrangian relaxation of the packing SDP and then suitably “round” the solution of the relaxation to an  $\epsilon$ -optimal feasible solution to the original packing SDP. In Section 3 we show how to recover an  $\epsilon$ -optimal solution for the packing SDP from an  $\epsilon$ -optimal solution of the Lagrangian dual. In Section 4 we show how to extend a first-order procedure proposed by Nesterov [23] to compute  $\epsilon$ -optimal solutions to the Lagrangian dual of the packing SDP. Our solution method is able to take advantage of any sparsity in the problem, i.e. sparsity in  $\mathbf{C}$  or sparsity in computing the packing functions  $g_i(\mathbf{X})$ . Since our method is a variant of the Nesterov procedure, the method computes an approximate solution even when the gradients are only approximately computed [6]. In addition, after reading in the problem data, the complexity of our method is *logarithmic* in the number of constraints.
- (c) In Section 5 we show that for specific applications the semidefinite packing approach yields faster solution algorithms. We show that an  $\epsilon$ -optimal solution to the semidefinite relaxation to the MAXCUT problem can be computed in  $\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$  time, where  $r$  denotes the number of non-zero elements in the Laplacian matrix of the graph. The previous result for a first-order technique is  $\mathcal{O}(nr \log^2(n) \cdot \epsilon^{-2})$  by Klein and Lu [17]. Recently, a result by Trevisan [27] has allowed a randomized algorithm of Arora and Kale [1] to be extended to general MAXCUT and runs in  $\mathcal{O}(r \log^2(n) \cdot \epsilon^{-6} \log^3(\epsilon^{-1}))$  time [15]. Thus, we have a trade-off – for moderate  $\epsilon$  the Klein-Lu and Arora-Kale-Trevisan bounds are superior, but as  $\epsilon$  decreases our approach is faster, and is more suited for applications where one requires fairly accurate solutions of the MAXCUT relaxation.

We show that an  $\epsilon$ -optimal solution to the semidefinite relaxation for the graph coloring problem can be computed in  $\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$  time. Our models for MAXCUT and the coloring SDP only differ by the number of constraints ( $\mathcal{O}(n)$  versus  $\mathcal{O}(r)$ ), and, consequently, our algorithm solves both these problems with the identical worst case complexity. The Klein-Lu bound [17] for graph coloring is  $\mathcal{O}(nr \log^3(n) \cdot \epsilon^{-4})$ , which is significantly slower than their bound for MAXCUT.

Our algorithm can compute an  $\epsilon$ -approximations of the Lovász- $\vartheta$  [18] and Szegedy number [26] in  $\mathcal{O}((n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})))$  (i.e., the same as MAXCUT and coloring) compared to the  $\mathcal{O}(n^5 r^3 \cdot \log(\epsilon^{-1}))$  runtime of the barrier method [4]. Thus, our tradeoff works in an opposite direction. For moderate  $\epsilon$  our bound is better for all graphs, and for dense graphs and moderate  $\epsilon$  our bound is much better.

We also show our methods compute an  $\epsilon$ -optimal solution to the sparse PCA problem in  $\mathcal{O}(n^4 \log(n) \cdot \epsilon^{-1})$  which matches the best known previous result for this problem [7]. Moreover, our method returns a feasible solution which is not true of [7].

- (d) In Section 6.3, we show our solution algorithm actually runs  $\Omega(n)$  faster than the theoretical bounds predict on test cases from Sparse PCA. We are able to solve SDPs with over  $10^7$  variables and constraints (i.e., problems where  $\mathbf{X}$  is of dimension up to  $6000 \times 6000$ ).

## 1.1 Notation and preliminaries

We denote vectors in lowercase bold, e.g.,  $\mathbf{x}$ , scalars in italics, e.g.,  $x$  or  $X$ , matrices in uppercase bold, e.g.,  $\mathbf{X}$ , and sets in uppercase calligraphic font, e.g.,  $\mathcal{X}$ . We use  $\mathbf{1}_n$  and  $\mathbf{0}_n$  to denote  $n$  dimensional vectors of all ones and zeros respectively, and omit the subscript  $n$  when the dimension is clear. We follow the same convention with the identity matrix,  $\mathbf{I}_n$ , and the matrix of all ones,  $\mathbf{J}_n$ . We use  $\mathcal{S}^n$  to denote the set of symmetric  $n \times n$  matrices and  $\mathcal{S}_+^n$  to denote the cone of positive semidefinite matrices, i.e. symmetric matrices with non-negative eigenvalues. We denote the partial order on  $\mathcal{S}^n$  induced by the cone  $\mathcal{S}_+^n$  by  $\succeq$ , i.e.  $\mathbf{A} \succeq \mathbf{0}$  denotes that the matrix  $\mathbf{A}$  is positive semidefinite, and  $\mathbf{A} \succeq \mathbf{B}$  denotes that  $\mathbf{A} - \mathbf{B} \succeq \mathbf{0}$ .

For a given vector  $\mathbf{v} \in \mathbb{R}^n$ , we let  $\|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|$ ,  $\|\mathbf{v}\|_\infty = \max_{i=1, \dots, n} |v_i|$ , and  $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2}$ , i.e., the  $\ell_1$ ,  $\ell_\infty$  and  $\ell_2$  norms, respectively. We define the  $\mathcal{L}_1$ ,  $\mathcal{L}_\infty$ , and  $\mathcal{L}_2$  for a symmetric matrix  $\mathbf{A}$  as follows. Let

$$\|\mathbf{A}\|_1 = \sum_{i=1}^n |\lambda_i(\mathbf{X})|, \quad \|\mathbf{A}\|_\infty = \max_{i=1, \dots, n} \{|\lambda_i(\mathbf{A})|\}, \quad \|\mathbf{A}\|_2 = \left( \sum_{i=1}^n \lambda_i^2(\mathbf{A}) \right)^{\frac{1}{2}},$$

where  $\{\lambda_i(\mathbf{X}) : i = 1, \dots, n\}$  denote the eigenvalues of  $\mathbf{A}$ .

We call a convex function  $f$  *strongly* convex with convex parameter  $\sigma$  if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{1}{2} \sigma \|\mathbf{y} - \mathbf{x}\|^2,$$

or, equivalently, for all  $\mathbf{x}, \mathbf{z}^\top \nabla^2 f(\mathbf{x}) \mathbf{z} \geq \sigma \|\mathbf{z}\|^2$ . Note that the value of the convexity parameter  $\sigma$  depends on the particular norm  $\|\cdot\|$ .

We say that  $\bar{\mathbf{z}}_a$  is  $\epsilon$ -optimal in the *absolute* sense for the optimization problem  $\max_{\mathbf{z} \in \mathcal{Z}} \{f(\mathbf{z})\}$  if  $f(\bar{\mathbf{z}}_a) \geq f^* - \epsilon$ , where  $f^* = \max_{\mathbf{z} \in \mathcal{Z}} \{f(\mathbf{z})\}$ , i.e.  $f(\bar{\mathbf{z}})$  is within an *additive* error  $\epsilon$  to the optimal value. We say that  $\bar{\mathbf{z}}_r$  is  $\epsilon$ -optimal in the *relative* sense if  $f(\bar{\mathbf{z}}_r) \geq (1 - \epsilon)f(\mathbf{z}^*)$ , i.e.  $f(\bar{\mathbf{z}}_r)$  is within a  $(1 - \epsilon)$  *multiplicative* factor of the optimal value. Note that the relative error measure has meaning only if  $f^* > 0$ . Suppose  $0 < C \leq f^*$  and  $\mathbf{z}_a$  is  $\epsilon$ -optimal in the absolute sense. Then  $f(\bar{\mathbf{z}}_a) \geq f^* - \epsilon = f^* - C(\epsilon/C) \geq (1 - \epsilon/C)f^*$ , i.e.  $\mathbf{z}_a$  is  $\epsilon/C$  optimal in the relative sense.

We use  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  to denote an undirected graph with  $n = |\mathcal{N}|$  nodes and  $m = |\mathcal{E}|$  edges. We assume all graphs are connected which implies that  $m = \Omega(n)$ .

## 2 The Packing SDP

In this section, we define the packing SDP and describe problems arising in combinatorial optimization, principal component analysis and maximum variance unfolding that can be reformulated as packing SDPs.

**Definition 1.** A function,  $g : \mathcal{S}^n \rightarrow \mathbb{R}$  is called a **packing function** if

1. (convexity)  $g$  is convex.
2. (positive homogeneity)  $g(\beta \mathbf{X}) = \beta g(\mathbf{X})$ , for all  $\beta \geq 0$ , and  $\mathbf{X} \succeq \mathbf{0}$ .
3. (non-negativity)  $g(\mathbf{X}) \geq 0$ , for all  $\mathbf{X} \succeq \mathbf{0}$ .

Examples of packing functions include:

1.  $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle$ ,  $\mathbf{A} \succeq \mathbf{0}$ .
2.  $g(\mathbf{X}) = \max \left\{ \sum_{i=1}^k z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \mathbf{z} \in \mathcal{P} \subseteq \mathbb{R}_+^k \right\}$ , where the matrices  $\mathbf{A}_i \succeq \mathbf{0}$ , for  $i = 1, \dots, k$  and  $\mathcal{P}$  is a convex set.
3.  $g(\mathbf{X}) = \sum_{i,j} |X_{ij}| = \max \left\{ \langle \mathbf{X}, \mathbf{Z} \rangle : |Z_{ij}| \leq 1, \forall i, j \right\}$ .
4.  $g(\mathbf{X}) = \left\| \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{X} \rangle \\ \vdots \\ \langle \mathbf{A}_k, \mathbf{X} \rangle \end{pmatrix} \right\|_2 = \max \left\{ \sum_{i=1}^k z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \|\mathbf{z}\|_2 \leq 1 \right\}$ .

The positive homogeneity condition (see 2. in Definition 1) is very restrictive; it essentially restricts  $g$  to norm-like functions. For example, the function

$$h(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle + b,$$

is *not* a packing function for any  $\mathbf{A} \in \mathcal{S}^n$  and  $b \in \mathbb{R}$  since it violates positive homogeneity. Similarly,

$$h(\mathbf{X}) = \left\| \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{X} \rangle \\ \vdots \\ \langle \mathbf{A}_k, \mathbf{X} \rangle \end{pmatrix} - \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix} \right\|$$

is also *not* a packing function.

**Definition 2.** A packing semidefinite program (packing SDP) is an optimization problem of the form

$$\begin{aligned} \rho^* = \max \quad & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} \quad & g_i(\mathbf{X}) \leq 1, \quad i = 1, \dots, m, \\ & \mathbf{Tr}(\mathbf{X}) \leq \omega_x, \\ & \mathbf{X} \succeq \mathbf{0}, \end{aligned} \quad (3)$$

where  $\mathbf{C} \succeq \mathbf{0}$  and the functions  $g_i(\mathbf{X})$  are packing functions for all  $i = 1, \dots, m$ . We also allow the trace constraint  $\mathbf{Tr}(\mathbf{X}) \leq \omega_x$  to be an equality.

Note that the trace constraint  $\mathbf{Tr}(\mathbf{X}) \leq \omega_x$  is equivalent to assuming the feasible region of the packing SDP (3) is compact. This is almost always true in problems of practical interest.

In the rest of this section we discuss specific instances of packing SDPs. In Section 3, we introduce the Lagrangian relaxation of the packing SDP and show how to recover an  $\epsilon$ -optimal feasible solution from the solution of the Lagrangian relaxation. In Section 4, we show how to adapt Nesterov's first-order procedure [23] to efficiently compute an  $\epsilon$ -optimal solution for the Lagrangian dual of the packing SDP by further restricting our class of packing functions. For all reformulations we discuss, however, our restrictions still allow the computation of  $\epsilon$ -optimal solutions for the Lagrangian dual.

## 2.1 Instances of packing SDP

Recall that  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  denotes a graph with  $|\mathcal{N}| = n$  nodes and  $|\mathcal{E}| = m$  edges.

### 2.1.1 The MAXCUT SDP

The SDP relaxation to the MAXCUT problem introduced by Goemans and Williamson [9] is given by

$$\begin{aligned} \max \quad & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \langle \mathbf{e}_i \mathbf{e}_i^\top, \mathbf{X} \rangle = 1, \quad i = 1, \dots, n, \\ & \mathbf{X} \succeq \mathbf{0}, \end{aligned} \quad (4)$$

where  $\mathbf{L}$  is the Laplacian of  $\mathcal{G}$  and  $\mathbf{e}_i$  is the  $i$ th column of the identity matrix. The Laplacian of a weighted graph with nonnegative edge weights  $w_{ij}$ ,  $(i, j) \in \mathcal{E}$ , is a matrix  $\mathbf{L} = [L_{ij}]$  where

$$L_{ij} = \begin{cases} -w_{ij}, & i \neq j, \\ \sum_{k=1}^n w_{ik}, & i = j. \end{cases} \quad (5)$$

We set  $w_{ij} = 0$  when  $(i, j) \notin \mathcal{E}$  and  $i \neq j$ . Then for any  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (x_i - x_j)^2 \geq 0$ , i.e.  $\mathbf{L} \succeq \mathbf{0}$ . Recall that we assume that  $\mathcal{G}$  is connected, which implies that for all  $i = 1, \dots, n$ , there exists an index  $k$  such that  $(i, k) \in \mathcal{E}$  and  $w_{ik} > 0$ . Then  $L_{ii} = \sum_{j=1}^n w_{ij} > 0$  for all  $i = 1, \dots, n$ .

Let  $\mathbf{D} = \mathbf{diag}(\mathbf{L})$ . Then the change of variables  $\mathbf{Y} = (\mathbf{Tr}(\mathbf{D})) \cdot \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$  implies that (4) is equivalent to

$$\begin{aligned} \max \quad & \langle \mathbf{L}_\mathbf{D}, \mathbf{Y} \rangle \\ \text{s.t.} \quad & \langle \mathbf{e}_i \mathbf{e}_i^\top, \mathbf{Y} \rangle = \frac{D_i}{\mathbf{Tr} \mathbf{D}}, \quad i = 1, \dots, n, \\ & \mathbf{Y} \succeq \mathbf{0}, \end{aligned} \quad (6)$$

where  $\mathbf{L}_\mathbf{D} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$  is the *normalized Laplacian* [24]. We claim that the packing SDP

$$\begin{aligned} \max \quad & \langle \mathbf{L}_\mathbf{D}, \mathbf{Y} \rangle \\ \text{s.t.} \quad & \frac{\mathbf{Tr}(\mathbf{D})}{D_i} \langle \mathbf{e}_i \mathbf{e}_i^\top, \mathbf{Y} \rangle \leq 1, \quad i = 1, \dots, n, \\ & \mathbf{Tr}(\mathbf{Y}) \leq 1, \\ & \mathbf{Y} \succeq \mathbf{0}, \end{aligned} \quad (7)$$

is equivalent to (6). Since the constraints,  $Y_{ii} \leq \frac{D_i}{\mathbf{Tr}(\mathbf{D})}, \forall i$ , imply  $\mathbf{Tr}(\mathbf{Y}) \leq 1$ , the packing SDP (7) and the original SDP formulation (6) are equivalent unless there exists an optimal solution  $\mathbf{X}^*$  to (7) with an index

$i$  such that  $X_{ii}^* < \frac{D_i}{\text{Tr}(\mathbf{D})}$ . Suppose this is the case. Define  $\mathbf{Y} = \mathbf{X}^* + (\frac{D_i}{\text{Tr}(\mathbf{D})} - X_{ii}^*)\mathbf{e}_i\mathbf{e}_i^\top$ . By construction,  $\mathbf{Y}$  is feasible for (6), and

$$\langle \mathbf{L}_D, \mathbf{Y} \rangle = \langle \mathbf{L}_D, \mathbf{X}^* \rangle + (\frac{D_i}{\text{Tr}(\mathbf{D})} - X_{ii}^*) > \langle \mathbf{L}_D, \mathbf{X}^* \rangle,$$

a contradiction. Thus, it follows that the packing SDP (7) is equivalent to the Max Cut SDP (6).

### 2.1.2 The Lovász $\vartheta$ function SDP

Lovász [18] defined the function  $\vartheta(\mathcal{G})$  as follows. Let

$$\begin{aligned} \vartheta(\mathcal{G}) &= \max \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t. } & x_{ij} = 0, \quad (i, j) \in \mathcal{E}, \\ & \text{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}, \end{aligned} \tag{8}$$

where  $\mathbf{J} \in \mathbb{R}^{n \times n}$  with all entries equal to 1. For each  $(i, j) \in \mathcal{E}$ , define  $\mathbf{E}^{(i,j)} = \mathbf{I} + \mathbf{e}_i\mathbf{e}_j^\top + \mathbf{e}_j\mathbf{e}_i^\top$  and  $\mathbf{F}^{(i,j)} = \mathbf{I} - \mathbf{e}_i\mathbf{e}_j^\top - \mathbf{e}_j\mathbf{e}_i^\top$ ,  $(i, j) \in \mathcal{E}$ . It is easy to show that  $\mathbf{E}^{(i,j)} \succeq \mathbf{0}$  and  $\mathbf{F}^{(i,j)} \succeq \mathbf{0}$ , for all  $(i, j) \in \mathcal{E}$ . Using the fact that  $\text{Tr}(\mathbf{X}) = 1$ , we can rewrite (8) as the packing SDP

$$\begin{aligned} \max \quad & \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t. } \quad & \langle \mathbf{E}^{(i,j)}, \mathbf{X} \rangle \leq 1, \quad (i, j) \in \mathcal{E}, \\ & \langle \mathbf{F}^{(i,j)}, \mathbf{X} \rangle \leq 1, \quad (i, j) \in \mathcal{E}, \\ & \text{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{9}$$

Note that in reformulating (8) as the packing SDP (9) it was extremely important that we allow trace equality constraints in packing SDPs (see Definition 2).

A related quantity to  $\vartheta(\mathcal{G})$  is the Szegedy number [26]

$$\begin{aligned} \vartheta^+(\mathcal{G}) &= \max \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t. } & x_{ij} \leq 0, \quad (i, j) \in \mathcal{E}, \\ & \text{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{10}$$

Gvozdenović and Laurent [11] show that  $\vartheta^+$  is a part of a family of graph parameters that approximate the clique and chromatic numbers. In particular,  $\vartheta^+(\mathcal{G})$  is a better approximation to the clique number to  $\mathcal{G}$  than  $\vartheta(\mathcal{G})$ . We can reformulate (10) as the packing SDP

$$\begin{aligned} \max \quad & \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t. } \quad & \langle \mathbf{E}^{(i,j)}, \mathbf{X} \rangle \leq 1, \quad (i, j) \in \mathcal{E}, \\ & \text{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{11}$$

### 2.1.3 The coloring SDP

Karger et al. [16] describe the following SDP relaxation for graph coloring problem on  $\mathcal{G}$ .

$$\begin{aligned} \max \quad & \zeta \\ \text{s.t. } \quad & x_{ii} = 1, \quad i = 1, \dots, n, \\ & \zeta \leq -x_{ij}, \quad (i, j) \in \mathcal{E}, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{12}$$

For each  $(i, j) \in \mathcal{E}$ , define

$$\mathbf{G}^{(i,j)} = \frac{1}{2}(\mathbf{e}_i\mathbf{e}_i^\top + \mathbf{e}_j\mathbf{e}_j^\top - (\mathbf{e}_i\mathbf{e}_j^\top + \mathbf{e}_j\mathbf{e}_i^\top)) = \frac{1}{2}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \succeq \mathbf{0}.$$

Then, for all  $\mathbf{X}$  feasible to (12), we have that  $\langle \mathbf{G}^{(i,j)}, \mathbf{X} \rangle = 1 - x_{ij}$ . Therefore,

$$\zeta = \min_{(i,j) \in \mathcal{E}} -x_{ij} = \min_{(i,j) \in \mathcal{E}} \left\{ \langle \mathbf{G}^{(i,j)}, \mathbf{X} \rangle \right\} - 1 = \min \left\{ \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} \langle \mathbf{G}^{(i,j)}, \mathbf{X} \rangle : \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} = 1 \right\} - 1.$$

From an argument similar to that used to show the equivalence of the Max Cut SDP to a packing SDP, it follows that (12) is equivalent to the max-min problem

$$\begin{aligned} \max \quad & \min \left\{ \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle \right\}, \\ \text{s.t.} \quad & x_{ii} \leq 1, \quad i = 1, \dots, n, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{13}$$

The optimization problem (13) is *not* a packing SDP. We compute an approximate solution to a packing SDP by Lagrangian relaxation, i.e. by converting packing SDP into a max-min problem (see Section 3 for details). We show that our solution algorithm can be easily adapted to solve a max-min problem of the form (13).

#### 2.1.4 Single factor Sparse Principal Component Analysis

Principal Component Analysis (PCA) is a popular tool for data analysis and dimensionality reduction. It has applications throughout science and engineering. In essence, PCA finds linear combinations of the variables (the so-called principal components) that correspond to the directions of maximal variance in the data. Sparse PCA is concerned with computing principal components that are sparse, a highly desirable feature when working with high dimensional data.

The single factor sparse principal component analysis problem reduces to

$$\begin{aligned} \max \quad & \mathbf{x}^\top \mathbf{C} \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\| = 1, \\ & \mathbf{Card}(\mathbf{x}) \leq \kappa. \end{aligned}$$

Here,  $\mathbf{C} \in \mathbb{R}^{n \times n} \succeq \mathbf{0}$  is a given covariance matrix,  $\mathbf{Card}(\mathbf{x})$  is a function that returns the number of nonzero components of  $\mathbf{x}$  and  $1 \leq \kappa \leq n$  is a given parameter. We refer the readers to the work of d’Aspremont et al. [7] for details. d’Aspremont et al. [7] formulate the following SDP relaxation for the above non-convex optimization problem

$$\begin{aligned} \max \quad & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \frac{1}{\kappa^2} \sum_{ij} |X_{ij}| \leq 1, \\ & \mathbf{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{14}$$

The optimization problem (14) is a packing SDP. In [7] the authors approximately solve (14) by dualizing the cardinality constraint  $\sum_{ij} |X_{ij}| \leq \kappa^2$ ; however, they do not guarantee that their solution is feasible. Our method computes feasible  $\epsilon$ -optimal solutions for the packing SDP (14).

#### 2.1.5 Maximum variance unfolding

Maximum Variance Unfolding (MVU) (also called semidefinite embedding) is a technique introduced by Weinberger and Saul [29] for computing low-dimensional representations that preserve distances between “local” points.

Suppose we are given  $n$  data points  $\{\mathbf{y}_i : i = 1, \dots, n\} \subseteq \mathbb{R}^\ell$  where the dimension  $\ell \gg 1$ . Let  $\mathcal{E} \subset \{(i, j) : 1 \leq i < j \leq n\}$  denote a set of tuples. We call a pair  $(i, j)$  “local” with respect to each other if, and only if,  $(i, j) \in \mathcal{E}$ . The goal of the MVU technique is to compute a representation  $\{\mathbf{u}_i : i = 1, \dots, n\} \subset \mathbb{R}^m$ ,  $m \ll \ell$ ,

that preserves distances and minimizes the effective dimension of the resulting manifold. Weinberger and Saul [29] propose constructing such a manifold by solving the optimization problem

$$\begin{aligned} \max \quad & \sum_{(i,j) \in \mathcal{E}} \|\mathbf{u}_i - \mathbf{u}_j\|^2 - \nu \sqrt{\sum_{(i,j) \in \mathcal{E}} (\|\mathbf{u}_i - \mathbf{u}_j\|^2 - d_{ij})^2} \\ \text{s.t.} \quad & \sum_{i=1}^n \mathbf{u}_i = \mathbf{0}. \end{aligned} \quad (15)$$

where  $\sum_{i=1}^n \mathbf{u}_i = \mathbf{0}$  is a centering constraint. Since (15) is not a convex optimization problem, Weinberger and Saul [29] approximately solve (15) by constructing a semidefinite programming relaxation.

We present a slightly modified version of the relaxation developed in [29]. Let  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{n-1}]$ , the Gram matrix  $\mathbf{K} = \mathbf{U}^\top \mathbf{U}$ , and for each  $(i, j) \in \mathcal{E}$ , define  $\mathbf{a}_{(i,j)} \in \mathbb{R}^{(n-1)}$  as follows. Let

$$\mathbf{a}_{ij} = \begin{cases} \mathbf{e}_i - \mathbf{e}_j, & i, j \neq n, \\ \mathbf{e}_i - \sum_{k=1}^{n-1} \mathbf{e}_k, & j = n, \\ \sum_{k=1}^{n-1} \mathbf{e}_k - \mathbf{e}_j, & i = n, \end{cases}$$

where  $\mathbf{e}_k$  denotes the  $k$ -th column of  $(n-1) \times (n-1)$  dimensional identity matrix. Then, for each  $i \neq j$ ,  $\mathbf{u}_i - \mathbf{u}_j = \mathbf{U} \mathbf{a}_{ij}$  and  $\|\mathbf{u}_i - \mathbf{u}_j\|^2 = \mathbf{a}_{ij}^\top \mathbf{K} \mathbf{a}_{ij} = \langle \mathbf{a}_{ij} \mathbf{a}_{ij}^\top, \mathbf{K} \rangle$ . In terms of the new variables, the optimization problem (15) is equivalent to

$$\begin{aligned} \max \quad & \min_{\|\mathbf{z}\|_2 \leq 1} \left\{ \left\langle \sum_{(i,j) \in \mathcal{E}} \mathbf{a}_{ij} \mathbf{a}_{ij}^\top, \mathbf{K} \right\rangle - \sum_{(i,j) \in \mathcal{E}} z_{ij} (\langle \mathbf{a}_{ij} \mathbf{a}_{ij}^\top, \mathbf{K} \rangle - d_{ij}) \right\}, \\ \text{s.t.} \quad & \text{rank}(\mathbf{K}) = m, \\ & \text{Tr}(\mathbf{K}) \leq \tau, \\ & \mathbf{K} \succeq \mathbf{0}, \end{aligned} \quad (16)$$

where  $\tau = \sum_{i=1}^n \|\mathbf{y}_i\|$ . The semidefinite relaxation is obtained by relaxing the rank constraint on  $\mathbf{K}$  as in (14). The optimization problem (16) is a max-min problem. We show that our solution algorithm can be easily adapted to solve a max-min problem of the form (16).

### 2.1.6 Improving Laplacian eigenvalues and locally linear embedding using MVU

Laplacian eigenmaps, locally linear embedding and Isomaps are different techniques for computing low-dimensional representations for high dimensional data that preserve proximity relations. Let

$$\mathbf{u}_i = \mathbf{V} \mathbf{y}_i, \quad i = 1 \dots, n,$$

where  $\mathbf{V} \in \mathbb{R}^{m \times \ell}$  denote the  $m$  dimensional representation for the set of  $\ell$ -dimensional vectors  $\{\mathbf{y}_i : 1 \leq i \leq n\}$  computed by any data mining technique. Xiao et al. [30] propose that this representation can be further refined via MVU post-processing.

Recall that the MVU approach reduces to computing an appropriate Gram matrix for the data vectors. Given the representation matrix  $\mathbf{V}$ , the data vectors are now  $r$ -dimensional vectors. In the Xiao et al [30] approach, the Gram matrix  $\mathbf{K}$  of the  $n$  vectors is approximated by  $\mathbf{K} = \mathbf{V}^\top \mathbf{Q} \mathbf{V}$ , where  $\mathbf{Q} \in \mathbb{R}^{r \times r}$  and  $\mathbf{Q} \succeq \mathbf{0}$ . The MVU post-processing step then reduces to the packing SDP

$$\begin{aligned} \max \quad & \langle \mathbf{V} \mathbf{V}^\top, \mathbf{Q} \rangle \\ \text{s.t.} \quad & \langle \mathbf{V}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{V}^\top, \mathbf{Q} \rangle \leq d_{ij}, \quad (i, j) \in \mathcal{E} \\ & \text{Tr}(\mathbf{Q}) \leq \tau, \\ & \mathbf{Q} \succeq \mathbf{0}, \end{aligned} \quad (17)$$

where  $\tau = \sum_{i=1}^n \|\mathbf{u}_i\|$ , and  $d_{ij}$ ,  $(i, j) \in \mathcal{E}$ , denotes the bound on the distance between the ‘‘local’’ node pair  $(i, j)$ .

### 3 Lagrangian formulation and rounding

In this section, we show how to construct an  $\epsilon$ -approximate solution for the Packing SDP (3) from its natural Lagrangian relaxation. The Lagrangian relaxation converts the packing SDP into a primal-dual problem where both the primal and the dual feasible sets are “simple”, i.e. sets over which optimization is easy. In Section 4, we show how to solve the Lagrangian relaxation.

Define the Lagrangian function  $\phi(\mathbf{X}, \mathbf{v})$  of the packing SDP (3) as follows. Let

$$\phi(\mathbf{X}, \mathbf{v}) = \langle \mathbf{C}, \mathbf{X} \rangle - \sum_{i=0}^m v_i (g_i(\mathbf{X}) - 1), \quad (18)$$

and define the set

$$\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \text{Tr}(\mathbf{X}) \leq \omega_x\}. \quad (19)$$

Recall that we assume either the packing SDP (3) has the trace constraint  $\text{Tr}(\mathbf{X}) \leq \omega_x$  or such a bound is implied by the packing constraints. When the trace constraint in the packing SDP is an equality constraint we set  $\text{Tr}(\mathbf{X}) = \omega_x$  in (19). Let

$$\mathcal{V} = \left\{ \mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_{j=0}^m v_j \leq \omega_v \right\}, \quad (20)$$

where

$$\omega_v = \max\{\langle \mathbf{C}, \mathbf{X} \rangle : \text{Tr}(\mathbf{X}) \leq 1, \mathbf{X} \succeq \mathbf{0}\} \leq \omega_x \text{Tr}(\mathbf{C}). \quad (21)$$

We need the “diameter”  $\omega_v$  of the dual set to be large enough to ensure the infeasible iterates are sufficiently penalized. The proof of Theorem 1 demonstrates that the bound  $\omega_v$  in (21) is sufficiently large.

Let  $\bar{\mathbf{v}} \in \mathcal{V}$ . Then for all  $\mathbf{X}$  feasible to the Packing SDP (3),  $\phi(\mathbf{X}, \bar{\mathbf{v}}) \geq \langle \mathbf{C}, \mathbf{X} \rangle$ . Therefore,

$$\max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) \geq \max_{\mathbf{X} \in \mathcal{X}} \langle \mathbf{C}, \mathbf{X} \rangle \geq \rho^*, \quad (22)$$

where the last inequality follows from the fact that the feasible region  $\{\mathbf{X} : g_i(\mathbf{X}) \leq 1, i = 1, \dots, m, \mathbf{X} \succeq \mathbf{0}\}$  of the packing SDP (3) is a subset of  $\mathcal{X}$ . Thus,

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) = \min_{\mathbf{v} \in \mathcal{V}} \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{v}) \geq \rho^*, \quad (23)$$

where the equality follows an appropriate saddle-point theorem applied to the function  $\phi(\mathbf{X}, \mathbf{v})$  and the inequality follows from (22). We refer to the max-min problem in (23) as the *Lagrangian relaxation* of the packing SDP, the maximization problem in  $\mathbf{X}$  as the *primal* problem and the minimization problem in  $\mathbf{v}$  as the *dual* problem. We call a pair  $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$ ,  $\bar{\mathbf{X}} \in \mathcal{X}$ ,  $\bar{\mathbf{v}} \in \mathcal{V}$ , an  $\epsilon$ -*saddle-point* for (18) if

$$0 \leq \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) - \min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}) \leq \epsilon.$$

**Theorem 1.** Fix  $\epsilon > 0$ . Suppose  $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$  is an  $\epsilon$ -saddle-point. Define

$$\hat{\mathbf{X}} = \begin{cases} (1/\bar{d})\bar{\mathbf{X}}, & \bar{d} > 1, \\ \bar{\mathbf{X}}, & \text{otherwise,} \end{cases} \quad (24)$$

where  $\bar{d} = \max_{i=1, \dots, m} \{g_i(\bar{\mathbf{X}})\}$  is the maximum constraint violation. Then  $\hat{\mathbf{X}}$  is an  $\epsilon$ -optimal solution for the packing SDP (3).

*Proof.* When  $\bar{d} < 1$ ,  $g_i(\bar{\mathbf{X}}) \leq 1$  for all  $i$ ; thus,  $\hat{\mathbf{X}} = \bar{\mathbf{X}}$  is feasible. When  $\bar{d} > 1$ , the positive homogeneity property of the packing functions  $g_i(\mathbf{X})$  implies that

$$1 \geq \left(\frac{1}{\bar{d}}\right)g_i(\bar{\mathbf{X}}) = g_i\left(\frac{1}{\bar{d}} \cdot \bar{\mathbf{X}}\right) = g_i(\hat{\mathbf{X}}).$$

Thus,  $\hat{\mathbf{X}}$  is always feasible to the packing SDP (3). Next, we show that  $\hat{\mathbf{X}}$  is  $\epsilon$ -optimal. Consider the following two cases:

(a)  $\bar{d} \leq 1$ . In this case,

$$\min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^m v_i (1 - g_i(\mathbf{X})) \right\} = 0.$$

Thus,

$$\langle \mathbf{C}, \widehat{\mathbf{X}} \rangle = \langle \mathbf{C}, \bar{\mathbf{X}} \rangle + \min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^m v_i (1 - g_i(\mathbf{X})) \right\} = \min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}),$$

where the last equality follows from the definition of  $\phi(\mathbf{X}, \mathbf{v})$ . Since  $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$  is an  $\epsilon$ -saddle-point, it follows that

$$\min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}) \geq \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) - \epsilon \geq \rho^* - \epsilon,$$

where the last inequality follows (22).

(b)  $\bar{d} > 1$ . Since  $\mathbf{C} \succeq \mathbf{0}$ ,  $\widehat{\mathbf{X}} \succeq \mathbf{0}$  and  $\frac{1}{\bar{d}} \geq 1 - (d - 1)$  for all  $d > 0$  we have that

$$\begin{aligned} \langle \mathbf{C}, \widehat{\mathbf{X}} \rangle &= \frac{1}{\bar{d}} \langle \mathbf{C}, \bar{\mathbf{X}} \rangle \\ &\geq \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - (\bar{d} - 1) \langle \mathbf{C}, \bar{\mathbf{X}} \rangle \\ &\geq \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - (\bar{d} - 1) \omega_v \end{aligned} \tag{25}$$

$$= \langle \mathbf{C}, \bar{\mathbf{X}} \rangle + \min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^m v_i (1 - g_i(\mathbf{X})) \right\} \tag{26}$$

$$\begin{aligned} &= \min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}) \\ &\geq \rho^* - \epsilon, \end{aligned} \tag{27}$$

where (25) follows from the fact that  $\langle \mathbf{C}, \mathbf{X} \rangle \leq \omega_v$  for all  $\mathbf{X} \in \mathcal{X}$  (see (21)), (26) follows from the fact that  $\omega_v(1 - \bar{d}) = \min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^m v_i (1 - g_i(\mathbf{X})) \right\}$ , whenever  $\bar{d} > 1$ , and (27) follows from an argument identical to that used in the previous case.

Thus, we have that  $\widehat{\mathbf{X}}$  is a feasible,  $\epsilon$ -optimal solution to (3).  $\square$

Typically, Lagrangian relaxation yields good bounds but does not yield feasible solutions. Theorem 1 shows that by setting the “diameter”  $\omega_v$  sufficiently large one can recover a *feasible*  $\epsilon$ -approximate solution for Packing SDP (3) from an  $\epsilon$ -saddle-point for (18). In the next section we show that for a restricted class of packing functions one can compute an  $\epsilon$ -saddle-point very efficiently.

We note that Theorem 1 does *not* find feasible solutions for packing SDPs with a trace *equality* constraint. In the case of MAXCUT, relaxing the original trace equality constraint is equivalent to restricting the main diagonal to the ones vector. However, the objective function in this case are non-decreasing in the main diagonal, so a feasible,  $\epsilon$ -optimal solution can be calculated by just replacing the main diagonal with ones.

**Lemma 1.** *Suppose  $F : \mathcal{S}^n \mapsto \mathbb{R}$  such that  $f(\mathbf{Y} + \alpha \mathbf{e}_i \mathbf{e}_i^\top) \geq f(\mathbf{Y})$  for all  $\mathbf{Y} \succeq \mathbf{0}$ ,  $\alpha > 0$  and all canonical basis vectors  $\mathbf{e}_i$ ,  $i = 1, \dots, n$ . Let  $\nu^* = \max\{f(\mathbf{X}) : \mathbf{diag}(\mathbf{X}) = \mathbf{1}, \mathbf{X} \succeq \mathbf{0}\}$ .*

*Suppose  $\mathbf{X} \succeq \mathbf{0}$  such that  $\mathbf{diag}(\mathbf{X}) \leq \mathbf{1}$  and  $f(\mathbf{X}) \geq \nu^* - \epsilon$  for  $\epsilon > 0$ , then  $\mathbf{Y} = \mathbf{X} + \mathbf{I} - \mathbf{diag}(\mathbf{diag}(\mathbf{X}))$  is a feasible  $\epsilon$ -optimal solution.*

*Proof.* This follows directly from the fact that  $\mathbf{I} - \mathbf{diag}(\mathbf{diag}(\mathbf{X})) \succeq \mathbf{0}$  and that  $f(\mathbf{Y}) \geq f(\mathbf{X})$ .  $\square$

Note that Lemma 1 can be used for both the MAXCUT packing SDP (in conjunction with Theorem 1) and the max-min optimization problem (13) for coloring.

Recall that the packing SDPs for the Lovász  $\vartheta$ -function, Szegedy’s function and Sparse PCA all have a trace equality constraint. We also provide a more general “additive rounding” when the packing SDP with a trace equality constraint has a *strictly* feasible point.

**Theorem 2.** Suppose  $\mathbf{Z}$  is a strictly feasible solution to a packing SDP (3) with the trace equality constraint, i.e.

$$\mathbf{Tr}(\mathbf{Z}) = \omega_x, \quad g_i(\mathbf{Z}) < 1, \forall i = 1, \dots, m.$$

Define the dual set parameter

$$\omega_v = \frac{\rho_u - \langle \mathbf{C}, \mathbf{Z} \rangle}{1 - g_{\max}(\mathbf{Z})}, \quad (28)$$

where the function

$$g_{\max}(\mathbf{X}) \triangleq \max_{1 \leq i \leq m} \{g_i(\mathbf{X})\},$$

and  $\rho_u$  is any upper bound on  $\rho^*$ , in particular, we can set  $\rho_u = \langle \mathbf{C}, \mathbf{Z} \rangle$ . Suppose that  $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$  is an  $\epsilon$ -optimal saddle point and assume that  $\langle \mathbf{C}, \mathbf{Z} \rangle < \langle \mathbf{C}, \bar{\mathbf{X}} \rangle$ .<sup>1</sup> Define

$$\hat{\mathbf{X}} = \frac{\bar{\mathbf{X}} + \beta(\bar{\mathbf{X}})\mathbf{Z}}{1 + \beta(\bar{\mathbf{X}})},$$

where the function

$$\beta(\mathbf{X}) \triangleq \frac{g_{\max}(\mathbf{X}) - 1}{1 - g_{\max}(\mathbf{Z})}.$$

Then  $\hat{\mathbf{X}}$  is a feasible  $\epsilon$ -optimal solution to (3) with  $\mathbf{Tr}(\hat{\mathbf{X}}) = \omega_x$ .

*Proof.* We first show that  $\hat{\mathbf{X}}$  is feasible. Since  $\mathbf{Z}$  is strictly feasible  $g_{\max}(\mathbf{Z}) < 1$  and  $g_{\max}(\bar{\mathbf{X}}) > 1$ ; therefore,  $\beta(\bar{\mathbf{X}}) > 0$  and  $\hat{\mathbf{X}}$  is a convex combination of  $\bar{\mathbf{X}}$  and  $\mathbf{Z}$ . Then  $\mathbf{Z}, \bar{\mathbf{X}} \in \mathcal{X}$  implies that  $\hat{\mathbf{X}} \in \mathcal{X}$ , i.e.  $\hat{\mathbf{X}} \succeq \mathbf{0}$  and  $\mathbf{Tr}(\hat{\mathbf{X}}) = \omega_x$ . Since each of the packing functions  $g_i(\mathbf{X})$  are convex, it follows that  $g_{\max}(\mathbf{X})$  is also a convex function. Therefore,

$$g_{\max}(\hat{\mathbf{X}}) \leq \frac{g_{\max}(\bar{\mathbf{X}}) + \beta(\bar{\mathbf{X}})g_{\max}(\mathbf{Z})}{1 + \beta(\bar{\mathbf{X}})}$$

Substituting for  $\beta(\bar{\mathbf{X}})$  we get

$$g_{\max}(\hat{\mathbf{X}}) \leq \frac{g_{\max}(\bar{\mathbf{X}})(1 - g_{\max}(\mathbf{Z})) + g_{\max}(\mathbf{Z})(g_{\max}(\bar{\mathbf{X}}) - 1)}{g_{\max}(\bar{\mathbf{X}}) - g_{\max}(\mathbf{Z})} = 1.$$

Thus,  $g_i(\hat{\mathbf{X}}) \leq 1$ , for all  $i = 1, \dots, m$ .

We now show that  $\hat{\mathbf{X}}$  is  $\epsilon$ -optimal. Define

$$\Psi(\mathbf{X}) = \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) = \langle \mathbf{C}, \mathbf{X} \rangle - \omega_v \max\{0, g_{\max}(\mathbf{X}) - 1\},$$

where the last equality follows from the definition of the dual set  $\mathcal{V}$ . We first show that  $\rho^* = \max_{\mathbf{X} \in \mathcal{X}} \Psi(\mathbf{X})$ . Fix  $\mathbf{X} \in \mathcal{X}$ , define

$$\mathbf{Y} = \frac{\mathbf{X} + \max\{\beta(\mathbf{X}), 0\}\mathbf{Z}}{1 + \max\{\beta(\mathbf{X}), 0\}}.$$

Then,  $\mathbf{Y}$  is feasible for (3). Also,

$$\begin{aligned} \Psi(\mathbf{X}) - \rho^* &= \langle \mathbf{C}, \mathbf{X} \rangle - \omega_v \max\{0, g_{\max}(\mathbf{X}) - 1\} - \rho^*, \\ &= \langle \mathbf{C}, \mathbf{X} \rangle + \omega_v(1 - g_{\max}(\mathbf{Z})) \max\{\beta(\mathbf{X}), 0\} - \rho^*, \end{aligned} \quad (29)$$

$$\begin{aligned} &= (1 + \max\{\beta(\mathbf{X}), 0\}) \langle \mathbf{C}, \mathbf{Y} \rangle - \max\{\beta(\mathbf{X}), 0\} \langle \mathbf{C}, \mathbf{Z} \rangle \\ &\quad + \omega_v(1 - g_{\max}(\mathbf{Z})) \max\{\beta(\mathbf{X}), 0\} - \rho^*, \end{aligned} \quad (30)$$

$$\begin{aligned} &\leq (1 + \max\{\beta(\mathbf{X}) - 1\}) (\langle \mathbf{C}, \mathbf{Y} \rangle - \rho^*) \\ &\quad + \max\{\beta(\mathbf{X}), 0\} (\rho^* - \langle \mathbf{C}, \mathbf{Z} \rangle - \omega_v(1 - g_{\max}(\mathbf{Z}))), \\ &= (1 + \max\{\beta(\mathbf{X}) - 1\}) (\langle \mathbf{C}, \mathbf{Y} \rangle - \rho^*) + \max\{\beta(\mathbf{X}), 0\} (\rho^* - \rho_u), \end{aligned} \quad (31)$$

<sup>1</sup>Otherwise, we just use  $\mathbf{Z}$  as our solution.

where (29) follows from the definition of  $\beta(\mathbf{X})$ , (30) follows from definition of  $\mathbf{Y}$ , (31) follows from the definition of  $\omega_v$  in (28). Since  $\mathbf{Y}$  is feasible for (3) and  $\rho_u \geq \rho^*$ , it follows that

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) = \rho^*.$$

Since  $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$  is an  $\epsilon$ -saddle point,

$$\Psi(\bar{\mathbf{X}}) = \min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}) \geq \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) - \epsilon \geq \rho^* - \epsilon.$$

Therefore, it follows that

$$\begin{aligned} (1 + \beta(\bar{\mathbf{X}})) \left( \langle \mathbf{C}, \hat{\mathbf{X}} \rangle - \rho^* \right) &= \langle \mathbf{C}, \bar{\mathbf{X}} \rangle + \beta(\bar{\mathbf{X}}) \langle \mathbf{C}, \mathbf{Z} \rangle - (1 + \beta(\bar{\mathbf{X}})) \rho^* \\ &= \left( \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - \omega_v (g_{\max}(\bar{\mathbf{X}}) - 1) - \rho^* \right) \\ &\quad + \beta(\bar{\mathbf{X}}) \left( \langle \mathbf{C}, \mathbf{Z} \rangle + \omega_v (1 - g_{\max}(\mathbf{Z})) \rho^* \right), \tag{32} \\ &\geq -\epsilon + (\rho_u - \rho^*), \tag{33} \end{aligned}$$

where (32) follows from the definition of  $\beta(\mathbf{X})$ , and (32) follows from the definition of  $\omega_v$  in (28) and the fact that  $\Psi(\bar{\mathbf{X}}) = \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - \omega_v (g_{\max}(\bar{\mathbf{X}}) - 1) \geq \rho^*$ . Thus,

$$\langle \mathbf{C}, \hat{\mathbf{X}} \rangle \geq \rho^* - \frac{\epsilon}{1 + \beta(\bar{\mathbf{X}})} \geq \rho^* - \epsilon,$$

i.e.,  $\hat{\mathbf{X}}$  is a feasible,  $\epsilon$ -optimal solution.  $\square$

Theorem 2 can be used to convert  $\epsilon$ -saddle points to both the Sparse PCA and Szegedy's number Lagrangian relaxations into feasible,  $\epsilon$ -optimal solutions to their respective packing SDPs. A version of Theorem 2 was also established by Z. Lu, Monteiro and Yuan [19].

Recall that the semidefinite relaxation for graph coloring problem (13) and the semidefinite relaxation for the maximum variance unfolding problem (16) are *not* packing SDPs. However, the structure of the Lagrangian relaxation of these problems is identical to that of a packing SDP. For instance, the Lagrangian relaxation for the coloring problem

$$\max_{\{\mathbf{X}: \text{Tr}(\mathbf{X}) \leq n, \mathbf{X} \succeq \mathbf{0}\}} \min_{\{(\mathbf{w}, \mathbf{z}) \geq \mathbf{0}: \sum_{(i,j) \in \mathcal{E}} w_{ij} = 1, \sum_{i=1}^n z_i \leq \tau\}} \left\{ \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle - \sum_{i=1}^n z_i (X_{ii} - 1) \right\},$$

where  $\tau = \max \left\{ \min_{(i,j) \in \mathcal{E}} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle : \mathbf{X} \succeq \mathbf{0}, \text{Tr}(\mathbf{X}) \leq n \right\} \leq 4$ , has the same structure as the Lagrangian relaxation of the packing problem.

## 4 Solving the saddle-point problem

The saddle-point problem (18) is a game. An  $\epsilon$ -saddle-point for (18) is an  $\epsilon$ -equilibrium for this game. One could, in principle, use fictitious play [5] or similar methods to compute such an  $\epsilon$ -equilibrium. Since the general methods for computing  $\epsilon$ -equilibria in minimax games rely on subgradient descent and the primal-dual objective functions are non-smooth, these general methods need  $\mathcal{O}(1/\epsilon^2)$  iterations to converge.

Nesterov proposed an iterative procedure for computing an  $\epsilon$ -saddle-point for the special case where each of the packing functions are linear, i.e.  $g_i(\mathbf{X}) = \langle \mathbf{A}_i, \mathbf{X} \rangle$  with  $\mathbf{A}_i \succeq \mathbf{0}$ , i.e. minimax problems of the form

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \left\{ \left\langle \mathbf{C} - \sum_{i=1}^m v_i \mathbf{A}_i, \mathbf{X} \right\rangle + \sum_{i=1}^m v_i \right\}. \tag{34}$$

NESTEROV PROCEDURE

Set  $N \leftarrow \frac{\Omega}{\epsilon} \sqrt{\frac{D_x D_v}{\sigma_x \sigma_v}}$ ,  $\mu_x \leftarrow \frac{\epsilon}{2D_x}$ ,  $\mu_v \leftarrow \frac{\Omega}{\mu_x \sigma_x \sigma_v} = \frac{\Omega D_x}{\sigma_x \sigma_v} \cdot \frac{1}{\epsilon}$ ,  $\mathbf{x}^{(0)} \leftarrow \frac{\omega_v}{n+1} \mathbf{1}$ .  
**for**  $k \leftarrow 0$  **to**  $N$   
**do**  
Set  $\mathbf{X}^{(k)} = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \left\{ \langle \mathbf{C}, \mathbf{X} \rangle - \sum_{i=1}^m x_i^{(k)} g_i(\mathbf{X}) + \mu_x d_x(\mathbf{X}) \right\}$ ;  
 $\gamma_i^{(k)} \leftarrow 1 - g_i(\mathbf{X}^{(k)})$ ,  $i = 1, \dots, m$ ;  
 $\mathbf{y}^{(k)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \left\{ (\boldsymbol{\gamma}^{(k)} - \mu_v \nabla d_v(\mathbf{x}^{(k)}))^\top \mathbf{v} + \mu_v d_v(\mathbf{v}) \right\}$ ;  
 $\mathbf{z}^{(k)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \left\{ \left( \sum_{i=0}^k \frac{i+1}{2} \boldsymbol{\gamma}^{(i)} \right)^\top \mathbf{v} + \mu_v d_v(\mathbf{v}) \right\}$ ;  
 $\mathbf{x}^{(k+1)} = \left( \frac{2}{k+3} \right) \mathbf{z}^{(k)} + \left( \frac{k+1}{k+3} \right) \mathbf{y}^{(k)}$ .  
**return**  $\left( \bar{\mathbf{X}} = \sum_{k=0}^N \frac{2(i+1)}{(N+1)(N+2)} \mathbf{X}^{(k)}, \mathbf{y}^{(N)} \right)$ .

Figure 1: Nesterov Procedure

**Theorem 3** ([23]). *The iterative procedure displayed in Figure 1 computes an  $\epsilon$ -saddle point for (34) in  $\mathcal{O}\left(\frac{\Omega}{\epsilon} \sqrt{\frac{D_x D_v}{\sigma_x \sigma_v}}\right)$  iterations where*

(i)  $\Omega^2 = \max_{\{\|\mathbf{v}\|_x \leq 1, \|\mathbf{X}\|_x \leq 1\}} \left| \left\langle \sum_{i=1}^m v_i \mathbf{A}_i, \mathbf{X} \right\rangle \right|^2$ , where  $\|\cdot\|_x$  and  $\|\cdot\|_v$  are appropriate norms on the primal and dual spaces, respectively,

(ii)  $D_x = \max_{\mathbf{X} \in \mathcal{X}} d_x(\mathbf{X})$  is the “diameter” of the set  $\mathcal{X}$  with respect to a strongly convex function  $d_x(\mathbf{X})$  that has a convexity parameter  $\sigma_x$  and is non-negative on the primal set  $\mathcal{X}$ ,

(iii)  $D_v = \max_{\mathbf{v} \in \mathcal{V}} d_v(\mathbf{v})$ , is the “diameter” of the set  $\mathcal{V}$  with respect to a strongly convex function  $d_v(\mathbf{v})$  that has a convexity parameter  $\sigma_v$  and is non-negative on the primal set  $\mathcal{V}$ , and

(iv) in each iteration the procedure needs to compute an exact solution to problems of the form

$$\min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=1}^m \gamma_i v_i + \mu_v d_v(\mathbf{v}) \right\}, \quad (35)$$

and

$$\max_{\mathbf{X} \in \mathcal{X}} \left\{ \langle \boldsymbol{\Gamma}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \right\}, \quad (36)$$

where  $\mu_v$  and  $\mu_x$  are functions of  $\sigma_x, \sigma_v, \Omega, D_x$  and  $\epsilon$  as described in Figure 1.

We call a strongly convex function that is non-negative on a given convex set  $\mathcal{S}$  a *prox-function* for the set. Prox-functions ensure that both the primal and the dual optimization problems are smooth. In order for the Nesterov algorithm to be efficient, one should choose the prox-function  $d_v$  and  $d_x$  so that the optimization problems (35) and (36) can both be solved in closed form. It is this requirement that restricts one to “simple” feasible sets  $\mathcal{X}$  and  $\mathcal{V}$ . Another requirement on the prox function is that the associated “diameter”  $D_x$  and  $D_v$  be modest.

We have made a few modifications to the standard version of the Nesterov procedure. We iterate in the dual, i.e.  $\mathcal{V}$ , space and then compute the approximate primal solution  $\bar{\mathbf{X}}$  by aggregating over all gradients. We compute the iterate  $\mathbf{y}^{(k)}$  using the Bregman distance associated with prox-function  $d_v$  for the  $\mathcal{V}$  space. This allows us to treat the  $\mathbf{y}^{(k)}$  and  $\mathbf{z}^{(k)}$  iterates in a unified manner. Note that the numerical value of the constants  $\mu_v$  and  $\mu_x$  are very different:  $\mu_x = \mathcal{O}(\epsilon)$  and  $\mu_v = \mathcal{O}\left(\frac{1}{\epsilon}\right)$ . This difference of scale plays a critical

role in computing solutions of the optimization problems and can be exploited to improve the runtimes of the algorithm.

We extend Nesterov’s method to packing functions of the form

$$\begin{aligned} g(\mathbf{X}) &= \max \sum_{j=1}^k z_j \langle \mathbf{A}_j, \mathbf{X} \rangle \\ \text{s.t.} \quad & \mathbf{P}\mathbf{z} \leq \mathbf{b}, \end{aligned} \quad (37)$$

where  $\mathbf{A}_j$ ,  $j = 1, \dots, k$ , are symmetric matrices,  $\mathbf{P} \in \mathbb{R}^{\ell \times k}$  and  $\mathbf{b} \in \mathbb{R}^\ell$ . All packing functions discussed in Section 2 are of the form described in (37). Recall that  $g(\mathbf{X})$  must satisfy the conditions in Definition 1; consequently, not all choices for  $\mathbf{A}_i$ ,  $i = 1, \dots, k$ ,  $\mathbf{P}$  and  $\mathbf{b}$  will result in a valid packing function.

For this restricted class of packing functions, the Lagrangian function  $\phi(\mathbf{X}, \mathbf{v})$  is given by

$$\phi(\mathbf{X}, \mathbf{v}, \mathbf{z}_1, \dots, \mathbf{z}_m) = \langle \mathbf{C}, \mathbf{X} \rangle + \sum_{i=1}^m v_i \left( 1 - \sum_{j=1}^{k_i} z_{ij} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle \right), \quad (38)$$

where  $\mathbf{z}_i \in \mathcal{P}_i = \{\mathbf{z} : \mathbf{A}_i \mathbf{z} \leq \mathbf{b}_i\}$  are the variables associated with the  $i$ -th packing function  $g_i(\mathbf{X}) = \max \left\{ \sum_{j=1}^{k_i} z_{ij} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle : \mathbf{P}_i \mathbf{z}_i \leq \mathbf{b}_i \right\}$ ,  $i = 1, \dots, m$ . The dual variables defining  $\phi(\mathbf{X})$  are  $\mathbf{v} \in \mathcal{V}$  and  $\mathbf{z}_i \in \mathcal{P}_i$ ,  $i = 1, \dots, m$ , and the dual objective in (38) is *quadratic* in these variables. We linearize the objective by defining a new set of variables  $\mathbf{y}_i = v_i \mathbf{z}_i$ . In terms of these new variables, the Lagrangian function is given by

$$\phi(\mathbf{X}, \mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) = \langle \mathbf{C}, \mathbf{X} \rangle + \sum_{i=1}^m \left( v_i - \sum_{j=1}^{k_i} y_{ij} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle \right). \quad (39)$$

Note that this linearization step works only because  $\mathbf{v} \geq \mathbf{0}$ . Now we are in position to apply the Nesterov procedure to compute the an  $\epsilon$ -saddle point for the function  $\phi(\mathbf{X}, \mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m)$  over the sets  $\mathcal{X} \times \mathcal{Y}$ , where

$$\mathcal{Y} = \left\{ (\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) : \mathbf{v} \geq \mathbf{0}, \sum_{i=1}^m v_i \leq 1, \mathbf{P}_i \mathbf{y}_i \leq v_i \mathbf{b}_i, i = 1, \dots, m \right\}. \quad (40)$$

As we have indicated earlier, such a procedure is efficient only if one is able to construct prox functions for the sets  $\mathcal{X}$  and  $\mathcal{Y}$ .

#### 4.1 Prox function for the primal set $\mathcal{X}$

A natural choice for the prox function  $d_x$  for the set  $\mathcal{X}$  is

$$d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \ln(\lambda_i(\mathbf{X})) + s_x \ln(s_x) - \omega_x \ln(\omega_x / (n + 1)), \quad (41)$$

where  $\{\lambda_i(\mathbf{X}) : i = 1, \dots, n\}$  denote the  $n$  eigenvalues of  $\mathbf{X}$  and  $s_x$  is a slack variable that satisfies  $s_x + \text{Tr}(\mathbf{X}) = s_x + \sum_{i=1}^n \lambda_i(\mathbf{X}) = \omega_x$ . We show in Appendix C.2 that the “diameter”  $D_x = \omega_x \ln(n + 1)$  and the convexity parameter  $\sigma_x = 1/\omega_x$  with respect to the  $\mathcal{L}_1$ -norm  $\|\mathbf{A}\|_1 = \sum_{i=1}^n |\lambda_i(\mathbf{A})|$ , and

$$\mathbf{X}^* = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \left\{ \langle \mathbf{\Gamma}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \right\} = \frac{\omega_x e^{\frac{1}{\mu_x} \mathbf{\Gamma}}}{1 + \text{Tr} \left( e^{\frac{1}{\mu_x} \mathbf{\Gamma}} \right)}, \quad (42)$$

where  $e^{\frac{1}{\mu_x} \mathbf{\Gamma}}$  denotes the matrix exponential of  $\frac{1}{\mu_x} \mathbf{\Gamma}$ . Computing the exact value of a matrix exponential is computationally expensive [21, 22]. It is known that an approximate value of the matrix exponential suffices for the Nesterov procedure [6] and in Corollary 6 of Appendix B we show how to efficiently approximate the matrix exponential in  $\mathcal{O}(nr \log^3 \epsilon^{-1})$  time.

## 4.2 Prox function for the dual set $\mathcal{Y}$

In every step of the Nesterov procedure we are required to solve

$$\min_{(\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) \in \mathcal{Y}} \left\{ \sum_{i=1}^m \left( v_i - \sum_{j=1}^{k_i} y_{ij} \gamma_{ij} \right) + \mu_y d_y(\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) \right\} \quad (43)$$

where  $\gamma_{ij}$ ,  $j = 1, \dots, k_i$ ,  $i = 1, \dots, m$ , are parameters that change in each iteration. Let  $d_i$  denote any prox function such that

$$\mathbf{f}_i(\boldsymbol{\gamma}) = \operatorname{argmin} \left\{ \sum_{j=1}^{k_i} \gamma_j z_j + \mu_y d_i(\mathbf{z}) : \mathbf{P}_i \mathbf{z} \leq \mathbf{b}_i \right\} \quad (44)$$

can be computed in closed form. Note that the optimization is over the  $\mathbf{z}_i$  variables and *not* the  $\mathbf{y}_i$  variables. Let  $d_v$  denote any prox function that allows one to compute

$$\mathbf{f}_v(\boldsymbol{\gamma}) = \operatorname{argmin} \left\{ \sum_{j=1}^{k_i} \gamma_j v_j + \mu_y d_v(\mathbf{v}) : \mathbf{v} \in \mathcal{V} \right\} \quad (45)$$

in closed form. Define

$$d_y(\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) = \sum_{i=1}^m v_i d_i(\mathbf{y}_i / v_i) + d_v(\mathbf{v}). \quad (46)$$

Then the optimization problem (43) can be decomposed as follows.

$$\min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=1}^m v_i \left( 1 - \max_{\mathbf{z}_i \in \mathcal{P}_i} \left\{ \sum_{j=1}^{k_i} z_j \gamma_{ij} - \mu_y d_i(\mathbf{z}_i) \right\} \right) + \mu_y d_v(\mathbf{v}) \right\}. \quad (47)$$

Let  $\boldsymbol{\gamma}_i = [\gamma_{i1}, \dots, \gamma_{ik_i}]^T$ ,  $i = 1, \dots, m$ ,

$$\nu_i = 1 - (\boldsymbol{\gamma}_i^T \mathbf{f}_i(\boldsymbol{\gamma}_i) - \mu_y d_i(\mathbf{f}_i(\boldsymbol{\gamma}_i))), \quad i = 1, \dots, m,$$

and  $\boldsymbol{\nu} = [\nu_1, \dots, \nu_m]$ . Then the optimization problem (43) has the following closed form solution

$$\begin{aligned} \mathbf{v}^* &= \mathbf{f}_v(\boldsymbol{\nu}), \\ \mathbf{y}_i^* &= v_i^* \mathbf{f}_i(-\boldsymbol{\gamma}_i), \quad i = 1, \dots, m. \end{aligned} \quad (48)$$

All that remains to be shown is that function  $d_y$  that satisfies (46) is, in fact, a prox function for the dual set  $\mathcal{Y}$ .

**Theorem 4** ([13]). *Suppose  $d_i(\mathbf{z}_i)$  is a prox-functions for the sets  $\mathcal{P}_i = \{\mathbf{z} : \mathbf{P}_i \mathbf{z} \leq \mathbf{b}_i\}$ ,  $i = 1, \dots, m$ , and  $d_v(\mathbf{v})$  is a prox function for the set  $\mathcal{V}$ .*

- (i)  $d_y(\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m) = \sum_{i=1}^m v_i d_i(\mathbf{y}_i / v_i) + d_v(\mathbf{v})$  is a prox function of the set  $\mathcal{Y}$ . The “diameter”  $D_y$  of the set  $\mathcal{Y}$  with respect to  $d_y$  is given by

$$D_y = \max_{i=1, \dots, m} \{D_i\} + D_v,$$

where  $D_i$  is the “diameter” of the set  $\mathcal{P}_i$  with respect to the prox-functions  $d_i(\mathbf{z}_i)$ ,  $i = 1, \dots, m$ , and  $D_v$  is the “diameter” of the set  $\mathcal{V}$  with respect to the prox-function  $d_v$ .

- (ii) The convexity parameter  $\sigma_y$  of the prox-function  $d_y$  is given by

$$\sigma_y = \frac{1}{\sum_{i=1}^m \frac{(1+M_i)}{\sigma_i} + \frac{1}{\sigma_y}},$$

where  $\sigma_i$  are the convexity parameters of the prox-function  $d_i(\mathbf{z}_i)$  with respect to the norm  $\|\cdot\|_i$ ,  $i = 1, \dots, m$ ,  $M_i = \max\{\|\mathbf{z}\| : \mathbf{z} \in \mathcal{P}_i\}$ ,  $i = 1, \dots, m$ ,  $\sigma_v$  is the convexity parameter of the prox-function  $d_v(\mathbf{v})$  with respect to the norm  $\|\cdot\|_v$ , and the  $\|\cdot\|_y$  in the  $\mathcal{Y}$ -space is given by  $\|(\mathbf{v}, \mathbf{y}_1, \dots, \mathbf{y}_m)\|_y = \|\mathbf{v}\|_v + \sum_{i=1}^m \|\mathbf{y}_i\|_i$ .

(iii) The parameter  $\Omega$  for the min-max problem (39) is given by

$$\Omega^2 = \max_{\sum_{i=1}^m \|\mathbf{y}_i\|_i \leq 1} \max_{\|\mathbf{X}\|_x \leq 1} \left| \sum_{i=1}^m \sum_{j=1}^{k_i} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle z_{ij} \right|^2 = \max_{i=1, \dots, m} \Omega_i^2,$$

$$\text{where } \Omega_i^2 = \max_{\|\mathbf{y}_i\|_i \leq 1} \max_{\|\mathbf{X}\|_x \leq 1} \left| \sum_{j=1}^{k_i} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle z_{ij} \right|^2, \quad i = 1, \dots, m.$$

The last result (iii) is not explicitly established in [13]; however, it follows from the results in the paper in a straightforward manner. Thus, our problem reduces to constructing prox functions for each of the packing constraints and the set  $\mathcal{V}$ . The natural prox-function for the set  $\mathcal{V}$  is

$$d_v(\mathbf{v}) = \sum_{i=1}^m v_i \ln(v_i) + s_v \ln(s_v) - \omega_v \ln(\omega_v / (m+1)). \quad (49)$$

We show in Appendix C.1 that this prox-function has a convexity parameter  $\sigma_v = 1/\omega_v$  with respect to the  $\ell_1$ -norm and the "diameter"  $D_v = \omega_v \ln(m+1)$ , and

$$\mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \{ \gamma^T \mathbf{v} + \mu_y d_v(\mathbf{v}) \} \quad \Rightarrow \quad v_i^* = \frac{\omega_v e^{-\gamma_i / \mu_v}}{1 + \sum_{k=1}^m e^{-\gamma_k / \mu_k}}, \quad i = 1, \dots, m. \quad (50)$$

Next we describe some prox-functions for the packing functions discussed in Section 2.

1.  $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle$ ,  $\mathbf{A} \succeq \mathbf{0}$ : This function is smooth and we do *not* need a prox-function.
2.  $g(\mathbf{X}) = \sum_{i,j} |x_{ij}| = \max \{ \langle \mathbf{X}, \mathbf{Z} \rangle : |Z_{ij}| \leq 1, \forall i, j \}$ : The simplest prox function is  $d(\mathbf{Z}) = \frac{1}{2} \sum_{i,j=1}^n |Z_{ij}|^2$ . For this prox-function

$$D = n^2/2, \quad \sigma = 1, \quad M = \max\{\|\mathbf{Z}\|_2 : \mathbf{Z} \in \mathcal{P}\} = n,$$

and the optimal solution  $\mathbf{Z}^* = \operatorname{argmax} \{ \langle \mathbf{Z}, \mathbf{X} \rangle - \mu d(\mathbf{Z}) : |Z_{ij}| \leq 1 \}$  is given by

$$Z_{ij}^* = \operatorname{sgn}(X_{ij}) \min \{ |X_{ij}| / \mu, 1 \}, \quad i, j = 1, \dots, n.$$

3.  $g(\mathbf{X}) = \left\| \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{X} \rangle \\ \vdots \\ \langle \mathbf{A}_k, \mathbf{X} \rangle \end{pmatrix} \right\|_2 = \max \left\{ \sum_{i=1}^k z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \|\mathbf{z}\|_2 \leq 1 \right\}$ : The simplest prox function is  $d(\mathbf{z}) = \frac{1}{2} \|\mathbf{z}\|_2^2$ . For this prox-function

$$D = \frac{1}{2}, \quad \sigma = 1, \quad M = 1,$$

and the optimal solution

$$\mathbf{z}^* = \operatorname{argmax} \left\{ \sum_{i=1}^k z_i \langle \mathbf{A}_i, \mathbf{X} \rangle - \mu d(\mathbf{z}) : \|\mathbf{z}\|_2 \leq 1 \right\} = \frac{1}{\mu + \beta} \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{X} \rangle \\ \vdots \\ \langle \mathbf{A}_k, \mathbf{X} \rangle \end{pmatrix},$$

where  $\beta = \max \left\{ \left\| \begin{pmatrix} \langle \mathbf{A}_1, \mathbf{X} \rangle \\ \vdots \\ \langle \mathbf{A}_k, \mathbf{X} \rangle \end{pmatrix} \right\|_2 - \mu, 0 \right\}$ .

Algorithm	packing SDP	Interior Point	Previous work
MAXCUT	$\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$	$\mathcal{O}(\log(\epsilon^{-1}) n^{3.5})$	$\mathcal{O}(nr \log^2(n) \cdot \epsilon^{-2} \log(\epsilon^{-1}))$ [17] $\mathcal{O}(r \log(n) \cdot \epsilon^{-6} \log^3(\epsilon^{-1}))$ [1, 27]
Coloring	$\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$	$\mathcal{O}(n^{.5} r^3 \cdot \log(\epsilon^{-1}))$	$\mathcal{O}(nr \log^3(n) \cdot \epsilon^{-4})$ [17]
Lovász $\vartheta$ -function and $\vartheta^+$	$\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$	$\mathcal{O}(n^{.5} r^3 \cdot \log(\epsilon^{-1}))$	n/a
Sparse PCA	$\mathcal{O}(n^4 \sqrt{\log(n)} \cdot \epsilon^{-1})$	$\mathcal{O}(n^{6.5} \cdot \log(\epsilon^{-1}))$	$\mathcal{O}(n^4 \sqrt{\log(n)} \cdot \epsilon^{-1})$ [7]

Table 1: Running time of SDP solvers. Interior point runtimes are from [4].  
 $n$  = number of nodes/dimension,  $r$  = number of edges/cost matrix sparsity

Theorem 4 allows one the flexibility of independently choosing approximate prox-functions for each of the packing functions and the set  $\mathcal{V}$ . However, this flexibility comes with a price, namely, that the convexity parameter  $\sigma_y$  is typically very small. Consequently, the number of iterations required to converge to an  $\epsilon$ -optimal solution increases and the numerical stability of the algorithm may be adversely affected. Consequently, it might be more efficient and numerically stable to directly define a prox-function on the  $\mathcal{Y}$  space.

## 5 Complexity results for specific packing SDPs

### 5.1 Max Cut, Graph coloring, and Lovasz- $\vartheta$ function

Recall that the packing SDP formulation of the MAXCUT problem is given by

$$\begin{aligned} \rho^* = \max & \quad \langle \mathbf{L}_D, \mathbf{X} \rangle \\ \text{s.t.} & \quad \frac{\text{Tr}(\mathbf{D})}{d_i} \langle \mathbf{e}_i \mathbf{e}_i^\top, \mathbf{X} \rangle \leq 1, \quad i = 1, \dots, n, \\ & \quad \text{Tr}(\mathbf{X}) \leq 1, \\ & \quad \mathbf{X} \succeq \mathbf{0}, \end{aligned}$$

where  $\mathbf{L}_D = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$  denotes the normalized Laplacian. Since  $\text{diag}(\mathbf{L}_D) = \mathbf{I}$  and  $\mathbf{D}/\text{Tr}(\mathbf{D})$  is feasible to the MAXCUT problem, it follows that  $\rho^* \geq 1$ . Then  $\omega_x = 1$  and  $\omega_v = n$ . Letting  $\mathbf{C} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$ , the Lagrangian relaxation of this packing SDP is

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \left\{ \mathbf{1}^\top \mathbf{v} + \langle \mathbf{C} - d(\mathbf{v}), \mathbf{X} \rangle \right\},$$

where  $d(\mathbf{v})$  denotes a diagonal matrix with  $\frac{\mathbf{D}}{d_i} v_i$  along the main diagonal. We use the prox-function  $d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \log(\lambda_i(\mathbf{X}))$  and the norm  $\|\mathbf{X}\|_x = \sum_{i=1}^n |\lambda_i(\mathbf{X})|$  for the  $\mathcal{X}$ -space, and the prox-function  $d_v(\mathbf{v}) = \sum_{i=1}^n v_i \log(v_i)$  and the norm  $\|\mathbf{v}\|_v = \sum_{i=1}^n |v_i|$  for the  $\mathcal{V}$ -space. For this choice of the prox-functions,

$$D_x = \log(n+1), \quad D_v = n \log(n+1),$$

and for this choice of norms,

$$\Omega^2 = \max_{\|\mathbf{X}\|_x \leq 1} \max_{\|\mathbf{v}\|_v \leq 1} \left| \sum_{i=1}^n v_i X_{ii} \right|^2 \leq 1, \quad \sigma_x = 1, \quad \sigma_v = 1/n.$$

Thus, the Nesterov procedure requires  $\mathcal{O}(n \log(n)/\epsilon)$  iterations to compute an  $\epsilon$ -approximate in the absolute sense. Since  $\rho^* \geq 1$ , it follows that such a solution is also  $\epsilon$ -approximate in the relative sense. Each iteration of the Nesterov procedure requires us to solve one problem of the form (35) and two optimization problems of the form (36). Thus, we have the following result.

**Corollary 1.** *The complexity of computing an  $\epsilon$ -optimal solution in the relative sense for the MAXCUT problem using the Nesterov procedure is  $\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$ , where  $r$  denotes the total number of non-zero elements in the Laplacian matrix  $\mathbf{L}$ .*

Recall that the Lagrangian relaxation of the packing SDP formulation for the graph coloring problem is given by

$$\max_{\{\mathbf{X}: \text{Tr}(\mathbf{X}) \leq n, \mathbf{X} \succeq \mathbf{0}\}} \min_{\{(\mathbf{w}, \mathbf{z}) \geq \mathbf{0}: \sum_{(i,j) \in \mathcal{E}} w_{ij} = 1, \sum_{i=1}^n z_i \leq \tau\}} \left\{ \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle - \sum_{i=1}^n z_i (X_{ii} - 1) \right\},$$

where  $\tau = \max \left\{ \min_{(i,j) \in \mathcal{E}} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle : \mathbf{X} \succeq \mathbf{0} \text{ Tr}(\mathbf{X}) \leq n \right\} \leq 4$ . We use the prox-function  $d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \log(\lambda_i(\mathbf{X}))$  and the norm  $\|\mathbf{X}\|_x = \sum_{i=1}^n |\lambda_i(\mathbf{X})|$  for the primal space, and the prox-function  $d_v(\mathbf{w}, \mathbf{v}) = \sum_{(i,j) \in \mathcal{E}} w_{ij} \log(w_{ij}) + \sum_{i=1}^n v_i \log(v_i)$  and the norm  $\|(\mathbf{w}, \mathbf{v})\|_v = \sum_{(i,j) \in \mathcal{E}} |w_{ij}| + \sum_{i=1}^n |v_i|$  for the dual space. For this choice of the prox-functions,

$$D_x = n \log(n+1), \quad D_v = \log(r) + \log(n+1) \leq 2 \log(n+1),$$

where in this case the sparsity equals the number of edges, i.e.,  $r = m$ . For this choice of norms,

$$\Omega^2 = \max_{\|\mathbf{X}\|_x \leq 1} \max_{\|(\mathbf{w}, \mathbf{v})\|_v \leq 1} \left| \sum_{(i,j) \in \mathcal{E}} w_{(i,j)} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle - \sum_{i=1}^n z_i (X_{ii} - 1) \right|^2 \leq 4, \quad \sigma_x = 1/n, \quad \sigma_v = 1.$$

Thus, the Nesterov procedure requires  $\mathcal{O}(n \log(n)/\epsilon)$  iterations to compute an  $\epsilon$ -approximate in the absolute sense. Karger et al [16] establish that  $\rho^* \geq 1/c^*$ , where  $c^*$  is optimal number of color required to color the graph. Thus, an  $\epsilon$ -approximate solution in the absolute sense is  $(c^* \epsilon)$ -approximate in the relative sense. Each iteration of the Nesterov procedure requires us to solve one problem of the form (35) and two optimization problems of the form (36). By Corollary 6 we have the following result.

**Corollary 2.** *The complexity of computing an  $\epsilon$ -optimal solution for the graph coloring SDP using the Nesterov procedure is  $\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$ , where  $r$  denotes the total number of edges in the graph.*

Recall that the packing SDP formulation for the Lovasz- $\vartheta$  function is given by

$$\begin{aligned} \max \quad & \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \langle \mathbf{E}^{(i,j)}, \mathbf{X} \rangle \leq 1, \quad (i,j) \in \mathcal{E}, \\ & \langle \mathbf{F}^{(i,j)}, \mathbf{X} \rangle \leq 1, \quad (i,j) \in \mathcal{E}, \\ & \text{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

We use the prox-function  $d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \log(\lambda_i(\mathbf{X}))$  and the norm  $\|\mathbf{X}\|_x = \sum_{i=1}^n |\lambda_i(\mathbf{X})|$  for the primal space. For the dual space we use the prox-function  $d_v(\mathbf{w}, \mathbf{v}) = \sum_{(i,j) \in \mathcal{E}} w_{ij} \log(w_{ij}) + \sum_{(i,j) \in \mathcal{E}} v_{ij} \log(v_{ij})$ , where  $\mathbf{w}$  are the dual-multipliers for  $\langle \mathbf{E}^{(i,j)}, \mathbf{X} \rangle \leq 1$  constraints and  $\mathbf{v}$  are the dual multipliers for  $\langle \mathbf{F}^{(i,j)}, \mathbf{X} \rangle \leq 1$  constraints. We use the norm  $\|(\mathbf{w}, \mathbf{v})\|_v = \sum_{(i,j) \in \mathcal{E}} |w_{ij}| + \sum_{i=1}^n |v_i|$  for the dual space. For this choice of the prox-functions,

$$D_x = \log(n+1), \quad D_v = n \log(2n+1)$$

where, as before,  $r = m$ . For this choice of norms,

$$\Omega^2 = 1, \quad \sigma_x = 1, \quad \sigma_v = 1/n.$$

Thus, the Nesterov procedure requires  $\mathcal{O}(n \log(n) \cdot \epsilon^{-1})$  iterations to compute an  $\epsilon$ -approximate in the absolute sense. Since  $\frac{1}{n} \mathbf{I}$  is feasible to (8) it follows that  $\vartheta(\mathcal{G}) \geq 1$ . Thus, an  $\epsilon$ -approximate solution in the absolute sense is  $\epsilon$ -approximate in the relative sense. Each iteration of the Nesterov procedure requires us to solve one problem of the form (35) and two optimization problems of the form (36). An analogous argument also works for Szegedy's number,  $\vartheta^+$ . Thus, we have the following result.

**Corollary 3.** *The complexity of computing an  $\epsilon$ -optimal solutions for the Lovász- $\vartheta$  and Szegedy numbers of a graph using the Nesterov procedure is  $\mathcal{O}(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1}))$ , where  $r$  denotes the total number of edges in the graph.*

We compare the best known algorithms for coloring, MAXCUT, the Lovasz- $\vartheta$  function and Szegedy’s number in Table 4.2. For moderate  $\epsilon \approx 10^{-3}$  and dense graphs,  $r = \Omega(n^{(1+\epsilon)})$  the packing SDP based methods are superior to other methods available in the literature. Another significant feature of our method is that we treat a very large class of semidefinite programming problems in a unified manner.

## 5.2 Sparse PCA

By Equations (39), (40) and (47), the Lagrangian relaxation of the sparse PCA packing SDP (14) is given by

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{(v, \mathbf{Y}) \in \mathcal{Y}} \left\{ \left\langle \mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}, \mathbf{X} \right\rangle + v \right\}, \quad (51)$$

where  $\omega_x = 1$ , and  $\mathcal{Y} = \{(v, \mathbf{Y}) : 0 \leq v \leq 1, |Y_{ij}| \leq v\}$ . See also the prox function described in 2 at the end of Section 4.2. Here we have assumed that the matrix  $\mathbf{C}$  has been scaled to ensure that  $\text{Tr}(\mathbf{C}) = 1$ . Note also, that

$$g(\mathbf{X}) = \frac{1}{\kappa^2} \sum_{i,j} |X_{ij}|.$$

For the primal set  $\mathcal{X}$ , we use the entropy prox function

$$d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \log(\lambda_i(\mathbf{X})), \quad \|\mathbf{X}\|_x = \sum_{i=1}^n |\lambda_i(\mathbf{X})|,$$

with the resulting  $D_x = \log(n+1)$ , and  $\sigma_x = 1$ . For the set  $\mathcal{Y}$ , we use the *quadratic* prox function

$$d_y(v, \mathbf{Y}) = \frac{1}{2}v^2 + \frac{1}{2} \sum_{ij} |Y_{ij}|^2, \quad \|(v, \mathbf{Y})\|_y = \left( |v|^2 + \sum_{ij} |Y_{ij}|^2 \right)^{\frac{1}{2}},$$

which results in width parameters  $D_y = n^2$ ,  $\sigma_y = 1$  and  $\Omega = 1$  by Lemma 4 in Appendix C.3. Note that the width parameters of the prox-function  $d_y(v, \mathbf{Y})$  can not be calculated with Theorem 4. We show in Appendix C.3 that the optimization problem

$$\min_{(v, \mathbf{Y}) \in \mathcal{Y}} \{ \langle \mathbf{X}, \mathbf{Y} \rangle + \ell v + \mu_y d_y(v, \mathbf{Y}) \}$$

can be solved with an active set method in  $\mathcal{O}(n^2 \log(n))$  time. The complexity is then determined by the cost of computing the matrix exponential, which could be dense at every iteration. Thus, the full eigenvalue-eigenvector decomposition is, theoretically, best.

**Corollary 4.** *The complexity of the computing an  $\epsilon$ -optimal solution for the sparse PCA problem using the Nesterov procedure is  $\mathcal{O}(n^4 \sqrt{\log(n)} \cdot \epsilon^{-1})$ .*

Our runtime matches the best known previous result of [7]. However, the procedure in [7] does not yield a feasible solution for the relaxation – one needs to conduct a 1-dimensional search over  $v$  to obtain a feasible solution.

## 6 Numerical experiments

We tested our algorithm on the packing SDP relaxation of the Sparse PCA problem (14). We solve the associated saddle point problem (51) described in Section 5.2 and then round the resulting solution to

$n$		$\kappa$		Ind. Vars. [Obs.]		Dep. Vars. [Obs.]		Noise Vars. [Obs.]	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	124	40	4	2 [40]	30 [4]	1 [20]	1 [2]	1 [20]	1 [2]
240	244	80	4	2 [80]	60 [4]	1 [40]	1 [2]	1 [40]	1 [2]
360	364	120	4	2 [120]	90 [4]	1 [60]	1 [2]	1 [60]	1 [2]
480	484	160	4	2 [160]	120 [4]	1 [80]	1 [2]	1 [80]	1 [2]
600	604	200	4	2 [200]	150 [4]	1 [100]	1 [2]	1 [100]	1 [2]
720	724	240	4	2 [240]	180 [4]	1 [120]	1 [2]	1 [120]	1 [2]
840	844	280	4	2 [280]	210 [4]	1 [140]	1 [2]	1 [140]	1 [2]
960	964	320	4	2 [320]	240 [4]	1 [160]	1 [2]	1 [160]	1 [2]
1080	1084	360	4	2 [360]	270 [4]	1 [180]	1 [2]	1 [180]	1 [2]
1200	1204	400	4	2 [400]	300 [4]	1 [200]	1 [2]	1 [200]	1 [2]
1320	1324	440	4	2 [440]	330 [4]	1 [220]	1 [2]	1 [220]	1 [2]
1440	1444	480	4	2 [480]	360 [4]	1 [240]	1 [2]	1 [240]	1 [2]
1560	1564	520	4	2 [520]	390 [4]	1 [260]	1 [2]	1 [260]	1 [2]
1680	1684	560	4	2 [560]	420 [4]	1 [280]	1 [2]	1 [280]	1 [2]
1800	1804	600	4	2 [600]	450 [4]	1 [300]	1 [2]	1 [300]	1 [2]
1920	1924	640	4	2 [640]	480 [4]	1 [320]	1 [2]	1 [320]	1 [2]
2040	2044	680	4	2 [680]	510 [4]	1 [340]	1 [2]	1 [340]	1 [2]
2160	2164	720	4	2 [720]	540 [4]	1 [360]	1 [2]	1 [360]	1 [2]
2280	2284	760	4	2 [760]	570 [4]	1 [380]	1 [2]	1 [380]	1 [2]
2400	2404	800	4	2 [800]	600 [4]	1 [400]	1 [2]	1 [400]	1 [2]
3600	3604	1200	4	2 [1200]	900 [4]	1 [600]	1 [2]	1 [600]	1 [2]
4800	4804	1600	4	2 [1600]	1200 [4]	1 [800]	1 [2]	1 [800]	1 [2]
6000	6004	2000	4	2 [2000]	1500 [4]	1 [1000]	1 [2]	1 [1000]	1 [2]

Table 2: Description of the artificial data.

feasibility via Corollary 5. We describe our implementation in detail in Section 6.1. We tested our implementation on random instances generated in a manner similar to that described in [7] (see also [31]) to test runtime performance. We describe data generation in Section 6.2, and report the results of our numerical experiments in Section 6.3. The code for both the solution algorithm and data generation was written in MATLAB 2008a <sup>®</sup> [20] and run serially on either one or four cores of an Opteron 2.6 GHz dual-core two processor machine with 8 GB of RAM<sup>2</sup>. In order to accurately report the CPU time we only report the single core runtimes in the main results. The runtime difference between single core and four cores is displayed in Figure 3.

## 6.1 Implementation details

Our proposed solution algorithm for packing SDP requires us to initialize with a *feasible* dual solution. The dual feasible set for the sparse PCA packing SDP (51) is

$$\mathcal{Y} = \{(v, \mathbf{Y}) : 0 \leq v \leq 1, |Y_{ij}| \leq v\}$$

In our numerical experiments, we randomly and uniformly sampled  $v^{(0)}$  from the  $[0, 1]$  interval and then randomly and uniformly sampled each  $Y_{ij}^{(0)}$  from  $[-v^{(0)}, v^{(0)}]$ .

The dual gradient is given by the optimal solution of the smoothed optimization problem

$$\max_{\mathbf{X} \in \mathcal{X}} \left\{ \left\langle \mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}^{(k)}, \mathbf{X} \right\rangle + \mu_x d_x(\mathbf{X}) \right\}$$

<sup>2</sup>Computational facilities were at the College of William and Mary which were provided with the assistance of the National Science Foundation, the Virginia Port Authority, Sun Microsystems, and Virginia's Commonwealth Technology Research Fund.

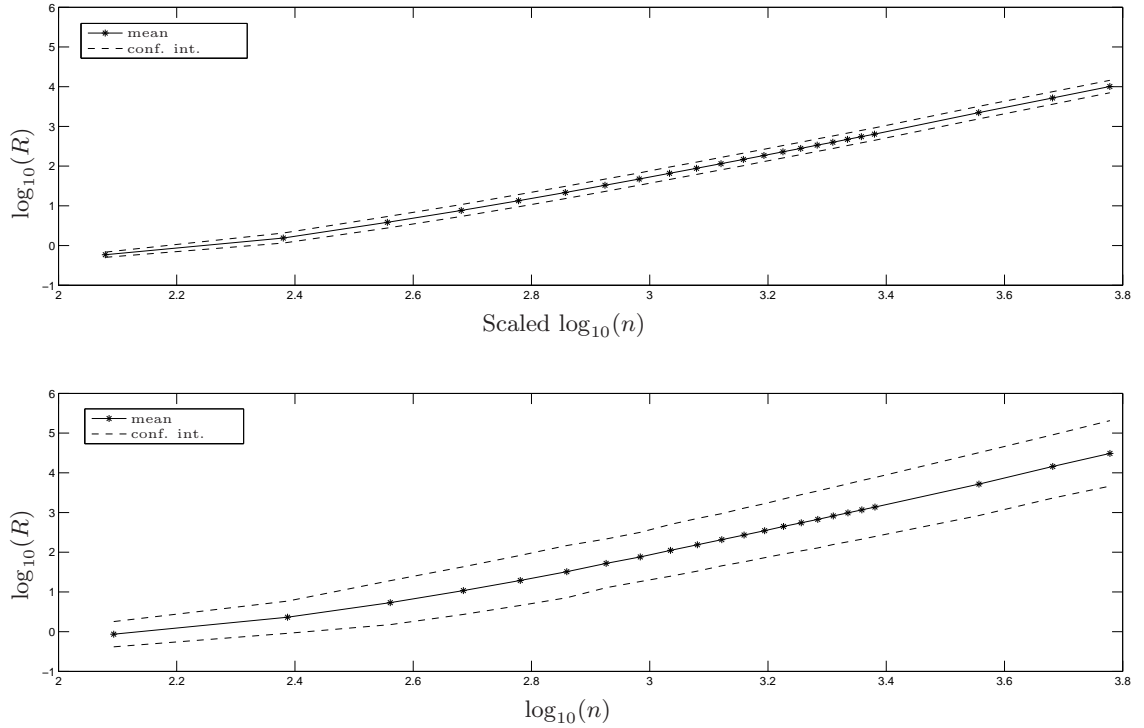


Figure 2: The average runtimes ( $R$ ) for the packing SDP on Scaled and Fixed. The dotted lines around the averages represent 95% confidence intervals. Runtime reported is in CPU seconds.

where the prox function  $d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \log(\lambda_i(\mathbf{X}))$ , and  $\{\lambda_i\}$  denotes the set of eigenvalues of  $\mathbf{X}$ . The optimal

$$\mathbf{X}^* = \frac{e^{\frac{1}{\mu_x} \cdot (\mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}^{(k)})}}{1 + \text{Tr} \left( e^{\frac{1}{\mu_x} \cdot (\mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}^{(k)})} \right)}.$$

We computed  $\mathbf{X}^*$  using the standard matrix exponential calculator `expm` in MATLAB<sup>®</sup>. In order to compute the two step sizes, we need to solve two minimization problems of form

$$\min_{(v, \mathbf{Y}) \in \mathcal{Y}} \{ \langle \mathbf{X}, \mathbf{Y} \rangle + \ell v + \mu_v d_v(v, \mathbf{Y}) \}.$$

We implemented the method described in Appendix C.3 to solve this optimization problem. We found that  $\mu_x < 0.005$  exceeded the precision capabilities of the MATLAB<sup>®</sup> `expm` function; therefore, we set  $\mu_x = 0.01 / \ln(n + 1)$  in our algorithm. The quadratic prox function used in the step-length calculation was very stable with respect to the  $\mu_v$  parameter. An interesting open direction would be an implementation of Nesterov's algorithm using the quadratic prox function for the primal optimization step.

The duality gap  $\eta^{(k)}$  at a given iterate  $(\mathbf{X}^{(k)}, \mathbf{v}^{(k)}, \mathbf{Y}^{(k)})$  is given by

$$\begin{aligned} \eta^{(k)} &= \min_{(v, \mathbf{Y}) \in \mathcal{V}} \left\{ \langle \mathbf{C}, \mathbf{X}^{(k)} \rangle - \frac{1}{\kappa^2} \langle \mathbf{Y}, \mathbf{X}^{(k)} \rangle + v \right\} - \max_{\mathbf{X} \in \mathcal{X}} \left\{ \langle \mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}^{(k)}, \mathbf{X} \rangle + v^{(k)} \right\} \\ &= \left( \langle \mathbf{C}, \mathbf{X}^{(k)} \rangle - \max_{i,j} \{0, 1 - |X_{ij}^{(k)}|\} \right) - \left( v^{(k)} + \max\{0, \lambda_{\max}(\mathbf{C} - \frac{1}{\kappa^2} \mathbf{Y}^{(k)})\} \right) \end{aligned}$$

$n$		$\mu$		$\sigma$		max	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	124	0.10	0.08	0.01	0.02	0.13	0.12
240	244	0.25	0.24	0.01	0.02	0.28	0.28
360	364	0.63	0.62	0.02	0.04	0.66	0.72
480	484	1.25	1.21	0.02	0.03	1.28	1.3
600	604	2.21	2.13	0.01	0.03	2.24	2.18
720	724	3.55	3.43	0.04	0.05	3.62	3.53
840	844	5.39	5.14	0.02	0.06	5.43	5.22
960	964	7.74	7.41	0.04	0.11	7.83	7.56
1080	1084	10.80	10.28	0.05	0.15	10.88	10.5
1200	1204	14.43	13.78	0.09	0.22	14.55	14.14
1320	1324	18.91	17.98	0.07	0.23	19.00	18.33
1440	1444	24.12	22.91	0.07	0.38	24.20	23.4
1560	1564	30.38	28.83	0.18	0.55	30.75	29.78
1680	1684	37.61	35.64	0.12	0.8	37.80	36.83
1800	1804	45.70	43.39	0.24	1	46.25	44.67
1920	1924	55.18	52.15	0.25	1.28	55.50	53.8
2040	2044	65.29	62.4	0.28	1.6	65.75	64.6
2160	2164	77.29	73.5	0.37	1.96	78.00	76.2
2280	2284	90.32	85.94	0.37	2.76	90.75	90
2400	2404	104.67	99.66	0.47	3.16	105.50	104
3600	3604	363.73	325.23	1.35	13.36	365.71	344.29
4800	4804	850.32	784.39	5.65	20.14	865.71	815.71
6000	6004	1651.49	1518.82	7.27	37.06	1661.67	1581.67

Table 3: CPU seconds per iteration for the Scaled and Fixed families of Sparse PCA SDP relaxations. Mean ( $\mu$ ), standard deviation ( $\sigma$ ) and maximum shown.

To calculate the maximum eigenvalue, we used the Matlab function `eigs`, which calls the Fortran library, ARPACK [2], a Lanczos method [10] when used with symmetric matrices. We terminated our algorithm when the duality gap  $\eta^{(k)}$  reached the prescribed precision  $\epsilon$ .

In order to guarantee monotone convergence we implemented the following modification suggested in [23]:

$$\mathbf{y}^{(k)} \leftarrow \operatorname{argmin} \left\{ \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{x}^{(k-1)}), \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{y}^{(k-1)}), \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{y}^{(k)}) \right\}$$

Finally, we compute *feasible*  $\epsilon$ -optimal solutions to the Sparse PCA packing SDP (14) using Corollary 5 (see Appendix A).

## 6.2 Artificial data

Since our method is not a full method for determining the principal components, but rather a solution algorithm for an SDP relaxation of the sparse PCA problem, we focused our experiments on SDP instances where the number of components and their sparsity were known. We based our simulated data on the following example introduced in [31] (see, also [7]).

(i) Descriptive variables

$$Y_1 \sim \mathcal{N}(0, \sigma_1), \quad Y_2 \sim \mathcal{N}(0, \sigma_2), \quad D = w_1 Y_1 + w_2 Y_2 + \delta, \quad \delta \sim \mathcal{N}(0, \sigma_3),$$

where  $Y_1, Y_2$  and  $\delta$  were independent random variables and  $\mathcal{N}(\mu, \sigma)$  denotes a Normal distribution with mean  $\mu$  and variance  $\sigma$ .

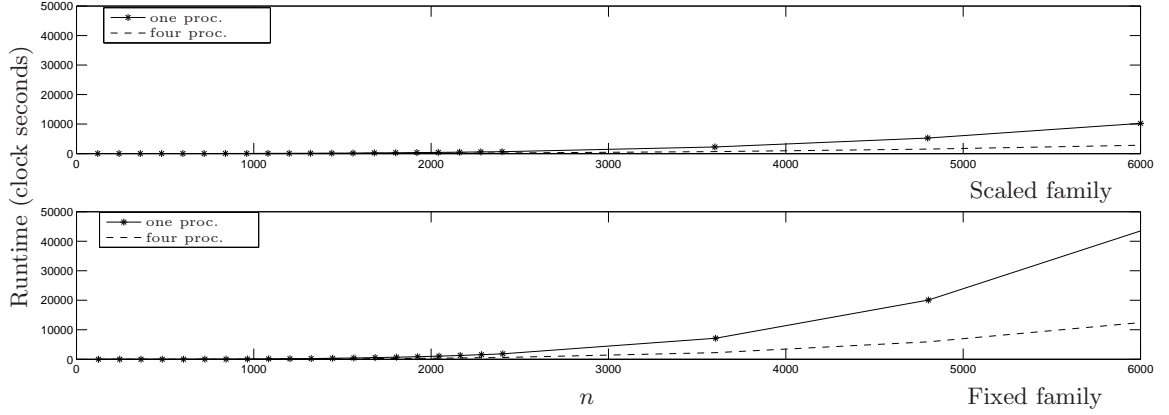


Figure 3: The average runtimes of the Packing SDP algorithm using one versus four processors. Runtime reported is in actual seconds.

(ii) The observations

$$X_i = \begin{cases} Y_1 + \eta_i, & i = 1, 2, 3, 4, \\ Y_2 + \eta_i, & i = 5, 6, 7, 8, \\ D + \eta_i, & i = 9, 10. \end{cases}$$

where each  $\eta_i \sim \mathcal{N}(0, 1)$ .

We generalized this example to create two instance families with covariance matrices varying in size from  $120 \times 120$  to  $6004 \times 6004$ .

1. *Scaled family*: In this instance family the descriptive variables were a particular instance of the example above, i.e. in particular

$$Y_1, Y_2 \sim \text{iid } \mathcal{N}(0, 200), \quad D = 0.1Y_1 - 0.35Y_2$$

The number of observations was scaled up by a positive scaling factor  $s$  as follows.

$$X_i = \begin{cases} Y_1 + \eta_i, & 1 \leq i \leq 4s, \\ Y_2 + \eta_i, & 4s + 1 \leq i \leq 8s, \\ D + \eta_i, & 8s + 1 \leq i \leq 10s, \\ \eta_i, & 10s + 1 \leq i \leq 12s. \end{cases}$$

For this family, the size of the covariance matrix is  $n = 12s$  and the sparsity variable  $\kappa = 4s$ . We choose ten instances each for  $s \in \{10, 20, \dots, 200\}$ , and  $\{300, 400, 500\}$ .

2. *Fixed family*: In this instance family we scaled the number of descriptive variables by a positive scale factor  $s$  as follows:

$$Y_i \sim \mathcal{N}(0, 200), i = 1, \dots, c, \quad D = \sum_{k=1}^s w_k Y_k, j = 1, 2,$$

where

$$w_k = \begin{cases} 0.1k, & k < s, \\ -0.35, & k = s. \end{cases}$$

The observations were generated as follows:

$$X_i = \begin{cases} Y_t + \eta_i, & 4(t-1) < i \leq 4t, t = 1, \dots, c, \\ D + \eta_i, & i = 4c + j, j = 1, 2 \\ \eta_i, & i = 4c + 2 + j, j = 1, 2. \end{cases}$$

$n$		$\mu$		$\sigma$		max	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	124	6.2	12.4	1.0	6.2	7	21
240	244	6.2	10.9	1.0	5.3	7	18
360	364	6.2	10.4	1.0	5.6	7	17
480	484	6.2	10.9	1.0	6.2	7	19
600	604	6.2	11.3	1.0	6.6	7	20
720	724	6.2	11.9	1.0	7.3	7	22
840	844	6.2	12.6	1.0	7.7	7	23
960	964	6.2	12.8	1.0	8.0	7	24
1080	1084	6.2	13.8	1.0	9.0	7	26
1200	1204	6.2	14.3	1.0	9.3	7	26
1320	1324	6.2	14.6	1.0	9.4	7	28
1440	1444	6.2	15.3	1.0	10.2	7	29
1560	1564	6.2	15.6	1.0	10.3	7	29
1680	1684	6.2	16.3	1.0	11.0	7	32
1800	1804	6.2	16.8	1.0	11.4	7	32
1920	1924	6.2	17.1	1.0	11.9	7	34
2040	2044	6.2	17.6	1.0	12.3	7	35
2160	2164	6.2	18	1.0	12.5	7	35
2280	2284	6.2	18.5	1.0	13.1	7	36
2400	2404	6.2	18.8	1.0	13.3	7	38
3600	3604	6.2	22.5	1.0	16.3	7	44
4800	4804	6.2	26	1.0	19.2	7	52
6000	6004	6.2	29.1	1.0	21.6	7	59

Table 4: Iteration counts for the packing SDP algorithm on the on the Scaled and Fixed families of the Sparse PCA SDP relaxations.

Thus, we have 4 observations per descriptive variable. For this family, the size of the covariance matrix  $n = 4s + 4$  and  $\kappa = 4$ . We choose ten instances each for  $c \in \{30, 60, \dots, 600\}$  and  $\{900, 1200, 1500\}$ .

We summarize the data generated in Table 2. We used the theoretical covariance matrices, but did not notice any significant runtime difference when the sampled covariance matrices were used.

### 6.3 Results

We report the average runtimes in Table 3 and the average iteration count in Table 4. In Table 3 (resp. Table 4) the column labeled  $\mu$  reports the CPU seconds (resp. number of iterations) averaged over 10 instances for each problem size, the column labeled  $\sigma$  reports the standard deviation of the runtimes (resp. iteration counts), and the column labeled ‘max’ reports the maximum CPU time (resp. iterations) over the 10 instances. In Figure 2 we display a plot of the runtimes as a function of the problem size  $n$ . These numerical results support the following observations.

- (a) The number of iterations required to solve instances from the the Scaled family was small and remained relatively constant, ranging from 4 to 7. The number of iterations required for solving instances from the Fixed family had more variance – requiring between 3 and 59 iterations. The number of iterations tended to grow with larger instances.
- (b) The best fit line for the average runtime (ignoring the three smallest instances) is as follows:

$$\begin{aligned} \text{Scaled family : } \log(R) &= -6.71 + 2.82 \log(n) \\ \text{Fixed family : } \log(R) &= -7.36 + 3.01 \log(n) \end{aligned}$$

Thus, the running time grows as some function which is  $\mathcal{O}(n^3)$ , outperforming the theoretical bound by  $\Omega(n)$ .

From the results reported in Table 3, it is easy to check that the average runtime per iteration grew at the same rate as the overall runtime. The Fixed family had a slightly smaller runtime per iteration growth than the Scaled family, which implies that the main bottleneck is the  $\mathcal{O}(n^3)$  operations required to compute the matrix exponential. The runtime per iteration (and, also the overall runtime) should decrease significantly if the Lanczos-Shift-Invert method is used to compute the matrix exponential. Another possibility is to use a quadratic prox function for the primal smoothing.

- (c) The instances from the Fixed family were more difficult when compared to the instances from the Scaled family. The principal difference between the two families was that the cardinality constraint,  $\kappa$  remained fixed at 4 for the Fixed family, whereas in the Scaled family,  $\kappa$  grew linearly with the scaling factor  $s$ .
- (d) In Figure 3 we display the CPU times for solving the problem using multi-threading on a four core machine. Multi-threading on larger problems resulted in a significant improvement – the runtime for a problem with  $n = 6000$  decreased by a factor of 4. Since the CPU time MATLAB® reported included all four cores’ CPU time, we report actual clock time. For the single core experiments, the CPU time and clock time did not differ significantly, with a 0.36% average percentage difference across all runs.

We compared the performance of our algorithm against SeDuMi [25], an interior-point based code for solving SDPs. In general, our algorithm was orders of magnitude faster than SeDuMi [25]. However, since we did not optimize the various settings of SeDuMi, and, for the settings we use, SeDuMi was not able to solve the instances we studied, e.g. SeDuMi crashed on instances of with covariance matrices larger than  $50 \times 50$ , we do not have SeDuMi runtimes to report. Our runtimes are also significantly superior to the runtimes reported in [7].

## Acknowledgements

We thank Satyen Kale and Michael Lewis for useful discussions and David Steurer for pointing out the reformulation of the MAXCUT problem. We also thanks Tom Crockett for his help setting up the computational experiments.

## References

- [1] S. ARORA AND S. KALE, *A combinatorial, primal-dual approach to semidefinite programs*, in Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007.
- [2] ARPACK is a set of Fortran77 implementations to solve large-scale eigenvalue problems. Code by R. Lehoucq, K. Maschhoff, D. Sorensen, and C. Yang. <http://www.caam.rice.edu/software/ARPACK/>.
- [3] A. BEN-TAL AND A. NEMIROVSKI, *Non-euclidean restricted memory level method for large-scale convex optimization*, Mathematical Programming, 102 (2005), pp. 407–456.
- [4] S. J. BENSON, Y. YE, AND X. ZHANG, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461 (electronic).
- [5] U. BERGER, *Brown’s original fictitious play*, Journal of Economic Theory, 135 (2007), pp. 572–578.
- [6] A. D’ASPREMONT, *Smooth optimization with approximate gradient*, SIAM Journal on Optimization, 19 (2008), pp. 1171–1183.
- [7] A. D’ASPREMONT, L. EL GHAOU, M. I. JORDAN, AND G. R. G. LANCKRIET, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Rev., 49 (2007), pp. 434–448 (electronic).
- [8] E. GALLOPOULOS AND Y. SAAD, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.

- [9] M. X. GOEMANS AND D. P. WILLIAMSON, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM, 42 (1995), pp. 1115–1145.
- [10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.
- [11] N. GVOZDENOVIĆ AND M. LAURENT, *The operator  $\Psi$  for the chromatic number of a graph*, SIAM J. Optim., 19 (2008), pp. 572–591.
- [12] M. HOCHBRUCK AND C. LUBICH, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.
- [13] S. HODA, A. GILPIN, AND J. PENA, *Smoothing techniques for computing Nash equilibria of sequential games*, tech. rep., Technical report, Carnegie Mellon University, 2008.
- [14] G. IYENGAR, D. J. PHILLIPS, AND C. STEIN, *Approximation algorithms for semidefinite packing problems with applications to maxcut and graph coloring*, in Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization, 2005, pp. 152–166. Submitted to SIAM J. Optim.
- [15] S. KALE. personal communication, 2009.
- [16] D. KARGER, R. MOTWANI, AND M. SUDAN, *Approximate graph coloring by semidefinite programming*, J. ACM, 45 (1998), pp. 246–265.
- [17] P. KLEIN AND H.-I. LU, *Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING*, in Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996), New York, 1996, ACM, pp. 338–347.
- [18] L. LOVÁSZ, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.
- [19] Z. LU, R. MONTEIRO, AND M. YUAN, *Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression*, Arxiv preprint arXiv:0904.0691, (2009).
- [20] MATLAB<sup>®</sup>, Mathworks, Inc. <http://www.mathworks.com>.
- [21] C. MOLER AND C. VAN LOAN, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (1978), pp. 801–836.
- [22] ———, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49 (electronic).
- [23] Y. NESTEROV, *Smooth minimization of nonsmooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.
- [24] D. STEURER. personal communication, 2009.
- [25] J. STURM, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization methods and software, 11 (1999), pp. 625–653. See also <http://sedumi.ie.lehigh.edu/>.
- [26] M. SZEGEDY, *A note on the  $\vartheta$  number of Lovász and the generalized Delsarte bound*, in SFCS '94: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Washington, DC, USA, 1994, IEEE Computer Society, pp. 36–39.
- [27] L. TREVISAN, *Max Cut and the Smallest Eigenvalue*, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2009. Arxiv preprint arXiv:0806.1978.
- [28] J. VAN DEN ESHOF AND M. HOCHBRUCK, *Preconditioning lanczos approximations to the matrix exponential*, 2004. To appear in SIAM J. of Sci. Comp.

- [29] K. WEINBERGER AND L. SAUL, *Unsupervised learning of image manifolds by semidefinite programming*, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 2 (2004), pp. II-988–II-995 Vol.2.
- [30] L. XIAO, J. SUN, AND S. BOYD, *A duality view of spectral methods for dimensionality reduction*, in Proceedings of the 23rd International Conference on Machine Learning (ICML), 2006, pp. 1041–1048.
- [31] H. ZOU, T. HASTIE, AND R. TIBSHIRANI, *Sparse principal component analysis*, Journal of computational and graphical statistics, 15 (2006), pp. 265–286.

## A Rounding sparse PCA solutions

**Corollary 5.** *Suppose  $\kappa > 1$ . Let  $\bar{\mathbf{X}}$  denote an  $\epsilon$ -saddle point for the sparse PCA saddle-point problem (51). Let  $\bar{\mathbf{Y}} = \mathbf{diag}(\bar{\mathbf{X}})$  and  $\bar{\mathbf{Z}} = \bar{\mathbf{X}} - \bar{\mathbf{Y}}$ . If  $\langle \mathbf{C}, \bar{\mathbf{Z}} \rangle \leq 0$  then set  $\hat{\mathbf{X}} = \bar{\mathbf{Y}}$ . Otherwise, let*

$$\gamma = \frac{g(\bar{\mathbf{Z}})}{1 - g(\bar{\mathbf{Y}})}$$

and set

$$\hat{\mathbf{X}} = \bar{\mathbf{Y}} + \min\{1, \frac{1}{\gamma}\}\bar{\mathbf{Z}}. \quad (52)$$

Then  $\hat{\mathbf{X}}$  is a feasible,  $\epsilon$ -optimal solution to the sparse PCA packing SDP(14).

*Proof.* The packing constraint in the sparse PCA problem is given by

$$g(\mathbf{X}) = \frac{1}{\kappa^2} \sum_{i,j} |X_{ij}|.$$

Since  $\mathbf{Tr}(\bar{\mathbf{X}}) = 1$  and  $\bar{\mathbf{Y}} = \mathbf{diag}(\bar{\mathbf{X}}) \succeq \mathbf{0}$  it follows that

$$\sum_{i,j} |\bar{Y}_{ij}| = \sum_{i=1}^n |\bar{Y}_{ii}| = 1 < \kappa^2,$$

i.e.  $\bar{\mathbf{Y}}$  is feasible for the sparse PCA packing SDP (14).

First consider the case when  $\langle \mathbf{C}, \bar{\mathbf{Z}} \rangle \leq 0$ . Then  $\langle \mathbf{C}, \bar{\mathbf{Y}} \rangle = \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle > \langle \mathbf{C}, \bar{\mathbf{X}} \rangle$ . Since  $\bar{\mathbf{Y}}$  is feasible for (14), it follows that  $\bar{\mathbf{Y}}$  is an  $\epsilon$ -optimal solution for the sparse PCA packing SDP (14).

Next, consider the case  $\langle \mathbf{C}, \bar{\mathbf{Z}} \rangle > 0$ . In this case, the proof of feasibility and  $\epsilon$ -optimality of  $\hat{\mathbf{X}}$  follows the same lines as in Theorem 1. We have the following two cases.

- (a)  $\gamma \leq 1$ . In this case, the argument for  $\epsilon$ -optimality is exactly as stated in Theorem 1 since  $\bar{\mathbf{X}}$  is already feasible.
- (b)  $\gamma > 1$ . Note that  $\min\{1, \frac{1}{\gamma}\} = \frac{1}{\gamma} < 1$ . Also, the convexity and positive homogeneity of  $g$  imply that

$$g(\bar{\mathbf{X}}) = g\left(\frac{1}{2}(2\bar{\mathbf{Y}}) + \frac{1}{2}(2\bar{\mathbf{Z}})\right) \leq \frac{1}{2}g(2\bar{\mathbf{Y}}) + \frac{1}{2}g(2\bar{\mathbf{Z}}) = g(\bar{\mathbf{Y}}) + g(\bar{\mathbf{Z}}).$$

Then,

$$\begin{aligned} g(\hat{\mathbf{X}}) &= g\left(\bar{\mathbf{Y}} + \frac{1}{\gamma}\bar{\mathbf{Z}}\right), \\ &\leq g(\bar{\mathbf{Y}}) + \frac{1}{\gamma}g(\bar{\mathbf{Z}}), \\ &= g(\bar{\mathbf{Y}}) + \frac{1 - g(\bar{\mathbf{Y}})}{g(\bar{\mathbf{Z}})} \cdot g(\bar{\mathbf{Z}}) = 1, \end{aligned}$$

where the last equality follows from the definition of  $\gamma$ . Also,

$$\widehat{\mathbf{X}} = \bar{\mathbf{Y}} + \frac{1}{\gamma}\bar{\mathbf{Z}} = (1 - \frac{1}{\gamma})\bar{\mathbf{Y}} + \frac{1}{\gamma}\bar{\mathbf{X}} \succeq \mathbf{0}.$$

Thus,  $\widehat{\mathbf{X}}$  is feasible for (14). To show that  $\bar{\mathbf{X}}$  is  $\epsilon$ -optimal, we first let  $\bar{d} = g(\bar{\mathbf{X}})$  as defined in Theorem 1. We must have  $g(\bar{\mathbf{X}}) > 1$ , or else  $\bar{\mathbf{X}}$  was feasible to begin with. Then, since  $1 < g(\bar{\mathbf{X}}) \leq g(\bar{\mathbf{Y}}) + g(\bar{\mathbf{Z}}) < g(\bar{\mathbf{Y}}) + g(\bar{\mathbf{X}})g(\bar{\mathbf{Z}})$ ,

$$\frac{1}{\bar{d}} = \frac{1}{g(\bar{\mathbf{X}})} \leq \frac{1}{g(\bar{\mathbf{X}})} \frac{g(\bar{\mathbf{X}})g(\bar{\mathbf{Z}})}{1 - g(\bar{\mathbf{Y}})} = \frac{g(\bar{\mathbf{Z}})}{1 - g(\bar{\mathbf{Y}})} = \frac{1}{\gamma}$$

Thus,

$$\begin{aligned} \langle \mathbf{C}, \widehat{\mathbf{X}} \rangle &= \langle \mathbf{C}, \bar{\mathbf{Y}} \rangle + \frac{1}{\gamma} \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle \\ &\geq \langle \mathbf{C}, \bar{\mathbf{Y}} \rangle + \frac{1}{\bar{d}} \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle \\ &\geq \langle \mathbf{C}, \bar{\mathbf{Y}} \rangle + \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle - (\bar{d} - 1) \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle \tag{53} \\ &\geq \langle \mathbf{C}, \bar{\mathbf{X}} \rangle - (\bar{d} - 1) \langle \mathbf{C}, \bar{\mathbf{X}} \rangle, \tag{54} \end{aligned}$$

where (53) follows from same argument as in Theorem 1, and (54) follows from that fact that  $\langle \mathbf{C}, \bar{\mathbf{Y}} \rangle + \langle \mathbf{C}, \bar{\mathbf{Z}} \rangle = \langle \mathbf{C}, \bar{\mathbf{X}} \rangle$  and  $\mathbf{C}, \bar{\mathbf{Y}} \succeq \mathbf{0}$  (so  $\langle \mathbf{C}, \bar{\mathbf{Y}} \rangle \geq 0$ ). Then, Equations (25), (26) and (27) from the proof of Theorem 1 imply that  $\langle \mathbf{C}, \widehat{\mathbf{X}} \rangle \geq \rho^* - \epsilon$ .

Thus, we have that  $\widehat{\mathbf{X}}$  is a feasible,  $\epsilon$ -optimal solution to (3).  $\square$

## B Matrix exponential via Lanczos iterations

The most expensive step in using the Nesterov procedure to solve the Lagrangian relaxation of the packing SDP is computing the optimal

$$\mathbf{X}^* = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \left\{ \langle \mathbf{\Gamma}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \right\} = \frac{\omega_x e^{\frac{1}{\mu_x} \mathbf{\Gamma}}}{1 + \operatorname{Tr} \left( e^{\frac{1}{\mu_x} \mathbf{\Gamma}} \right)}, \tag{55}$$

where  $e^{\frac{1}{\mu} \mathbf{\Gamma}}$  denotes the matrix exponential. Let  $\mathbf{\Gamma} = \mathbf{V} \mathbf{diag}(\boldsymbol{\gamma}) \mathbf{V}^\top$ , where  $\boldsymbol{\gamma}$  denotes the vector of eigenvalues of  $\mathbf{\Gamma}$  and  $\mathbf{V}$  denotes the eigenvectors of  $\mathbf{\Gamma}$ . Then

$$\mathbf{X}^* = \frac{\mathbf{V} \mathbf{diag}(e^{\frac{1}{\mu} \boldsymbol{\gamma}}) \mathbf{V}^\top}{1 + \sum_{i=1}^n e^{\frac{\gamma_i}{\mu}}},$$

where  $\mathbf{diag}(e^{\frac{1}{\mu} \boldsymbol{\gamma}})$  denotes a diagonal matrix with the  $i$ -th entry equal to  $e^{\frac{\gamma_i}{\mu}}$ . Thus, we can compute  $\mathbf{X}^*$  by first computing the eigendecomposition of  $\mathbf{\Gamma}$ . However, the complexity of this procedure is  $\mathcal{O}(n^3)$ .

Matrix exponentials appear in solving discrete approximations of elliptic partial differential equations. Therefore, there has been a lot of interest in the applied numerical mathematics community to efficiently compute approximations to a matrix exponential. Currently, the best known techniques for efficiently computing the matrix exponential rely on using the Lanczos method to computing the basis of the Krylov subspaces associated with the matrix  $\mathbf{\Gamma}$  [8, 12] or  $(\mathbf{I} + \theta \mathbf{\Gamma})^{-1}$  [28] for an appropriately chosen  $\theta$ . Theorem 3.3 of [28] indicates that  $\mathcal{O}(\log^2(1/\epsilon))$  Lanczos iterations are required to approximate the matrix-vector product  $\exp(\mathbf{\Gamma})\mathbf{v}$  for any  $\mathbf{v} \in \mathbb{R}^n$ . Setting  $\mathbf{v} = \mathbf{e}_i, i = 1, \dots, n$  results in an overall complexity of  $\mathcal{O}(nr \log^3(1/\epsilon))$ , where  $r$  denotes the number of non-zero elements in  $\mathbf{\Gamma}$ . Thus, we have the following corollary.

**Corollary 6.** *The complexity of computing  $\exp(\mathbf{\Gamma}/\mu)$  via SHIFT-INVERT-LANCZOS procedure proposed in [28] is  $\mathcal{O}(nr) \log^3(\epsilon^{-1})$ , where  $r$  denotes the number of non-zero terms in the matrix  $\mathbf{\Gamma}$ . Also, computing  $\exp(\mathbf{\Gamma}/\mu)\mathbf{v}$  for any  $\mathbf{v} \in \mathbb{R}^n$  requires  $\mathcal{O}(r \log^3(\epsilon^{-1}))$  time.*

## C Details of our prox functions

### C.1 The dual prox function

In order to keep the notation simple, we relabel the slack variable  $s_v$  in (49) as  $v_{m+1}$ .

**Lemma 2.** Let  $d_v(\mathbf{v}) = \sum_{i=1}^{m+1} v_i \ln(v_i) - \omega_v \ln(\omega_v/(m+1))$ , where  $\mathbf{v} \in \mathcal{V} = \{\mathbf{v} : \mathbf{v} \geq 0, \sum_{i=1}^{m+1} v_i = \omega_v\}$ .

1.  $d_v$  is strongly convex with convexity parameter  $\sigma_v = \frac{1}{\omega_v}$  on the interior of  $\mathcal{V}$ .

2. Let

$$\mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \{\boldsymbol{\gamma}^\top \mathbf{v} + \mu_v d_v(\mathbf{v})\}. \quad (56)$$

Then

$$v_i^* = \frac{\omega_v e^{-\gamma_i/\mu_v}}{\sum_{k=1}^{m+1} e^{-\gamma_k/\mu_v}}, \quad i = 1, \dots, m+1.$$

3.  $d_v(\cdot) \geq 0$  on  $\mathcal{V}$  and  $D_v = \omega_v \log(m+1)$ .

*Proof.* The Hessian

$$\nabla^2 d_v(\mathbf{v}) = \mathbf{diag}([1/v_1, \dots, 1/v_{m+1}])$$

is clearly positive definite on the interior of  $\mathcal{V}$ . For any  $\mathbf{w} \in \mathbb{R}^{m+1}$ ,

$$\begin{aligned} \mathbf{w}^\top \nabla^2 d_v(\mathbf{v}) \mathbf{w} &= \sum_{i=1}^{m+1} \frac{w_i^2}{v_i}, \\ &= \left( \sum_{i=1}^m \frac{w_i^2}{v_i} \right) \left( \frac{1}{\omega_v} \sum_{i=1}^{m+1} v_i \right), \\ &\geq \frac{1}{\omega_v} \left( \sum_{i=1}^m \frac{|w_i|}{\sqrt{v_i}} \sqrt{v_i} \right)^2, \end{aligned} \quad (57)$$

$$= \frac{1}{\omega_v} \|\mathbf{w}\|_1^2, \quad (58)$$

where (57) follows from the Cauchy-Schwartz inequality applied to the vector  $\mathbf{s} = [w_1/\sqrt{v_1}, \dots, w_{m+1}/\sqrt{v_{m+1}}]^\top$  and  $\mathbf{s} = [\sqrt{v_1}, \dots, \sqrt{v_{m+1}}]^\top$ .

Since objective function of the optimization problem (56) is strongly convex and the Slater condition holds, it follows that the optimum solution is the unique Karush-Kuhn-Tucker point for the problem. Lagrangian function for the optimization problem (56) is given by

$$L(\mathbf{v}, \beta) = \boldsymbol{\gamma}^\top \mathbf{v} + \mu_v d_v(\mathbf{v}) + \beta \left( \omega_v - \sum_{i=1}^{m+1} v_i \right) + \boldsymbol{\rho}^\top \mathbf{v},$$

where  $\beta$  and  $\boldsymbol{\rho}$  are the penalty multipliers. Setting the gradient of the Lagrangian function to zero, we get

$$\mathbf{0} = \nabla_{\mathbf{v}} L(\mathbf{v}, \boldsymbol{\rho}, \beta) = \begin{bmatrix} \gamma_i + \mu_v(1 + \ln(v_i^*)) - \beta \\ \dots \\ \gamma_{m+1} + \mu_v(1 + \ln(v_{m+1}^*)) - \beta \end{bmatrix} \Leftrightarrow v_i^* = e^{-(\gamma_i/\mu_v)} e^{\beta + \rho_i}, \quad i = 1, \dots, m+1.$$

Since  $v_i^* > 0$  for all choices of  $\beta$  and  $\boldsymbol{\rho}$ , the complementary slackness condition  $\rho_i v_i^* = 0$  implies that  $\rho_i = 0$ . Thus,  $v_i^* = e^{-\gamma_i/\mu_v} e^\beta$ ,  $i = 1, \dots, m+1$ . Since  $\sum_{i=1}^{m+1} v_i^* = \omega_v$ , it follows that

$$e^\beta = \frac{\omega_v}{\sum_{i=1}^{m+1} e^{-\gamma_i/\mu_v}}.$$

and the optimal  $v_i^* = \omega_v e^{-\gamma_i/\mu_v} / \sum_{k=1}^{m+1} e^{-\gamma_k/\mu_v}$ ,  $i = 1, \dots, m+1$ .

By setting  $\boldsymbol{\gamma} = \mathbf{0}$ , we see that  $\bar{\mathbf{v}} = \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} d_v(v) = \frac{\omega_v}{m+1} \mathbf{1}$ , and  $d_v(\bar{\mathbf{v}}) = 0$ . Thus, it follows that  $d_v(\mathbf{v}) \geq d(\bar{\mathbf{v}}) = 0$  on  $\mathcal{V}$ .

Since  $d_v(\mathbf{v})$  is a convex function, it follows that the optimal value of  $\max_{\mathbf{v} \in \mathcal{V}} d_v(\mathbf{v})$  is achieved at an extreme point of  $\mathcal{V}$ . The extreme points of  $\mathcal{V}$  are given by  $\omega_v \mathbf{e}_i$ ,  $i = 1, \dots, m+1$ , and at any of these point  $d_v(\omega_v \mathbf{e}_i) = \omega_v \ln(m+1)$ . Thus,  $D_v = \omega_v \ln(m+1)$ .  $\square$

## C.2 The primal prox function

Recall that the primal prox function in (41) is

$$d_x(\mathbf{X}) = \sum_{i=1}^n \lambda_i(\mathbf{X}) \ln(\lambda_i(\mathbf{X})) + s_x \ln(s_x) - \omega_x \ln(\omega_x/(n+1)),$$

for  $\mathbf{X} \in \mathcal{X} = \{\mathbf{X} \succeq 0 : \operatorname{Tr}(\mathbf{X}) \leq \omega_x\}$ , and  $s_x = \omega_x - \operatorname{Tr}(\mathbf{X})$ . In order to keep the notation simple, we will work with the matrix

$$\widehat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} & \mathbf{0}^\top \\ \mathbf{0} & s_x \end{bmatrix}.$$

In terms of the new variables the prox-function  $d_x(\widehat{\mathbf{X}}) = \sum_{i=1}^{n+1} \lambda_i(\widehat{\mathbf{X}}) \ln(\lambda_i(\widehat{\mathbf{X}})) - \omega_x \log(\omega_x/(n+1))$ . The prox-function  $d_x$  is simply the dual prox-function  $d_v$  evaluated on the eigenvalues of  $\widehat{\mathbf{X}}$ .

**Lemma 3.** Let  $d_x(\mathbf{X}) = \sum_{i=1}^{n+1} \lambda_i(\mathbf{X}) \ln(\lambda_i(\mathbf{X})) - \omega_x \ln(\omega_x/(n+1))$  where  $\mathbf{X} \in \mathcal{X} = \{\mathbf{X} \succeq 0 : \operatorname{Tr}(\mathbf{X}) = \omega_x\}$ .

1.  $d_x$  is strongly convex with convexity parameter  $\sigma_v = \frac{1}{\omega_x}$  with respect to the norm  $\|\mathbf{X}\|_1 = \sum_{i=1}^n |\lambda_i(\mathbf{X})|$  on the interior of  $\mathcal{X}$ .

2. Let

$$\mathbf{X}^* = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \{\langle \boldsymbol{\Gamma}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X})\} = \frac{\omega_x e^{\frac{1}{\mu} \boldsymbol{\Gamma}}}{\operatorname{Tr}(e^{\frac{1}{\mu} \boldsymbol{\Gamma}})}. \quad (59)$$

Then

3.  $d_x(\cdot) \geq 0$  on  $\mathcal{X}$  and  $D_x = \omega_x \log(n+1)$ .

*Proof.* From results in [3] it follows that  $d_x(\mathbf{X})$  is strongly convex with respect to the  $\ell_1$ -norm,  $\|\mathbf{X}\|_1 = \sum_{i=1}^{n+1} |\lambda_i(\mathbf{X})|$ .

Let  $\boldsymbol{\lambda} \in \mathbb{R}^{n+1}$  such that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n+1} \geq 0$  and  $\sum_{i=1}^{n+1} \lambda_i = 1$  denote the ordered eigenvalues of a matrix  $\mathbf{X} \in \mathcal{X}$ . Note that the value of the function  $d_x(\mathbf{X})$  is completely determined by the eigenvalues of  $\mathbf{X}$ . Thus, the eigenvectors of the optimal  $\mathbf{X}$  are completely determined by the matrix  $\boldsymbol{\Gamma}$ .

Let  $\boldsymbol{\Gamma} = \mathbf{U}^\top \operatorname{diag}(\boldsymbol{\gamma}) \mathbf{U}$  denote the eigendecomposition of the matrix  $\boldsymbol{\Gamma}$ , where we  $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_{n+1}$ . Then

$$\langle \boldsymbol{\Gamma}, \mathbf{X} \rangle = \sum_{i=1}^{n+1} \gamma_i \mathbf{u}_i^\top \mathbf{X} \mathbf{u}_i \leq \sum_{i=1}^{n+1} \gamma_i \lambda_i$$

where equality holds only if  $\mathbf{u}_i$ ,  $i = 1, \dots, k$ , are the eigenvectors corresponding to the  $k$ -largest eigenvalues of  $\mathbf{X}$ . Thus, it follows that for a fixed  $\boldsymbol{\lambda}$  the optimal set of eigenvectors for the matrix  $\mathbf{X}$  is given by the eigenvectors of  $\boldsymbol{\Gamma}$ .

Now, our problem reduces to computing the optimal set of eigenvalues  $\boldsymbol{\lambda}$ . From Lemma 2 it follows that the optimal  $\boldsymbol{\lambda}^*$  is given by

$$\lambda_i^* = \frac{e^{\gamma_i/\mu_x}}{\sum_{k=1}^{n+1} e^{\gamma_k/\mu_x}}, \quad i = 1, \dots, n+1.$$

Thus,

$$\mathbf{X}^* = \frac{\mathbf{U}^\top \text{diag}([e^{\gamma_1/\mu_x}, \dots, e^{\gamma_{n+1}/\mu_x}])\mathbf{U}}{\text{Tr}(\mathbf{U}^\top \text{diag}([e^{\gamma_1/\mu_x}, \dots, e^{\gamma_{n+1}/\mu_x}])\mathbf{U})} = \frac{e^{\frac{1}{\mu_x}\boldsymbol{\Gamma}}}{\text{Tr}(e^{\frac{1}{\mu_x}\boldsymbol{\Gamma}})}.$$

Setting  $\boldsymbol{\Gamma} = \mathbf{0}$ , it follows that  $\text{argmin}_{\mathbf{X} \in \mathcal{X}} d_x(\mathbf{X}) = \frac{\omega_x}{n+1}\mathbf{I}$ . Consequently, for all  $\mathbf{X} \in \mathcal{X}$  we have that  $d_x(\mathbf{X}) \geq d_x(\omega_x/(n+1)\mathbf{I}) = 0$ .

Since  $d_x(\mathbf{X})$  is a convex function, it follows that the optimal value of  $\max_{\mathbf{X} \in \mathcal{X}} d_x(\mathbf{X})$  is achieved at an extreme point of  $\mathcal{X}$ . The extreme points of  $\mathcal{X}$  are given by  $\omega_x \mathbf{u}\mathbf{u}^\top$ , where  $\mathbf{u} \in \mathbb{R}^{n+1}$  with  $\|\mathbf{u}\| = 1$ , and at any of these point  $d_x(\omega_x \mathbf{u}\mathbf{u}^\top) = \omega_x \ln(n+1)$ . Thus,  $D_x = \omega_x \ln(n+1)$ .  $\square$

### C.3 The sparse PCA dual prox function

Here, we show the details for the sparse PCA dual prox function. Recall that, in this case,  $\mu_y \geq 1$ .

**Lemma 4.** *Let*

$$d_y(v, \mathbf{Y}) = \frac{1}{2}|v|^2 + \frac{1}{2} \sum_{i,j} |Y_{ij}|^2$$

and  $\mathcal{Y} = \{(v, \mathbf{Y}) : 0 \leq v \leq 1, |Y_{ij}| \leq v\}$ . Then,

1.  $d_y$  is strongly convex with  $\sigma_y = 1$ .
2. For a given matrix  $\mathbf{X} \in \mathcal{S}_n$ , and scalars  $\ell \in \mathbb{R}$  and  $\mu_y > 0$  let

$$(v^*, \mathbf{Y}^*) = \arg \min_{(v, \mathbf{Y}) \in \mathcal{Y}} \{ \langle \mathbf{X}, \mathbf{Y} \rangle + \ell v + \mu_y d_y(v, \mathbf{Y}) \}.$$

Then,

$$v^* = \begin{cases} 1, & -\ell + \sum_{i,j} \max\{0, |X_{ij}| - \mu_y\} \geq \mu_y, \\ \alpha_t, & \text{if there is some } t \text{ with } \beta_{t+1} \leq \mu\alpha_t \leq \beta_t \leq 1 \\ 0 & \text{otherwise, and} \end{cases} \quad (60)$$

$$Y_{ij}^* \leftarrow -\text{sgn}(X_{ij}) \cdot \min\left\{\frac{|X_{ij}|}{\mu_y}, v^*\right\},$$

where  $(\beta_t)$  is  $\text{Vec}(|\mathbf{X}|)^\top$  sorted in descending order with  $\beta_{n^2+1} = 0$  and  $\beta_0 = 1$  and, for  $t = 0, \dots, n^2+1$ ,

$$\alpha_t = \frac{1}{(t+1)\mu_y} \left(-\ell + \sum_{i=1}^t \beta_i\right).$$

Moreover,  $-\ell + \sum_{i,j} \max\{0, |X_{ij}| - \mu_y\} \geq \mu_y$ ,  $\sum_{i,j} |X_{ij}| \leq \ell$  or, for some  $t$ , either  $\beta_{t+1} \leq \alpha_t \leq \beta_t$ . Also,

$(v^*, \mathbf{Y}^*)$  can be computed in  $\mathcal{O}(n^2 \log n)$  time.

*Proof.* The strong convexity of  $d_y$  with  $\sigma_y = 1$  follows immediately from the fact that  $\nabla^2 d_y(v, \mathbf{Y}) = \mathbf{I}_{n^2}$ .

For a given  $\mathbf{X}$  and  $\ell$ ,  $\arg \min_{(v, \mathbf{Y}) \in \mathcal{Y}} \{ \langle \mathbf{X}, \mathbf{Y} \rangle + \ell v + \mu_y d_y(v, \mathbf{Y}) \} =$

$$\arg \min \left\{ \ell v + \sum_{i,j} X_{ij} Y_{ij} + \frac{\mu_y}{2} (v^2 + \sum_{i,j} Y_{ij}^2) : 0 \leq v \leq 1, -v \leq Y_{ij} \leq v \right\}. \quad (61)$$

Then, the Lagrangian function is  $L(v, \mathbf{Y}, p, q, \mathbf{r}, \mathbf{s}) = \ell v + \sum_{i,j} X_{ij} Y_{ij} + \frac{\mu_y}{2} (v^2 + \sum_{i,j} Y_{ij}^2) + p(v-1) - qv + \sum_{i,j} (r_{ij}(Y_{ij}-v) - s_{ij}(Y_{ij}+v))$ . Setting the gradient of the Lagrangian function to zero, we get

$$0 = \nabla_{Y_{ij}} L(v, \mathbf{Y}) = X_{ij} + \mu_y Y_{ij} + r_{ij} - s_{ij}, \quad 0 = \nabla_v L(v, \mathbf{Y}) = \ell + \mu_y v + p - q - \sum_{i,j} (r_{ij} + s_{ij}) \quad (62)$$

for multipliers  $p^*, q^*, r_{ij}^*, s_{ij}^* \geq 0$  satisfying  $p^*(1 - v^*) = q^*v^* = r_{ij}^*(v^* - Y_{ij}^*) = s_{ij}^*(Y_{ij}^* + v^*) = 0$ .

Using (60) and (62) to calculate  $(v^*, \mathbf{Y}^*)$ , we set

$$(r_{ij}^*, s_{ij}^*) = \begin{cases} (|X_{ij}| - \mu v^*, 0), & X_{ij} \geq \mu v^*, \\ (0, |X_{ij}| - \mu v^*), & X_{ij} \leq -\mu v^*, \\ (0, 0) & \text{otherwise.} \end{cases}$$

and

$$p^* = \begin{cases} -\ell + \sum_{ij} \max\{0, |X_{ij}| - \mu y\}, & v^* = 1, \\ 0, & \text{otherwise,} \end{cases} \quad q^* = \begin{cases} \ell - \sum_{i,j} (r_{ij}^* + s_{ij}^*), & v^* = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (63)$$

Note that

$$r_{ij}^* + s_{ij}^* = \max\{0, |X_{ij}| - \mu y v^*\} \quad \text{and} \quad r_{ij}^* - s_{ij}^* = \text{sgn}(X_{ij}) \max\{0, |X_{ij}| - \mu y v^*\}. \quad (64)$$

We now show that  $(v^*, \mathbf{Y}^*, \mathbf{r}^*, \mathbf{s}^*, p^*, q^*)$  satisfy the KKT conditions. From (60),  $(v^*, \mathbf{Y}^*) \in \mathcal{Y}$  since  $0 \leq v^* \leq 1$  and  $|Y_{ij}^*| \leq v^*$ . Similarly,  $(r_{ij}^*, s_{ij}^*) \geq 0$  by definition. For any  $(i, j)$ ,  $Y_{ij}^* = -\text{sgn}(X_{ij}) \min\{\frac{|X_{ij}|}{\mu y}, v^*\}$ , so by (64),

$$X_{ij} + \mu y Y_{ij}^* + r_{ij}^* - s_{ij}^* = 0.$$

Note that  $p^*$  and  $q^*$  similarly satisfy complementary slackness by definition. To satisfy (62) as well, note that, by (64),  $v^*, p^*, q^*$  must satisfy

$$v^* = \frac{1}{\mu y} (-\ell - p^* + q^* + \sum_{ij} \max\{0, |X_{ij}| - \mu y v^*\}).$$

If  $-\ell + \sum_{ij} \max\{0, |X_{ij}| - \mu y\} \geq \mu y$  then  $q^* = 0$  and  $v^* = \frac{1}{\mu y} (-\ell - p^* + \sum_{ij} \max\{0, |X_{ij}| - \mu y\})$ . Thus, for this case,  $q^*$  and  $p^*$  are both set correctly by (63) and  $v^*$  and  $Y_{ij}^*$  are set correctly by (60). If  $\sum_{ij} |X_{ij}| \leq \ell$  then  $p^* = 0$  and

$$v^* = \frac{1}{\mu y} (-\ell + q^* + \sum_{ij} \max\{0, |X_{ij}| - \mu y v^*\}) \leq q^*.$$

Also,  $q^* > 0$  only if  $v^* = 0$ . In this case, both  $p^*$  and  $q^*$  are set correctly by (63) and  $v^*$  and  $Y_{ij}^*$  are set correctly by (60)

So assume  $-\ell + \sum_{ij} \max\{0, |X_{ij}| - \mu y\} < \mu y$  and  $\sum_{ij} |X_{ij}| > \ell$ . Then, since  $p^* = q^* = 0$ ,

$$v^* = \frac{1}{(t^* + 1)\mu y} (-1 + \sum_{\{(i,j): |X_{ij}| > \mu y v^*\}} |X_{ij}|),$$

for some  $t^*$  such that  $1 \leq t^* \leq |\{(i, j) : |X_{ij}| > \mu y v^*\}|$ . However, we must show that  $t^*$  exists so that

$$1 \geq v^* \geq 0 \quad (65)$$

Let  $\beta_t$  be as defined in the Lemma statement so that  $\sum_{ij} |X_{ij}| = \sum_{t=1}^{n^2} \beta_t$  and assume, without loss of generality, that  $\beta_t > 0$  for all  $t$ . We show recursively that this implies  $\alpha_t > \beta_t$  until some  $t^*$  with  $\alpha_{t^*} \leq \beta_{t^*}$  whereupon  $\alpha_{t^*} \geq \beta_{t^*+1}$  is also satisfied. Since  $0 \leq \beta_t \leq 1$  for all  $t$ , our claim ensures (65) is satisfied *so long as a  $t$  exists with  $\alpha_t \leq \beta_t$*  which we show below. For  $t = n^2$ , note that  $-\ell + \sum_{i=1}^{n^2} \beta_i > 0$  implies

$$\alpha_{n^2} = \frac{1}{\mu y(n^2 + 1)} (-\ell + \sum_{i=1}^{n^2} \beta_i) > 0 = \beta_{n^2+1}.$$

So either  $t^* = n^2$  or  $\alpha_{n^2} > \beta_{n^2}$ . Now assume that  $\alpha_t > \beta_t$  for some  $n^2 + 1 \geq t \geq 1$  and suppose, for the sake of a contradiction, that  $\alpha_{t-1} < \beta_t \leq \beta_{t-1}$ . Then, since  $\mu_y > 1$ ,

$$\begin{aligned} \frac{\mu_y t}{\mu_y(t+1)} \beta_t &> \frac{\mu_y t}{\mu_y(t+1)} \alpha_{t-1} \\ &= \frac{1}{\mu_y(t+1)} \left( -\ell + \sum_{i=1}^{t-1} \beta_i \right) \end{aligned} \tag{66}$$

$$\begin{aligned} &= \alpha_t - \frac{1}{\mu_y(t+1)} \beta_t \\ &> \left( 1 - \frac{1}{\mu_y(t+1)} \right) \beta_t \\ &= \frac{\mu_y(t+1) - 1}{\mu_y(t+1)} \beta_t, \end{aligned} \tag{67}$$

where (66) follows from the definition of  $\alpha_t$  and (67) follows since  $\alpha_t > \beta_t$ . So, since  $\beta_t > 0$  for  $t = 1, \dots, n^2$ ,

$$\frac{\mu_y t}{\mu_y(t+1)} > \frac{\mu_y(t+1) - 1}{\mu_y(t+1)} = \frac{\mu_y t + \mu_y - 1}{\mu_y(t+1)},$$

a contradiction, since  $\mu_y > 1$ . All that remains is to show that there has to be one  $t$  where  $\alpha_t \leq \beta_t$ . However, this follows immediately from the assumption that  $\alpha_0 = -\frac{1}{\mu_y} \ell < 1 = \beta_0$ . □