Approximating semidefinite packing programs [*]

G. Iyengar [†] D. J. Phillips [‡] 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

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

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: $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle$, where $\mathbf{A} \succeq \mathbf{0}$, $g(\mathbf{X}) = \big(\sum_{i=1}^{k}(\langle \mathbf{A}_i, \mathbf{X}\rangle)^2\big)^{\frac{1}{2}}$, and $g(\mathbf{X}) = \sum_{i,j=1}^{n} |X_{ij}|$. The *Frobenius inner product* $\langle \mathbf{A}, \mathbf{B} \rangle$ between symmetric matrices $\mathbf{A}$, and $\mathbf{B}$ is defined as

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

The constraint $\mathbf{X} \succeq \mathbf{0}$ indicates that the matrix $\mathbf{X}$ is symmetric and *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 techniques for dimension reduction. See Section 2.1 for a detailed discussion of optimization problems that can be reformulated as packing SDPs. The term packing SDP is derived from the fact that (1) is a *packing problem*, as defined in [24]. We believe the first published reference to an SDP in the context of packing was by Klein and Lu [17] in reference to the Maxcut and coloring SDPs.

Our contributions in this paper are as follows.

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

[†] The Department of Industrial Engineering & Operations Research, Columbia University`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`phillips. Supported in part by NSF grants CCF-0728733, DMS-0703532, DGE-0086390 and DMI-9970063.

[§] The Department of Industrial Engineering & Operations Research, Columbia University`cliff@ieor.columbia.edu`. Supported in part by NSF grants CCF-0728733 and DMI-9970063.

(a) In Section 2 we define the class of packing SDPs, and in Section 2.1 we show that SDPs arising in many important optimization problems can be reformulated as packing SDPs. Using our algorithm we are able to solve all the packing SDPs in a unified manner.

(b) We propose a new technique for solving a packing SDP to an absolute error ϵ. Our solution approach relies on Lagrangian relaxation. We dualize the hard packing constraints to construct a relaxation where the primal feasible set is defined as $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq \omega_x\}$. In Section 3, we show how to recover an ϵ-optimal *feasible* solution from the optimal solution of the Lagrangian relaxation of the packing SDP. Unlike usual Lagrangian approaches which are only able to compute a bound for the optimal value, we produce a feasible solution. The results in this section apply to all packing SDPs.

In Section 4 we consider the problem of computing an optimal solution of a Lagrangian relaxation for (1). We show that the resulting nonlinear Lagrangian objective function, which has form $\langle \mathbf{C}, \mathbf{X} \rangle - \sum_{i=1}^{m} v_i(g_i(\mathbf{X}) - 1)$, can be linearized if we restrict the packing functions to the form

$$g(\mathbf{X}) = \max\Big\{ \sum_{i=1}^{k} z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \mathbf{P}\mathbf{z} \leq \mathbf{d}, \mathbf{z} \geq \mathbf{0} \Big\}, \tag{2}$$

where the symmetric matrices $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, matrix $\mathbf{P} \in \mathbb{R}^{\ell \times k}$, and vectors $\mathbf{d} \in \mathbb{R}^{\ell}$ are such that $g(\mathbf{X}) \geq 0$ for all $\mathbf{X} \succeq \mathbf{0}$. The packing functions arising in the examples discussed in Section 2.1 are all of this form. Note also that this set of functions is much larger than just functions of the form $g(X) = \max_{1 \leq i \leq m} \langle \mathbf{A}_i, \mathbf{X} \rangle$. We show that Nesterov's first-order procedure [21] can be used to efficiently compute a feasible, ϵ-optimal solution for the Lagrangian relaxation of the packing SDP where all functions are of the form (2). Our algorithm 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 based on 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 describe the complexity results for the specific instances discussed in Section 2.1. We show that an ϵ-optimal solution to the SDP relaxation to the MAXCUT problem can be computed in $\mathcal{O}\big(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})\big)$ time, where r denotes the number of non-zero elements in the Laplacian matrix of the graph. The previous best known 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 [29] 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 ϵ the Klein-Lu and Arora-Kale-Trevisan bounds are superior, but as ϵ 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 ϵ-optimal solution to the semidefinite relaxation for the graph coloring problem can be computed in $\mathcal{O}\big(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})\big)$ 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 ϵ-approximations of the Lovász-ϑ function [18] and Szegedy number [28] in $\mathcal{O}(\big(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})\big)$ (i.e., the same as MAXCUT and coloring) compared to the $\mathcal{O}(n^{.5}(n^3 + r^3) \cdot \log(\epsilon^{-1}))$ runtime of the barrier method [23]. Thus, our tradeoff works in an opposite direction. For moderate ϵ our bound is better for all graphs, and for dense graphs and moderate ϵ our bound is much better. Chan et al. [5] extend the results of [1] to compute the Lovász-ϑ function in $\mathcal{O}(n^5 \cdot \epsilon^{-2})$. Their method does compute a feasible solution to the SDP for the Lovász-ϑ function SDP, while our method does not.

We also show our methods compute an ϵ-optimal solution to the sparse PCA problem in $\mathcal{O}\big(n^4 \sqrt{\log(n)} \cdot \epsilon^{-1}\big)$ which matches the best known previous result for this problem [7]. Unlike the method in [7], our method always returns a feasible solution.

(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 6000×6000) .

In [22] Nesterov describes how to extend the smoothing technique that he proposed in [21] to minimizing the maximal eigenvalue and the spectral radius of symmetric matrices. Nesterov establishes that one can efficiently compute an ϵ-optimal solution to the non-smooth semidefinite optimization problem by solving a sequence of penalized gradient descent problems where the step is penalized by an appropriately chosen smooth, strongly convex function. In the method proposed in [21, 22], the penalized gradient descent step has to be solved over the feasible set of the original non-smooth problem. This restriction limits one to non-smooth problems where the constraint set is "simple" [21, 22]. Note that computing a penalized gradient step over the feasible set of the packing SDP, as would be required by the method proposed in [22], is, in fact, as hard as solving the packing SDP. Thus, the method proposed in [21, 22] cannot be directly used to solve packing SDPs. A main contribution of our paper is that we show one can dualize the packing constraints and compute the smoothed gradient step for a large class of packing SDPs over the "simple" set $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq \omega_x\}$ and still converge to an ϵ-optimal *feasible* solution to the packing SDP in $\mathcal{O}(\epsilon^{-1})$ operations.

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, $\mathbf{J}_n$. When the dimension is clear, we define $\mathbf{0}$ as the matrix of all zeros and for all i, we define $\mathbf{e}_i$ as the ith column of the identity matrix. We use $\mathcal{S}^n$ to denote the set of symmetric $n \times n$ matrices and $\mathcal{S}^n_+$ to denote the cone of symmetric, 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. $\mathbf{A} \succeq \mathbf{0}$ indicates that the matrix $\mathbf{A}$ is symmetric and positive semidefinite, and $\mathbf{A} \succeq \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B} \succeq \mathbf{0}$.

For a given vector $\mathbf{v} \in \mathbb{R}^n$, we let $\|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|$, $\|\mathbf{v}\|_\infty = \max_{i=1,\ldots,n} |v_i|$, and $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2}$ denote the ℓ_1, ℓ_∞ and ℓ_2 norms, respectively. We define the $\mathcal{L}_1, \mathcal{L}_\infty$, and $\mathcal{L}_2$ norms for a symmetric matrix $\mathbf{A}$ as follows. Let

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

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

We call a differentiable, convex function f *strongly* convex with convexity parameter σ 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 if $\nabla^2 f$ exists, for all $\mathbf{x}$ and $\mathbf{z}$, $\mathbf{z}^\top \nabla^2 f(\mathbf{x})\mathbf{z} \geq \sigma \|\mathbf{z}\|^2$. Note that the value of the convexity parameter σ depends on the particular norm $\|\cdot\|$.

For $\mathcal{Z} \subseteq \mathbb{R}^n$, we say that $\bar{\mathbf{z}}_a \in \mathcal{Z}$ is ϵ-optimal in the *absolute* sense for the optimization problem $\max_{\mathbf{z} \in \mathcal{Z}}\{f(\mathbf{z})\}$ if $f(\bar{\mathbf{z}}_a)$ is within an *additive* error ϵ to the optimal value, i.e., if the inequality $f(\bar{\mathbf{z}}_a) \geq f^* - \epsilon$ is satisfied, where we define $f^* = \max_{\mathbf{z} \in \mathcal{Z}}\{f(\mathbf{z})\}$. We say that $\bar{\mathbf{z}}_r \in \mathcal{Z}$ is ϵ-optimal in the *relative* sense if $f(\bar{\mathbf{z}}_r) \geq (1-\epsilon)f^*$, 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 $\bar{\mathbf{z}}_a$ is ϵ-optimal in the absolute sense. Then $\bar{\mathbf{z}}_a$ is ϵ/C optimal in the relative sense since the definitions imply that $f(\bar{\mathbf{z}}_a) \geq f^* - \epsilon = f^* - C(\epsilon/C) \geq (1 - \epsilon/C)f^*$.

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 Packing SDP

We begin by formally defining the packing SDP. Next, we show that many important optimization problems arising in combinatorial optimization, principal component analysis and maximum variance unfolding can be reformulated as packing SDPs.

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

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

Packing functions are similar to *gauge* functions (see page 28 of [25] and [8]) – note that unlike gauge functions we only require non-negativity and positive homogeneity on $\mathcal{S}^n_+$, e.g., $g(\mathbf{X}) = \mathbf{Tr}(\mathbf{X})$ is a packing function but not a gauge function. Also, *Minkowski* functions of convex subsets of $\mathcal{S}^n$ are packing functions but not necessarily gauge functions. Examples of packing functions include:

1. $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle, \mathbf{A} \succeq \mathbf{0}$.
2. $g(\mathbf{X}) = \sum_{i,j} |X_{ij}| = \max\big\{ \langle \mathbf{X}, \mathbf{Z} \rangle : |Z_{ij}| \leq 1, \forall i, j \big\}$.
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 previous three packing functions are all special cases of the packing function

   $$g(\mathbf{X}) = \max\big\{ \sum_{i=1}^k z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \mathbf{z} \in \mathcal{P} \subseteq \mathbb{R}^n_+ \big\}, \tag{3}$$

   where the $\mathbf{A}_i \in \mathcal{S}^n$, for $i = 1, \ldots, n$ and convex $\mathcal{P}$ are such that $g(\mathbf{X}) \geq 0$ for all $\mathbf{X} \succeq \mathbf{0}$.
4. $g(\mathbf{X}) = \|\boldsymbol{\lambda}(\mathbf{X})\|$ where $\boldsymbol{\lambda}(\mathbf{X})$ denotes the vector of eigenvalues of $\mathbf{X}$ and $\|\cdot\|$ is any vector norm.

The positive homogeneity condition (see 2. in Definition 1) is 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 $b \in \mathbb{R} - \{0\}$ since it violates positive homogeneity. General symmetric functions of eigenvalues are *not* packing functions, e.g. $g(\mathbf{X}) = \sum_i \lambda_i(\mathbf{X}) \ln(\lambda_i(\mathbf{X}))$ is *not* a packing function [22].

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

$$\begin{array}{rll} \rho^* = & \max & \langle \mathbf{C}, \mathbf{X} \rangle \\ & \text{s.t.} & g_i(\mathbf{X}) \leq 1, \quad i = 1, \ldots, m, \\ & & \mathbf{Tr}(\mathbf{X}) \leq \omega_x, \\ & & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{4}$$

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

The trace constraint $\mathbf{Tr}(\mathbf{X}) \leq \omega_x$ is equivalent to assuming the feasible region of the packing SDP (4) is compact. This is almost always true in problems of practical interest. The results in Section 3 hold for all packing SDPs. In Section 4 we restrict ourselves to packing functions of the form (2), i.e., we have

$$g(\mathbf{X}) = \max\Big\{ \sum_{i=1}^{k} z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \mathbf{P}\mathbf{z} \leq \mathbf{d}, \mathbf{z} \geq \mathbf{0} \Big\}.$$

where $\mathbf{A}_i \in \mathcal{S}^n$, $\mathbf{P} \in \mathbb{R}^{\ell \times k}$ and $\mathbf{d} \in \mathbb{R}^{\ell}$ are such that $g(\mathbf{X}) \geq 0$ for all $\mathbf{X} \succeq \mathbf{0}$. Note that g is a packing function in the form of (3) where the convex sets are polyhedral. All the packing constraints arising in the applications discussed in Section 2.1 belong to this class of packing functions. The results in Section 4 continue to hold for packing functions of the form

$$g(\mathbf{X}) = \max\Big\{ \sum_{i=1}^{k} z_i \langle \mathbf{A}_i, \mathbf{X} \rangle : \mathbf{z} \in \mathcal{Q} \Big\}, \mathcal{Q} \text{ is compact \& convex,} \tag{5}$$

provided there exists a smooth, strongly convex function $d(\mathbf{z})$ such that $\min\{\mathbf{c}^T\mathbf{z} + d(\mathbf{z}) : \mathbf{z} \in \mathcal{Q}\}$ can be efficiently computed. We leave this extension to the reader.

2.1 Instances of packing SDP

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

2.1.1 The Maxcut SDP

The SDP relaxation to the Maxcut problem introduced by Goemans and Williamson [10] is given by

$$\begin{array}{ll} \max & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{e}_i \mathbf{e}_i^{\top}, \mathbf{X} \rangle = 1, \quad i = 1, \ldots, n, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{6}$$

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 symmetric matrix $\mathbf{L} = [L_{ij}]$ where

$$L_{ij} = \begin{cases} -w_{ij}, & i \neq j, \\ \sum_{k=1}^{n} w_{ik}, & i = j. \end{cases} \tag{7}$$

We set $w_{ij} = 0$ when $(i,j) \notin \mathcal{E}$ and $i \neq j$. Then for any $\mathbf{x} \in \mathbb{R}^n$, (7) indicates that $\mathbf{x}^{\top}\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2 \geq 0$, i.e. $\mathbf{L} \succeq \mathbf{0}$. Recall that we assume that $\mathcal{G}$ is connected, which implies that for all $i = 1, \ldots, n$, there exists an index k such that $(i,k) \in \mathcal{E}$ and $w_{ik} > 0$. Then (7) implies that $L_{ii} = \sum_{j=1}^{n} w_{ij} > 0$ for all $i = 1, \ldots, n$.

Let $\mathbf{D}$ be a diagonal matrix with $\mathbf{diag}(\mathbf{L})$ as the main diagonal. Then the change of variables $\mathbf{X} = \big(\mathbf{Tr}(\mathbf{D})\big) \cdot \mathbf{D}^{-\frac{1}{2}}\mathbf{Y}\mathbf{D}^{-\frac{1}{2}}$ implies that (6) is equivalent to

$$\begin{array}{ll} \max & \langle \mathbf{L}_{\mathbf{D}}, \mathbf{Y} \rangle \\ \text{s.t.} & \langle \mathbf{e}_i \mathbf{e}_i^{\top}, \mathbf{Y} \rangle = \frac{D_{ii}}{\mathbf{Tr}\,\mathbf{D}}, \quad i = 1, \ldots, n, \\ & \mathbf{Y} \succeq \mathbf{0}, \end{array} \tag{8}$$

where $\mathbf{L}_{\mathbf{D}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ is the *normalized Laplacian* [26]. We claim that the packing SDP

$$\begin{array}{ll} \max & \langle \mathbf{L}_{\mathbf{D}}, \mathbf{Y} \rangle \\ \text{s.t.} & \frac{\mathbf{Tr}(\mathbf{D})}{D_{ii}} \langle \mathbf{e}_i \mathbf{e}_i^{\top}, \mathbf{Y} \rangle \leq 1, \quad i = 1, \ldots, n, \\ & \mathbf{Tr}(\mathbf{Y}) \leq 1, \\ & \mathbf{Y} \succeq \mathbf{0}, \end{array} \tag{9}$$

is equivalent to (8). Since the constraints, $Y_{ii} \leq \frac{D_{ii}}{\mathbf{Tr}(\mathbf{D})}$, for all i, imply $\mathbf{Tr}(\mathbf{Y}) \leq 1$, the packing SDP (9) and the original SDP formulation (8) are equivalent unless there exists an optimal solution $\mathbf{Y}^*$ to (9) with an index i such that $Y^*_{ii} < \frac{D_{ii}}{\mathbf{Tr}(\mathbf{D})}$. Suppose this is the case and define $\mathbf{Y} = \mathbf{Y}^* + (\frac{D_{ii}}{\mathbf{Tr}(\mathbf{D})} - Y^*_{ii})\mathbf{e}_i\mathbf{e}_i^\top$. By construction, $\mathbf{Y}$ is feasible for (8), and we have

$$\langle \mathbf{L}_\mathbf{D}, \mathbf{Y} \rangle = \langle \mathbf{L}_\mathbf{D}, \mathbf{Y}^* \rangle + (\frac{D_{ii}}{\mathbf{Tr}(\mathbf{D})} - Y^*_{ii}) > \langle \mathbf{L}_\mathbf{D}, \mathbf{Y}^* \rangle ,$$

a contradiction. Thus, it follows that the packing SDP (9) is equivalent to the Maxcut SDP (8).

2.1.2 The Lovász-ϑ function SDP

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

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

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 $\mathbf{F}^{(i,j)} = \mathbf{I} - (\mathbf{e}_i\mathbf{e}_j^\top + \mathbf{e}_j\mathbf{e}_i^\top)$. 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 $\mathbf{Tr}(\mathbf{X}) = 1$, we can rewrite (10) as the packing SDP

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

Note that in reformulating (10) as the packing SDP (11) it was extremely important that we allow trace equality constraints in packing SDPs (see Definition 2).

A related quantity to $\vartheta(\mathcal{G})$ is Szegedy's number [28], defined as

$$\begin{array}{rcll} \vartheta^+(\mathcal{G}) & = & \max & \langle \mathbf{J}, \mathbf{X} \rangle \\ & & \text{s.t.} & x_{ij} \leq 0, \quad (i,j) \in \mathcal{E}, \\ & & & \mathbf{Tr}(\mathbf{X}) = 1, \\ & & & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{12}$$

Gvozdenović and Laurent [11] show that ϑ^+ 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 (12) as the packing SDP

$$\begin{array}{rll} \vartheta^+(\mathcal{G}) = \max & \langle \mathbf{J}, \mathbf{X} \rangle \\ \text{s.t.} & \left\langle \mathbf{E}^{(i,j)}, \mathbf{X} \right\rangle \leq 1, \quad (i,j) \in \mathcal{E}, \\ & \mathbf{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{13}$$

2.1.3 The coloring SDP

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

$$\begin{array}{rl} \max & \zeta \\ \text{s.t.} & x_{ii} = 1, \quad i = 1, \ldots, n, \\ & \zeta \leq -x_{ij}, \quad (i,j) \in \mathcal{E}, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{14}$$

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

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

Then, for all $\mathbf{X}$ feasible to (14), we have that $\left\langle \mathbf{G}^{(i,j)}, \mathbf{X} \right\rangle = 1 - x_{ij}$. Therefore, it follows that

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

From an argument similar to that used to show the equivalence of the Maxcut SDP to a packing SDP, it follows that (14) is equivalent to the max-min problem

$$\begin{array}{ll} \max & \min \left\{ \sum_{(i,j)\in\mathcal{E})} w_{(i,j)} \left\langle \mathbf{G}_{ij}, \mathbf{X} \right\rangle \right\} \\ \text{s.t.} & x_{ii} \le 1, \quad i = 1, \ldots, n, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{15}$$

The optimization problem (15) is *not* a packing SDP. We compute an approximate solution to a packing SDP by using a Lagrangian relaxation, i.e. by converting packing SDP into a max-min problem. In Section 3, we describe this conversion and show that our solution algorithm can be easily adapted to solve a max-min problem of the form (15).

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

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

Here, $\mathbf{C} \in \mathcal{S}_+^n$ is a given covariance matrix, $\mathbf{Card}(\mathbf{x})$ is a function that returns the number of nonzero components of $\mathbf{x}$ and $1 < \kappa < n$ is a given parameter ($\kappa = 1$ is the variable with maximum variance and $\kappa = n$ is ordinary PCA, an eigenvalue problem). Further details about Sparse PCA can be found in d'Aspremont et al. [7], who formulate the following SDP relaxation for the above non-convex optimization problem

$$\begin{array}{ll} \max & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} & \frac{1}{\kappa} \sum_{ij} |X_{ij}| \le 1, \\ & \mathbf{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{16}$$

The optimization problem (16) is a packing SDP. In [7] the authors approximately solve (16) by dualizing the cardinality constraint $\sum_{ij} |X_{ij}| \le \kappa$; however, they do not guarantee that their solution is feasible. Our method computes feasible ϵ-optimal solutions for the packing SDP (16).

2.1.5 Maximum variance unfolding

Maximum Variance Unfolding (MVU) (also called semidefinite embedding) is a technique introduced by Weinberger and Saul [32] for computing low-dimensional representations that preserves distances between "local" points while seeking to maximize the overall distance between all points.

Suppose we are given n data points $\mathcal{D} = \{\mathbf{y}_i : i = 1, \dots, n\} \subseteq \mathbb{R}^\ell$ where the dimension $\ell \gg 1$. Let $\mathcal{E} \subset \{(i,j) : 1 \le i < j \le n\}$ 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 an m-dimensional representation of $\mathcal{D}$ that preserves distances and minimizes the effective dimension of the resulting manifold, where $m \ll \ell$. To formulate as a mathematical program, denote the m-dimensional representation by $\{\mathbf{u}_i : i = 1, \dots, n\} \subset \mathbb{R}^m$. Weinberger and Saul [32] propose constructing such a manifold by solving the optimization problem

$$\begin{array}{ll} \max & \sum\limits_{i,j \in \mathcal{V}} \|\mathbf{u}_i - \mathbf{u}_j\|^2 - \nu \sqrt{\sum\limits_{(i,j)\in\mathcal{E}} (\|\mathbf{u}_i - \mathbf{u}_j\|^2 - d_{ij})^2} \\ \text{s.t.} & \sum\limits_{i=1}^{n} \mathbf{u}_i = \mathbf{0}. \end{array} \tag{17}$$

where $\sum_{i=1}^n \mathbf{u}_i = 0$ is a centering constraint. Since (17) is not a convex optimization problem, Weinberger and Saul [32] approximately solve (17) by constructing a semidefinite programming relaxation.

We present a slightly modified version of the relaxation developed in [32]. Let $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{n-1}]$ and set $\mathbf{K} = \mathbf{U}^\top \mathbf{U}$. For each $(i,j) \in \mathcal{E}$, define $\mathbf{a}_{(i,j)} \in \mathbb{R}^{(n-1)}$ as follows. Let

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

where $\mathbf{e}_k$ denotes the k^{th} column of $\mathbf{I}_{n-1}$. Then, for each $i \ne j$, $\mathbf{u}_i - \mathbf{u}_j = \mathbf{U}\mathbf{a}_{ij}$ 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 (17) is equivalent to

$$\begin{array}{ll} \max & \min\limits_{\|\mathbf{z}\|_2 \le 1} \Big\{ \Big\langle \sum_{i,j\in\mathcal{V}} \mathbf{a}_{ij}\mathbf{a}_{ij}^\top, \mathbf{K} \Big\rangle - \sum\limits_{(i,j)\in\mathcal{E}} z_{ij} \big(\langle \mathbf{a}_{ij}\mathbf{a}_{ij}^\top, \mathbf{K} \rangle - d_{ij} \big) \Big\}, \\ \text{s.t.} & \text{rank}(\mathbf{K}) = m, \\ & \mathbf{Tr}(\mathbf{K}) \le \tau, \\ & \mathbf{K} \succeq \mathbf{0}, \end{array} \tag{18}$$

where $\tau = \sum_{i=1}^n \|\mathbf{y}_i\|$. The semidefinite relaxation is obtained by relaxing the rank constraint on $\mathbf{K}$ as in (16). The optimization problem (18) is a max-min problem which has the same structure as the Lagrangian relaxation we use in our solution algorithm for packing SDPs (see Section 3).

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 ℓ-dimensional vectors $\{\mathbf{y}_i : 1 \le i \le n\}$ computed by any data mining technique. Xiao et al. [33] 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. [33] 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

$$\begin{array}{ll} \max & \langle \mathbf{V}\mathbf{V}^\top, \mathbf{Q} \rangle \\ \text{s.t.} & \langle \mathbf{V}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{V}^\top, \mathbf{Q} \rangle \le d_{ij}, \quad (i,j) \in \mathcal{E} \\ & \mathbf{Tr}(\mathbf{Q}) \le \tau, \\ & \mathbf{Q} \succeq \mathbf{0}, \end{array} \tag{19}$$

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 ϵ-approximate solution for the packing SDP (4) using Lagrangian penalization on the packing constraints. Penalizing the packing constraints 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. Since the dual sets we use are bounded, the penalization results in a relaxation of the original packing SDP and in this section we show how to convert approximate solutions to the relaxation into approximate solutions to the corresponding packing SDP. In Section 4, we show the Lagrangian relaxation can be efficiently solved for the class of packing functions defined in (2).

Define the Lagrangian function $\phi : \mathcal{S}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ of the packing SDP (4) as follows. For $(\mathbf{X}, \mathbf{v}) \in \mathcal{S}^n \times \mathbb{R}^m$, let

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

Consider computing a saddle-point (i.e., an exact solution) to the maximin problem

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}), \tag{20}$$

where we need to specify the sets $\mathcal{X}$ and $\mathcal{V}$. Define

$$\mathcal{X} = \left\{ \mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq \omega_x \right\}. \tag{21}$$

Recall that we assume either the packing SDP (4) has the trace constraint $\mathbf{Tr}(\mathbf{X}) \leq \omega_x$ or such a bound is implied by the packing constraints. When the trace constraint in the packing SDP is an equality constraint we set $\mathbf{Tr}(\mathbf{X}) = \omega_x$ in (21). If we let $\mathcal{V} = \mathbb{R}^m_+$ then (20) would be the Lagrangian dual for (4). However, we require a compact, i.e., bounded set for $\mathcal{V}$. Thus, let

$$\mathcal{V} = \Big\{ \mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_{j=0}^{m} v_j \leq \omega_v \Big\}, \tag{22}$$

where

$$\omega_v = \max\{\langle \mathbf{C}, \mathbf{X} \rangle : \mathbf{X} \in \mathcal{X}\} \leq \omega_x \, \mathbf{Tr}(\mathbf{C}). \tag{23}$$

We need the "diameter" ω_v of the dual set to be large enough to ensure that infeasible solutions to (4) are sufficiently penalized. The proof of Theorem 1 demonstrates that the bound ω_v in (23) is sufficiently large.

Let $\bar{\mathbf{v}} \in \mathcal{V}$. Then for all $\mathbf{X}$ feasible to the packing SDP (4), $\phi(\mathbf{X}, \bar{\mathbf{v}}) \geq \langle \mathbf{C}, \mathbf{X} \rangle$. Thus, we have

$$\max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) \geq \max_{\mathbf{X} \in \mathcal{X}} \langle \mathbf{C}, \mathbf{X} \rangle \geq \rho^*, \tag{24}$$

where the last inequality follows since the feasible region of the packing SDP (4), which is $\{\mathbf{X} : g_i(\mathbf{X}) \leq 1, i = 1, \ldots, m, \mathbf{Tr}(\mathbf{X}) \leq \omega_x, \mathbf{X} \succeq \mathbf{0}\}$, is a subset of $\mathcal{X}$. Thus, it follows that

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) = \min_{\mathbf{v} \in \mathcal{V}} \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{v}) \geq \rho^*, \tag{25}$$

where the equality follows from an appropriate saddle-point theorem applied to the function $\phi(\mathbf{X}, \mathbf{v})$ and the inequality follows from (24). We refer to the max-min problem in (25) 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 (20) if the pair satisfies

$$0 \leq \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) - \min_{\mathbf{v} \in \mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v}) \leq \epsilon. \tag{26}$$

The main result in this section establishes that one can compute an ϵ-optimal solution of the packing SDP (4) by appropriately scaling the ϵ-saddle-point $\bar{\mathbf{X}}$. We defer the problem of computing the ϵ-saddle-point $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$ to Section 4.

Theorem 1. *Fix $\epsilon > 0$. Let $(\overline{\mathbf{X}}, \overline{\mathbf{v}})$ denote any ϵ-saddle-point to (20) with ω_v defined as in (23). Define $\overline{d} = \max_{i=1,\ldots,m} \{g_i(\overline{\mathbf{X}})\}$ as the maximum value of the packing constraints. Then*

$$\widehat{\mathbf{X}} = \begin{cases} (1/\bar{d})\overline{\mathbf{X}}, & \overline{d} > 1, \\ \overline{\mathbf{X}}, & \textit{otherwise}, \end{cases} \tag{27}$$

is an ϵ-optimal solution for the packing SDP (4).

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

$$1 \geq \Big(\frac{1}{\overline{d}}\Big) g_i(\overline{\mathbf{X}}) = g_i\Big(\frac{1}{\overline{d}} \cdot \overline{\mathbf{X}}\Big) = g_i(\widehat{\mathbf{X}}).$$

Thus, $\widehat{\mathbf{X}}$ is always feasible to the packing SDP (4). Next, we show that $\widehat{\mathbf{X}}$ is ϵ-optimal. Consider the following two cases:

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

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

Thus,

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

where the last equality follows from the definition of ϕ. Since $(\overline{\mathbf{X}}, \overline{\mathbf{v}})$ is an ϵ-saddle-point, it follows from (26) and (24) that

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

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

$$\begin{aligned} \Big\langle \mathbf{C}, \widehat{\mathbf{X}} \Big\rangle &= \frac{1}{\overline{d}} \langle \mathbf{C}, \overline{\mathbf{X}} \rangle \\ &\geq \langle \mathbf{C}, \overline{\mathbf{X}} \rangle - (\overline{d} - 1) \langle \mathbf{C}, \overline{\mathbf{X}} \rangle \\ &\geq \langle \mathbf{C}, \overline{\mathbf{X}} \rangle - (\overline{d} - 1)\omega_v & (28) \\ &= \langle \mathbf{C}, \overline{\mathbf{X}} \rangle + \min_{\mathbf{v}\in\mathcal{V}} \left\{ \sum_{i=0}^{m} v_i(1 - g_i(\overline{\mathbf{X}})) \right\} & (29) \\ &= \min_{\mathbf{v}\in\mathcal{V}} \phi(\overline{\mathbf{X}}, \mathbf{v}) \\ &\geq \rho^* - \epsilon, & (30) \end{aligned}$$

where (28) follows from (23) since $\overline{\mathbf{X}} \in \mathcal{X}$, (29) follows from the fact that $\omega_v(1 - \bar{d}) = \min_{\mathbf{v}\in\mathcal{V}} \big\{ \sum_{i=0}^{m} v_i(1 - g_i(\mathbf{X}))\big\}$, whenever $\bar{d} > 1$, and (30) follows from (26) and (24).

Thus, we have that $\widehat{\mathbf{X}}$ is a feasible, ϵ-optimal solution to (4). □

Lagrangian relaxations typically yield good bounds but do not yield feasible solutions. Theorem 1 shows that by setting the "diameter" ω_v sufficiently large one can recover a *feasible* ϵ-approximate solution for any packing SDP (4) from an ϵ-saddle-point for (20). In the next section we show that for a restricted class of packing functions one can compute an ϵ-saddle-point efficiently.

Theorem 1 can be used to convert ϵ-saddle-points to ϵ-approximate solutions for the packing SDPs for maximum variance unfolding and Laplacian eigenmaps (Sections 2.1.5 and 2.1.6, respectively). However, 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 is non-decreasing in the main diagonal, so a feasible, ϵ-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 $\mathbf{e}_i$, $i = 1, \ldots, n$. 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 $\mathbf{Y} = \mathbf{X} + \mathbf{I} - \mathbf{D}_\mathbf{X}$ is a feasible ϵ-optimal solution (where $\mathbf{D}_\mathbf{X}$ is a diagonal matrix with $\mathbf{diag}(\mathbf{X})$ along the main diagonal).

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

Note that Lemma 1 can be used for both the Maxcut packing SDP (in conjunction with Theorem 1) and the max-min optimization problem (15) for coloring.

Recall that the packing SDPs for the Lovász ϑ-function, Szegedy's number and Sparse PCA all have a trace equality constraint so Theorem 1 does not apply. We therefore 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 (4) with the trace equality constraint,*

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

Define the dual set parameter

$$\omega_v = \frac{\rho_u - \langle \mathbf{C}, \mathbf{Z} \rangle}{1 - g_{\max}(\mathbf{Z})}, \tag{31}$$

where for $\mathbf{X} \in \mathcal{X}$, we set

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

and ρ_u is any upper bound on ρ^*, in particular, we can set $\rho_u = \langle \mathbf{C}, \mathbf{Z} \rangle$. Suppose that $(\overline{\mathbf{X}}, \overline{\mathbf{v}})$ is an ϵ-optimal saddle-point and assume that $\langle \mathbf{C}, \mathbf{Z} \rangle < \langle \mathbf{C}, \overline{\mathbf{X}} \rangle$ and $g_{\max}(\overline{\mathbf{X}}) > 1$[1]. Define

$$\widehat{\mathbf{X}} = \frac{\overline{\mathbf{X}} + \beta(\overline{\mathbf{X}})\mathbf{Z}}{1 + \beta(\overline{\mathbf{X}})},$$

where for $\mathbf{X} \in \mathcal{X}$ we set

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

Then $\widehat{\mathbf{X}}$ is a feasible ϵ-optimal solution to (4) with $\mathbf{Tr}(\widehat{\mathbf{X}}) = \omega_x$.

Proof. We first show that $\widehat{\mathbf{X}}$ is feasible. Since $\mathbf{Z}$ is strictly feasible, we have $g_{\max}(\mathbf{Z}) < 1$. By assumption, $g_{\max}(\overline{\mathbf{X}}) > 1$ so $\beta(\overline{\mathbf{X}}) > 0$ Thus, $\widehat{\mathbf{X}}$ is a convex combination of $\overline{\mathbf{X}}$ and $\mathbf{Z}$. Then $\mathbf{Z}, \overline{\mathbf{X}} \in \mathcal{X}$ and the convexity of $\mathcal{X}$ imply that $\widehat{\mathbf{X}} \in \mathcal{X}$, i.e. $\widehat{\mathbf{X}} \succeq \mathbf{0}$ and $\mathbf{Tr}(\widehat{\mathbf{X}}) = \omega_x$. Since each of the packing functions $g_i(\mathbf{X})$ are convex, it follows that $g_{\max}(\mathbf{X})$ is also a convex function. Thus, we have

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

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

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

[1] Otherwise, we just use $\mathbf{Z}$ or $\overline{\mathbf{X}}$ as our solution.

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

We now show that $\widehat{\mathbf{X}}$ is ϵ-optimal. Define

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

where the last equality follows from the definition of the dual set $\mathcal{V}$. We first show that $\rho^* = \max_{\mathbf{X} \in \mathcal{X}} \Psi(\mathbf{X})$. For a fixed $\mathbf{X} \in \mathcal{X}$, define $\beta^+ = \max\{\beta(\mathbf{X}), 0\}$ and

$$\mathbf{Y} = \frac{\mathbf{X} + \beta^+ \mathbf{Z}}{1 + \beta^+} \qquad \Leftrightarrow \qquad \mathbf{X} + \beta^+ \mathbf{Z} = (1 + \beta^+) \mathbf{Y}.$$

Then, $\mathbf{Y}$ is feasible for (4). Also, by the definition of $\beta(\mathbf{X})$, it follows that

$$\Psi(\mathbf{X}) - \rho^* = \langle \mathbf{C}, \mathbf{X} \rangle - \omega_v \max\{0, g_{\max}(\mathbf{X}) - 1\} - \rho^* = \langle \mathbf{C}, \mathbf{X} \rangle - \omega_v \max\{\beta(\mathbf{X})(1 - g_{\max}(\mathbf{Z}), 0\}) - \rho^*.$$

Since $g_{\max}(\mathbf{Z}) < 1$, we can factor and then substitute $\rho_u - \langle \mathbf{C}, \mathbf{Z} \rangle$ for $\omega_v(1 - g_{\max}(\mathbf{Z}))$ to obtain

$$\Psi(\mathbf{X}) - \rho^* = \left\langle \mathbf{C}, \mathbf{X} + \beta^+ \mathbf{Z} \right\rangle - \beta^+ \rho_u - \rho^*.$$

Now we can use the definition of $\mathbf{Y}$ to obtain

$$\Psi(\mathbf{X}) - \rho^* = (1 + \beta^+) \langle \mathbf{C}, \mathbf{Y} \rangle - \beta^+ \rho_u - \rho^* = (1 + \beta^+)(\langle \mathbf{C}, \mathbf{Y} \rangle - \rho^*) - \beta^+(\rho_u - \rho^*) \leq 0,$$

where the inequality follows since $\mathbf{Y}$ is feasible for (4) and $\rho_u \geq \rho^*$. Then, (25) implies that

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

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

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

Therefore, the definition of $\widehat{\mathbf{X}}$ implies that

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

where (32) follows from the definition of $\beta(X)$, and (33) follows from the definition of ω_v in (31) and the fact that $\Psi(\overline{\mathbf{X}}) = \left\langle \mathbf{C}, \overline{\mathbf{X}} \right\rangle - \omega_v(g_{\max}(\overline{\mathbf{X}}) - 1) \geq \rho^* - \epsilon$. Thus, $\widehat{\mathbf{X}}$ is a feasible, ϵ-optimal solution since we have

$$\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle \geq \rho^* - \frac{\epsilon}{1 + \beta(\overline{\mathbf{X}})} \geq \rho^* - \epsilon,$$

□

Theorem 2 can be used to "round" ϵ-saddle-points to both the Sparse PCA and Szegedy's number Lagrangian relaxations into feasible, ϵ-optimal solutions to their respective packing SDPs. A version of Theorem 2 was also established by Z. Lu, Monteiro and Yuan [19]. Note that the Lovász-ϑ function cannot be rounded with Theorem 2.

Recall that the semidefinite relaxation for graph coloring problem (15) and the semidefinite relaxation for the maximum variance unfolding problem (18) 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

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

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

4 Solving the saddle-point problem

The saddle-point problem (20) is a game. An ϵ-saddle-point for (20) is an ϵ-equilibrium for this game. One could, in principle, use fictitious play [4] or similar methods to compute such an ϵ-equilibrium. Since the general methods for computing ϵ-equilibria in minimax games rely on subgradient descent and the primal-dual objective functions are non-smooth, these general methods need $\mathcal{O}(\epsilon^{-2})$ iterations to converge [34].

4.1 Nesterov Procedure for non-smooth optimization

Nesterov proposed an iterative procedure for computing an ϵ-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. saddle-point problems of the form

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

We note that Nesterov [22] has adapted his method from [21] to solve functions of form (34). We provide the relevant theorem from [21] for completeness.

Theorem 3 ([21])**.** *The Nesterov iterative procedure computes an ϵ-saddle-point for (34) in $\mathcal{O}\Big(\epsilon^{-1}\cdot\Omega\sqrt{\frac{D_x D_v}{\sigma_x\sigma_v}}\Big)$ iterations where*

(i) $\Omega^2 = \max_{\{\|\mathbf{v}\|_x\leq 1, \|\mathbf{X}\|_x\leq 1\}} \Big|\Big\langle \sum_{i=1}^{m} v_i\mathbf{A}_i, \mathbf{X}\Big\rangle\Big|^2$ is the "size" of the constraint matrices and $\|\cdot\|_x$ 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 σ_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 σ_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*

$$\max_{\mathbf{X}\in\mathcal{X}} \Big\{ \langle \mathbf{\Gamma}, \mathbf{X}\rangle - \mu_x d_x(\mathbf{X})\Big\}, \tag{35}$$

and

$$\min_{\mathbf{v}\in\mathcal{V}} \Big\{ \sum_{i=1}^{m} \gamma_i v_i + \mu_v d_v(\mathbf{v})\Big\}, \tag{36}$$

for given $\mathbf{\Gamma} \in \mathcal{S}^n$ and $\boldsymbol{\gamma} \in \mathbb{R}^m$. The parameters μ_x and μ_v are functions of $\sigma_v, \sigma_x, \Omega, D_v$ and ϵ 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_x and d_v 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-functions is that the associated "diameters" D_x and D_v are modest.

As is typical in the Nesterov procedure, the numerical value of the constants $\mu_x = \mathcal{O}(\epsilon)$ and $\mu_v = \mathcal{O}(\frac{1}{\mu_x}) = \mathcal{O}(\epsilon^{-1})$ are very different. The multiplier μ_x is the Lipschitz constant of a smoothed approximator of the saddle-point function whereas μ_v is a penalty parameter on the change between dual iterates. In order to find an ϵ-saddle-point, μ_x must be set to the order of ϵ. This, in turn, requires the raising the smoothing constant μ_v to $\mathcal{O}(\frac{1}{\mu_x})$ (see [21] for further details). In practice, reducing μ_v can improve the runtimes of the algorithm (see also [30]).

4.2 Extension of Nesterov procedure to packing constraints

We extend Nesterov's method to packing functions of the form described in (2). A large set of useful packing functions belong to this class – all the packing constraints that arise in the optimization problems described in Section 2 are of this form. For this restricted class of packing functions, the Lagrangian function ϕ is given by

$$\phi(\mathbf{X}, \mathbf{v}, \mathbf{z}_1, \ldots, \mathbf{z}_m) = \langle \mathbf{C}, \mathbf{X} \rangle + \sum_{i=1}^{m} v_i \Big(1 - \sum_{j=1}^{k_i} z_{ij} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle \Big), \tag{37}$$

where for $i = 1, \ldots, m$, we denote $\mathbf{z}_i \in \mathcal{P}_i = \{\mathbf{z} : \mathbf{A}_i \mathbf{z} \leq \mathbf{b}_i\}$ as the variables associated with the i^{th} packing function $g_i(\mathbf{X}) = \max \Big\{ \sum_{j=1}^{k_i} z_{ij} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle : \mathbf{P}_i \mathbf{z}_i \leq \mathbf{b}_i \Big\}$, and v_i as the variables associated with the constraint $g_i(\mathbf{X}) \leq 1$. Since ϕ is quadratic in $\mathbf{v}$ and $\mathbf{z}_i$, we linearize the objective by defining a new set of variables for all i and j as

$$\mathbf{y}_{ij} = v_i \mathbf{z}_{ij}. \tag{38}$$

In terms of these new variables, the Lagrangian function is given by

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

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

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

In [21, 22] the primal feasible set is of the form $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq \omega_x\}$; however, the dual set is of the form $\mathcal{Y} = \{\mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_i v_i \leq 1\}$. We show that the Nesterov procedure can be extended to the larger set of dual spaces $\mathcal{Y}$ in the form of (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 $\mathcal{Y}$. For the primal space, $\mathcal{X}$, we have

$$\max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{v}, \mathbf{y}_1, \ldots, \mathbf{y}_m) = \omega_x \sum_{i=i}^{m} y_i + \omega_x \lambda_{\max} \Big(\mathbf{C} - \sum_{i=1}^{m} \sum_{j=1}^{k_i} y_{ij} \mathbf{A}_{ij} \Big),$$

so the results of [22] (see §4 of that paper) can be used, i.e., we can use the spectral entropy function,

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

We note that the use of the spectral entropy function for smoothing is not new, e.g., see Ben-Tal and Nemirovski [2]. For completeness, we show in Appendix A.2 that the "diameter" is $D_x = \omega_x \ln(n+1)$. What remains is to determine a prox-function for the dual set.

4.2.1 Prox-function for the dual set $\mathcal{Y}$

In every step of the Nesterov procedure we are required to solve a smoothed version of the following problem

$$\min_{(\mathbf{v},\mathbf{y}_1,\ldots,\mathbf{y}_m)\in\mathcal{Y}} \Big\{ \sum_{i=1}^{m} \Big(v_i - \sum_{j=1}^{k_i} y_{ij}\gamma_{ij} \Big) \Big\} \tag{42}$$

where γ_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, k_i$, are parameters that change in each iteration. Here, we have $y_{ij} = v_i z_{ij}$ as described in (38).

Recall that in this section we assume that for all i, the packing function $g_i(\mathbf{X})$ is of the form (2), i.e., for all j, we are given matrices $\mathbf{A}_{ij}$ and $\mathbf{P}_i$ so that $g_i(\mathbf{X}) = \max\{\sum_j z_j \langle \mathbf{A}_{ij}, \mathbf{X}\rangle : \mathbf{P}_i\mathbf{z} \leq \mathbf{b}_i\}$. Let d_i denote any prox-function such that the vector valued function,

$$\mathbf{f}_i(\boldsymbol{\gamma}) = \operatorname{argmin}\Big\{ \sum_{j=1}^{k} \gamma_j z_j + \mu_y d_i(\mathbf{z}) : \mathbf{P}_i\mathbf{z} \leq \mathbf{b}_i \Big\} \tag{43}$$

can be computed efficiently. 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

$$\mathbf{f}_v(\boldsymbol{\gamma}) = \operatorname{argmin}\Big\{ \sum_{j=1}^{k} \gamma_i v_i + \mu_y d_v(\mathbf{v}) : \mathbf{v} \in \mathcal{V} \Big\} \tag{44}$$

in closed form. We smooth (42) using the prox-function

$$d_y(\mathbf{v}, \mathbf{y}_1, \ldots, \mathbf{y}_m) = \sum_{i=1}^{m} v_i d_i(\mathbf{y}_i/v_i) + d_v(\mathbf{v}), \tag{45}$$

to obtain the smooth optimization problem

$$\min_{(\mathbf{v},\mathbf{y}_1,\ldots,\mathbf{y}_m)\in\mathcal{Y}} \Big\{ \sum_{i=1}^{m} \Big(v_i - \sum_{j=1}^{k_i} y_{ij}\gamma_{ij} \Big) \Big\} + \mu_y d_y(\mathbf{v}, \mathbf{y}_1, \ldots, \mathbf{y}_m) \Big\} \tag{46}$$

The optimization problem (46) can be decomposed into the form

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

Let $\boldsymbol{\gamma}_i = [\gamma_{i1}, \ldots, \gamma_{ik_i}]^T$, for $i = 1, \ldots, m$, and $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_m]$, where

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

Then the optimal solution to (46) is given by

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

where the functions $\mathbf{f}_i$, $i = 1, \ldots, m$, are defined in (43) and $\mathbf{f}_v$ is defined in (44). All that remains to be shown is that the function d_y that satisfies (45) is, in fact, a prox-function for the dual set $\mathcal{Y}$. We can now use the following result of [13].

Theorem 4 ([13]). *For all $i = 1, \ldots, m$, suppose that $d_i(\mathbf{z}_i)$ is a prox-function for the set $\mathcal{P}_i = \{\mathbf{z} : \mathbf{P}_i\mathbf{z} \leq \mathbf{b}_i\}$, and $d_v(\mathbf{v})$ is a prox-function for the set $\mathcal{V}$. Then the following statements are true.*

(i) The function defined by $d_y(\mathbf{v}, \mathbf{y}_1, \ldots, \mathbf{y}_m) = \sum_{i=1}^m v_i d_i(\mathbf{y}_i/v_i) + d_v(\mathbf{v})$ is a prox-function of the set $\mathcal{Y}$. For all $i = 1, \ldots, m$, let D_i be the "diameter" of the set $\mathcal{P}_i$ with respect to the prox-function $d_i(\mathbf{z}_i)$, and D_v be the "diameter" of the set $\mathcal{V}$ with respect to the prox-function d_v. Then, the "diameter" of the set $\mathcal{Y}$ with respect to d_y is given by

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

(ii) For $i = 1, \ldots, m$, let σ_i be the convexity parameters of the prox-function $d_i(\mathbf{z}_i)$ with respect to the norm $\|\cdot\|_i$ and define $M_i = \max\{\|\mathbf{z}\| : z \in \mathcal{P}_i\}$. Let σ_v be the convexity parameter of the prox-function $d_v(\mathbf{v})$ with respect to the norm $\|\cdot\|_v$, Then, the convexity parameter of the prox-function d_y is

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

and the norm, $\|\cdot\|_y$, in the $\mathcal{Y}$-space is given by $\|(\mathbf{v}, \mathbf{y}_1, \ldots, \mathbf{y}_m)\|_y = \|\mathbf{v}\|_v + \sum_{i=1}^m \|\mathbf{y}_i\|_i$.

(iii) The parameter Ω for the saddle-point problem associated with (39) is given by

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

where $\Omega_i^2 = \max_{\|\mathbf{y}_i\|_i \leq 1} \max_{\|\mathbf{X}\|_x \leq 1} \Big| \sum_{j=1}^{k_i} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle z_{ij} \Big|^2$, for $i = 1, \ldots, m$.

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

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

We show in Appendix A.1 that this prox-function has a convexity parameter $\sigma_v = 1/\omega_v$ with respect to the ℓ_1-norm and the "diameter" $D_v = \omega_y \ln(m+1)$, and the dual solutions have form

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

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

1. $g(\mathbf{X}) = \langle \mathbf{A}, \mathbf{X} \rangle$, for $\mathbf{A} \succeq \mathbf{0}$: This function is smooth and we do *not* need a prox-function.

2. $g(\mathbf{X}) = \sum_{i,j} |x_{ij}| = \max\big\{ \langle \mathbf{X}, \mathbf{Z} \rangle : |Z_{ij}| \leq 1, \forall i, j\big\}$: The simplest prox-function is $d(\mathbf{Z}) = \frac{1}{2}\sum_{i,j=1}^n |Z_{ij}|^2$. For this prox-function the parameters are

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

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

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

```
NESTEROV PROCEDURE
    Set $N = \frac{\Omega}{\epsilon}\sqrt{\frac{D_x D_v}{\sigma_x \sigma_v}}$,     $\mu_x = \frac{\epsilon}{2D_x}$,     $\mu_v = \frac{\Omega}{\mu_x \sigma_x \sigma_v} = \frac{\Omega D_x}{\sigma_x \sigma_v} \cdot \frac{1}{\epsilon}$ ,
    Fix $(\mathbf{u}^{(0)}, \mathbf{w}^{(0)}) \in \mathcal{Y}$.
    Set $\mathbf{X}^{(0)} = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \Big\{ \langle \mathbf{C}, \mathbf{X} \rangle - \sum_{i=1}^{m} \sum_{j=1}^{k_i} w_{ij}^{(0)} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \Big\}$;
        $\gamma_{ij}^{(0)} = \big\langle -\mathbf{A}_{ij}, \mathbf{X}^{(0)} \big\rangle$, $j = 1, \ldots, k_i$, $i = 1, \ldots, m$;
    Set $(\mathbf{v}_L^{(0)}, \mathbf{y}_L^{(0)}) = (\mathbf{v}_G^{(0)}, \mathbf{y}_G^{(0)}) = \operatorname{argmin}_{(\mathbf{v},\mathbf{y}) \in \mathcal{Y}} \Big\{ \frac{1}{2}(\mathbf{1}_m, \boldsymbol{\gamma}^{(0)})^\top (\mathbf{v}, \mathbf{y}) + \mu_v d_y(\mathbf{v}, \mathbf{y}) \Big\}$;
    for $t = 0$ to $N$
        do
            Set $(\mathbf{u}^{(t+1)}, \mathbf{w}^{(t+1)}) = \Big(\frac{2}{t+3}\Big)(\mathbf{v}_G^{(t)}, \mathbf{y}_G^{(t)}) + \Big(\frac{t+1}{t+3}\Big)(\mathbf{v}_L^{(t)}, \mathbf{y}_L^{(t)})$;
              $\mathbf{X}^{(t+1)} = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} \Big\{ \langle \mathbf{C}, \mathbf{X} \rangle - \sum_{i=1}^{m} w_{ij}^{(t+1)} \langle \mathbf{A}_{ij}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \Big\}$;
              $\gamma_{ij}^{(t+1)} = \big\langle -\mathbf{A}_{ij}, \mathbf{X}^{(t+1)} \big\rangle$, $j = 1, \ldots, k_i$, $i = 1, \ldots, m$;
              $(\widehat{\mathbf{u}}^{(t+1)}, \widehat{\mathbf{w}}^{(t+1)}) = \operatorname{argmin}_{(\mathbf{v},\mathbf{y}) \in \mathcal{Y}} \Big\{ \Big( \frac{2}{\mu_v (t+3)} (\mathbf{1}_m, \boldsymbol{\gamma}^{(t+1)}) - \nabla d_y\big((\mathbf{v}_G^{(t)}, \mathbf{y}_G^{(t)})\big) \Big)^\top (\mathbf{v}, \mathbf{y}) + d_y(\mathbf{v}, \mathbf{y}) \Big\}$;
              $(\mathbf{v}_L^{(t+1)}, \mathbf{y}_L^{(t+1)}) = \Big(\frac{2}{t+3}\Big)(\widehat{\mathbf{u}}^{(t+1)}, \widehat{\mathbf{w}}^{(t+1)}) + \Big(\frac{t+1}{t+3}\Big)(\mathbf{v}_L^{(t)}, \mathbf{y}_L^{(t)})$;
              $(\mathbf{v}_G^{(t+1)}, \mathbf{y}_G^{(t+1)}) = \operatorname{argmin}_{(\mathbf{v},\mathbf{y}) \in \mathcal{Y}} \Big\{ \Big( \sum_{i=0}^{t+1} \frac{i+1}{2} (\mathbf{1}, \boldsymbol{\gamma}^{(i)}) \Big)^\top (\mathbf{v}, \mathbf{y}) + \mu_v d_y(\mathbf{v}, \mathbf{y}) \Big\}$;

    return $\Big( \overline{\mathbf{X}} = \sum_{t=0}^{N} \frac{2(t+1)}{(N+1)(N+2)} \mathbf{X}^{(t)}, (\mathbf{v}_L^{(N)}, \mathbf{y}_L^{(N)}) \Big)$.
```

Figure 1: Nesterov Procedure: Here, $(\mathbf{1}_m, \boldsymbol{\gamma}^{(t)})$ represents the gradient calculation, $(\mathbf{v}_G^{(t)}, \mathbf{y}_G^{(t)})$ and $(\widehat{\mathbf{u}}^{(t)}, \widehat{\mathbf{w}}^{(t)})$ are each calculated via Equations (46), (47), and (48) for the prox-functions of the form described in (45).

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

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

and the optimal solution is

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

where $\beta = \max \Big\{ \Big\| \big(\langle \mathbf{A}_1, \mathbf{X} \rangle \quad \ldots \quad \langle \mathbf{A}_k, \mathbf{X} \rangle \big)^\top \Big\|_2 - \mu, 0 \Big\}$.

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 has the tradeoff that the convexity parameter σ_y is typically very small. Consequently, the number of iterations required to converge to an ϵ-optimal solution increases and the numerical stability of the algorithm can be adversely affected. Therefore, for certain applications it might be more efficient and numerically stable to directly define a prox-function on the $\mathcal{Y}$ space.

Algorithm	packing SDP	Interior Point	Previous work
MAXCUT	$\mathcal{O}\big(n^2 r\log(n)\cdot\epsilon^{-1}\log^3(\epsilon^{-1})\big)$	$\mathcal{O}(\log(\epsilon^{-1})n^{3.5})$ [3]	$\mathcal{O}(nr\log^2(n)\cdot\epsilon^{-2}\log(\epsilon^{-1}))$ [17] $\mathcal{O}(r\log(n)\cdot\epsilon^{-6}\log^3(\epsilon^{-1}))$ [1, 29]
Coloring	$\mathcal{O}\big(n^2 r\log(n)\cdot\epsilon^{-1}\log^3(\epsilon^{-1})\big)$	$\mathcal{O}(n^{.5}(n^3+r^3)\cdot\log(\epsilon^{-1}))$	$\mathcal{O}(nr\log^3(n)\cdot\epsilon^{-4})$ [17]
Lovász ϑ, ϑ^+	$\mathcal{O}\big(n^2 r\log(n)\cdot\epsilon^{-1}\log^3(\epsilon^{-1})\big)$	$\mathcal{O}(n^{.5}(n^3+r^3)\cdot\log(\epsilon^{-1}))$	$\mathcal{O}(n^5\cdot\epsilon^{-2})$ [5]
Sparse PCA	$\mathcal{O}\big(n^4\sqrt{\log(n)}\cdot\epsilon^{-1}\big)$	$\mathcal{O}(n^{6.5}\cdot\log(\epsilon^{-1}))$	$\mathcal{O}\big(n^4\sqrt{\log(n)}\cdot\epsilon^{-1}\big)$ [7]

Table 1: Running time of SDP solvers. n = number of nodes/dimension, r = number of edges/cost matrix sparsity. Except for the case of MAXCUT, the interior point runtimes are from [23].

4.3 Solution algorithm for packing SDPs

The algorithm for solving our saddle-point problems (37) is described in Figure 1. After executing the Nesterov procedure, we then apply Theorem 1, Lemma 1 and/or Theorem 2 as appropriate. We have made a few modifications to the standard version of the Nesterov procedure. We iterate in the dual space, i.e., in $\mathcal{Y}$, and then compute the approximate primal solution $\overline{\mathbf{X}}$ by aggregating over all gradients. We compute the iterate $\mathbf{y}^{(k)}$ using the Bregman distance associated with the prox-function d_y. For prox-functions of the form (45), the dual update decomposes into separate updates of the $\mathbf{v}$ and $\mathbf{z}$ variables (see (47) and (48) for details).

5 Complexity results for specific packing SDPs

5.1 MAXCUT, Graph coloring, and Lovász-ϑ function

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

$$\begin{array}{rll} \rho^* = & \max & \langle \mathbf{L_D}, \mathbf{X}\rangle \\ & \text{s.t.} & \frac{\mathbf{Tr}(\mathbf{D})}{d_i}\left\langle \mathbf{e}_i\mathbf{e}_i^\top, \mathbf{X}\right\rangle \le 1, \quad i = 1,\ldots,n, \\ & & \mathbf{Tr}(\mathbf{X}) \le 1, \\ & & \mathbf{X}\succeq \mathbf{0}, \end{array}$$

where $\mathbf{L_D} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$ denotes the normalized Laplacian. Note that $\mathbf{diag}(\mathbf{L_D}) = \mathbf{I}$ holds. Then since $\mathbf{D}/\mathbf{Tr}(\mathbf{D})$ is feasible to the MAXCUT problem, it follows that $\rho^* \ge 1$. The trace constraint and (23) imply $\omega_x = 1$ and $\omega_v = n$, respectively. Letting $\mathbf{C} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, the Lagrangian relaxation of this packing SDP is

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

where $d(\mathbf{v})$ denotes a diagonal matrix with $\frac{\mathbf{Tr}(\mathbf{D})}{d_i}v_i$ 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 these prox-functions, we have

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

and for these norms, we have

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

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

Corollary 1. *The complexity of computing an ϵ-optimal solution in the relative sense for the* Maxcut *problem using the Nesterov procedure and rounding is $\mathcal{O}\big(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})\big)$, 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

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

where $\tau = \max\Big\{ \min_{(i,j)\in\mathcal{E}} \langle \mathbf{G}_{ij}, \mathbf{X}\rangle : \mathbf{X} \succeq \mathbf{0}\; \mathbf{Tr}(\mathbf{X}) \leq n \Big\} \leq 4$. 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 $d_y(\mathbf{w},\mathbf{v}) = \sum_{(i,j)\in\mathcal{E}} w_{ij} \log(w_{ij}) + \sum_{i=1}^n v_i \log(v_i)$ 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 these prox-functions, the "diameters" are

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

where in this case the sparsity equals the number of edges, i.e., $r = m$. For these norms, the convexity parameters are given by

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

The Nesterov procedure and rounding via Theorem 1 and Lemma 1 require $\mathcal{O}(n\log(n)\cdot\epsilon^{-1})$ iterations to compute an ϵ-approximate solution in the absolute sense. Karger et al. [16] establish that $\rho^* \geq 1/c^*$, where c^* denotes the optimal number of colors required to color the graph. Thus, an ϵ-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 (36) and two optimization problems of the form (35). By Corollary 5 we have the following result.

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

Recall that the packing SDP formulation for the Lovasz-ϑ function is given by

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

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 $d_y(\mathbf{v},\mathbf{w}) = \sum_{(i,j)\in\mathcal{E}} w_{ij}\log(w_{ij}) + \sum_{(i,j)\in\mathcal{E}} v_{ij}\log(v_{ij})$, where $\mathbf{w}$ are the dual-multipliers for the $\big\langle \mathbf{E}^{(i,j)}, \mathbf{X}\big\rangle \leq 1$ constraints and $\mathbf{v}$ are the dual multipliers for the $\big\langle \mathbf{F}^{(i,j)}, \mathbf{X}\big\rangle \leq 1$ 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 these prox-functions, the "diameters" are given by

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

where, as before, $r = m$. For these norms, the convexity parameters are given by

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

The Nesterov procedure requires $\mathcal{O}(n \log(n) \cdot \epsilon^{-1})$ iterations to approximate $\vartheta(\mathcal{G})$ to within ϵ in the absolute sense. Since $\frac{1}{n}\mathbf{I}$ is feasible to (10) it follows that $\vartheta(\mathcal{G}) \geq 1$, so an absolute ϵ-approximation is also ϵ-approximate in the relative sense. Each iteration of the Nesterov procedure requires us to solve one problem of the form (36) and two optimization problems of the form (35). An analogous argument also works for Szegedy's number, ϑ^+. Moreover, Theorem 2 can be used to round the SDP solution to feasibility as $\frac{1}{n}(\mathbf{I} - \frac{1}{n}\mathbf{J})$ is a strict feasible solution to (13). Thus, we have the following result.

Corollary 3. *The complexity of approximating the Lovász-ϑ function and Szegedy's number of a graph to within ϵ (absolutely or relatively) using the Nesterov procedure is $\mathcal{O}\big(n^2 r \log(n) \cdot \epsilon^{-1} \log^3(\epsilon^{-1})\big)$, where r denotes the total number of edges in the graph.*

We compare the best known algorithms for coloring, Maxcut, the Lovasz-ϑ function and Szegedy's number in Table 4.3. 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 large class of SDPs in a unified manner.

5.2 Sparse PCA

In this application the packing function is given by

$$g(\mathbf{X}) = \frac{1}{\kappa} \sum_{i,j} |X_{ij}| = \max_{\{\mathbf{Y}: |Y|_{ij} \leq 1\}} \Big\{ \Big\langle \frac{1}{\kappa}\mathbf{Y}, \mathbf{X} \Big\rangle \Big\}.$$

Therefore, (39), (40) and (47) imply that the Lagrangian relaxation of the sparse PCA packing SDP (16) is given by

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{(v,\mathbf{Y}) \in \mathcal{Y}} \Big\{ \Big\langle \mathbf{C} - \frac{1}{\kappa}\mathbf{Y}, \mathbf{X} \Big\rangle + v \Big\}, \tag{51}$$

where the dual set is of the form $\mathcal{Y} = \{(v, \mathbf{Y}) : 0 \leq v \leq 1, |Y_{ij}| \leq v\}$, and the primal set is of the form $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) = 1\}$. In the sparse PCA formulation one typically assumes that $\mathbf{C}$ has been scaled to ensure that $\mathbf{Tr}(\mathbf{C}) = 1$; therefore, we have that $\omega_x = \max\{\langle \mathbf{C}, \mathbf{X} \rangle : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) = 1\} = 1$. 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 which has a "diameter" $D_x = \log(n+1)$, and convexity parameter $\sigma_x = 1$. For the set $\mathcal{Y}$, we use the *quadratic* prox-function

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

Since the prox-function $d_y(v, \mathbf{Y})$ is not of the form $v d_1(\mathbf{Y}/v) + d_v(v)$, we cannot use Theorem 4 to compute the "diameter" D_y, the convexity parameter σ and the parameter Ω. In Lemma 4 in Appendix C.1 we directly compute that these parameters are $D_y = n^2$, $\sigma_y = 1$, and $\Omega = 1$.

We show in Appendix C.1 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. Therefore, the complexity per iteration is dominated by the cost of computing the exponential of a matrix that is generally dense so the full eigenvalue-eigenvector decomposition is, theoretically, best. Since $\mathbf{I}$ is a strictly feasible solution, we can use Theorem 2 to round.

n		κ		Ind. Vars. [Obs.]		Dep. Vars. [Obs.]	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	122	40	4	2 [40]	30 [4]	1 [20]	1 [2]
240	242	80	4	2 [80]	60 [4]	1 [40]	1 [2]
360	362	120	4	2 [120]	90 [4]	1 [60]	1 [2]
480	482	160	4	2 [160]	120 [4]	1 [80]	1 [2]
600	602	200	4	2 [200]	150 [4]	1 [100]	1 [2]
720	722	240	4	2 [240]	180 [4]	1 [120]	1 [2]
840	842	280	4	2 [280]	210 [4]	1 [140]	1 [2]
960	962	320	4	2 [320]	240 [4]	1 [160]	1 [2]
1080	1082	360	4	2 [360]	270 [4]	1 [180]	1 [2]
1200	1202	400	4	2 [400]	300 [4]	1 [200]	1 [2]
1320	1322	440	4	2 [440]	330 [4]	1 [220]	1 [2]
1440	1442	480	4	2 [480]	360 [4]	1 [240]	1 [2]
1560	1562	520	4	2 [520]	390 [4]	1 [260]	1 [2]
1680	1682	560	4	2 [560]	420 [4]	1 [280]	1 [2]
1800	1802	600	4	2 [600]	450 [4]	1 [300]	1 [2]
1920	1922	640	4	2 [640]	480 [4]	1 [320]	1 [2]
2040	2042	680	4	2 [680]	510 [4]	1 [340]	1 [2]
2160	2162	720	4	2 [720]	540 [4]	1 [360]	1 [2]
2280	2282	760	4	2 [760]	570 [4]	1 [380]	1 [2]
2400	2402	800	4	2 [800]	600 [4]	1 [400]	1 [2]
3600	3602	1200	4	2 [1200]	900 [4]	1 [600]	1 [2]
4800	4802	1600	4	2 [1600]	1200 [4]	1 [800]	1 [2]
6000	6002	2000	4	2 [2000]	1500 [4]	1 [1000]	1 [2]

Table 2: Description of the artificial data.

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

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 general algorithm for solving packing SDPs on the Sparse PCA problem (16). We describe our implementation in detail in Section 6.1. We tested the runtime performance of our implementation on random instances generated in a manner similar to that described in [35] (see also [7]). 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 completely in Matlab [20]. Each experiment was run in Matlab release R2009a on an Opteron 2.6 GHz dual-core two processor machine with 20 GB of RAM[2] and the default multi-threading capabilities of Matlab enabled. Since we use the multi-threading capability of Matlab, we report the actual time (using Matlab functions `tic` and `toc`). For the per-iteration cost, we report the more conservative CPU time which includes the overhead incurred by the multi-threading.

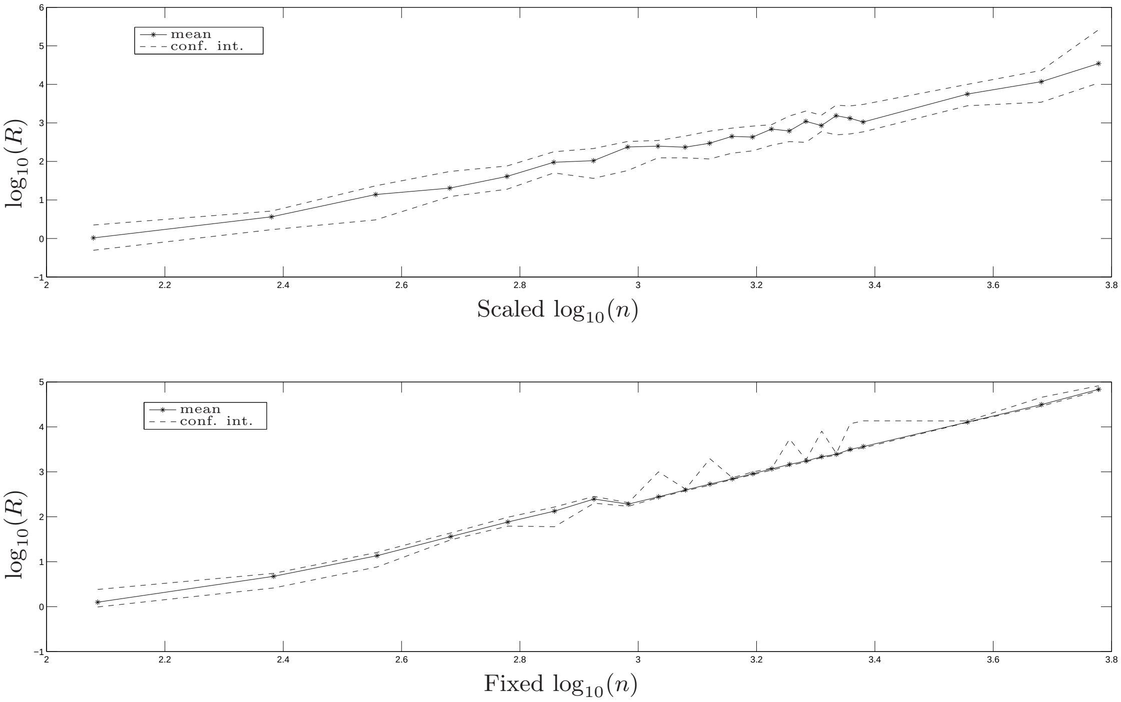

Figure 2: The median runtimes (R) for the packing SDP on Scaled and Fixed. The dotted lines around the medians represent 97.5% confidence intervals. Runtime reported is in real seconds.

6.1 Implementation details

Initial dual iterate. Our solution algorithm for packing SDP needs an initial *feasible* dual solution. In our numerical experiments, we used the initial solution $v = 0.8$ and $\mathbf{Y} = 0.2\,\mathbf{sgn}(\mathbf{C})$, where $\mathbf{sgn}(\mathbf{C})$ is a matrix with $\mathbf{sgn}(C_{ij})$ as the $(i,j)^{\text{th}}$ entry.

Primal iterate, dual gradients and the matrix exponential. The dual gradient is given by the optimal solution of the smoothed primal optimization problem

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

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 solution is

$$\mathbf{X}^* = \frac{e^{\frac{1}{\mu_x}\cdot\left(\mathbf{C}-\frac{1}{\kappa}\mathbf{Y}^{(k)}\right)}}{\mathbf{Tr}\left(e^{\frac{1}{\mu_x}\cdot\left(\mathbf{C}-\frac{1}{\kappa}\mathbf{Y}^{(k)}\right)}\right)} = \frac{\mathbf{V}e^{\Omega}\mathbf{V}}{\sum_i e^{\omega_i}} = \frac{\mathbf{V}e^{\Omega-\nu\mathbf{I}}\mathbf{V}}{\sum_i e^{(\omega_i-\nu)}}$$

where $\Omega = \mathbf{diag}(\omega)$ and ω_i is the i^{th} eigenvalue of the matrix $\mu_x^{-1}(\mathbf{C} - \frac{1}{\kappa}\mathbf{Y}^{(k)})$ and ν is an arbitrary user-defined parameter. Shifting eigenvalues by ν was suggested in Section 5.2 in [21].

We computed $\mathbf{X}^*$ using three different methods: the standard matrix exponential calculator `expm` in MATLAB, a full eigenvalue-vector decomposition, and a partial eigenvalue-vector decomposition (using Lanczos

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

iterations). The latter two methods allow for a finer control over the precision of the exponential computation. We found that the full eigenvalue-vector decomposition performed best with $\nu = \max_i \omega_i$. We also found that $\mu_x < 0.02$ exceeded the precision capabilities for all methods; therefore, we use $\mu_x \in \{.03, .04\}$ depending on the problem class. The quadratic prox-function used in the dual space was stable with respect to the μ_y parameter. An interesting open direction would be to use a quadratic prox-function for the primal optimization step.

Scaling covariance matrix C. Typically in sparse PCA applications one assumes that the matrix $\mathbf{C}$ is scaled so that $\mathbf{Tr}(\mathbf{C}) = 1$. We found that this scaling did not perform well in our numerical experiments. The main reason was the numerical difficulties in computing the exponential of a dense matrix. Since the primal parameter was set so that $\mu_x \geq 0.03$, the primal iterate $X^{(k)}$, i.e. the smoothed optimal solution, is not very close to the true (non-smooth) optimal solution. We found that this inaccuracy did not hamper the progress of the algorithm when the dual iterate was set so that $(v^{(k)}, \mathbf{Y}^{(k)}) \neq (0, \mathbf{0})$. However, when the dual iterates were small, i.e., $(v^{(k)}, \mathbf{Y}^{(k)}) \approx (0, \mathbf{0})$, we need to ensure that the primal iterate

$$\mathbf{X_C} = \underset{\mathbf{X} \in \mathcal{X}}{\operatorname{argmax}} \Big\{ \langle \mathbf{C}, \mathbf{X} \rangle - \mu_x d_x(\mathbf{X}) \Big\}$$

is close to the true (non-smooth) optimal solution in order to correctly compute the dual iterates. If not, it is possible that the smoothed solution is primal feasible whereas the true solution is, in fact, infeasible. In such a case, the successive dual iterates remain close to zero and the algorithm progresses slowly. We can avoid this problem by either decreasing μ_x or scaling up the matrix $\mathbf{C}$. In either case, we increase the computational cost since the effective Lipschitz constant increases. We chose to scale up $\mathbf{C}$ because such a scaling only significantly affects the computational cost iterate when $(v^{(k)}, \mathbf{Y}^{(k)}) \approx (0, \mathbf{0})$. We scale $\mathbf{C}$ by $\nu / \mathbf{Tr}(\mathbf{C})$ where ν is chosen so $g(\mathbf{X_C}) \in [1 + \delta, 1 + \gamma]$ for some positive $\delta < \gamma$. In our numerical experiments we set $\delta = .3$ and $\gamma = .4$. Since each iteration of the search for the correct scaling is equivalent (modulo a constant) to one step of the algorithm, we include these search iterations in our runtimes and iteration counts (which added 4-9 iterations to our iteration counts).

Termination conditions. We used several different early termination conditions concurrently.

(a) At any iteration the duality gap is

$$\begin{aligned} \eta^{(t)} &= \min_{(\mathbf{v},\mathbf{y}) \in \mathcal{Y}} \Big\{ \Big\langle \mathbf{C}, \mathbf{X}^{(t)} \Big\rangle - \sum_{ij} \Big\langle y_{ij} \mathbf{A}_{ij}, \mathbf{X}^{(t)} \Big\rangle + \sum_i v_i \Big\} - \max_{\mathbf{X} \in \mathcal{X}} \Big\{ \Big\langle \mathbf{C} - \sum_{ij} y^{(t)}, \mathbf{X} \Big\rangle + v^{(t)} \Big\} \\ &= \Big(\Big\langle \mathbf{C}, \mathbf{X}^{(t)} \Big\rangle - \max\{0, 1 - \sum_{i,j} |X_{ij}^{(t)}|\} \Big) - \Big(v^{(t)} + \max\{0, \lambda_{\max}(\mathbf{C} - \frac{1}{\kappa} \mathbf{Y}^{(t)})\}\} \Big). \end{aligned}$$

We terminate whenever the duality gap satisfies $\eta^{(t)} \leq \epsilon$. We "round" the output $\overline{\mathbf{X}}$ from the Nesterov procedure into a feasible solution $\widehat{\mathbf{X}}$ using Lemma 5 in Appendix C.2.

(b) Let $\mathbf{X}^{(k)}$ denote the primal iterate computed in iteration k. Let $\overline{\mathbf{X}}^{(t)} = \sum_{k=0}^{t} \frac{2(k+1)}{(t+1)(t+2)} \mathbf{X}^{(k)}$ denote the primal saddle-point solution returned by the Nesterov procedure if it were to be terminated at iteration t. Suppose $\overline{\mathbf{X}}^{(t)}$ is δ-feasible, i.e. $g(\overline{\mathbf{X}}^{(t)}) \leq 1 + \delta$, and the cumulative infeasibility satisfied $\sigma^{(t)} = \sum_{k=0}^{t} \frac{2(k+1)}{(t+1)(t+2)} \big(v^{(k)} - \frac{1}{\kappa} \big\langle \mathbf{Y}^{(k)}, \mathbf{X}^{(k)} \big\rangle \big) \leq \delta$. Then Lemma 6 in Appendix C.2 shows how to construct a feasible solution $\widehat{\mathbf{X}}$ such that $\Big\langle \mathbf{C}, \widehat{\mathbf{X}} \Big\rangle \geq \big(1 - \frac{\kappa\delta}{\kappa - 1}\big) \big(\langle \mathbf{C}, \mathbf{X}^* \rangle - \frac{\epsilon}{2} - \delta \big)$. Thus, we are guaranteed that $\widehat{\mathbf{X}}$ is ϵ-optimal if we set $\delta = \frac{\epsilon(\kappa - 1)}{2\big((\nu+1)\kappa - 1\big)}$, where ν denotes the factor by which we scale $\mathbf{C}$.

n		mean		stand. dev.	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	122	0.10	0.07	0.02	0.05
240	242	0.43	0.45	0.01	0.33
360	362	0.97	1.03	0.05	0.72
480	482	1.56	1.68	0.06	1.04
600	602	2.46	2.49	0.06	1.32
720	722	3.66	3.89	0.11	1.92
840	842	5.38	5.52	0.28	2.51
960	962	7.16	8.01	0.37	3.59
1080	1082	9.89	10.49	0.66	4.61
1200	1202	12.35	14.54	0.62	6.13
1320	1322	15.73	18.24	0.67	7.64
1440	1442	19.02	22.93	0.62	9.41
1560	1562	23.03	28.90	0.53	11.68
1680	1682	27.72	35.39	1.69	14.22
1800	1802	32.68	41.63	0.64	16.54
1920	1922	39.08	50.04	1.30	19.56
2040	2042	45.11	55.74	0.68	21.78
2160	2162	52.54	67.57	1.73	25.83
2280	2282	61.43	78.20	2.98	29.85
2400	2402	69.04	88.90	1.58	33.25
3600	3602	202.10	268.69	7.64	90.80
4800	4802	495.11	635.32	32.32	189.18
6000	6002	923.58	1269.51	52.17	335.83

Table 3: CPU seconds per iteration for the Scaled and Fixed families of Sparse PCA SDP relaxations. Mean and standard deviation are shown.

6.2 Problem data

We focused our experiments on random SDP instances where the number of components and their sparsity were known. The following instance generator was introduced in [35] (see, also [7]).

(i) Descriptive variables: The family is generated from the random variables

$$Y_1 \sim \mathcal{N}(0, \sigma_1^2), \qquad Y_2 \sim \mathcal{N}(0, \sigma_2^2), \qquad D = w_1 Y_1 + w_2 Y_2 + \delta, \quad \delta \sim \mathcal{N}(0, \sigma_3^2),$$

where Y_1, Y_2 and δ are independent random variables and $\mathcal{N}(\mu, \sigma^2)$ denotes a Normal distribution with mean μ and variance σ^2, and

(ii) Observations: The family has the following "observed variables"

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

where each $\eta_i \sim \mathcal{N}(0, 1)$. Thus, there are 4 observation for each Y_i, $i = 1, 2$, and 2 observations for the mixed variable D.

We modified this methodology to construct two instance families of covariance matrices varying in size from 120×120 to 6002×6002.

1. *Scaled family*: In this instance family the descriptive variables are as follows

$$Y_1, Y_2 \sim \text{independently } \mathcal{N}(0, \sigma_i^2), \quad D = 0.8Y_1 - 0.35Y_2,$$

n		mean		stand. dev.		maximum	
Scaled	Fixed	Scaled	Fixed	Scaled	Fixed	Scaled	Fixed
120	122	46.7	73.7	26.1	18.9	104	121
240	242	29.0	37.1	11.1	9.1	47	48
360	362	45.7	41.9	23.5	10.8	75	53
480	482	42.5	55.9	22.8	7.6	83	74
600	602	33.9	67.0	16.3	12.2	62	86
720	722	51.7	66.6	20.0	15.7	93	86
840	842	38.6	81.8	17.5	8.2	75	94
960	962	56.8	43.1	22.1	1.2	90	46
1080	1082	42.0	73.7	15.1	56.7	75	187
1200	1202	35.1	47.1	10.7	0.9	63	49
1320	1322	36.0	64.5	18.3	45.5	74	201
1440	1442	42.2	52.0	14.4	1.5	65	55
1560	1562	36.3	52.8	14.2	2.0	64	56
1680	1682	41.5	54.4	15.1	1.3	61	57
1800	1802	43.7	74.2	24.0	52.0	89	230
1920	1922	56.0	57.4	23.3	0.9	100	59
2040	2042	42.0	133.3	15.5	89.0	66	243
2160	2162	53.7	58.8	24.4	1.4	98	62
2280	2282	43.8	101.7	22.9	75.7	85	254
2400	2402	38.4	102.2	20.4	76.9	80	256
3600	3602	50.1	72.4	22.3	1.2	92	75
4800	4802	39.5	76.1	20.4	19.1	88	117
6000	6002	50.5	74.0	17.1	5.0	81	86

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

where $\sigma_1^2 = 200$ and $\sigma_2^2 = 250$. Each instance of size s was generated by scaling the number of observations by a positive scalar 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 dimension of the covariance matrix is $n = 12s$ and the sparsity variable is $\kappa = 4s$. The theoretical optimal sparse principal component has loadings on the variables in the set $\{X_i : 4s+1 \leq i \leq 8s\}$, i.e., the variables associated with Y_2. We choose ten instances each for $s \in \{10, 20, \ldots, 200\} \cup \{300, 400, 500\}$.

2. *Fixed family*: In this instance family, we fix the number of observations associated with each descriptive variable Y_i at 4 and those associated with D at 2, and scale the number of descriptive variables up by a positive factor c. The variance of the normal random variables was scaled to ensure a dominant component. In particular, we set

$$Y_i \sim \mathcal{N}(0, 4i^2), i = 1, \ldots, c, \quad D = \sum_{i=1}^{c} \frac{1}{\sqrt{c}} Y_i,$$

and generated the observations as follows,

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

For this family, the size of the covariance matrix is $n = 4c + 2$ and the sparsity variable is set to $\kappa = 4$. The theoretical optimal sparse principal component has loadings on the variables in the set $\{X_i : 4c - 3 \leq i \leq 4c\}$, i.e., the variables associated with Y_c. We chose ten instances each for $c \in \{30, 60, \ldots, 600\} \cup \{900, 1200, 1500\}$.

We summarize the data generated in Table 2.

6.3 Results

We report the average runtimes in Table 3 and the average iteration count in Table 4 in order to find relative ϵ-optimal solutions with $\epsilon = .001$. In Table 3 (resp. Table 4) the column labeled 'mean' reports the CPU seconds (resp. number of iterations) averaged over 10 instances for each problem size, the column labeled 'stand. dev.' 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 average number of iterations required to solve instances from the Scaled family was relatively small, ranging from 29 to 56. Also, the standard deviation remained fairly consistent, ranging from 11 to 25. The average number of iterations required to solve the Fixed family was larger and varied more, ranging between 37 and 134 iterations. The standard deviation varied more as well, ranging from 48 and 256.

(b) The best fit line for the average runtime in real seconds is as follows:

$$\text{Scaled family} : \log(R) = -5.61 + 2.61 \log(n)$$
$$\text{Fixed family} : \log(R) = -6.16 + 2.88 \log(n)$$

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 Scaled family had a slightly smaller runtime per iteration growth than the Fixed 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, both in terms of runtime-per-iteration and iteration growth. The principal difference between the two families was that the cardinality constraint, κ remained fixed at 4 for the Fixed family, whereas in the Scaled family, κ grew linearly with the scaling factor s.

We compared the performance of our algorithm against SeDuMi [27], an interior-point based code for solving SDPs. In general, our algorithm was orders of magnitude faster than SeDuMi [27]. However, with default settings SeDuMi was not able to solve the instances we studied – SeDuMi crashed on instances of with covariance matrices larger than 50 × 50. Consequently, we do not have SeDuMi runtimes to report. Our runtimes are also significantly superior to the runtimes reported in [7].

Acknowledgements

We thank the referees and associate editor for their helpful comments. Also, we thank Satyen Kale and Michael Lewis for useful discussions and David Steurer for pointing out the reformulation of the Max-cut problem. We also thank 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] A. Ben-Tal and A. Nemirovski, *Non-Euclidean restricted memory level method for large-scale convex optimization*, Mathematical Programming, 102 (2005), pp. 407–456.

[3] 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).

[4] U. Berger, *Brown's original fictitious play*, Journal of Economic Theory, 135 (2007), pp. 572–578.

[5] T. Chan, K. Chang, and R. Raman, *An SDP primal-dual algorithm for approximating the Lovász-theta function*, in Proceedings of the 2009 IEEE international conference on Symposium on Information Theory-Volume 4, Citeseer, 2009, pp. 2808–2812.

[6] A. d'Aspremont, *Smooth optimization with approximate gradient*, SIAM Journal on Optimization, 19 (2008), pp. 1171–1183.

[7] A. d'Aspremont, L. El Ghaoui, 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] R. Freund, *Dual gauge programs, with applications to quadratic programming and the minimum-norm problem*, Mathematical Programming, 38 (1987), pp. 47–67.

[9] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.

[10] 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.

[11] N. Gvozdenović and M. Laurent, *The operator Ψ 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 Journal on Optimization.

[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®, Mathworks, Inc. http://www.mathworks.com.

[21] Y. Nesterov, *Smooth minimization of nonsmooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.

[22] Y. Nesterov, *Smoothing technique and its applications in semidefinite optimization*, Mathematical Programming, 110 (2007), pp. 245–259.

[23] Y. Nesterov and A. Nemirovski, *Interior-point polynomial algorithms in convex programming*, vol. 13 of SIAM Studies in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[24] S. Plotkin, D. B. Shmoys, and E. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Mathematics of Operations Research, 20 (1995), pp. 257–301.

[25] R. T. Rockafellar, *Convex analysis*, Princeton Mathematical Series, No. 28, Princeton University Press, Princeton, N.J., 1970.

[26] D. Steurer. personal communication, 2009.

[27] 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/.

[28] M. Szegedy, *A note on the ϑ 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.

[29] 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.

[30] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, submitted to SIAM Journal on Optimization, (2008).

[31] J. van den Eshof and M. Hochbruck, *Preconditioning lanczos approximations to the matrix exponential*, 2004. To appear in SIAM J. of Sci. Comp.

[32] 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.

[33] 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.

[34] D. B. Yudin and A. S. Nemirovskiĭ, *Informational complexity and effective methods for the solution of convex extremal problems*, Èkonom. i Mat. Metody, 12 (1976), pp. 357–369.

[35] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, Journal of computational and graphical statistics, 15 (2006), pp. 265–286.

A Details of our prox-functions

A.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*

$$\mathbf{v}^* = \underset{\mathbf{v}\in\mathcal{V}}{\operatorname{argmin}}\{\boldsymbol{\gamma}^\top \mathbf{v} + \mu_v d_v(\mathbf{v})\}. \tag{52}$$

   Then

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

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

Proof. The Hessian

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

is positive definite on any $\mathbf{v} \in \mathcal{V} \cup \mathbb{R}^{m+1}_{++}$. Fix such a $\mathbf{v}$. Then for any $\mathbf{w} \in \mathbb{R}^{m+1}$,

$$\begin{aligned}
\mathbf{w}^\top \nabla^2 d_v(\mathbf{v})\mathbf{w} &= \sum_{i=1}^{m+1} \frac{w_i^2}{v_i}, \\
&= \Big(\sum_{i=0}^{m} \frac{w_i^2}{v_i}\Big)\Big(\frac{1}{\omega_v}\sum_{i=1}^{m+1} v_i\Big), \\
&\geq \frac{1}{\omega_v}\big(\sum_{i=0}^{m} \frac{|w_i|}{\sqrt{v_i}}\sqrt{v_i}\big)^2, && (53)\\
&= \frac{1}{\omega_v}\|\mathbf{w}\|_1^2, && (54)
\end{aligned}$$

where (53) follows from the Cauchy-Schwartz inequality applied to the vector $\mathbf{s} = [w_1/\sqrt{v_1}, \ldots, w_{m+1}/\sqrt{v_{m+1}}]^\top$ and $\mathbf{s} = [\sqrt{v_1}, \ldots, \sqrt{v_{m+1}}]^\top$.

Since the objective function of the optimization problem (52) is strongly convex and the Slater condition holds, it follows that the optimum solution is the unique Karush-Kuhn-Tucker point for the problem. The Lagrangian function for the optimization problem (52) is given by

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

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

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

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

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

Since $d_v(\mathbf{v})$ is a convex function and $\mathcal{V}$ is a simplex, 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 $\omega_v \mathbf{e}_i$, $i = 1, \ldots, m+1$, and at any of these point $d_v(\omega_v \mathbf{e}_i) = \omega_v \ln(m+1)$. Thus, $D_v = \omega_v \ln(m+1)$. □

A.2 The primal prox-function

The primal prox-function in (41) (which is also used in [22]) 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 $\mathbf{X} \in \mathcal{X} = \{\mathbf{X} \succeq 0 : \mathbf{Tr}(\mathbf{X}) \leq \omega_x\}$, and $s_x = \omega_x - \mathbf{Tr}(\mathbf{X})$. 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* $d_x(\mathbf{X}) = \sum\limits_{i=1}^{n+1} \lambda_i(\mathbf{X}) \ln(\lambda_i(\mathbf{X})) - \omega_x \ln(\omega_x/(n+1))$ *where* $\mathbf{X} \in \mathcal{X} = \{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) = \omega_x\}$*.*

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*

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

Then

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

Proof. From results in [2] it follows that $d_x(\mathbf{X})$ is strongly convex with respect to the ℓ_1-norm, $\|\mathbf{X}\|_1 = \sum_{i=1}^{n+1} |\lambda_i(\mathbf{X})|$.

Let $\boldsymbol{\lambda} \in \mathbb{R}^{n+1}$ such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n+1} \geq 0$ 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 $\mathbf{\Gamma}$.

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

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

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

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

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

Thus,

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

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 on the eigenvalues of $\mathbf{X} \in \mathcal{X}$ which lie in a simplex, 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 $d_x(\omega_x \mathbf{u}\mathbf{u}^T) = \omega_x \ln(n+1)$. Thus, $D_x = \omega_x \ln(n+1)$. □

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

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathcal{X}}{\text{argmax}} \Big\{ \langle \boldsymbol{\Gamma}, \mathbf{X}\rangle - \mu_x d_x(\mathbf{X}) \Big\} = \frac{\omega_x e^{\frac{1}{\mu_x}\boldsymbol{\Gamma}}}{1 + \mathbf{Tr}\left(e^{\frac{1}{\mu_x}\boldsymbol{\Gamma}}\right)}, \tag{56}$$

where $e^{\frac{1}{\mu}\boldsymbol{\Gamma}}$ denotes the matrix exponential for a matrix $\boldsymbol{\Gamma} \in \mathcal{S}^n$ scaled by a positive constant $\mu_x \in \mathbb{R}_{++}$. Let $\boldsymbol{\Gamma} = \mathbf{V}\,\mathbf{diag}(\boldsymbol{\gamma})\mathbf{V}^\top$, where $\boldsymbol{\gamma}$ denotes the vector of eigenvalues of $\boldsymbol{\Gamma}$ and $\mathbf{V}$ denotes the matrix with rows equal to the corresponding eigenvectors of $\boldsymbol{\Gamma}$. 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 $\boldsymbol{\Gamma}$. 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 $\boldsymbol{\Gamma}$ [9, 12] or $(\mathbf{I} + \theta\boldsymbol{\Gamma})^{-1}$ [31] for an appropriately chosen θ. Theorem 3.3 of [31] indicates that $\mathcal{O}(\log^2 \cdot \epsilon^{-1})$ Lanzos iterations are required to approximate the matrix-vector product $\exp(\boldsymbol{\Gamma})\mathbf{v}$ for any $\mathbf{v} \in \mathbb{R}^n$. Setting $\mathbf{v} = \mathbf{e}_i, i = 1, \ldots, n$ results in an overall complexity of $\mathcal{O}(nr \log^3 \epsilon^{-1})$, where r denotes the number of non-zero elements in $\boldsymbol{\Gamma}$. Thus, we have the following corollary.

Corollary 5. *The complexity of computing* $\exp(\boldsymbol{\Gamma}/\mu)$ *via* Shift-Invert-Lanczos *procedure proposed in [31] is* $\mathcal{O}\big(nr) \log^3(\epsilon^{-1})\big)$*, where* r *denotes the number of non-zero terms in the matrix* $\boldsymbol{\Gamma}$*. Also, computing* $\exp(\boldsymbol{\Gamma}/\mu)\mathbf{v}$ *for any* $\mathbf{v} \in \mathbb{R}^n$ *requires* $\mathcal{O}\big(r \log^3(\epsilon^{-1})\big)$ *time.*

In practice, Corollary 5 is of limited value for calculating the *full* matrix exponential. However, as noted in [6], a partial matrix exponential can be used to approximate the gradient successfully.

C The sparse PCA packing SDP

C.1 Sparse PCA dual prox-function

Lemma 4. *Let*

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

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

1. *d_y is strongly convex with $\sigma_y = 1$.*
2. *Fix $\mathbf{X} \in \mathcal{S}_n$, $\ell \in \mathbb{R}$ and $\mu_y > 0$. Let*

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

Let $\{\beta_t : t = 1, \ldots, \tau\}$ denote the distinct values in the set $\{\mu_y^{-1} |X_{ij}| : 0 < \mu_y^{-1} |X_{ij}| < 1\}$ sorted in increasing order. Set $\beta_0 = 0$. For $k = 0, \ldots, \tau - 1$, define

$$\alpha_k = \frac{-\ell + \mu_y \sum_{\{ij:|X_{ij}|>\mu_y\beta_k} |X_{ij}|}{\mu_y\big(1 + |\{(ij : |X_{ij}| > \beta_k|\big)}.$$

Then

$$\begin{array}{rcl} Y_{ij}^* & = & -\operatorname{sgn}(X_{ij}) \cdot \min\{\frac{|X_{ij}|}{\mu_y}, v^*\}, \\ v^* & = & \left\{ \begin{array}{ll} 0, & \ell + \sum_{ij} |X_{ij}| \geq 0, \\ 1, & \ell + \mu_y + \sum_{ij} \max\{0, |X_{ij}| - \mu_y\} \leq 0, \\ \alpha_k, & \textit{for some } k \in \{0, \ldots, \tau - 1\}. \end{array} \right. \end{array} \quad (57)$$

3. *(v^*, Y^*) can be computed in $\mathcal{O}(n^2 \ln(n))$ operations.*

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

From the definition of d_y it follows that

$$\begin{aligned} &\min_{(v,\mathbf{Y})\in\mathcal{Y}} \Big\{ \langle \mathbf{X}, \mathbf{Y}\rangle + \ell v + \mu_y d_y(v, Y)\Big\} \\ = \;& \min\Big\{\ell v + \sum_{i,j} X_{ij}Y_{ij} + \frac{\mu_y}{2}(v^2 + \sum_{i,j} Y_{ij}^2) : 0 \leq v \leq 1, -v \leq Y_{ij} \leq v\Big\}. \end{aligned} \quad (58)$$

Since d_y is strongly convex, it follows that (58) has a unique solution. The Lagrangian function of the quadratic program (58) is given by

$$L(v, \mathbf{Y}, p, q, \mathbf{r}, \mathbf{s}) = \ell v + \sum_{i,j} X_{ij}Y_{ij} + \frac{\mu_y}{2}(v^2 + \sum_{i,j} Y_{ij}^2) + p(v-1) - qv + \sum_{i,j}(r_{ij}(Y_{ij} - v) - s_{ij}(Y_{ij} + v)),$$

where $p, q, r_{ij}, s_{ij} \geq 0$. Then (v^*, Y^*) is optimal for (58) if, and only if, (v^*, Y^*) is feasible and there exist multipliers $p^*, q^*, r_{ij}^*, s_{ij}^* \geq 0$ such that

$$\begin{array}{rcrcl} \nabla_{Y_{ij}} L(v, \mathbf{Y}) \mid_{(v^*,Y^*)} & = & X_{ij} + \mu_y Y_{ij}^* + r_{ij}^* - s_{ij}^* & = & 0, \\ \nabla_v L(v, \mathbf{Y}) \mid_{(v^*,Y^*)} & = & \ell + \mu_y v^* + p^* - q^* - \sum_{i,j}(r_{ij}^* + s_{ij}^*) & = & 0, \end{array} \quad (59)$$

and the complementary slackness conditions

$$p^*(1 - v^*) = q^* v^* = r_{ij}^*(v^* - Y_{ij}^*) = s_{ij}^*(Y_{ij}^* + v^*) = 0.$$

hold. From the gradient condition for Y_{ij}^* in (59) and the complementary slackness conditions for r_{ij}^* and s_{ij}^* it follows that

$$Y_{ij}^* = -\operatorname{sgn}(X_{ij}) \cdot \min\Big\{\frac{|X_{ij}|}{\mu_y}, v^*\Big\}, \quad r_{ij}^* = \max\{X_{ij} - \mu_y v^*, 0\}, \quad s_{ij}^* = \max\{-\mu_y v^* - X_{ij}, 0\}. \quad (60)$$

Thus,

$$r_{ij}^* + s_{ij}^* = \max\{0, |X_{ij}| - \mu_y v^*\} \quad \text{and} \quad r_{ij}^* - s_{ij}^* = \operatorname{sgn}(X_{ij}) \max\{0, |X_{ij}| - \mu_y v^*\}. \quad (61)$$

Using the gradient condition for v^* in (59), we have that

$$f(v^*) \triangleq \ell + \mu_y v^* - \sum_{ij}(r^*_{ij} - s^*_{ij}) = \ell + \mu_y v^* - \sum_{ij} \max\{0, |X_{ij}| - \mu_y v^*\} = q^* - p^*.$$

Note that $f(v^*)$ is the gradient of the objective with respect to v^*. We now compute the optimal v^* using case analysis.

(i) $f(0) = \ell - \sum_{ij} |X_{ij}| \geq 0$. Set $v^* = 0$, $p^* = 0$, and $q^* = f(0) \geq 0$. Then (v^*, p^*, q^*) satisfy the gradient condition and the complementary slackness conditions.

(ii) $f(1) = \ell + \mu_y - \sum_{ij} \max\{0, |X_{ij}| - \mu_y\} \leq 0$. Set $v^* = 1$, $p^* = -f(1) \geq 0$, and $q^* = 0$. Then (v^*, p^*, q^*) satisfy the gradient condition and the complementary slackness conditions.

(iii) $f(1) > 0 > f(0)$. Since $f(v^*)$ is a continuous function of v^*, it follows that there exists $v^* \in (0, 1)$ with $f(v^*) = 0$. Since $f(v^*)$ is piece-wise linear, we can compute v^* by sorting $\big\{\, |X_{ij}| : 1 \leq i, j \leq n\big\}$.

Let $\{\beta_t : t = 1, \ldots, \tau\}$ denote the distinct values in the set $\big\{\mu_y^{-1} |X_{ij}| : 0 < \mu_y^{-1} |X_{ij}| < 1\big\}$ sorted in increasing order. Let $\beta_0 = 0$. Since $f(\alpha) = \ell + \mu_y \alpha - \sum_{\{ij:|X_{ij}|>\mu_y\beta_k} \mu_y(|X_{ij}| - \alpha)$, for $\alpha \in (\beta_k, \beta_{k+1}]$, $k = 0, \ldots, \tau - 1$, there exists $v^* \in (\beta_k, \beta_{k+1}]$ with $f(v^*) = 0$ if, and only if,

$$v^* = \alpha_k \triangleq \frac{-\ell + \mu_y \sum_{\{ij:|X_{ij}|>\mu_y\beta_k} |X_{ij}|}{\mu_y\big(1 + |\{(ij : |X_{ij}| > \beta_k|\big)}.$$

The computational cost of computing v^* is dominated by the cost of sorting $\{\mu_y^{-1} |X_{ij}|\}$ and can, therefore, be computed in $\mathcal{O}(n^2 \ln(n))$ time.

C.2 Rounding sparse PCA solutions

Recall that we assume $\kappa > 1$ since the sparse PCA problem reduces to $\arg\max\{C_{ii} : 1 \leq i \leq n\}$ when $\kappa = 1$.

Lemma 5. *Suppose $\kappa > 1$. Let $\overline{\mathbf{X}}$ denote an ϵ-saddle-point for the sparse PCA saddle-point problem* (51)*. Let $\overline{\mathbf{W}} = \mathbf{diag}(\overline{\mathbf{X}})$ and $\overline{\mathbf{Z}} = \overline{\mathbf{X}} - \overline{\mathbf{W}}$. Set*

$$\widehat{\mathbf{X}} = \begin{cases} \overline{\mathbf{W}} & \langle \mathbf{C}, \overline{\mathbf{Z}} \rangle \leq 0, \\ \overline{\mathbf{W}} + \gamma \overline{\mathbf{Z}}, & \textit{otherwise}, \end{cases} \tag{62}$$

where $\gamma = \min\big\{1, \frac{1-g(\overline{\mathbf{W}})}{g(\overline{\mathbf{Z}})}\big\}$. Then $\widehat{\mathbf{X}}$ is a feasible, $\big(\frac{\kappa\epsilon}{\kappa-1}\big)$-optimal solution to the sparse PCA packing SDP (16).

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

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

For $\mathbf{A} = \mathbf{D} + \mathbf{E}$, where $\mathbf{D}$ and $\mathbf{E}$ are disjoint components of $\mathbf{A}$, i.e. $D_{ij}E_{ij} = 0$ for all $1 \leq i, j \leq n$. For example, $\mathbf{D} = \mathbf{diag}(\mathbf{A})$ and $\mathbf{E} = \mathbf{A} - \mathbf{D}$ are disjoint components of $\mathbf{A}$. Then

$$g(\mathbf{A}) = g(\mathbf{D}) + g(\mathbf{E}). \tag{63}$$

Since $\mathbf{Tr}(\overline{\mathbf{X}}) = 1$ and $\overline{\mathbf{W}} = \mathbf{diag}(\overline{\mathbf{X}}) \succeq \mathbf{0}$ it follows that $g(\overline{\mathbf{W}}) = \frac{1}{\kappa} \mathbf{Tr}(\overline{\mathbf{W}}) = \frac{1}{\kappa} < 1$. i.e. $\overline{\mathbf{W}}$ is strictly feasible for the sparse PCA packing SDP (16).

We now show the theorem by case analysis depending on the value of $\overline{\mathbf{Z}}$ at the objective.

(a) $\langle \mathbf{C}, \overline{\mathbf{Z}} \rangle \leq 0$. Then $\langle \mathbf{C}, \overline{\mathbf{W}} \rangle = \langle \mathbf{C}, \overline{\mathbf{X}} \rangle - \langle \mathbf{C}, \overline{\mathbf{Z}} \rangle \geq \langle \mathbf{C}, \overline{\mathbf{X}} \rangle$. Since $\overline{\mathbf{W}}$ is feasible for (16), it follows that $\overline{\mathbf{W}}$ is an ϵ-optimal solution for the sparse PCA packing SDP (16).

(b) $\left\langle \mathbf{C}, \overline{\mathbf{Z}} \right\rangle > 0$. Note that

$$\gamma = \max\{\alpha : g(\overline{\mathbf{W}} + \alpha\overline{\mathbf{Z}}) \leq 1, \alpha \leq 1\}.$$

Consider the following two cases:

(i) $\gamma = 1$. In this case $g(\overline{\mathbf{X}}) \leq 1$. Thus, $\overline{\mathbf{X}}$ is feasible for the sparse PCA SDP. The ϵ-optimality follows as in the proof of Theorem 1.

(ii) $\gamma < 1$. Then, (63) implies

$$g(\widehat{\mathbf{X}}) = g(\overline{\mathbf{W}} + \gamma\overline{\mathbf{Z}}) = g(\overline{\mathbf{W}}) + \gamma g(\overline{\mathbf{Z}}) = g(\overline{\mathbf{W}}) + \frac{1 - g(\overline{\mathbf{W}})}{g(\overline{\mathbf{Z}})} \cdot g(\overline{\mathbf{Z}}) = 1.$$

Also,

$$\widehat{\mathbf{X}} = \overline{\mathbf{W}} + \gamma\overline{\mathbf{Z}} = \overline{\mathbf{W}} + \gamma(\overline{\mathbf{X}} - \overline{\mathbf{W}}) = (1-\gamma)\overline{\mathbf{W}} + \gamma\overline{\mathbf{X}} \succeq \mathbf{0}.$$

Thus, $\widehat{\mathbf{X}}$ is feasible for (16).

Let $\overline{d} = g(\overline{\mathbf{X}})$. Since $\gamma < 1$, it follows that $\overline{\mathbf{X}}$ is infeasible, i.e. $\overline{d} = g(\overline{\mathbf{X}}) > 1$. Since $g(\overline{\mathbf{W}}) = \frac{1}{\kappa}$ and $g(\overline{\mathbf{Z}}) = g(\overline{\mathbf{X}}) - g(\overline{\mathbf{W}}) = \overline{d} - \frac{1}{\kappa}$, it follows that

$$\gamma = \frac{\kappa - 1}{\kappa\overline{d} - 1} = \frac{1}{1 + \frac{\kappa(\overline{d}-1)}{\kappa-1}} \geq 1 - \frac{\kappa}{\kappa - 1}(\overline{d} - 1).$$

Thus,

$$\begin{aligned}
\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle &= \left\langle \mathbf{C}, \overline{\mathbf{W}} \right\rangle + \gamma \left\langle \mathbf{C}, \overline{\mathbf{Z}} \right\rangle \\
&\geq \left\langle \mathbf{C}, \overline{\mathbf{W}} \right\rangle + \big(1 - \frac{\kappa}{\kappa - 1}(\overline{d} - 1)\big) \left\langle \mathbf{C}, \overline{\mathbf{Z}} \right\rangle \\
&\geq \left\langle \mathbf{C}, \overline{\mathbf{X}} \right\rangle - \frac{\kappa}{\kappa - 1}(\overline{d} - 1) \left\langle \mathbf{C}, \overline{\mathbf{Z}} \right\rangle \qquad (64) \\
&\geq \left\langle \mathbf{C}, \overline{\mathbf{X}} \right\rangle - \frac{\kappa}{\kappa - 1}(\overline{d} - 1) \left\langle \mathbf{C}, \overline{\mathbf{X}} \right\rangle, \qquad (65)
\end{aligned}$$

where (64) follows from same argument as in Theorem 1, and (65) follows from that fact that $\left\langle \mathbf{C}, \overline{\mathbf{W}} \right\rangle + \left\langle \mathbf{C}, \overline{\mathbf{Z}} \right\rangle = \left\langle \mathbf{C}, \overline{\mathbf{X}} \right\rangle$ and $\mathbf{C}, \overline{\mathbf{W}} \succeq \mathbf{0}$ (so $\left\langle \mathbf{C}, \overline{\mathbf{W}} \right\rangle \geq 0$). Then, Equations (28), (29) and (30) from the proof of Theorem 1 imply that $\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle \geq \rho^* - \frac{\kappa}{\kappa-1}\epsilon$.

Thus, we have that $\widehat{\mathbf{X}}$ is a feasible, $\left(\frac{\kappa\epsilon}{\kappa-1}\right)$-optimal solution to (4). □

The following lemma establishes the correctness of a stopping criterion used in our sparse PCA code.

Lemma 6. *Let $(v^{(k)}, \mathbf{Y}^{(k)})$, $k \geq 0$, denote the sequence dual iterates generated by the Nesterov algorithm displayed in Figure 1 applied to the sparse PCA problem. Then the primal iterates*

$$\mathbf{X}^{(k)} = \underset{\mathbf{X} \in \mathcal{X}}{\operatorname{argmax}} \Big\{ \langle \mathbf{C}, \mathbf{X} \rangle + \nu^{(k)} - \frac{1}{\kappa} \left\langle \mathbf{Y}^{(k)}, \mathbf{X} \right\rangle - \mu_x d_x(\mathbf{X}) \Big\}, \qquad (66)$$

Let $\overline{\mathbf{X}}^{(t)} = \sum_{k=0}^{t} \frac{2(k+1)}{(t+1)(t+2)} \mathbf{X}^{(t)}$ denote the primal solution returned by the Nesterov algorithm if it were to be terminated at iteration t. Let $\overline{\mathbf{W}}^{(t)} = \mathbf{diag}(\overline{\mathbf{X}}^{(t)})$ and $\overline{\mathbf{Z}}^{(t)} = \overline{\mathbf{X}}^{(t)} - \overline{\mathbf{W}}^{(t)}$. Define

$$\widehat{\mathbf{X}}^{(t)} = \begin{cases} \overline{\mathbf{W}}^{(t)} & \left\langle \mathbf{C}, \overline{\mathbf{Z}}^{(t)} \right\rangle \leq 0, \\ \overline{\mathbf{W}}^{(t)} + \gamma^{(t)} \overline{\mathbf{Z}}^{(t)}, & \textit{otherwise}, \end{cases}$$

where $\gamma^{(t)} = \min\{1, \frac{1-g(\overline{\mathbf{W}}^{(t)})}{g(\overline{\mathbf{Z}}^{(t)})}\}$.

Suppose $\overline{\mathbf{X}}^{(t)}$ *is* δ*-feasible for the packing constraint, i.e.* $g(\overline{\mathbf{X}}^{(t)}) \leq 1 + \delta$*. Then*

$$\left\langle \mathbf{C}, \widehat{\mathbf{X}}^{(t)} \right\rangle \geq \Big(1 - \frac{\kappa}{\kappa - 1}\delta\Big)\Big(\langle \mathbf{C}, \mathbf{X}^* \rangle - \frac{\epsilon}{2} - \sigma^{(t)}\Big),$$

where

$$\sigma^{(t)} = \sum_{k=0}^{t} \frac{2(k+1)}{(t+1)(t+2)}\Big(\nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X} \right\rangle\Big)$$

denotes the average infeasibility of the iterates $\{\mathbf{X}^{(k)}\}$ *and* $\mathbf{X}^*$ *denotes the optimal solution to sparse PCA packing SDP*(16)*.*

Proof. From (66) it follows that

$$\left\langle \mathbf{C}, \mathbf{X}^{(k)} \right\rangle + \nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X}^{(k)} \right\rangle \geq \langle \mathbf{C}, \mathbf{X}^* \rangle + \nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X}^* \right\rangle - \mu_x\big(d_x(\mathbf{X}^*) - d_x(\mathbf{X}^{(k)})\big).$$

Since $\mathbf{X}^*$ is feasible, i.e. $\|\mathbf{X}^*\|_1 = \max_{\{\mathbf{Y}:|Y_{ij}|\leq 1\}} \langle \mathbf{Y}, \mathbf{X}^* \rangle \leq 1$, it follows that $\nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X}^* \right\rangle \geq 0$, and

$$\left\langle \mathbf{C}, \mathbf{X}^{(t)} \right\rangle \geq \langle \mathbf{C}, \mathbf{X}^* \rangle - \big(\nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X}^{(k)} \right\rangle\big) - \mu_x\big(d_x(\mathbf{X}^*) - d_x(\mathbf{X}^{(k)})\big)$$

Since $\mu_x = \frac{\epsilon}{2D_x}$ and $d_x(\mathbf{X}^*) - d_x(\mathbf{X}^{(k)}) \leq D_x$, it follows that

$$\left\langle \mathbf{C}, \mathbf{X}^{(t)} \right\rangle \geq \langle \mathbf{C}, \mathbf{X}^* \rangle - \big(\nu^{(k)} - \frac{1}{\kappa}\left\langle \mathbf{Y}^{(k)}, \mathbf{X}^{(k)} \right\rangle\big) - \frac{\epsilon}{2}.$$

Hence,

$$\left\langle \mathbf{C}, \overline{\mathbf{X}}^{(t)} \right\rangle \geq \langle \mathbf{C}, \mathbf{X}^* \rangle - \sigma^{(t)} - \frac{\epsilon}{2}. \tag{67}$$

Next, consider the two cases

(a) $\left\langle \mathbf{C}, \overline{\mathbf{Z}}^{(t)} \right\rangle < 0$: Since (63) implies that $\langle \mathbf{C}, \overline{\mathbf{X}}(t) \rangle = \left\langle \mathbf{C}, \overline{\mathbf{W}}^{(t)} \right\rangle + \left\langle \mathbf{C}, \overline{\mathbf{Z}}^{(t)} \right\rangle = \left\langle \mathbf{C}, \widehat{\mathbf{X}}^{(t)} \right\rangle + \left\langle \mathbf{C}, \overline{\mathbf{Z}}^{(t)} \right\rangle$, it follows that $\left\langle \mathbf{C}, \widehat{\mathbf{X}}^{(t)} \right\rangle > \left\langle \mathbf{C}, \overline{\mathbf{X}}^{(t)} \right\rangle$. Thus, the result follows.

(b) $\left\langle \mathbf{C}, \overline{\mathbf{Z}}^{(t)} \right\rangle \geq 0$: In this case, an argument similar to that employed in the proof of Lemma 5 establishes that

$$\left\langle \mathbf{C}, \widehat{\mathbf{X}}^{(t)} \right\rangle \geq \Big(1 - \frac{\kappa}{\kappa - 1}\max\{g(\overline{\mathbf{X}}^{(t)}) - 1, 0\}\Big)\left\langle \mathbf{C}, \overline{\mathbf{X}}^{(t)} \right\rangle.$$

The result follows from the fact that $g(\overline{\mathbf{X}}^{(t)}) - 1 \leq \delta$.

□

□