Feasible and accurate algorithms for covering semidefinite programs

G. Iyengar*, D. J. Phillips **, and C. Stein* ***

The Department of Industrial Engineering & Operations Research, Columbia University, cliff,garud@ieor.columbia.edu, Mathematics Department, The College of William & Mary, phillips@math.wm.edu

Abstract. In this paper we describe an algorithm to approximately solve a class of semidefinite programs called *covering semidefinite programs*. This class includes many semidefinite programs that arise in the context of developing algorithms for important optimization problems such as Undirected SPARSEST CUT, wireless multicasting, and pattern classification. We give algorithms for covering SDPs whose dependence on ϵ is only ϵ^{-1}. These algorithms, therefore, have a better dependence on ϵ than other combinatorial approaches, with a tradeoff of a somewhat worse dependence on the other parameters. For many reasons, including numerical stability and a variety of implementation concerns, the dependence on ϵ is critical, and the algorithms in this paper may be preferable to those of the previous work. Our algorithms exploit the structural similarity between covering semidefinite programs, packing semidefinite programs and packing and covering linear programs.

1 Introduction

Semidefinite programming (SDP) is a powerful tool for designing approximation algorithms for NP-hard problems. Early uses of SDP include Lovász's work on the Shannon capacity of a graph [19] and that of Grötschel et al on the stable set of a perfect graph [12]. The Goemans and Williamson MAXCUT algorithm [10] demonstrated the power of SDP-relaxations, and subsequent SDP-based approximation algorithms include those for the SPARSEST CUT [3] and coloring [14]. SDP-relaxations are also used in a variety of important applications such as multicast beam-forming [25], pattern classification [26] and sparse principal component analysis [7].

Solving SDPs remains a significant theoretical and practical challenge. Interior point algorithms compute an approximate solution for an SDP with $n \times n$ decision matrices with an absolute error ϵ in $\mathcal{O}(\sqrt{n}(m^3 + n^6)) \cdot \log(\epsilon^{-1}))$ time, where m is the number of constraints [23]. These algorithms have a very low dependence on ϵ, but a high dependence on the other parameters; for example, interior point algorithms require $\mathcal{O}(n^{9.5} \cdot \log(\epsilon^{-1}))$ time to solve the SDPs that arise in Undirected SPARSEST

* Supported in part by NSF grants CCR-00-09972, DMS-01-04282 and ONR grant N000140310514

** Supported in part by NSF grant DMS-0703532 and a NASA/VSGC New Investigator grant.

*** Supported in part by NSF grants CCF-0728733 and CCF-0915681.

CUT. Thus, interior point algorithms have some drawbacks, and there has been significant work on designing faster algorithms to approximately solve the SDPs that arise in important applications.

One general class of SDPs with efficient solutions are known as *packing SDPs*. (Packing SDPs are analogous to packing linear programs, see [13] for a precise definition). Examples of packing SDPs include those used to solve MAXCUT, COLORING, Shannon capacity and sparse principal component analysis. For these problems there are essentially three known solution methods, each with its own advantages and drawbacks. The first method is specialized interior point methods which have a good dependence on ϵ, but a poor dependence on the other parameters. A second method, due to Klein and Lu [15, 16], extends ideas of, among others, Plotkin, Shmoys and Tardos [24] for packing linear programs to give algorithms for MAXCUT and coloring that have running times of $\mathcal{O}(nm\log^2(n)\cdot\epsilon^{-2}\log(\epsilon^{-1}))$ and $\mathcal{O}(nm\log^3(n)\cdot\epsilon^{-4})$ respectively on graphs with n nodes and m edges. The drawback of these algorithms is the dependence on ϵ of at least ϵ^{-2}; this bound is inherent in these methods [17] and a significant bottleneck in practice on these types of problems [18, 5]. A third approach, due to Iyengar, Phillips and Stein[13] also extends ideas from packing linear programs, but starts from the more recent work of Bienstock and Iyengar [6] who build on techniques of Nesterov [21]. Nesterov [22] has also adapted his method to solve the associated saddle point problem with general SDP, although his method does not find feasible solutions. These results for packing SDPs have a dependence of only ϵ^{-1}, but slightly larger dependence on the other parameters than the algorithms of Klein and Lu. The work in [13] is also significant in that it explicitly defines and approximately solves all packing SDPs in a unified manner.

A related approach for a more general class of SDPs is known as the "multiplicative weights method," which appears in several papers [1, 2] and generalizes techniques used in packing and covering linear programs, e.g. [8, 9]. The work of Arora and Kale [2] gives faster algorithms for solving SDPs (in terms of the problem size), and also extends results to more applications, including Undirected SPARSEST CUT. Their running times achieve a better dependence on n and m than the previous combinatorial algorithms do, but the dependence on ϵ grows to ϵ^{-6} for Undirected SPARSEST CUT. Moreover, they only satisfy each constraint to within an additive error ϵ, and thus, do not necessarily find a feasible solution to the SDP. Their algorithms do, however, incorporate a rounding step so that finding a feasible solution to the SDP is not required.

The authors are not aware of any other work that efficiently finds approximate solutions to large classes of SDPs.

1.1 New results

In this work, we first define a new class of SDPs that we call *covering SDPs*. Analogous to *covering linear programs* which require that all variables be non-negative and all the constraints are of the form $\mathbf{a}^\top\mathbf{x} \geq 1$, with $\mathbf{a} \geq 0$; in a covering SDP the variable matrix $\mathbf{X}$ is positive semidefinite, and all constraints are of the form $\langle \mathbf{A}, \mathbf{X}\rangle \geq 1$ for positive semidefinite $\mathbf{A}$. We show that several SDPs, including those used in diverse applications such as Undirected SPARSEST CUT, Beamforming, and k-nearest neighbors can be expressed as covering SDPs. We then describe an algorithm for computing a feasible solution to a covering SDP with objective value at most $(1+\epsilon)$ times the optimal

Problem	m	This paper	Previous work
Undirected SPARSEST CUT	$\mathcal{O}(n^3)$	$\tilde{\mathcal{O}}(n^4 \cdot \epsilon^{-1})$	$\tilde{\mathcal{O}}(\min\{E^2 \cdot \epsilon^{-4}, n^2 \cdot \epsilon^{-6}\})$ [2]
Beamforming	$\mathcal{O}(R)$	$\tilde{\mathcal{O}}((n^4 + nR) \cdot \epsilon^{-1})$	$\tilde{\mathcal{O}}((R + n^2)^{3.5} \cdot \log(\epsilon^{-1}))$ [25]
k-Nearest Neighbor	$\mathcal{O}(T^2)$	$\tilde{\mathcal{O}}((T^2 n^3 + T^4) \cdot \epsilon^{-1})$	$\tilde{\mathcal{O}}((n^7 + nT^6) \cdot \log(\epsilon^{-1}))$ [26]

Table 1. Running time comparison. $n =$ matrix dimension, $E =$ number of edges, $R =$ number of receivers, $T =$ number of inputs. We use $\tilde{\mathcal{O}}$ to suppress $\log^k(n)$ factors and all but the highest dependence on ϵ.

objective value. We call such a solution a relative ϵ-*optimal solution*. (We will use the term absolute approximation to refer to an additive approximation bound.) Our algorithm has an ϵ^{-1} dependence, and the dependence on other parameters depends on the specific application. The running times of our algorithm and previous works, applied to three applications, are listed in Table 1. To obtain these results, we first give new SDP formulations of each problem, showing that they are examples of covering SPD. We then apply our main theorem (see Theorem 1 below).

Our algorithm for covering SDP is *not* a simple extension of that for packing SDPs [13]. Several steps that were easy for packing SPDs are not so for covering SDPs and give rise to new technical challenges.

- Computing feasible solutions for covering SDPs is non-trivial; and previous work does *not* compute true feasible solutions. For packing SDPs a feasible solution can be constructed by simply scaling the solution of a Lagrangian relaxation; whereas in covering SDPs, both scaling and shifting by a known strictly feasible point is required.
- In both packing and covering, the SDP is solved by solving a Lagrangian relaxation. In packing, it is sufficient to solve a single Lagrangian relaxation; whereas in covering, we need to solve a sequence of Lagrangian relaxations. The iterative approach is necessary to ensure feasibility. The convergence analysis of this iterative scheme is non-trivial.
- Our algorithms use *quadratic* prox functions as opposed to the more usual logarithmic functions [2, 13]. Quadratic prox functions avoid the need to compute matrix exponentials and are numerically more stable. The quadratic prox function was motivated by the numerical results in [13].

Relative to barrier interior point methods, our algorithm represents an increased dependence on ϵ from $O(\log(\frac{1}{\epsilon})$ to $O(\frac{1}{\epsilon}$ in order to receive a considerable improvement in runtime with respect to the other problem parameters. In comparison to other "combinatorial" algorithms, our approximation algorithm reduces the dependence on ϵ at the expense of an increase in some of the other terms in the running time. At first glance, a decrease in the dependence on ϵ may not seem very significant, since for constant ϵ the decrease is only by a constant (albeit a potentially large constant) factor. However, experience with implementations of approximation algorithms for packing and covering type problems has repeatedly shown that the dependence on ϵ is a major bottleneck in designing an efficient algorithm (See, e.g. [4, 5, 11, 13, 18]), as it impacts the numerical stabilty and the convergence rate of the algorithm. Therefore, the reduction of the dependence on ϵ is an important technical endeavor, as is understanding the tradeoff between dependence on ϵ and other problem parameters.

2 Preliminaries

A semidefinite program(SDP) is an optimization problem of the form

$$\begin{array}{ll} \min & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{subject to} & \langle \mathbf{A}_i, \mathbf{X} \rangle \geq b_i, \quad i = 1, \ldots, m, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_i \in \mathbb{R}^{n \times n}, i = 1, \ldots, m$, decision variable $\mathbf{X} \in \mathbb{R}^{n \times n}$ and $\langle ., . \rangle$ denotes the usual *Frobenius* inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{Tr}(\mathbf{A}^\top \mathbf{B}) = \sum_{j=1}^{n} \sum_{i=1}^{n} A_{ij} B_{ij}$. The constraint $\mathbf{X} \succeq \mathbf{0}$ indicates that the symmetric matrix $\mathbf{X}$ is *positive semidefinite*, i.e. $\mathbf{X}$ has nonnegative eigenvalues. We use $\mathcal{S}^n$ and $\mathcal{S}^n_+$ to denote the space of $n \times n$ symmetric and positive semidefinite matrices, respectively, and omit the superscript n when the dimension is clear. For a matrix $\mathbf{X} \in \mathcal{S}^n$, we let $\lambda_{\max}(\mathbf{X})$ denote the largest eigenvalue of $\mathbf{X}$.

3 The covering SDP

We define the *covering SDP* as follows:

$$\begin{array}{rcll} \nu^* & = & \min & \langle \mathbf{C}, \mathbf{X} \rangle \\ & & \text{subject to} & \langle \mathbf{A}_i, \mathbf{X} \rangle \geq 1, \quad i = 1, \ldots, m \\ & & & \mathbf{X} \subseteq \mathcal{X} := \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq \tau\} \end{array} \tag{2}$$

where $\mathbf{A}_i \succeq \mathbf{0}, i = 1, \ldots, m$, $\mathbf{C} \succeq \mathbf{0}$ and the $\mathcal{X} \subset \mathcal{S}_+$ is a set over which linear optimization is "easy". We refer to the constraints of the form $\langle \mathbf{A}, \mathbf{X} \rangle \geq 1$ for $\mathbf{A} \succeq \mathbf{0}$ as *cover constraints*. Before describing our algorithms, we make a set of assumptions. Each appliction we consider satisfies these assumptions. We assume the following about the covering SDP (2):

(a) We can compute $\mathbf{Y} \in \mathcal{X}$ such that $\min_{1 \leq i \leq m} \{\langle \mathbf{A}_i, \mathbf{Y} \rangle\} \geq 1 + \frac{1}{q(n,m)}$ for some positive function $q(n, m)$, i.e. $\mathbf{Y}$ is *strictly* feasible with the *margin of feasibility* at least $\frac{1}{q(n,m)}$.

(b) We can compute a lower bound $\nu_L \in \mathbb{R}$ for (2) such that

$$\nu_U := \langle \mathbf{C}, \mathbf{Y} \rangle \leq \nu_L \cdot p(n, m)$$

for some positive function $p(n, m)$, i.e. the *relative error* of $\mathbf{Y}$ can be bounded above by $p(n, m)$.

Our main result is that we can find ϵ-optimal solutions to covering SDPs efficiently.

Theorem 1. *Suppose a covering SDP (2) satisfies Assumptions (a) and (b). Then a relative ϵ-optimal solution can be found in $\mathcal{O}\Big(\tau q(n, m) \log(p(n, m)\frac{1}{\epsilon})(T_G + \kappa(m)) \|A\| \sqrt{\ln(m)} \cdot \frac{1}{\epsilon}\Big)$ time where $\|A\| = \max_{1 \leq i \leq m} \lambda_{\max}(\mathbf{A}_i)$, T_G denotes the running time for computing the negative eigenvalues and the corresponding eigenvectors for a matrix of the form $\mathbf{C} - \sum_{i=1}^{m} v_i \mathbf{A}_i$, $\mathbf{v} \in \mathbb{R}^n$, and $\kappa(m)$ the running time to calculate $\langle \mathbf{A}_i, \mathbf{Z} \rangle$ for all i and any $\mathbf{Z} \in \mathcal{X}$.*

We prove this theorem to Section 4.3. For the results in Table 1 we set $T_G = n^3$ and calculate $\kappa(m)$ as described in their individual sections. In the remainder of this section, we describe how to formulate our applications as covering SDPs.

3.1 Undirected SPARSEST CUT

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a connected undirected graph with $n = |\mathcal{V}|$ nodes and $E = |\mathcal{E}|$ edges. Let $\mathbf{L}$ denote the Laplacian of $\mathcal{G}$, $\mathbf{K} = n\mathbf{I} - \mathbf{J}$ denote the Laplacian of the complete graph with unit edge weights and $\mathbf{K}_{ij}$ denote the Laplacian of the graph with a single undirected edge (i, j).

As in [3], we assume that $\mathcal{G}$ is unweighted and connected. In Appendix A.1 we show that the ARV formulation of [3] is equivalent to the following new covering SDP formulation.

USC ARV Formulation

$$\begin{aligned} \min \quad & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \langle \mathbf{K}, \mathbf{X} \rangle = 1 \\ \forall i,j,k \ & \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{X} \rangle \geq 0 \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

USC Covering SDP Formulation

$$\begin{aligned} \min \ & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} \ & \langle \mathbf{K}, \mathbf{X} \rangle \geq 1 \\ & \tfrac{n}{4} \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik} + 2\mathbf{I}, \mathbf{X} \rangle \geq 1, \quad \forall i,j,k \in \mathcal{V} \\ & \mathbf{X} \in \mathcal{X} = \left\{ \mathbf{Y} : \mathbf{Y} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{Y}) \leq \tfrac{2}{n} \right\} \end{aligned}$$

Both formulations have $m = \mathcal{O}(n^3)$ constraints. To obtain the covering SDP formulation we relax the first set of constraints, add a trace constraint $\mathbf{Tr}(\mathbf{X}) \leq \frac{2}{n}$, and then shift the second set of linear constraints. Although a trace bound of $\frac{1}{n}$ is sufficient for equivalence, we need the bound to be $\frac{2}{n}$ in order to construct a feasible solution that satisfies Assumption (a).

We now show that Assumption (a) is satisfied. Let $\mathbf{Y} = \frac{2}{n^2(n-1)}\mathbf{K} \succeq \mathbf{0}$. Then $\mathbf{Tr}(\mathbf{Y}) = \frac{2}{n^2(n-1)}(n(n-1)) = \frac{2}{n}$, thus, $\mathbf{Y} \in \mathcal{X}$. For all i, j, k,

$$\frac{n}{4} \langle \mathbf{K}ij + \mathbf{K}_{jk} - \mathbf{K}_{ik} + 2\mathbf{I}, \mathbf{Y} \rangle = \frac{1}{2n(n-1)} \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{K} \rangle + 1 = \frac{1}{n-1} + 1.$$

Also, $\langle \mathbf{K}, \mathbf{Y} \rangle = 2 > 1 + \frac{1}{n}$. Thus, $\mathbf{Y}$ satisfies Assumption (a) with $q(n, m) = n$.

Since $\mathcal{G}$ is unweighted, the upper bound $\nu_U = \langle \mathbf{L}, \mathbf{Y} \rangle = \langle \mathbf{L}, \mathbf{K} \rangle = 2$. It was shown in [3] that the ARV formulation has a lower bound of $\frac{1}{\sqrt{\log(n)}}$, i.e., $\nu_L = \frac{1}{\sqrt{\log(n)}}$; thus, $p(m, n) = 2\sqrt{\log(n)}$. Finally, note that $\|A\| = \lambda_{\max}(\mathbf{K}) = n$. Then, since $\tau = \frac{2}{n}$, Theorem 1 implies the following result.

Corollary 1. *An ϵ-optimal solution to the Undirected* SPARSEST CUT *SDP can be found in*
$\mathcal{O}\left(n^4 \sqrt{\ln(n)} \cdot \frac{1}{\epsilon}\right)$ *time.*

3.2 Beamforming

Sidiropoulos et al [25] consider a wireless multicast scenario with a single transmitter with n antenna elements and R receivers each with a single antenna. Let $\mathbf{h}_i \in \mathbb{C}^n$

denote the complex vector that models the propagation loss and the phase shift from each transmit antenna to the receiving antenna of user $i = 1, \ldots, R$. Let $\mathbf{w}^* \in \mathbb{C}^n$ denote the beamforming weight vector applied to the n antenna elements. Suppose the transmit signal is zero-mean white noise with unit power and the noise at receiver i is zero-mean with variance σ_i^2. Then the received signal to noise ratio (SNR) at receiver i is $\left|\mathbf{w}^*\mathbf{h}_i\right|^2 / \sigma_i^2$. Let ρ_i denote the minimum SNR required at receiver i. Then the problem of designing a beamforming vector $\mathbf{w}^*$ that minimizes the transmit power subject to constraints on the received SNR of each user can be formulated as the following optimization problem.

$$\begin{aligned} &\min \left\|\mathbf{w}\right\|_2^2, \\ &\text{subject to } \left|\mathbf{w}^*\mathbf{h}_i\right|^2 \geq \rho_i \sigma_i^2, \quad i = 1, \ldots, R. \end{aligned}$$

In [25], the authors show that this optimization problem is NP-hard and formulate the following SDP relaxation

$$\begin{aligned} &\min \left\langle \mathbf{I}, \mathbf{X} \right\rangle, \\ &\text{subject to } \left\langle \mathbf{Q}_i, \mathbf{X} \right\rangle \geq 1, \quad i = 1, \ldots, R, \\ &\mathbf{X} \succeq \mathbf{0}, \end{aligned} \tag{3}$$

$$\mathbf{Q}_i := \frac{\gamma}{\rho_i \sigma_i^2} \left(\mathbf{g}_i \mathbf{g}_i^\top + \bar{\mathbf{g}}_i \bar{\mathbf{g}}_i^\top \right), \quad \mathbf{g}_i = \begin{pmatrix} \Re(\mathbf{h}_i)^\top & \Im(\mathbf{h}_i)^\top \end{pmatrix}^\top, \qquad \bar{\mathbf{g}}_i = \begin{pmatrix} \Im(\mathbf{h}_i)^\top & -\Re(\mathbf{h}_i)^\top \end{pmatrix}^\top,$$

where $\mathbf{X} \in \mathcal{S}^{2n}$, and and γ is chosen to ensure that $\min_{1 \leq i \leq R} \mathbf{Tr}(\mathbf{Q}_i) = 1$.

We will now show that (3) is a covering SDP that satisfies Assumptions (a) and (b). Since $\mathbf{Tr}(\mathbf{Q}_i) \geq 1$, it follows that for any $c \geq 2$, $\mathbf{Y} = c\mathbf{I}$ is a strictly feasible for (3) with $q(m, n) = c - 1 = \mathcal{O}(1)$ and $\left\langle \mathbf{I}, \mathbf{Y} \right\rangle = cn$. Thus, we can assume without loss of generality that the optimal solution of (3) belongs to the set $\mathcal{X} = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq cn\}$. The dual of (3) is given by

$$\begin{aligned} &\max \textstyle\sum_{i=1}^R v_i, \\ &\text{subject to } \textstyle\sum_{i=1}^R v_i \mathbf{Q}_i \preceq \mathbf{I}, \\ &\mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Let k denote any index such that $\mathbf{Tr}(\mathbf{Q}_k) = 1$. Let $\bar{\mathbf{v}} = 2\mathbf{e}_k$, where $\mathbf{e}_k$ denote a vector with 1 is the k-th position and zeroes everywhere else. It is easy to check that both the non-zero eigenvalues of the rank-2 matrices $\mathbf{Q}_i$ are equal to $\gamma \left\|\mathbf{g}_i\right\|^2 / \rho_i \sigma_i^2$. Therefore, $\lambda_{\max}(\mathbf{Q}_k) = \frac{1}{2}$. Thus, $\bar{\mathbf{v}}$ is feasible for the dual with an objective value 2. Thus, we have that $\nu_L = \left\langle \mathbf{I}, \mathbf{Y} \right\rangle = 2n \leq n \cdot 2$, i.e. $p(n, R) = n$ and satisfies Assumption b. Thus, since $\tau = 2n$, Theorem 1 implies the following result.

Corollary 2. *A relative ϵ-optimal feasible solution for the beamforming SDP (3) can be computed in $\mathcal{O}\left(n(n^3 + R)\sqrt{\ln(R)} \cdot \frac{1}{\epsilon} \right)$ time.*

3.3 *k*-Nearest Neighbor Classification

Weinberger et al [26] consider the following model for pattern classification. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph where $T = |\mathcal{V}|$ and each node $i \in \mathcal{V}$ has an *input*, $(\mathbf{v}_i, y_i), i =$

$1, \ldots, T$, where $\mathbf{v}_i \in \mathbb{R}^n$ and y_i are labels from a finite discrete set. For each i, there are a constant k adjacent nodes in $\mathcal{G}$, i.e., $\mathcal{G}$ is k-regular, and if $(i,j) \in \mathcal{E}$, then i and j are "near" with respect to some distance function on $\mathbf{v}_i$ and $\mathbf{v}_j$. The goal is to use the T inputs to derive a linear transformation, $\mathbf{H} \in \mathbb{R}^{n \times n}$, so that for input i, $\mathbf{H}\mathbf{v}_i$ is still near its k nearest neighbors but "far" away from inputs j that do not share the same label, i.e., $y_i \neq y_j$. Let $\mathcal{F} = \{(i,j,\ell) : (i,j) \in \mathcal{E}, y_i \neq y_\ell\}$. Let $\mathbf{L}$ denote the Laplacian associated with $\mathcal{G}$ and $\mathbf{C} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & c\mathbf{I} \end{pmatrix}$ where $\mathbf{0}$ denotes an appropriately sized matrix of all zeros and $c > 0$ is a given constant. Also, let $\hat{\mathbf{A}}_{ij\ell}$ denote that diagonal block matrix with $\mathbf{K}_{i\ell} - \mathbf{K}_{ij}$ (the edge Laplacians to (i,j) and (i,ℓ) respectively). Finally, let $\mathbf{A}_{ij\ell} = \hat{\mathbf{A}}_{ij\ell} + \hat{\mathbf{I}}$ where $\hat{\mathbf{I}}$ denotes the block diagonal matrix with $\mathbf{I}$ in the upper $n \times n$ block and zeros everywhere else. In Appendix A.2, we show that the following two formulations are equivalent.

kNN WBS Formulation

$$\begin{aligned} \min\ & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.}\ & \left\langle \hat{\mathbf{A}}_{ij\ell}, \mathbf{X} \right\rangle \geq 1 \ (i,j,\ell) \in \mathcal{E}_D \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned}$$

kNN Covering SDP Formulation

$$\begin{aligned} \min\ & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.}\ & \langle \mathbf{A}_{ij\ell}, \mathbf{X} \rangle \geq 1 \ (i,j,\ell) \in \mathcal{E}_D \\ & \mathbf{Tr}(\mathbf{X}) \leq kn \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

The WBS formulation is due to Weinberger, Blitzer and Saul [26]. To obtain the covering SDP formulation, we add the trace constraint trace constraint $\mathbf{Tr}(\mathbf{X}) \leq kT$ and shift the second set of constraints as we did in Undirected SPARSEST CUT. Note the number of constraints is $m = kT^2 = \mathcal{O}(T^2)$ Arguments similar to those used to construct covering SDP formulations for Undirected SPARSEST CUT and Beamforming show that $\nu_L = 1$, $\nu_U = p(n,m) = q(n,m) = \mathcal{O}(T)$ (see Appendix A.2). Thus, we have the following corrollary to Theorem 1.

Corollary 3. *An ϵ-optimal solution to the k-Nearest Neighbors covering SDP can be found in*
$\mathcal{O}(T^2(n^3 + T^2)\sqrt{\log(T)} \cdot \frac{1}{\epsilon})$ *time.*

4 Computing a relative ϵ-optimal solution for a covering SDP

In this section we describe the steps of our solution algorithm SOLVECOVERSDP (See Figure 1).

– We start with the Lagrangian relaxation of (2)

$$\nu_\omega^* = \min_{\mathbf{X} \in \mathcal{X}} \max_{\mathbf{v} \in \mathcal{V}} \left\{ \left\langle \mathbf{C} - \omega \sum_{i=1}^m v_i \mathbf{A}_i, \mathbf{X} \right\rangle + \omega \sum_{i=1}^m v_i \right\}, \tag{4}$$

where $\mathcal{V} = \{\mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_{i=1}^n v_i \leq 1\}$ and penalty multiplier ω controls the magnitude of the dual variables. Lagrange duality implies that we need $\omega \to \infty$ to ensure strict feasibility.

– In Section 4.2 we show that an adaptation of an algorithm due to Nesterov [21] allows us to compute an ϵ-saddle-point, i.e. $(\hat{\mathbf{v}}, \hat{\mathbf{X}})$ such that

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

in $\tilde{\mathcal{O}}(\frac{\|\mathbf{A}\|\omega\tau}{\epsilon})$ iterations of a Nesterov non-smooth optimization algorithm [21], where
$\|\mathbf{A}\| = \max_{1\leq i\leq m} \lambda_{\max}(\mathbf{A}_i)$; thus, large ω leads to larger running times.
– In Section 4.1 we show that an ϵ-saddle-point can be converted into a relative ϵ-optimal solution provided $\omega \geq \frac{1}{g(\mathbf{Y})} \cdot \left(\frac{\nu_U - \nu_L}{\nu_L} \right)$, where ν_U (resp. ν_L) is an upper (resp. lower) bound on the optimal value ν^* of the covering SDP (2), and $g(\mathbf{Y}) = \min_{1\leq i\leq m} \langle \mathbf{A}_i, \mathbf{Y} \rangle - 1$ denotes the feasibility margin for any strictly feasible point $\mathbf{Y}$. Assumptions (a) and (b) guarantee that $\frac{\nu_U - \nu_L}{\nu_L} \leq p(n,m)$ and that there exists a feasible $\mathbf{Y}$ with $g(\mathbf{Y}) \geq \frac{1}{q(n,m)}$. Thus, it follows that one can compute an ϵ-optimal solution in $\tilde{\mathcal{O}}\big(\frac{p(n,m)q(n,m)\|\mathbf{A}\|}{\epsilon}\big)$ iterations.
– In Section 4.3 we show that by solving a sequence of covering SDPs we can reduce the overall number of iterations to $\tilde{\mathcal{O}}\big(\frac{q(n,m)\|\mathbf{A}\|}{\epsilon}\big)$, i.e. reduce it by a factor $p(n,m)$.

The running time per iteration is dominated by an eigenvalue computation and the solution of an optimization problem via an active set method.

4.1 Rounding the Lagrangian saddle point problem

Let $g(\mathbf{Y}) = \min_{i=1,\ldots,m} \left\{ \langle \mathbf{A}_i, \mathbf{Y} \rangle - b_i \right\}$ denote the feasibility margin with respect to the covering constraint in (2) . Then $\mathbf{Y} \in \mathcal{X}$ is *strictly* feasible for (2) if, and only if, $g(\mathbf{Y}) > 0$.

Lemma 1. *Let $\mathbf{Y} \in \mathcal{X}$ be a strictly feasible solution to (2), so $g(\mathbf{Y}) > 0$. Define $\bar{\omega} := \frac{\langle \mathbf{C}, \mathbf{Y} \rangle - \nu^*}{g(\mathbf{Y})}$, and assume $\bar{\omega} > 0$. Choose $\omega \geq \bar{\omega}$, and suppose $(\hat{\mathbf{X}}, \hat{\mathbf{v}})$ is a δ-saddle-point for (4) i.e.*

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

Then $\overline{\mathbf{X}} = \frac{\hat{\mathbf{X}} + \beta(\hat{\mathbf{X}})\mathbf{Y}}{1+\beta(\hat{\mathbf{X}})}$, where $\beta(\hat{\mathbf{X}}) = g(\hat{\mathbf{X}})^- / g(\mathbf{Y})$, is feasible and absolute δ-optimal for (2).

Proof. We first show that $\overline{\mathbf{X}}$ is feasible to (2). Since $\overline{\mathbf{X}}$ is a convex combination of $\hat{\mathbf{X}}$ and $\mathbf{Y}$, $\overline{\mathbf{X}} \in \mathcal{X}$. The definition of $\beta(\hat{\mathbf{X}})$ together with the concavity of g implies

$$g(\overline{\mathbf{X}}) \geq \frac{1}{1+\beta(\hat{\mathbf{X}})} \cdot g(\hat{\mathbf{X}}) + \frac{\beta(\hat{\mathbf{X}})}{1+\beta(\hat{\mathbf{X}})} \cdot g(\mathbf{Y}) = \frac{1}{1+\beta(\hat{\mathbf{X}})} \left(g(\hat{\mathbf{X}}) + g(\hat{\mathbf{X}})^- \right) \geq 0.$$

Thus, $\overline{\mathbf{X}}$ is feasible for (2). All that remains is to show that $\left\langle \hat{\mathbf{C}}, \overline{\mathbf{X}} \right\rangle \leq \nu^* + \delta$. First, we show that if the penalty $\omega \geq \bar{\omega}$, then $\nu^*_\omega = \nu^*$. Since $g(\mathbf{X})^- = 0$ for all $\mathbf{X}$ feasible

for (2) it follows that $\nu^*_\omega \leq \nu^*$; therefore, we must show that $\nu^*_\omega \geq \nu^*$ when $\omega \geq \bar{\omega}$. Fix $\mathbf{X} \in \mathcal{X}$ and let $\mathbf{X}^\beta = (\mathbf{X}+\beta(\mathbf{X})\mathbf{Y})/(1+\beta(\mathbf{X}))$, which means $\mathbf{X} = \big(1+\beta(\mathbf{X})\big)\mathbf{X}^\beta - \beta(\mathbf{X})\mathbf{Y}$. Then, by the previous argument, $g(\mathbf{X}^\beta) \geq 0$ so $\left\langle \hat{\mathbf{C}}, \mathbf{X}^\beta \right\rangle \geq \nu^*$. Also,

$$\begin{aligned}
\left\langle \hat{\mathbf{C}}, \mathbf{X} \right\rangle + \omega g(\mathbf{X})^- - \nu^* &= (1+\beta(\mathbf{X}))\left\langle \hat{\mathbf{C}}, \mathbf{X}^\beta \right\rangle - (1+\beta(\mathbf{X}))\nu^* - \beta(\mathbf{X})(\left\langle \hat{\mathbf{C}}, \mathbf{Y} \right\rangle - \nu^*) + \omega g(\mathbf{X})^- \\
&= (1+\beta(\mathbf{X}))\Big(\left\langle \hat{\mathbf{C}}, \mathbf{X}^\beta \right\rangle - \nu^*\Big) - \bar{\omega}\beta(\mathbf{X})g(\mathbf{Y}) + \omega g(\mathbf{X})^- \\
&= (1+\beta(\mathbf{X}))\Big(\underbrace{\left\langle \hat{\mathbf{C}}, \mathbf{X}^\beta \right\rangle - \nu^*}_{\geq 0}\Big) + \underbrace{(\omega-\bar{\omega})}_{\geq 0} g(\mathbf{X})^-. \qquad (5)
\end{aligned}$$

Thus, if $\omega \geq \bar{\omega}$ then $\nu^*_\omega = \nu^*$.

We can now show that an $\overline{\mathbf{X}}$ is δ-optimal for (2) when $\omega \geq \bar{\omega}$. Since $\hat{\mathbf{X}}$ is a δ-saddle-point, it follows that

$$\max_{\mathbf{v}\in\mathcal{V}} \left\{ \left\langle \hat{\mathbf{C}} - \omega \sum_{i=1}^m v_i \mathbf{A}_i, \hat{\mathbf{X}} \right\rangle + \omega \sum_{i=1}^m v_i \right\} = \left\langle \hat{\mathbf{C}}, \hat{\mathbf{X}} \right\rangle + \omega g(\hat{\mathbf{X}})^- \leq \nu^*_\omega + \delta = \nu^* + \delta.$$

Thus, the same argument used in (5) indicates that

$$\delta \geq \left\langle \hat{\mathbf{C}}, \hat{\mathbf{X}} \right\rangle + \omega g(\hat{\mathbf{X}})^- - \nu^* = (1+\beta(\hat{\mathbf{X}}))\Big(\left\langle \hat{\mathbf{C}}, \overline{\mathbf{X}} \right\rangle - \nu^*\Big) + (\omega-\bar{\omega})g(\hat{\mathbf{X}})^-$$

Since $\omega \geq \bar{\omega}$ and $g(\hat{\mathbf{X}})^- \geq 0$, it follows that $\left\langle \hat{\mathbf{C}}, \overline{\mathbf{X}} \right\rangle - \nu^* \leq \frac{\delta}{1+\beta(\hat{\mathbf{X}})} \leq \delta$. □

A version of Lemma 1 was established independently by Lu and Monteiro [20].

4.2 Solving the saddle point problems

In this section we describe how to use a variant of the Nesterov non-smooth optimization algorithm [21] to compute δ-optimal saddle-points for the minimax problem (4). We assume some familiarity with the Nesterov algorithm [21]. Let $f(\mathbf{v})$ denote the dual function (or, equivalently the objective function of the $\mathbf{v}$-player): $f(\mathbf{v}) = \sum_{i=1}^m v_i + \omega\tau \min_{\mathbf{X}\in\bar{\mathcal{X}}} \left\{ \left\langle \mathbf{C} - \sum_{i=1}^m v_i \mathbf{A}_i, \mathbf{X} \right\rangle \right\}$, where $\bar{\mathcal{X}} = \{\mathbf{X} \in \mathcal{S}^n_+ : \mathbf{Tr}(\mathbf{X}) \leq 1\}$. We wish to compute an approximate solution for $\max_{\mathbf{v}\in\mathcal{V}} f(\mathbf{v})$.

In order to use the Nesterov algorithm we need to smooth the non-smooth function f using a strongly convex *prox* function. We smooth f using the spectral quadratic prox function $\sum_{i=1}^n \lambda_i^2(\mathbf{X})$, where $\{\lambda_i(\mathbf{X}) : i = 1, \ldots, n\}$ denotes the eigenvalues of $\mathbf{X}$. Let

$$f_\alpha(\mathbf{v}) := \sum_{i=1}^m v_i + \omega\tau \min_{\mathbf{X}\in\bar{\mathcal{X}}} \left\{ \left\langle \mathbf{C} - \sum_{i=1}^m v_i \mathbf{A}_i, \mathbf{X} \right\rangle + \frac{\alpha}{2} \sum_{i=1}^n \lambda_i^2(\mathbf{X}) \right\}, \qquad (6)$$

The Nesterov algorithm requires that $\alpha = \frac{\delta}{D_x}$, where $D_x = \max_{\mathbf{X}\in\bar{\mathcal{X}}} \frac{1}{2}\lambda_i^2(\mathbf{X}) = \frac{1}{2}$.

To optimize $f_\alpha(\mathbf{v})$ the Nesterov algorithm uses a particular regularized gradient descent method where each step involves solving a problem of the form

$$\max_{\mathbf{v}\in\mathcal{V}} \left\{ \bar{\mathbf{g}}^\top \mathbf{v} - \frac{L}{\epsilon} \cdot d(\mathbf{v}, \overline{\mathbf{v}}) \right\}, \qquad (7)$$

where $\bar{\mathbf{g}}$ is either a gradient computed at the current iterate ($\mathbf{y}^{(k)}$ computation in Figure 3 in Appendix B.2) or a convex combination of gradients of f_α computed at all the previous iterates ($\mathbf{z}^{(k)}$ computation in Figure 3 in Appendix B.2), and $d(\mathbf{v}, \overline{\mathbf{v}})$ denotes the Bregman distance associated with a strongly convex function $\phi(\mathbf{v})$. We use $d(\mathbf{v}, \overline{\mathbf{v}}) = \sum_{i=1}^{n} v_i \ln(v_i/\bar{v}_i)$. Using results from [21], it is easy to show that for this choice of the Bregman distance, the constant $L = \omega^2\tau^2 \left\|\mathbf{A}\right\|^2$, where $\left\|\mathbf{A}\right\| = \max_{1\leq i\leq m} \lambda_{\max}(\mathbf{A}_i)$ and (7) can be solved in closed form in $\mathcal{O}(m)$ operations.

From the envelope theorem it follows that $\nabla f_\alpha(\mathbf{v})^\top = \mathbf{1}^\top + \tau\omega \left(\langle \mathbf{A}_1, \mathbf{X}^* \rangle \ldots \langle \mathbf{A}_m, \mathbf{X}^* \rangle \right)^\top$ where $\mathbf{X}^* = \operatorname{argmin}_{\mathbf{X}\in\bar{\mathcal{X}}} \left\{ \left\langle \hat{\mathbf{C}} - \sum_{i=1}^{m} v_i \mathbf{A}_i, \mathbf{X} \right\rangle + \frac{\alpha}{2}\lambda_i^2(\mathbf{X}) \right\}$. We show below how to compute $\mathbf{X}^*$.

Let $\mathbf{X} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$ and $\hat{\mathbf{C}} - \sum_{i=1}^{m} v_i \mathbf{A}_i = \sum_{i=1}^{n} \theta_i \mathbf{w}_i \mathbf{w}_i^\top$ denote the eigen-decompositions of $\mathbf{X}$ and $\mathbf{C} - \sum_{i=1}^{m} v_i \mathbf{A}_i$, respectively. Suppose we fix the eigenvalues $\{\lambda_i\}$ of $\mathbf{X}$ and optimize over the choice of eigenvectors $\{\mathbf{u}_i\}$. Then it is easy to show that $\min_{\mathbf{U}:\mathbf{U}^T\mathbf{U}=\mathbf{I}} \left\{ \left\langle \hat{\mathbf{C}} - \sum_{i=1}^{m} v_i \mathbf{A}_i, \mathbf{X} \right\rangle \right\} = \sum_{i=1}^{n} \lambda_{(i)}\theta_{[i]}$, where $\lambda_{(i)}$ denotes the i-th *smallest* eigenvalue of $\mathbf{X}$ and $\theta_{[i]}$ denotes the i-th *largest* eigenvalue of $\hat{\mathbf{C}} - \sum_{i=1}^{m} v_i \mathbf{A}_i$, and the minimum is achieved by setting eigenvector $\mathbf{u}_{(i)}$ corresponding to $\lambda_{(i)}$ equal to the eigenvector $\mathbf{w}_{[i]}$ corresponding to $\theta_{[i]}$. Since prox-function $\frac{1}{2}\sum_{i=1}^{n} \lambda_i^2(\mathbf{X})$ is invariant with respect to the eigenvectors of $\mathbf{X}$, it follows that, by suitably relabeling indices, the expression for the function f_α simplifies to

$$f_\alpha(\mathbf{v}) = \sum_{i=1}^{m} v_i + \omega\tau \min \left\{ \sum_{i=1}^{n} \theta_i\lambda_i + \frac{\alpha}{2}\sum_{i=1}^{n} \lambda_i^2 : \sum_{i=1}^{n} \lambda_i \leq 1, \lambda_i \geq 0 \right\}, \tag{8}$$

and $\mathbf{X}^* = \sum_{i=1}^{n} \lambda_i^* \mathbf{w}_i \mathbf{w}_i^T$, where l^* achieves the minimum in (8).

Lemma 2. ACTIVESOLVE *in Appendix B.1 solves (8) in* $\mathcal{O}(n \log n)$ *time.*

The computational effort in calculating ∇f_α is the dominated by the effort required to compute the eigen-decomposition $(\nu_i, \mathbf{u}_i)$. However, ACTIVESOLVE implies that we only have to compute the negative eigenvalues the corresponding eigenvectors. Since $\mathbf{Tr}(\mathbf{A}_k^\top \mathbf{X}^*) = \mathbf{Tr}(\mathbf{A}_k \sum_{i=1}^{n} \lambda_i^* \mathbf{u}_i^* (\mathbf{u}_i^*)^\top) = \sum_{i=1}^{n} \lambda_i^* (\mathbf{u}_i^*)^\top \mathbf{A}_k \mathbf{u}_i^*$, it follows that in order to compute the gradient, we only need to compute the product $(\mathbf{u}_i^*)^\top \mathbf{A}_k \mathbf{u}_i^*$. If the constraint matrices $\mathbf{A}_k$ are sparse we can compute the product $\mathbf{A}_k\mathbf{u}_i^*$ in $\mathcal{O}(s_k)$ time, where s_k denotes the number of non-zero elements in $\mathbf{A}_k$ and the product $(\mathbf{u}_i^*)^\top \mathbf{A}_k \mathbf{u}_i^*$ can be computed in $\mathcal{O}(n+s_k)$ time. Thus, the k-th component of the gradient can be computed in $\mathcal{O}(n(s_k+n))$ time, and all the m terms in $\mathcal{O}(m(n+n^2) + n\sum_{k=1}^{m} s_k)$ time.

When a constraint matrix is the Laplacian of a completely connected graph, i.e. $\mathbf{A}_k = \mathbf{K}$, as in Undirected SPARSEST CUT, although there is no sparsity to exploit, we can exploit the fact that $\mathbf{K} = n\mathbf{I} - \mathbf{J}$, where $\mathbf{J}$ is the all ones matrix. We then have that $\mathbf{Tr}(\mathbf{K}^\top \mathbf{X}^*) = \sum_{i=1}^{n} l_i^* \mathbf{Tr}((n\mathbf{I} - \mathbf{J})\mathbf{u}_i^*)(\mathbf{u}_i^*)^\top)$. Since $\mathbf{J}\mathbf{u}_i^*$ is a vector with identical entries (each the sum of the components of $\mathbf{u}_i^*$), the computational cost is $\mathcal{O}(n)$ additions. Also, $\mathbf{I}\mathbf{u}_i^* = \mathbf{u}_i^*$, so the total cost of computing each of the n summand's trace

is $\mathcal{O}(n)$ additions and multiplies (by the term n) plus $\mathcal{O}(n)$ additional multiplies and additions to scale and add the main diagonal terms. Thus, the total cost is $\mathcal{O}(n^2)$.

Theorem 2 (Theorem 3 in [21]). *The Nesterov non-smooth optimization procedure described in Figure 3 in Appendix B.2 computes an δ-saddle-point in $\mathcal{O}\big(\omega\tau(T_G + \kappa(m))\,\|A\|\,\sqrt{\ln(m)}\cdot\frac{1}{\delta}\big)$ time, where ω denote the penalty multiplier used in defining the saddle-point problem, $\|A\| = \max_{1\le i\le m}\lambda_{\max}(\mathbf{A}_i)$, T_G denotes the running time for of computing the gradient ∇f_α and $\kappa(m)$ denotes the running time to compute $\langle \mathbf{A}_i, \mathbf{Z}\rangle$ for $i = 1,\ldots,m$ and $\mathbf{Z}\in\mathcal{X}$.*

Note that T_G is typically dominated by the complexity of computing the negative eigenvalues and the corresponding eigenvectors, and is, in the worst case, $\mathcal{O}(n^3)$.

Since $\nu_L \le \nu^* \le \nu_U = \langle \mathbf{C}, \mathbf{Y}\rangle$, Lemma 1 implies that $\omega = \frac{\langle \mathbf{C},\mathbf{Y}\rangle - \nu_L}{g(\mathbf{Y})}$ suffices. Since an absolute $(\epsilon\nu_L)$-optimal solution is a relative ϵ-optimal solution, Assumption (b) implies that

$$\frac{\omega}{\nu_L} = \Big(\frac{\nu_U - \nu_L}{\nu_L}\Big)\cdot\frac{1}{g(\mathbf{Y})} \le p(n,m)q(n,m).$$

Thus, Theorem 2 implies that a relative ϵ-optimal solution can be computed in $\mathcal{O}\big(\tau q(n,m)p(n,m)(T_G+m)\,\|A\|\,\sqrt{\ln(m)}\cdot\frac{1}{\epsilon}\big)$ time. In the next section, we show that bisection can help us improve the overall running time by a factor $p(n,m)$.

4.3 Bisection on the gap improves running time

Let $\mathbf{Y}\in\mathcal{X}$ denote the strictly point specified in Assumption (a), i.e. $g(\mathbf{Y}) = \min_{1\le i\le m}\langle \mathbf{A}_i, \mathbf{Y}\rangle > 1 + \frac{1}{q(n,m)}$. In this section, we will assume that $q(n,m)\ge 8$. Let $\Delta = \nu_U - \nu_L$. We initialize the algorithm with $\nu_L^0 = \nu_L$, and $\nu_U^0 = \nu_U$. Let $\nu_L^{(t)}$ and $\nu_U^{(t)}$ denote lower and upper bounds on the optimal value ν^* of (2) and let $\epsilon^{(t)} = \frac{\Delta^{(t)}}{\nu_L^{(t)}}$ denote the relative error at the beginning of the iteration t of SOLVECOVERSDP. Note that Assumption (b) implies $\epsilon^{(0)} = O(p(n,m))$. In iteration t we approximately solve the SDP

$$\begin{aligned}\nu^{(t)} = \min\ & \Big\langle \tfrac{1}{\nu_L^{(t)}}\mathbf{C}, \mathbf{X}\Big\rangle \\ \text{s.t. } & \big\langle \bar{\mathbf{A}}_i, \mathbf{X}\big\rangle \ge q(n,m) + \gamma^{(t)}, \\ & \mathbf{X}\in\mathcal{X}\end{aligned} \tag{9}$$

where $\gamma^{(t)} = \min\{\alpha^t, \epsilon^{(t)}\}$ for some $\frac{5}{6} < \alpha < 1$ and $\bar{\mathbf{A}}_i = q(n,m)\mathbf{A}_i$. Thus, $\big\|\bar{\mathbf{A}}\big\| = q(n,m)\,\|\mathbf{A}\|$.

Since the right hand sides of the cover constraints in (9) are not all equal to 1, it is *not* a covering SDP. However, (9) can be converted into a covering SDP by rescaling; so, we abuse notation and refer to (9) as a covering SDP. In in each iteration we solve (9) with slightly different right hand sides, therefore, we find it more convenient to work with the unscaled version of the problem. Let $g^{(t)}(\mathbf{X}) \triangleq \min_{i=1,\ldots,m}\big\{\big\langle \bar{\mathbf{A}}_i, \mathbf{X}\big\rangle - q(n,m)\big\} - \gamma^{(t)}$ denote the margin of feasibility of $\mathbf{X}$ with respect to the cover constraints in (9). In this section, we let $g(\mathbf{X}) = \min_{i=1,\ldots,m}\big\{\big\langle \bar{\mathbf{A}}_i, \mathbf{X}\big\rangle - q(n,m)\big\}$; thus, $g^{(t)}(\mathbf{X}) = g(\mathbf{X}) - \gamma^{(t)}$.

SOLVECOVERSDP

Inputs: $\mathbf{C}, \mathbf{A}_K, \mathbf{X}^{(0)}, \epsilon, \nu_U^{(0)}, \nu_L^{(0)}, \delta_t$

Outputs: $\mathbf{X}^*$

Set $t \leftarrow 0 \qquad \gamma_t \leftarrow \frac{\nu_U^{(t)} - \nu_L^{(t)}}{\nu_L^{(t)}}$

while $(\gamma_t > \epsilon)$

do

Compute $\mathbf{Y}^{(t)}$, $\left(\frac{\gamma_t}{3}\right)$-optimal solution to $\nu^{(t)}$ using NESTEROV PROCEDURE

Set $(\nu_L^{(t+1)}, \nu_U^{(t+1)}) \leftarrow \begin{cases} (\nu_L^{(t)}, \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)}), \text{ if } \left\langle \mathbf{C}, \mathbf{Y}^{(t)} \right\rangle \leq \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)} \\ (\frac{\nu_L^{(t)} + \Delta^{(t)}/3}{1+\gamma_t \nu_L^{(t)}/\Delta^{(t)}}, \nu_U^{(t)}) \text{ otherwise} \end{cases}$

$t \leftarrow t+1, \qquad \gamma_t = \frac{\nu_U^{(t)} - \nu_L^{(t)}}{\nu_L^{(t)}}$

return $\mathbf{X}^{(t)}$

Fig. 1. Our algorithm for solving the covering SDP

Lemma 3. *For $k \geq 0$, let $\hat{\mathbf{X}}^{(t)} \in \mathcal{X}$ denote an* absolute $\frac{\epsilon^{(t)}}{3}$*-optimal solution, i.e. $\left\langle \mathbf{C}, \hat{\mathbf{X}}^{(t)} \right\rangle \leq \nu^{(t)} + \frac{1}{3}\Delta^{(t)}$. Update $(\nu_L^{(t+1)}, \nu_U^{(t+1)})$ as follows:*

$$(\nu_L^{(t+1)}, \nu_U^{(t+1)}) = \begin{cases} \left(\nu_L^{(t)}, \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)}\right), \text{ if } \left\langle \mathbf{C}, \hat{\mathbf{X}}^{(t)} \right\rangle \leq \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)}, \\ \left(\frac{\nu_L^{(t)} + \frac{1}{3}\Delta^{(t)}}{1+\gamma \frac{\Delta^{(t)}}{\nu_L^{(t)}}}, \nu_U^{(t)}\right), \quad \text{otherwise}. \end{cases}$$

Then, for all $k \geq 0$,

(i) $\frac{\Delta^{(t+1)}}{\nu_L^{(t+1)}} \leq \left(\frac{5}{6}\right)\left(\frac{\Delta^{(t)}}{\nu_L^{(t)}}\right) \leq \left(\frac{5}{6}\right)^{t+1}\left(\frac{\Delta^0}{\nu_L^0}\right)$*, i.e. the gap converges to geometrically to* 0*.*

(ii) $\hat{\mathbf{X}}^{(t)}$ *is feasible for $\nu^{(t+1)}$, and $g^{(t+1)}(\hat{\mathbf{X}}^{(t)}) \geq \gamma^{(t)} - \gamma^{(t+1)} \geq (1-\alpha)\gamma^{(t)}$.*

The proof of Lemma 3 is in Appendix B.1.

Theorem 3. *Fix $\frac{5}{6} < \alpha < 1$. Then for all ϵ satisfying (10)* SOLVECOVERSDP *computes a relative ϵ-optimal solution for the Cover SDP (2) in $\mathcal{O}\Big(\tau q(n,m)\log(p(n,m)\frac{1}{\epsilon})(T_G + m)\|A\|\sqrt{\ln(m)} \cdot \frac{1}{\epsilon}\Big)$ time, where $q(n,m)$ is the polynomial that satisfies Assumption (a), and T_G denotes the running time for computing the gradient ∇f_α in the Nesterov procedure.*

Proof. From Lemma 3 (i) it follows that SOLVECOVERSDP terminates after at most

$$T = \left\lceil \frac{\ln(\frac{\epsilon^{(0)}}{\epsilon})}{\ln(\frac{6}{5})} \right\rceil$$

iterations. From the analysis in the previous sections, we know that the run time for computing an absolute $\frac{1}{3}\epsilon^{(t)}$-optimal solution is

$$\tilde{\mathcal{O}}\Big(\frac{\epsilon^{(t)}}{g^{(t)}(\hat{\mathbf{X}}^{(t)})} \cdot \frac{3}{\epsilon^{(t)}}\Big) = \tilde{\mathcal{O}}\Big(\frac{1}{\gamma^{(t-1)}}\Big),$$

where we have ignored polynomial factors. Since $\alpha > \frac{5}{6}$, it follows that $\gamma^{(t+1)} \leq \alpha\gamma^{(t)}$, i.e. $\gamma^{(t)}$ decreases geometrically. Thus, the overall running time of SOLVECOVERSDP is $\mathcal{O}(\frac{1}{\gamma^{(T)}})$, and all that remains to show is that $\gamma^{(T)} = \epsilon$.

Let $T_\gamma = \inf\{t : \epsilon^{(0)}(\frac{5}{6})^t < \alpha^t\}$. Since $\epsilon^{(t)} \leq \epsilon^{(0)}(\frac{5}{6})^t$, it follows that for all $t \geq T_\gamma$, $\gamma^{(t)} = \epsilon^{(t)}$. From the definitions of T_γ and T it follows that $T \geq T_\gamma$ if

$$\epsilon \leq \frac{5}{6}\big(\epsilon^{(0)}\big)^{-\frac{\ln(\frac{1}{\alpha})}{\ln(\frac{6\alpha}{5})}}. \tag{10}$$

For $\alpha = \frac{6}{7} > \frac{5}{6}$, the bound is $\epsilon \leq 5.85\big(\epsilon^{(0)}\big)^{-0.84}$. □

Bibliography

[1] S. ARORA, E. HAZAN, AND S. KALE, *Fast algorithms for approximate semidefinite programming using the multiplicative weights update method*, in Proceedings of the 46th Annual Symposium on Foundations of Computer Science, 2005.

[2] 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.

[3] S. ARORA, S. RAO, AND U. VAZIRANI, *Expander flows, geometric embeddings, and graph partitionings*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 222–231.

[4] D. BIENSTOCK, *Experiments with a network design algorithm using epsilon-approximate lps*. Talk at ISMP97.

[5] D. BIENSTOCK, *Potential function methods for approximately solving linear programming problems: theory and practice*, International Series in Operations Research & Management Science, 53, Boston, MA, 2002.

[6] D. BIENSTOCK AND G. IYENGAR, *Solving fractional packing problems in $O^*(\frac{1}{\epsilon})$ iterations*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 146–155.

[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] L. FLEISCHER, *Fast approximation algorithms for fractional covering problems with box constraint*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms, 2004.

[9] N. GARG AND J. KONEMANN, *Faster and simpler algorithms for multicommodity flow and other fractional packing problems*, in Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 300–309.

[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] A. V. GOLDBERG, J. D. OLDHAM, S. A. PLOTKIN, AND C. STEIN, *An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow*, in Proceedings of the 4th Conference on Integer Programming and Combinatorial Optimization, 1998, pp. 338–352. Published as Lecture Notes in Computer Science 1412, Springer-Verlag",.

[12] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *Polynomial algorithms for perfect graphs*, in Topics on perfect graphs, vol. 88 of North-Holland Math. Stud., North-Holland, Amsterdam, 1984, pp. 325–356.

[13] 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.

[14] D. KARGER, R. MOTWANI, AND M. SUDAN, *Approximate graph coloring by semidefinite programming*, J. ACM, 45 (1998), pp. 246–265.

[15] 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.

[16] ——, *Space-efficient approximation algorithms for MAXCUT and COLORING semidefinite programs*, in Algorithms and computation (Taejon, 1998), vol. 1533 of Lecture Notes in Comput. Sci., Springer, Berlin, 1998, pp. 387–396.

[17] P. KLEIN AND N. YOUNG, *On the number of iterations for Dantzig-Wolfe optimization and packing-covering approximation algorithms*, in Integer programming and combinatorial optimization (Graz, 1999), vol. 1610 of Lecture Notes in Comput. Sci., Springer, Berlin, 1999, pp. 320–327.

[18] T. LEONG, P. SHOR, AND C. STEIN, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science: Volume 12, Network Flows and Matching*, Oct. 1991, ch. Implementation of a combinatorial multicommodity flow algorithm.

[19] L. LOVÁSZ, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[20] 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).

[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] N. SIDIROPOULOS, T. DAVIDSON, AND Z. LUO, *Transmit beamforming for physical-layer multicasting*, IEEE Transactions on Signal Processing, 54 (2006), p. 2239.

[26] K. WEINBERGER AND L. SAUL, *Distance metric learning for large margin nearest neighbor classification*, The Journal of Machine Learning Research, 10 (2009), pp. 207–244.

A Application details

A.1 Undirected SPARSEST CUT.

We first consider two equivalent SDP relaxations for Undirected SPARSEST CUT [3], and then show how a related covering SDP can be used to solve the SDP relaxations. As in [3], we assume that $\mathcal{G}$ is unweighted. Recall that we assume the graph $\mathcal{G}$ is connected. Here are two equivalent Undirected SPARSEST CUT SDP relaxations.

Formulation (ARV)

$$\begin{aligned} \min \quad & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \langle \mathbf{K}, \mathbf{X} \rangle = 1 \\ \forall i,j,k \ & \langle \mathbf{K}_{ij} + \mathbf{K}_{jk}, \mathbf{X} \rangle \geq \langle \mathbf{K}_{ik}, \mathbf{X} \rangle \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

Formulation (ARVT)

$$\begin{aligned} \min \quad & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} \quad & \langle \mathbf{K}, \mathbf{X} \rangle = 1 && \text{(a)} \\ \forall i,j,k \ & \langle \mathbf{K}_{ij} + \mathbf{K}_{jk}, \mathbf{X} \rangle \geq \langle \mathbf{K}_{ik}, \mathbf{X} \rangle && \text{(b)} \\ & \mathbf{Tr}(\mathbf{X}) \leq \tfrac{2}{n} && \text{(c)} \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

Formulation (ARV) is the original formulation due to Arora, Rao and Vazarani [3] and (ARVT) is the same formulation with a trace constraint added. We first show that the additional trace bound does not change the objective at optimality.

Lemma 4. *Formulation (ARVT) is equivalent to formulation (ARV) in the sense that if one is infeasible than so is the other and they share the same optimal objective if feasible.*

Proof. Since (ARVT) is simply (ARV) with an additional constraint, we must only show that a feasible solution to (ARV) can be transformed into a feasible solution to (ARVT) with the same objective value. Let $\mathbf{X} = \mathbf{V}^\top \mathbf{V}$ be feasible to (ARV) where the positive semidefiniteness of $\mathbf{X}$ implies the existence of the decomposition $\mathbf{V}$. In order to transform $\mathbf{X}$, we shift all the columns of $\mathbf{V}$ so they are centered at the origin. That is, let

$$\mathbf{w} = \frac{1}{n}\mathbf{V}\mathbf{1}, \qquad \mathbf{W} = \mathbf{V} - \mathbf{w}\mathbf{1}^\top,$$

and

$$\mathbf{Y} = \mathbf{W}^\top \mathbf{W} = \mathbf{X} - \mathbf{V}^\top \mathbf{w}\mathbf{1}^\top - \mathbf{1}\mathbf{w}^\top \mathbf{V} + \mathbf{w}^\top \mathbf{w}\mathbf{J}.$$

We must show that $\mathbf{Y}$ is feasible to (ARVT). We first note that any weighted Laplacian matrix $\mathbf{M}$ on a graph with n nodes has

$$\langle \mathbf{M}, \mathbf{Y} \rangle = \langle \mathbf{M}, \mathbf{X} \rangle - \left\langle \mathbf{M}, \mathbf{V}^\top \mathbf{w}\mathbf{1}^\top \right\rangle - \left\langle \mathbf{1}\mathbf{w}^\top \mathbf{V}, \mathbf{M} \right\rangle + \left\langle \mathbf{M}, \mathbf{w}^\top \mathbf{w}\mathbf{J} \right\rangle = \langle \mathbf{M}, \mathbf{X} \rangle ,$$

since $\mathbf{M}\mathbf{1} = \mathbf{1}^\top \mathbf{M} = \mathbf{0}$. Thus,

$$\langle \mathbf{L}, \mathbf{Y} \rangle = \langle \mathbf{L}, \mathbf{X} \rangle , \qquad \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{Y} \rangle = \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{X} \rangle \geq 0, \forall i,j,k,$$

and $\langle \mathbf{K}, \mathbf{Y} \rangle = \langle \mathbf{K}, \mathbf{X} \rangle = 1$. The final equality implies that $\mathbf{Tr}(\mathbf{Y}) = 1/n$ since $\mathbf{K} = n\mathbf{I} - \mathbf{J}$ and

$$\langle \mathbf{J}, \mathbf{Y} \rangle = \left\langle \mathbf{1}\mathbf{1}^\top, \mathbf{W}^\top \mathbf{W} \right\rangle = \left\langle \mathbf{W}\mathbf{1}\mathbf{1}^\top, \mathbf{W}^\top \right\rangle = 0,$$

as $\mathbf{W1} = \mathbf{V1} - \mathbf{w1}^\top\mathbf{1} = n\mathbf{w} - n\mathbf{w} = \mathbf{0}$. Finally, $\mathbf{Y} \succeq \mathbf{0}$ by construction. So a feasible solution to (ARV) implies the existence of a feasible solution to (ARVT) with the same objective, or, equivalently, if (ARVT) is infeasible so is (ARV) and the optimal objective of (ARV) is the same as that of (ARVT). □

We then use the following covering SDP, which is a relaxation of (ARVT).

$$\begin{array}{ll} \min & \langle \mathbf{L}, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{T}_{ijk}, \mathbf{X} \rangle \geq 1, \forall i, j, k \in \mathcal{V} \\ & \langle \mathbf{K}, \mathbf{X} \rangle \geq 1 \\ & \frac{n}{2}\,\mathbf{Tr}(\mathbf{X}) \leq 1 \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{11}$$

where $\mathbf{T}_{ijk} = \frac{n}{4}(\mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik} + 2\mathbf{I})$. Note that $\mathbf{T}_{ijk} \succeq \mathbf{0}$ and we use the last two constraints to set $\mathcal{X} = \left\{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) \leq \frac{2}{n}\right\}$. The need for a trace bound of $\frac{2}{n}$ is so that the feasible solution will satisfy Assumption (a) as we show below.

By suitably shifting the solution, we can solve (ARV) via (11).

Lemma 5. *Formulations (ARV) and (11) share the same objective value at optimality. Moreover, (ARV) is infeasible if, and only if, (11) is.*

Proof. Since (11) is a relaxation of (ARVT), Lemma 4 indicates we need only show a feasible solution to (11) can be converted to one for (ARV) with the same objective. So, suppose $\mathbf{Y}$ is a feasible solution to (11) with $\langle \mathbf{K}, \mathbf{Y} \rangle = \alpha > 1$. Since $\langle \mathbf{M}, \mathbf{J} \rangle = 0$ for any Laplacian matrix $\mathbf{M}$, we can use a shifting by $\mathbf{J}$ to achieve our result. Let

$$\mathbf{Y} = \frac{1}{\alpha}\mathbf{Y} + (1 - \frac{1}{\alpha})\mathbf{J}.$$

Then

$$\langle \mathbf{K}, \mathbf{Y} \rangle = \left\langle \mathbf{K}, \frac{1}{\alpha}\mathbf{Y} + (1 - \frac{1}{\alpha})\mathbf{J} \right\rangle = \frac{1}{\alpha} \langle \mathbf{K}, \mathbf{Y} \rangle = 1.$$

The feasibility of $\mathbf{Y}$ to (11) implies that $\langle \mathbf{T}_{ijk}, \mathbf{Y} \rangle = \left\langle \frac{n}{4}(\mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik} + 2\mathbf{I}), \mathbf{Y} \right\rangle \geq 1$. Thus, we have

$$\langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{Y} \rangle \geq \frac{4}{n} - 2\,\mathbf{Tr}(\mathbf{Y}) \geq \frac{4}{n} - \frac{4}{n} = 0,$$

where the last inequality follows since $\mathbf{Tr}(\mathbf{Y}) \leq \frac{2}{n}$. Then, since $\alpha > 0$, this implies that

$$\langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{Y} \rangle = \frac{1}{\alpha} \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{Y} \rangle \geq 0.$$

So $\mathbf{Y}$ is feasible to (ARV). Note that the objective at $\mathbf{Y}$ only improves since $\langle \mathbf{L}, \mathbf{Y} \rangle = \frac{1}{\alpha} \langle \mathbf{L}, \mathbf{Y} \rangle < \langle \mathbf{L}, \mathbf{Y} \rangle$. Thus, if a feasible $\mathbf{y}$ to (11) is not feasible to (ARVT), it can be converted to a feasible solution to (ARV) with an improved objective. If $\mathbf{y}$ is feasible to (ARVT), then Lemma 4 can be used to convert it to a feasible solution to (ARV). □

We then have, as a corollary to Theorem 1, the following result.

Corollary 4. *An ϵ-optimal solution to the Undirected* SPARSEST CUT *SDP (ARV) can be found in $\tilde{\mathcal{O}}(n^4 \cdot \epsilon^{-1} \log(\epsilon^{-1})$ operations.*

Proof. We show that Assumption (a) is satisfied with $q(n,m) = n$. Note that $\mathbf{Y} = \frac{2}{n^2(n-1)}\mathbf{K} \succeq \mathbf{0}$ has

$$\mathbf{Tr}(\mathbf{Y}) = \frac{2}{n^2(n-1)}(n(n-1)) = \frac{2}{n},$$

and so $\mathbf{Y} \in \mathcal{X}$.

$$\langle \mathbf{T}_{ijk}, \mathbf{Y} \rangle = \frac{1}{2n(n-1)} \langle \mathbf{K}_{ij} + \mathbf{K}_{jk} - \mathbf{K}_{ik}, \mathbf{K} \rangle + 1 = \frac{1}{n-1} + 1.$$

Also, since $\langle \mathbf{K}, \mathbf{K} \rangle = n^2(n-1)$, we have $\langle \mathbf{K}, \mathbf{Y} \rangle = 2 > 1 + \frac{1}{n}$. Note that a trace bound of $\frac{1}{n}$ would not allow a scaled version of $\mathbf{K}$ to satisfy the Assumption (a).

Thus, $\mathbf{Y}$ is feasible to (11) and $\nu_U = \langle \mathbf{L}, \mathbf{Y} \rangle \leq \langle \mathbf{K}, \mathbf{Y} \rangle = 2$ since $\mathcal{G}$ is unweighted. Assumption (b) is by results in [3], where it was shown that the (ARV) formulation has a lower bound of $\frac{1}{\sqrt{\log(n)}}$, i.e., $\nu_L = \frac{1}{\sqrt{\log(n)}}$; thus, $p(m,n) = 2\sqrt{\log(n)} = \tilde{\mathcal{O}}(1)$. Finally, $\mathcal{X} = \{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) \leq \frac{2}{n}\}$ which satisfies Assumption (a). Finally, note that that $\|A\| = \lambda_{\max}(\mathbf{K}) = n$, $T_G = \tilde{\mathcal{O}}(n^3)$ and $m = \mathcal{O}(n^3)$. □

A.2 Distance metric learning

For a positive integer, k, the k-Nearest Neighbor problem can be stated as follows. We are given a set of vertices $\mathcal{V} = \{1, \ldots, T\}$ with $T > k$ and for each $i \in \mathcal{V}$, there is a vector, $\mathbf{v}_i \in \mathbb{R}^n$ and a label $y_i \in \{0, 1, \ldots, R\}$ for some integer $R \geq 1$. We are also given an edge set, $\mathcal{E}_N = \{(i,j) : \mu(\mathbf{v}_i, \mathbf{v}_j) \leq \alpha\}$, which denotes that inputs i and j are near with respect to some distance function μ on $\mathbb{R}^d$. Each node $i \in \mathcal{V}$ has k neighbors, i.e., for fixed i, $|\{(u,v) \in \mathcal{E} : u = i\}| = k$. We also let $\mathcal{E}_D = \{(i,j,\ell) : (i,j) \in \mathcal{E}_L, y_i \neq y_\ell\}$ denote the triple of a nearest neighbor pair, (i,j), along with an input ℓ with a different label than i. Finally, let $L = |\mathcal{E}_L| \leq kT$ and $D = |\mathcal{E}_D| \leq kT^2$. Then, the objective is to find a transformation $\mathbf{H} \in \mathbb{R}^{n \times n}$ so that after transforming the vectors to $\mathbf{H}v_i$, the sum of the nearest neighbor distances is minimized but that the distance between inputs with different labels is maximized. To solve this problem, Weinberger, et al [26] formulate the following SDP with the matrix variable $\mathbf{Y} = \mathbf{H}^\top \mathbf{H}$ and additional variables $z_{ij\ell}$.

$$\begin{array}{ll} \min & \sum_{(i,j) \in \mathcal{E}_N} (\mathbf{v}_i - \mathbf{v}_j)^\top \mathbf{Y} (\mathbf{v}_i - \mathbf{v}_j) + c \sum_{(i,j) \in \mathcal{E}_N, (i,\ell) \in \mathcal{E}_D} z_{ij\ell} \\ \text{s.t.} & (\mathbf{v}_i - \mathbf{v}_\ell)^\top \mathbf{Y} (\mathbf{v}_i - \mathbf{v}_\ell) - (\mathbf{v}_i - \mathbf{v}_j)^\top \mathbf{Y} (\mathbf{v}_i - \mathbf{v}_j) + z_{ij\ell} \geq 1 \quad (i,j,\ell) \in \mathcal{E}_D \\ & z_{ij\ell} \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (i,j,\ell) \in \mathcal{E}_D \\ & \mathbf{Y} \succeq \mathbf{0}. \end{array} \tag{12}$$

Note that $\mathbf{Y} \in \mathcal{S}_n$ and that there are $D = \mathcal{O}(T^2)$ constraints. Also, c is a given, positive constant. To convert this to a Cover SDP, we introduce the variable matrix $\mathbf{X} \in \mathbb{R}^{n+D}$ where

$$\mathbf{X} = \begin{pmatrix} \mathbf{Y} & \mathbf{D} \\ \mathbf{D} & \mathbf{Z} \end{pmatrix}$$

so that $\mathbf{D}$ are dummy variables and $\mathbf{Z}$ is a matrix whose main diagonal is $\mathbf{z} \in \mathbb{R}_+^D$ and off-diagonal entries are dummy variables. Although we make this formulation change, we will not explicitly use the dummy variables and so, not incur more computational costs when we need to perform expensive calculations such as eigenvalue-eigenvector decompositions. Also, we use $\mathbf{C}$ to denote the objective coefficient matrix and $\mathbf{A}_{ij\ell}$ the constraint matrices where

$$\mathbf{C} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & c\mathbf{I} \end{pmatrix} \qquad \text{and} \qquad \hat{\mathbf{A}}_{ij\ell} = \begin{pmatrix} (\mathbf{e}_i - \mathbf{e}_\ell)(\mathbf{e}_i - \mathbf{e}_\ell)^\top - (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{diag}(\mathbf{e}_{ij\ell}) \end{pmatrix}$$

Then, (12) becomes

$$\begin{aligned} &\min \langle \mathbf{C}, \mathbf{X} \rangle \\ &\text{s.t.} \quad \left\langle \hat{\mathbf{A}}_{ij\ell}, \mathbf{X} \right\rangle \geq 1 \ (i,j,\ell) \in \mathcal{E}_D \\ &\qquad \mathbf{X} \succeq \mathbf{0} \end{aligned} \tag{13}$$

Also, we require upper and lower bounds on (13). Since $\hat{\mathbf{I}} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ is (strictly) feasible, we can set $\nu_U = L \leq kT = O(T)$. To find a lower bound, we consider the dual to (13),

$$\begin{aligned} &\max \textstyle\sum_{(i,j,\ell) \in \mathcal{E}_D} p_{ij\ell} \\ &\text{s.t.} \quad \mathbf{C} - \textstyle\sum_{(i,j,\ell) \in \mathcal{E}_D} p_{ij\ell} \hat{\mathbf{A}}_{ij\ell} \succeq \mathbf{0}. \end{aligned}$$

The key is to note that for any given (i,j,ℓ), $\mathbf{C} - \hat{\mathbf{A}}_{ij\ell}$ is the Laplacian formed from removing the edge (i,j) and adding the edge (i,ℓ). In particular, $\mathbf{C} - \hat{\mathbf{A}} \succeq \mathbf{0}$ and $\nu_L = 1$. We now convert (13) into a covering SDP. Note that $\mathbf{C} \succeq \mathbf{0}$, but $\hat{\mathbf{A}}_{ij\ell} \not\succeq \mathbf{0}$ and we do not have a trace bound. To obtain the latter, since $\hat{\mathbf{I}}$ is a strictly feasible solution, it is without loss of generality to assume $\mathbf{Tr}(\mathbf{X}) \leq L = O(T)$. Moreover, note that if a solution $\mathbf{X}$ has $\mathbf{Tr}(\mathbf{X}) < L$, since $\hat{\mathbf{J}} = \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ has $\left\langle \hat{\mathbf{A}}_{ij\ell}, \hat{\mathbf{J}} \right\rangle = \left\langle \mathbf{C}, \hat{\mathbf{J}} \right\rangle = 0$ we can $(L - \mathbf{Tr}(\mathbf{X}))\hat{\mathbf{J}}$ to $\mathbf{X}$ in a manner analogous to Undirected SPARSEST CUT, and keep the objective the same. Thus, we can also let $\mathbf{A}_{ij\ell} = \frac{1}{L}(\hat{\mathbf{A}}_{ij\ell} + \mathbf{I})$ and obtain the covering SDP

$$\begin{aligned} &\min \langle \mathbf{C}, \mathbf{X} \rangle \\ &\text{s.t.} \quad \langle \mathbf{A}_{ij\ell}, \mathbf{X} \rangle \geq 1 \ (i,j,\ell) \in \mathcal{E}_D \\ &\qquad \mathbf{Tr}(\mathbf{X}) \leq L \\ &\qquad \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{14}$$

We similarly scale $\mathbf{C}$ so that $\lambda_{\max}(\mathbf{C}) = O(1)$ (note the scaling effects both ν_U and ν_L the same). Then $\left\langle \mathbf{A}_{ij\ell}, \hat{\mathbf{I}} \right\rangle = \frac{L+1}{L} \geq 1 + \frac{1}{kT}$, so we have $q(n,m) = \tau = p(n,m) = L = O(T)$. Moreover, note that computing eigenvalues of $\mathbf{C} - \sum_{ij\ell} p_{ij\ell} \mathbf{A}_{ij\ell}$ is proportionate to $O(n^3)$ since the matrix is diagonal for the last T elements. Finally, there are $T = O(T^2)$ constraints in (14) as there was in (12) and $\|A\| = \mathcal{O}(1)$ since each constraint matrix is a just an identity plus one edge Laplacian minus another edge Laplacian. We can therefore state the following corollary to Theorem 1.

ACTIVESOLVE

Inputs: $\theta_i, i = 1, ..., n, \alpha$
Outputs: l^*
Set $\hat{\lambda}_i \leftarrow \max\{-\theta_i/\alpha, 0\}, \quad k \leftarrow |\{i : \hat{\lambda}_i > 0\}|$
if $(\sum_{i=1}^n \hat{\lambda}_i) \leq 1$
then
Set $\lambda_i^* \leftarrow \hat{\lambda}_i$
else
Sort $\hat{\lambda}_i$ in ascending order, update indices to sorted order
Set $l \leftarrow n - k + 1, \quad \hat{q} \leftarrow \hat{\lambda}_l$
while ($\sum_{i=l}^n \hat{\lambda}_i - k\hat{q} > 1$)
do
$k \leftarrow k - 1, \quad l \leftarrow l + 1$
$\hat{q} \leftarrow \hat{\lambda}_l, \quad \hat{\lambda}_l \leftarrow 0$

$\hat{q} \leftarrow \frac{1}{k}\big(\sum_{i=l}^n \hat{\lambda}_i - 1\big)$
Set $\lambda_i^* \leftarrow (\hat{\lambda}_i - \hat{q})^+, \forall i.$

return l^*

Fig. 2. The Active Set method for finding the optimal solution to (6).

Corollary 5. *An ϵ-optimal solution to the k-Nearest Neighbors covering SDP can be found in*
$\tilde{\mathcal{O}}(T^2(n^3 + T^2)\sqrt{\log(T)} \cdot \epsilon^{-1})$ *time.*

B Proofs

B.1 Algorithm details for Section 4.2

Proof (Proof of Theorem 2). The optimization problem in (8) is convex and satisfies the Slater condition. Thus, all optimal solutions satisfy Karush-Kuhn-Tucker (KKT) conditions, and since the objective function is strongly convex, the optimal point is unique. Thus, all we have to do is to show that the output of ACTIVESOLVE satisfies the KKT conditions:

$$\lambda_i^* = \Big(-\frac{\theta_i}{\alpha} - q\Big)^+, \quad \big(\sum_{i=1}^n \lambda_i^* - 1\big)q = 0, \quad q \geq 0.$$

If $\sum_{i=1}^n \Big(-\frac{\theta_i}{\alpha}\Big)^+ \leq 1$, ACTIVESOLVE returns $l^* = \frac{1}{\alpha}(-\theta)^+$. Clearly, l^* and $q = 0$ satisfy the KKT conditions.

If $\sum_{i=1}^{n}\Big(-\frac{\theta_i}{\alpha}\Big)^+ > 1$. Then on termination, $q > \hat{\lambda}_{l-1} \geq 0$. The output $\lambda_i^* = \Big(-\frac{\theta_i}{\alpha} - q\Big)^+$, and

$$\sum_{i=1}^{n}\lambda_i^* = \sum_{i=l}^{n}\lambda_i^* - kq = 1.$$

Thus, l^* and q again satisfy the KKT conditions.

The runtime of ACTIVESOLVE is determined by $\mathcal{O}(1)$ initialization, an $\mathcal{O}(n\log n)$ sorting call and an $\mathcal{O}(n)$ loop. Thus, the total runtime is $\mathcal{O}(n\log n)$. □

Proof (Proof of Lemma 3). Consider the two cases specified in the update rules for $\nu_L^{(t)}$ and $\nu_U^{(t)}$.

- $\Big\langle \mathbf{C}, \hat{\mathbf{X}}^{(t+1)} \Big\rangle \leq \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)}$: In this case, $\frac{\Delta^{(t+1)}}{\nu_L^{(t+1)}} = \frac{2}{3}\frac{\Delta^{(t)}}{\nu_L^{(t)}} \leq \frac{5}{6}\frac{\Delta^{(t)}}{\nu_L^{(t)}}$
- $\Big\langle \mathbf{C}, \hat{\mathbf{X}}^{(t+1)} \Big\rangle > \nu_L^{(t)} + \frac{2}{3}\Delta^{(t)}$: In this case, $\nu_L^{(t)} + \frac{1}{3}\Delta^{(t)}$ is a lower bound for $\nu^{(t)}$. Assumption (a) and the fact that $\mathbf{A}_i \succeq \mathbf{0}$ imply that $\nu_L^{(t+1)} = \frac{\omega}{1+\frac{\gamma^{(t)}}{q(n,m)}} \leq \nu^*$. Thus,

  $$\nu_L^{(t+1)} \geq \Big(\nu_L^{(t)} + \frac{1}{3}\Delta^{(t)}\Big)\Big(1 - \gamma\frac{\Delta^{(t)}}{\nu_L^{(t)}}\Big) \geq \nu_L^{(t)} + \Big(\frac{1}{3} - \frac{4}{3q(n,m)}\Big)\Delta^{(t)} \geq \nu_L^{(t)} + \frac{\Delta^{(t)}}{6},$$

  where use the fact that $\nu_L^{(t)}\gamma^{(t)} \leq \nu_L^{(t)}\epsilon^{(t)} = \Delta^{(t)}$, $\gamma^{(t)} \leq 1$ and $q(n,m) \geq 8$. Thus, $\frac{\Delta^{(t+1)}}{\nu_L^{(t+1)}} \leq \frac{\nu_U^{(t)} - (\nu_L^{(t)} + \frac{1}{6}\Delta^{(t)})}{\nu^{(t)}} \leq \frac{5}{6}\frac{\Delta^{(t)}}{\nu_L^{(t)}}$.

Since $\epsilon^{(t)} - \epsilon^{(t+1)} \geq \frac{1}{6}\epsilon^{(t)} \geq (1-\alpha)\epsilon^{(t)}$, a simple case analysis establishes that $\gamma^{(t+1)} - \gamma^{(t)} \geq (1-\alpha)\gamma^{(t)}$. This establishes (ii). □

B.2 Nesterov Procedure

NESTEROV PROCEDURE

Inputs: $\mathbf{C}, \mathbf{A}_i, \tau, \omega, \epsilon$

Outputs: $(\mathbf{Y}, \mathbf{y})$

Set $N \leftarrow \left\lceil \tau\omega \left\|\mathbf{A}\right\| \sqrt{\ln(m)} \cdot \frac{1}{\epsilon} \right\rceil$, $\alpha \leftarrow \frac{\epsilon}{2}$, $L \leftarrow \omega^2\tau^2 \left\|\mathbf{A}\right\|^2 \cdot \frac{1}{\epsilon}$, $\mathbf{x}^{(0)} \leftarrow \frac{1}{n+1}\mathbf{1}$.

for $k \leftarrow 0$ **to** N

do

Compute eigendecomposition: $\mathbf{C} - \sum_{i=1}^{m} w_i^{(k)} \mathbf{A}_i = \sum_{j=1}^{n} \theta_j \mathbf{u}_j \mathbf{u}_j^\top$

Set $\boldsymbol{l} = \text{ACTIVESOLVE}\big(\boldsymbol{\theta}, \boldsymbol{\alpha}\big)$

Set $\mathbf{X}^{(k)} = \sum_{j=1}^{n} \lambda_j \mathbf{u}_j \mathbf{u}_j^\top$;

$g_i^{(k)} \leftarrow 1 - \left\langle \mathbf{A}_i, \mathbf{X}^{(k)} \right\rangle, i = 1, \ldots, m;$

$\gamma_i^{(k)} \leftarrow 1 + \log(w_i), i = 1, \ldots, m;$

$\mathbf{y}^{(k)} \leftarrow \text{argmax}_{\mathbf{v} \in \mathcal{V}} \left\{ \left(\mathbf{g}^{(k)} - \frac{1}{\epsilon} L \boldsymbol{\gamma}^{(k)})\right)^\top \mathbf{v} - \frac{1}{\epsilon} L d_v(\mathbf{v}) \right\};$

$\mathbf{z}^{(k)} \leftarrow \text{argmax}_{\mathbf{v} \in \mathcal{V}} \left\{ \left(\sum_{i=0}^{k} \frac{i+1}{2} \mathbf{g}^{(i)}\right)^\top \mathbf{v} - \frac{1}{\epsilon} L d_v(\mathbf{v}) \right\};$

$\mathbf{w}^{(k+1)} = \left(\frac{2}{k+3}\right)\mathbf{z}^{(k)} + \left(\frac{k+1}{k+3}\right)\mathbf{y}^{(k)}.$

return $\left(\mathbf{Y} = \sum_{k=0}^{N} \frac{2(i+1)}{(N+1)(N+2)} \mathbf{X}^{(k)}, \mathbf{y}^{(N)}\right)$.

Fig. 3. Nesterov Procedure