Subgraph Sparsification and Nearly Optimal Ultrasparsifiers

Alexandra Kolla[*]
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
akolla@math.ias.edu

Yury Makarychev
Toyota Technological Institute
at Chicago
Chicago, IL 60637
yury@ttic.edu

Amin Saberi[†]
Management Science and
Engineering Department
Stanford University
Stanford, CA 94305
saberi@stanford.edu

Shang-Hua Teng[‡]
Dept of Computer Science
University of Southern
California
Los Angeles, CA 90089
shanghua@usc.edu

ABSTRACT

We consider a variation of the spectral sparsification problem where we are required to keep a subgraph of the original graph. Formally, given a union of two weighted graphs G and W and an integer k, we are asked to find a k-edge weighted graph W_k such that $G + W_k$ is a good spectral sparsifer of $G + W$. We will refer to this problem as the subgraph (spectral) sparsification. We present a nontrivial condition on G and W such that a good sparsifier exists and give a polynomial-time algorithm to find the sparsifer.

As a significant application of our technique, we show that for each positive integer k, every n-vertex weighted graph has an $(n - 1 + k)$-edge spectral sparsifier with relative condition number at most $\frac{n}{k} \log n \, \tilde{O}(\log \log n)$ where $\tilde{O}()$ hides lower order terms. Our bound nearly settles a question left open by Spielman and Teng about ultrasparsifiers, which is a key component in their nearly linear-time algorithms for solving diagonally dominant symmetric linear systems.

We also present another application of our technique to spectral optimization in which the goal is to maximize the algebraic connectivity of a graph (e.g. turn it into an expander) with a limited number of edges.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms

General Terms

Algorithms, Theory

Keywords

Graph sparsification, ultrasparsifiers, graph Laplacian, approximation algorithm

1. INTRODUCTION

Sparsification is an important technique for designing efficient graph algorithms, especially for dense graphs. Informally, a graph $\tilde{G}$ is a sparsifer of G if they are similar in a particular measure (which is important to the application that one has in mind), and that $\tilde{G}$ has linear or nearly linear number of edges. Various notions of graph approximation and sparsification have been considered in the literature. For example, Chew's [6] spanners (for shortest path planning) have the property that the distance between every pair of vertices in $\tilde{G}$ is approximately the same as in G. Benczur and Karger's [4] cut-sparsifiers (for cuts and flows) have the property that the weight of the boundary of every set of vertices is approximately the same in G as in $\tilde{G}$.

In this paper, we will mainly be interested in the spectral notion of graph similarity introduced by Spielman and Teng [18], [20]: we say that a weighted undirected graph H is a κ-approximation of another G if for all $x \in \mathbf{R}^V$,

$$x^T \mathcal{L}_G x \leq x^T \mathcal{L}_{\tilde{G}} x \leq \kappa x^T \mathcal{L}_G x \qquad (1)$$

where for a weighted undirected graph G, $\mathcal{L}_G$ is the *Laplacian matrix* of G defined as the following: For each i, $\mathcal{L}_G(i,i)$ is equal to the sum of weights of all edges incident to vertex i and for $i \neq j$, $\mathcal{L}_G(i,j) = -w_{i,j}$, where $w_{i,j}$ is the weight on edge (i,j).

In [18, 20], the following spectral sparsification problem is considered. Given a weighted graph $G = (V, E, w)$, an integer $\tilde{m} \leq |E|$, and $\kappa \geq 1$, find a graph $\tilde{G} = \{V, \tilde{E}, \tilde{w}\}$ such that $|\tilde{E}| \leq \tilde{m}$ and $\tilde{G}$ is a κ-approximation of G. We will refer to this problem and its corresponding optimization problem as the Spectral Sparsification. Spielman and Teng showed that every weighted graph has a nearly linear-sized

[*] Research supported by NSF grant CCF-0832797.
[†] Research supported by AHPCRC and NSF
[‡] Research supported by NSF grant CCF-0635102.

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.
Copyright 2010 ACM 978-1-4503-0050-6/10/06 ...$10.00.

spectral sparsifier and gave a nearly linear-time algorithm for computing such a sparsifier. Recently, Batson, Spielman, and Srivastava [3] gave a beautiful, polynomial-time construction to produce a linear-sized spectral sparsifier.

In this paper, we introduce a variation of the spectral sparsification problem which we will refer to as the SUBGRAPH SPARSIFICATION. In our version, we are given two weighted graphs G and W, an integer k and $\kappa \geq 1$. The goal is to find a k-edge weighted graph W_k such that $(G+W_k)$ is a κ-approximation of $(G+W)$. The challenge in the new version of the sparsification problem is that we have to respect part of the graph, i.e., G, and only modify part of graph given in W.

As the main technical contribution of the paper, we give a nontrivial condition about G and W such that a good sparsifier exists. Our proof critically uses the intuition of Batson, Spielman, and Srivastava [3], that uses potential functions that guide an incremental process for selecting the edges of the sparisifier. We will refer to that as as the *BSS process.* We have enhanced their approach with new understanding about subspace sparsification and spectral approximation.

Our challenge, at high level, is the following. The BSS process uses two carefully chosen barriers (see Section 2) so that at each step, all eigenvalues can be kept far enough from these barriers. They have $\Theta(n)$ edges to select. So they consider the entire n-dimensional space and have step size $\Theta(1/n)$ on these barriers.

On the other hand, we can only add k edges, where k can be arbitrarily smaller than n. The addition of each edge can only increase smallest eigenvalue to the second smallest eigenvalue. Therefore the addition of k edges can only improve the subspace defined by the k smallest eigenvalue. Now, the critical part of the argument is that to build a good sparsifier, we need to ensure that the addition of the edges does not increase the high spectra by too much. So in our incremental process, we need to keep track of two subspaces, a fixed one defined by the k smallest eigenvalues and a floating one defined by the higher spectra.

We developed an analysis for performing spectral analysis in the projection of a sequence of two subspaces, which might be interesting on its own right. Our analysis also provide a nice example for using majorization.

Our ability to conduct sparsification on a subgraph enables us to obtain improved results for a few problems on spectral optimization. The first application that we consider is the problem of finding ultrasparsifiers as defined in Spielman and Teng [18]. For parameters $\kappa \geq 1$ and $k \geq 1$, a weighted undirected graph U is a (κ, k)-*ultrasparsifier* of another graph G, if U has at most $n-1+k$ edges, and G is a κ-approximation of U. Ultrasparsifiers are essential in the application of the preconditioning techniques for solving linear systems [18, 20]. It has been shown in [18] that every weighted undirected graph G has a $(\frac{n}{k}\log^{O(1)} n, k)$ ultrasparsifiers, for any k.

As an application of our subgraph sparsification technique, we will show that for every positive integer k, every n-vertex weighted graph has a $(\frac{n}{k}\log n\,\tilde{O}(\log\log n), k)$-ultrasparsifier. Our bound almost settles the previous question about ultrasparsifiers left open by Spielman and Teng.

At high level, our solution to ultrasparsification is quite simple, once we have our subgraph sparsification result. Given a weighted graph$G$, we first construct a low-stretch spanning tree [2, 7, 1] T of G. We then apply an elegant result of Spielman and Woo [21] which states that the sum of the relative condition numbers of $\mathcal{L}_G$ and $\mathcal{L}_T$ is equal to the total stretch to embed G onto T. We will also use Spielman–Woo's tail distribution bound on the number of relative eigenvalues of $\mathcal{L}_G$ and $\mathcal{L}_T$ that are larger than a given parameter.

Algorithmically, we start with the best available [1] low-stretch spanning tree T of G whose total stretch is $n\log n\,\tilde{O}(\log\log n)$. We then consider the subgraph sparsification problem defined by T and $W = \frac{k}{n\log n\tilde{O}(\log\log n)}G$. We apply the structure theorem of Spielman and Woo [21] to show that (T, W) satisfy our condition for subgraph sparsification and apply our result to show that there exists a k-edge weighted graph W_k whose edges are in W such that $T+W_k$ is a spectral approximation of $T+W$. It is then not hard to prove that $T+W_k$ is an a $(\frac{n}{k}\log n\,\tilde{O}(\log\log n), k)$-ultrasparsifier.

As another application of our technique on subgraph sparsification, we consider the following spectral optimization problem studied in [5]: Given a graph G and a parameter k, we are asked to find k edges amongst a set of candidate edges to add to G so as to maximize its algebraic connectivity. Algebraic connectivity has emerged as an important parameter for measuring the robustness and stability of a network and is an essential factor in the performance of various search, routing and information diffusion algorithms.

The spectral optimization considered in this paper is known to be NP-hard [15] and no approximation guarantee for it was known prior to our work. We give an SDP-based approximation algorithm for the problem. Our techniques for subgraph sparsification enable us to develop a novel *rounding* scheme in order to find a combinatorial solution. Since the integrality gap of the SDP is unbounded, our analysis involves adding a separate upper bound, which is roughly the k-th largest eigenvalue of the Laplacian of G to approximate the optimum solution.

2. PRELIMINARIES

Matrix Notation and Definitions. We write $A \succeq 0$ to denote that symmetric matrix A is positive semidefinite; similarly, we write $A \succeq B$ to denote that matrix $A - B$ is positive semidefinite.

We denote the (non-normalized) Laplacian matrix of a graph G by $\mathcal{L}_G$. Recall that $\mathcal{L}_G$ is the matrix with $\mathcal{L}_G(i,i)$ equal to the sum of weights of all edges incident to vertex i, and $\mathcal{L}_G(i,j) = -w_{ij}$ where w_{ij} is the weight of the edge (i,j). For brevity, we write $G_1 \preceq G_2$ to denote $\mathcal{L}_{G_1} \preceq \mathcal{L}_{G_2}$.

For an $n \times n$ matrix A, let $\lambda_{\min}(A) \equiv \lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A) \equiv \lambda_{\max}(A)$ be the set of eigenvalues in the increasing order.

We denote the column space of a matrix A (which is equal to the image or range of the corresponding linear operator) by $\operatorname{Im} A$. We denote the kernel or nullspace of A by $\ker A$. Let $A^\dagger$ be the pseudoinverse of A. If A is symmetric, $A^\dagger$ is also symmetric and $AA^\dagger = A^\dagger A = P_{\operatorname{Im}(A)}$, where $P_{\operatorname{Im}(A)}$ is the orthogonal projection on $\operatorname{Im}(A)$. Let $A \bullet B \equiv \operatorname{tr} A^T B$ be the Frobenius product of matrices A and B. We define the *condition number* of a non-singular matrix A as $\kappa = \|A\|\|A^{-1}\|$, which is equal to $\lambda_{\max}(A)/\lambda_{\min}(A)$ if A is a positive definite matrix. For positive definite matrices A and B with $\operatorname{Im} A = \operatorname{Im} B$, we define the relative condition number as

$$\kappa(A,B) = \max_{x \notin \ker B} \frac{x^T A x}{x^T B x} \cdot \max_{x \notin \ker A} \frac{x^T B x}{x^T A x}.$$

Ultrasparsifiers. We say that a graph is k–ultrasparse if it has at most $n-1+k$ edges. We note that a spanning tree is 0–ultrasparse. A (κ, k)-*ultrasparsifier* of a graph $G = (V, E, w)$ is a k–ultrasparse subgraph of G such that $U \preceq G \preceq \kappa \cdot U$ [18].

3. MATRIX SPARSIFIERS

In this section, we prove an analog of the sparsification theorem of Batson, Spielman, and Srivastava [3].

DEFINITION 3.1 (GRAPH PATCH). *Let G be a (weighted) graph. A graph W on the vertices of G is a (k, T, λ^*)-patch for G if the following properties hold*[1]*,*

1. $\lambda_{k+1}(\mathcal{L}_G \mathcal{L}_{G+W}^\dagger) \equiv \lambda_{k+1}((\mathcal{L}_{G+W}^\dagger)^{1/2} \mathcal{L}_G (\mathcal{L}_{G+W}^\dagger)^{1/2}) \geq \lambda^*$;
2. $\operatorname{tr}(\mathcal{L}_W \mathcal{L}_{G+W}^\dagger) \leq T$.

We prove that for every patch, there exists a "patch sparsifier" supported on $O(k)$ edges. Specifically, we prove the following theorem.

CLAIM 3.2. *Let $W = (V, E_W, \{w_e\}_{e \in E_W})$ be a (k, T, λ^*)-patch for G with edge weights w_e and $N \geq 8k$. Then there is a weighted graph $W_k = (V, E_{W_k}, \{\tilde{w}_e\}_{e \in E_{W_k}})$ with edge weights $\tilde{w}_e$ such that*

1. *W_k has at most N edges; $E_{W_k} \subseteq E_W$.*
2. *$c_1 \min(N/T, 1)\lambda^* \mathcal{L}_{G+W} \preceq \mathcal{L}_{G+W_k} \preceq c_2 \mathcal{L}_{G+W}$, for some absolute constants c_1 and c_2.*
3. *$\sum_{e \in E_{W_k}} \tilde{w}_k \leq \min(1, N/T) \sum_{e \in E_W} w_e$.*

We say that W_k is a patch sparsifier of W with respect to G.

The claim will follow immediately from the following theorem, which is is of independent interest. We will also show another (related) application of this theorem in Section 5.

THEOREM 3.3. *Suppose we are given a positive definite $n \times n$ matrix X and a sequence of matrices $Y_i = v_i v_i^T$ ($i = 1, \ldots, m$) with*

$$X + \sum_{i=1}^m Y_i = M^*,$$

and $\lambda_{\max}(M^*) \leq 1$. Additionally, suppose each matrix Y_i has cost $cost_i \geq 0$ and $\sum_{i=1}^m cost_i = 1$. Let $\lambda^* = \lambda_{k+1}(X)$, and $T = \lceil \operatorname{tr}(M^* - X) \rceil$. Then for every $N > 8k$ there exists a set of weights w_i with $|\{w_i : w_i \neq 0\}| = N$ such that the matrix $M = X + \sum_{i=1}^m w_i Y_i$ satisfies,

$$c_1 \min(N/T, 1) \cdot \lambda^* \cdot \lambda_{\min}(M^*) \leq \lambda_{\min}(M) \leq \lambda_{\max}(M) \leq c_2,$$

where c_1 and c_2 are some absolute constants, and

$$\sum_{i=1}^m w_i cost_i \leq \min(1, N/T).$$

Proof Overview. Our proof closely follows the approach of Batson, Spielman, and Srivastava [3]. We construct matrix M in N steps; at each step we choose an index i and weight w_i and add $w_i Y_i$ to the sum $X + \sum_{i=1}^m w_i Y_i$. Recall that Batson, Spielman, and Srivastava define two "barriers" l and u and maintain the property that all eigenvalues of M lie between l and u. At each step, they increase l and u and update matrix M so that this property still holds. Finally, the ratio between u and l becomes very close to 1, which means that $\lambda_{\min}(M)$ is very close to $\lambda_{\max}(M)$. During this process, they keep track not only of the smallest and largest eigenvalues of M but of all n eigenvalues to avoid accumulation of eigenvalues in neighborhoods of l and u. To this end, they define two potential functions, the lower potential function $\Phi_l(M) = \sum_{i=1}^n \frac{1}{\lambda_i(M) - l}$ and the upper potential function $\Phi^u(M) = \sum_{i=1}^n \frac{1}{u - \lambda_i(M)}$, and then ensure that $\Phi_l(M)$ and $\Phi^u(M)$ do not increase over time. That guarantees that all eigenvalues of M stay far away from l and u.

In our proof, however, we cannot keep an eye on all eigenvalues. After each step, only one eigenvalue increases, and thus we need $\theta(n)$ steps to increase all eigenvalues participating in the definition of $\Phi_l(M)$. But our goal is to "patch" X in roughly k steps. So we focus our attention only on k smallest and T largest eigenvalues.

Let S be the eigenspace of X corresponding to k smallest eigenvalues, and P_S be the projection onto S. We define the lower potential function as follows,

$$\Phi_l(A) = \operatorname{tr}(P_S(A - lI)P_S)^\dagger = \sum_{i=1}^k \frac{1}{\lambda_i(A\big|_S) - l},$$

where $A\big|_S$ denotes the restriction of A to the space S ($A\big|_S$ is a $k \times k$ matrix). Note that the space S is fixed, and the eigenvector corresponding to the smallest eigenvalue will not necessarily lie in S after a few steps. We want to ensure that after N steps,

$$\sum_{i=1}^m w_i Y_i\big|_S \succeq c \min(N/T, 1) \sum_{i=1}^m Y_i\big|_S = c \min(N/T, 1)(M^* - X)\big|_S,$$

or in other words, $\lambda_{\min}(\,(Z(\sum_{i=1}^m w_i Y_i)Z)\big|_S) \geq c \min(N/T, 1)$, where $Z = \big((P_S(M^* - X)P_S)^\dagger\big)^{1/2}$. To this end, we show how to update M and l so that $\Phi_l(Z(\sum_{i=1}^m w_i Y_i)Z)$ does not increase, and l equals $c \min(N/T, 1)$ after N steps. It remains to lower bound $\lambda_{\min}(M)$ in the entire space. We know that all eigenvalues of X (and therefore, of M) in $S^\perp$ are at least λ^*. We show that that together with an upper bound on $\lambda_{\max}(M)$ implies that $\lambda_{\min}(M) \geq c_1 \min(N/T, 1) \cdot \lambda^* \lambda_{\min}(M^*)$ (the product of the lower bounds on $\lambda_{\min}$ in spaces S and $S^\perp$ divided by the upper bound on $\lambda_{\max}$).

Similarly, we amend the definition of the upper potential function. Since we need to bound $\lambda_{\max}$ in the entire space, we cannot restrict $\Phi^u(M)$ to a fixed subspace. For a matrix A, we consider the eigenspace of A corresponding to its largest T eigenvalues. Denote it by $L_A(A)$; denote the projection onto $L(A)$ by $P_{L(A)}$. Then

$$\Phi^u(A) = \operatorname{tr}(P_{L(A)}(uI - A)^{-1} P_{L(A)}) = \operatorname{tr}(P_{L(A)}(uI - A)P_{L(A)})^\dagger = \sum_{i=n-T+1}^N \frac{1}{u - \lambda_i(A)}.$$

Note that both definitions of $\Phi^u(A)$ — in terms of regular inverse and in terms of pseudoinverse — are equivalent

[1] $\lambda_{k+1}(\mathcal{L}_G \mathcal{L}_{G+W}^\dagger) = \lambda_{k+1}((\mathcal{L}_{G+W}^\dagger)^{1/2} \mathcal{L}_G (\mathcal{L}_{G+W}^\dagger)^{1/2})$ since $\lambda_i(AB) = \lambda_i(BA)$ for every two square matrices A and B

since $L(A)$ is an invariant subspace of A. However, $\Phi_l(A)$ is not equal to $\operatorname{tr}(P_S(A-lI)^{-1}P_S)$ in general since S is not necessarily an invariant subspace of A.

Our algorithm and analysis are similar to those of Batson, Spielman, and Srivastava [3]. However, several complications arise because we are controlling eigenvalues in different subspaces and, moreover, one of these subspaces, $L(A)$, is not fixed.

Let us summarize the proof. We construct the matrix M iteratively in N steps. Let $A^{(q)}$ be the matrix and $w_i^{(q)}$ be the weights after q steps. We define an auxiliary matrix $B^{(q)}$ as $Z(A^{(q)}-X)Z$. We have,

$$A^{(q)} = X + \sum_i w_i^{(q)} Y_i;$$
$$B^{(q)} = \sum_i w_i^{(q)} Z Y_i Z = Z(A^{(q)} - X)Z.$$

We will ensure that the following properties hold after each step (for some values of constants l_0, δ_L, u_0, δ_U, ϵ_L, ϵ_U, which we will specify later).

1. $\Phi_{l_0}(B^{(0)}) \le \epsilon_L$ and $\Phi^{u_0}(A^{(0)}) \le \epsilon_U$.

2. Each matrix $A^{(q)}$ and $B^{(q)}$ is obtained by a rank-one update of the previous one:

   $$A^{(q+1)} = A^{(q)} + tY_i,$$
   $$B^{(q+1)} = B^{(q)} + tZY_iZ$$

   for some i.

3. Lower and upper potentials do not increase. Namely, for every $q = 0, 1, \ldots, N$,

   $$\Phi^{u_0+(q+1)\delta_U}(A^{(q+1)}) \le \Phi^{u_0+q\delta_U}(A^{(q)}) \le \epsilon_U \text{ and}$$
   $$\Phi_{l+(q+1)\delta_L}(B^{(q+1)}) \le \Phi_{l_0+q\delta_L}(B^{(q)}) \le \epsilon_L.$$

4. At each step q, $\lambda_{\min}(B^{(q)}\big|_S) > l \equiv l_0 + q\delta_L$ and $\lambda_{\max}(A^{(q)}) < u \equiv u_0 + q\delta_U$. In particular, this condition ensures that all terms in the definitions of upper and lower potentials are positive.

5. At each step q, the total cost is at at most $q/\max(N,T)$: $\sum w_i^{(q)} cost_i \le q/\max(N,T)$.

We present the complete proof in Sections 3.2 and 3.3. In Section 3.2, we first find conditions under which we can update $A^{(q)}$ and u (Lemma 3.10), and $B^{(q)}$ and l (Lemma 3.11). Then we show that both conditions can be simultaneously satisfied (Lemma 3.12). In Section 3.1, we prove several theorems that we need later to deal with a non-fixed subspace $L(A)$. Finally, in Section 3.3, we combine all pieces of the proof together.

3.1 Some Basic Facts about Matrices

3.1.1 Sherman–Morrison Formula

We use the Sherman–Morrison Formula, which describes the behavior of the inverse of a matrix under rank-one updates. We first state the formula for regular inverse [8], and then we show that a similar expression holds for the pseudoinverse.

LEMMA 3.4 (SHERMAN–MORRISON FORMULA). *If A is a nonsingular $n \times n$ matrix and $Y = vv^T$ is a rank-one update, then*

$$(A+Y)^{-1} = A^{-1} - \frac{A^{-1}YA^{-1}}{1 + A^{-1} \bullet Y}$$

LEMMA 3.5. *If A is a symmetric (possibly singular) $n \times n$ matrix, $Y = vv^T$ is a rank-one update, then*

$$(A + PYP)^{\dagger} = A^{\dagger} - \frac{A^{\dagger}YA^{\dagger}}{1 + A^{\dagger} \bullet Y},$$

where P is the orthogonal projection on $\operatorname{Im}(A)$.

PROOF. Let $\bar v = Pv$ and $\bar Y = PYP = \bar v \bar v^T$. Note that $A^{\dagger}YA^{\dagger} = A^{\dagger}\bar YA^{\dagger}$, since $PA^{\dagger} = P$, and

$$A^{\dagger} \bullet \bar Y = \operatorname{tr} A^{\dagger}\bar Y = \operatorname{tr} A^{\dagger}(PYP) = \operatorname{tr}(PA^{\dagger}P)Y = A^{\dagger} \bullet Y.$$

We need to verify that

$$(A+\bar Y)\left(A^{\dagger} - \frac{A^{\dagger}\bar YA^{\dagger}}{1 + A^{\dagger} \bullet \bar Y}\right) = \left(A^{\dagger} - \frac{A^{\dagger}\bar YA^{\dagger}}{1 + A^{\dagger} \bullet \bar Y}\right)(A+\bar Y) = P.$$

Since A is a symmetric matrix, $AA^{\dagger} = A^{\dagger}A = P$. Since $P^2 = P$, $P\bar YP = \bar Y$ and $\bar YA^{\dagger}\bar Y = \bar v\bar v^T A\bar v\bar v^T = \bar v(A \bullet \bar Y)\bar v^T = (A \bullet \bar Y)\bar Y$. We calculate,

$$\begin{aligned}(A+\bar Y)\left(A^{\dagger} - \frac{A^{\dagger}\bar YA^{\dagger}}{1 + A^{\dagger} \bullet \bar Y}\right) &= \\ AA^{\dagger} + \bar YA^{\dagger} &- \frac{\bar P(YA^{\dagger} + \bar YA^{\dagger}\bar YA^{\dagger})}{1 + A^{\dagger} \bullet \bar Y} \\ &= P + \bar YA^{\dagger} - \frac{(1 + A^{\dagger} \bullet \bar Y)\bar YA^{\dagger}}{1 + A^{\dagger} \bullet \bar Y} \\ &= P + \bar YA^{\dagger} - \bar YA^{\dagger} = P.\end{aligned}$$

Similarly, $(A^{\dagger} - \frac{A^{\dagger}\bar YA^{\dagger}}{1+A^{\dagger}\bullet\bar Y})(A + \bar Y) = P$. □

3.1.2 Majorization

LEMMA 3.6 (MAJORIZATION). *For every positive semidefinite matrix A, every projection matrix P, and every $r \in \{1, \ldots, n\}$*

$$\sum_{i=n-r+1}^{n} \lambda_i(A) \ge \sum_{i=n-r+1}^{n} \lambda_i(PAP). \tag{2}$$

In particular, $\lambda_{\max}(A) \ge \lambda_{\max}(PAP)$.

PROOF. Let $e_1, \ldots, e_n$ be an orthonormal eigenbasis of A so that e_i has eigenvalue $\lambda_i(A)$. Similarly, let $\tilde e_1, \ldots, \tilde e_n$ be an orthonormal eigenbasis of PAP so that $\tilde e_i$ has eigenvalue $\lambda_i(PAP)$. Write

$$\tilde e_i = \sum_{j=1}^{n} \langle e_j, \tilde e_i \rangle e_j.$$

Note that if $\lambda_i(PAP) \ne 0$ then $\tilde e_i \in \operatorname{Im}(PAP) \subseteq \operatorname{Im}(P)$ and $P\tilde e_i = \tilde e_i$. Then

$$\lambda_i(PAP) = \tilde e_i^T PAP \tilde e_i = \tilde e_i A \tilde e_i = \sum_{j=1}^{n} \langle e_j, \tilde e_i \rangle^2 \lambda_j(A).$$

If $\lambda_i(PAP) = 0$ then trivially

$$\lambda_i(PAP) = 0 \le \sum_{j=1}^{n} \langle e_j, \tilde e_i \rangle^2 \lambda_j(A).$$

Therefore,

$$\sum_{i=n-r+1}^{n} \lambda_i(PAP) \leq \sum_{i=n-r+1}^{n} \sum_{j=1}^{n} \langle e_j, \tilde{e}_i\rangle^2 \lambda_j(A)$$
$$= \sum_{j=1}^{n} \left(\sum_{i=n-r+1}^{n} \langle e_j, \tilde{e}_i\rangle^2 \right) \lambda_j(A).$$

That is, $\sum_{i=n-r+1}^{n} \lambda_j(PAP)$ is at most the sum of $\lambda_j(A)$ with weights $\sum_{i=n-r+1}^{n} \langle e_j, \tilde{e}_i\rangle^2$. The total weight of all $\lambda_1(A), \ldots, \lambda_n(A)$ is r:

$$\sum_{i=n-r+1}^{n} \underbrace{\sum_{j=1}^{n} \langle e_j, \tilde{e}_i\rangle^2}_{\|\tilde{e}_i\|^2} = \sum_{i=n-r+1}^{n} \|\tilde{e}_i\|^2 = r.$$

The weight of each $\lambda_j(A)$ in the sum is at most 1:

$$\sum_{i=n-r+1}^{n} \langle e_j, \tilde{e}_i\rangle^2 \leq \sum_{i=1}^{n} \langle e_j, \tilde{e}_i\rangle^2 = 1.$$

Therefore, the sum does not exceed the sum of the r largest eigenvalues $\sum_{i=n-r+1}^{n} \lambda_r(A)$. □

Corollary 3.7. *For every positive semidefinite matrix A, every projection matrix P and $u > \lambda_{\max}(A)$, the following inequality holds.*

$$\Phi^u(PAP) = \sum_{i=n-T+1}^{n} \frac{1}{u - \lambda_i(PAP)}$$
$$\leq \sum_{i=n-T+1}^{n} \frac{1}{u - \lambda_i(A)} = \Phi^u(A)$$

Proof. The statement follows from the Karamata Majorization Inequality. The inequality claims that for every two non-increasing sequences that satisfy (2) and for every increasing convex function f,

$$\sum_{i=n-k+1}^{n} f(\lambda_i(A)) \geq \sum_{i=n-k+1}^{n} f(\lambda_i(PAP)).$$

Plugging in $f(x) = \frac{1}{u-x}$ (defined on $(0, u)$), we obtain the desired inequality. □

Lemma 3.8. *Let A be a positive semidefinite matrix such that $A \preceq I_n$. Assume* $\mathrm{Tr}(A) \leq r \in \mathbb{N}$. *Then for every positive semidefinite matrix M, $A \bullet M \leq \sum_{i=N-r+1}^{N} \lambda_i(M)$.*

Proof. By von Neumann's inequality [14],

$$A \bullet M = \mathrm{tr}(AM) \leq \sum_{i=1}^{n} \lambda_i(A)\lambda_i(M).$$

Since $\sum_{i=1}^{n} \lambda_i(A) \leq r$ and all $\lambda_i(A) \leq 1$, we can easily see that the above product achieves its maximum when the largest r eigenvalues of A are 1 and the rest are 0. In this case, we have,

$$A \bullet M \leq \sum_{i=1}^{n} \lambda_i(A)\lambda_i(M) = \sum_{i=n-r+1}^{n} \lambda_i(M).$$

□

As a corollary we get the following result.

Corollary 3.9. *Let X, M^* and T be as in Theorem 3.3. Then for any positive semidefinite matrix U, we have $U \bullet (M^* - X) \leq \sum_{i=n-T+1}^{n} \lambda_i(U)$.*

3.2 Barrier Shifts

In this section, we analyze how we can update matrices $A^{(q)}$ and $B^{(q)}$, and increment barriers l and r so that the upper and lower potentials do not increase. Let us think of $\Phi^u(A)$ as a function of an n^2 dimensional vector (consisting of entries of A). Then in the first approximation $\Phi^{u+\delta_U}(A + tY) \approx \Phi^{u+\delta_U}(A) + tY \bullet U$, where U is the gradient of $\Phi^{u+\delta_U}$ at A (U is an $n \times n$ matrix). Thus the potential function does not increase, $\Phi^{u+\delta_U}(A + tY) \leq \Phi^u(A)$, roughly when $tY \bullet \frac{U}{\Phi^u(A) - \Phi^{u+\delta_U}(A)} \leq 1$.

Similarly, $\Phi_{l+\delta_L}(B + tY) \leq \Phi_l(B)$, roughly when $tY \bullet \frac{L}{\Phi_{l+\delta_L}(B) - \Phi_l(B)} \geq 1$, where L is the gradient of $\Phi_{l+\delta_L}$ at B.

Following [3], we make these statements precise (we need to take into account lower order terms). We define matrices U_A and L_B,

$$U_A = \frac{((u + \delta_U)I - A)^{-2}}{\Phi^u(A) - \Phi^{u+\delta_U}(A)} + ((u + \delta_U)I - A)^{-1};$$

$$L_B = \frac{(P_S(B - (l + \delta_L)I)P_S)^{\dagger 2}}{\Phi_{l+\delta_L}(B) - \Phi_l(B)} - (P_S(B - (l + \delta_L)I)P_S)^{\dagger}$$

Lemma 3.10 (Upper Barrier Shift). *Suppose $\lambda_{max}(A) < u$ and $Y = vv^T$ is a rank-one update. If $U_A \bullet Y \leq \frac{1}{t}$ then $\Phi^{u+\delta_U}(A + tY) \leq \Phi^u(A)$ and $\lambda_{\max}(A + tY) < u + \delta_U$.*

Proof. Let $u' = u + \delta_U$ and $P = P_{L(A+tY)}$. By the Sherman–Morrison formula (Lemma 3.4), we can write the updated potential as:

$$\Phi^{u+\delta_U}(A + tY) = \mathrm{tr}\, P(u'I - A - tY)^{-1}P$$
$$= \mathrm{tr}\, P \left((u'I - A)^{-1} + \frac{t(u'I - A)^{-1}Y(u'I - A)^{-1}}{1 - t(u'I - A)^{-1} \bullet Y} \right) P$$
$$= \mathrm{tr}\, P(u'I - A)^{-1}P + \mathrm{tr}\, \frac{tP(u'I - A)^{-1}Y(u'I - A)^{-1}P}{1 - t(u'I - A)^{-1} \bullet Y}$$
$$\leq \Phi^{u+\delta_U}(PAP) + \frac{t(u'I - A)^{-2} \bullet Y}{1 - t(u'I - A)^{-1} \bullet Y}$$
$$\leq \Phi^{u+\delta_U}(A) + \frac{t(u'I - A)^{-2} \bullet Y}{1 - t(u'I - A)^{-1} \bullet Y}$$
$$= \Phi^u(A) - (\Phi^u(A) - \Phi^{u+\delta_U}(A)) + \frac{(u'I - A)^{-2} \bullet Y}{1/t - (u'I - A)^{-1} \bullet Y}.$$

Here, we used Corollary 3.7 for the inequality on line 4.

Substituting $U_A \bullet Y \leq 1/t$ gives $\Phi^{u+\delta_U}(A + tY) \leq \Phi^u(A)$. The statement about $\lambda_{\max}$ follows from continuity of eigenvalues. □

Lemma 3.11 (Lower Barrier Shift). *Suppose $\lambda_{\min}(B|_S) > l + \delta_L$ and $Y = vv^T$ is a rank-one update. If $L_B \bullet Y \geq 1/t$ then $\Phi_{l+\delta_L}(B + tY) \leq \Phi_l(B)$ and $\lambda_{\min}((B + tY)|_S) > l + \delta_L$.*

Proof. We proceed as in the proof for the upper potential. Let $l' = l + \delta_L$ and $P = P_S$. By the Sherman–Morrison formula for the pseudoinverse (Lemma 3.5), we have:

$$\Phi_{l+\delta_L}(B + tY)$$
$$= \mathrm{tr}(P(B + tY - l'I)P)^{\dagger} = \mathrm{tr}(P(B - l'I)P + tPYP)^{\dagger}$$
$$= \mathrm{tr}(P(B - l'I)P)^{\dagger} - \frac{t\, \mathrm{tr}((P(B - l'I)P)^{\dagger}Y(P(B - l'I)P)^{\dagger})}{1 + t(P(B - l'I)P)^{\dagger} \bullet Y}$$
$$= \Phi_l(B) + (\Phi_{l+\delta_L}(B) - \Phi_l(B)) - \frac{t(P(B - l'I)P)^{\dagger 2} \bullet Y}{1 + t(P(B - l'I)P)^{\dagger} \bullet Y}$$

Note that matrix U_A is positive semidefinite. Rearranging shows that $\Phi_{l+\delta_L}(B+Y) \le \Phi_l(B)$ when $L_A(\pi) \ge 1/t$. It is immediate that $\lambda_{\min}(P_S(A+t\pi\pi^T)P_S) > l+\delta_L$ since $\lambda_{\min}(P_SAP_S) > l+\delta_L$. □

Now we prove that we can choose Y_i and t so that conditions of both lemmas are satisfied.

LEMMA 3.12. *(Both Barriers) If $\Phi^u(A) \le \epsilon_U$ and $\Phi_l(B) \le \epsilon_L$ and $\epsilon_U, \epsilon_L, \delta_U, \delta_L$ satisfy*

$$0 \le \frac{1}{\delta_U} + \epsilon_U + \max(N,T) \le \frac{1}{\delta_L} - \epsilon_L,$$

and X, Y_i, $cost_i$, Z, T and N as in Theorem 3.3, $M^* - X$ is non-singular on S, then there exists i and positive t for which

$$L_B \bullet (ZY_iZ) \ge 1/t \ge U_A \bullet Y_i, \text{ and} \tag{3}$$

$$cost_i \cdot t \le 1/\max(N,T). \tag{4}$$

We will use the following lemma

LEMMA 3.13. *$\sum_{i=1}^m U_A \bullet Y_i \le \frac{1}{\delta_U} + \epsilon_U$ and $\sum_{i=1}^m L_B \bullet (ZY_iZ) \ge \frac{1}{\delta_L} - \epsilon_L$.*

PROOF. 1. We use Corollary 3.9 to bound the Frobenius product of Y_i with each of the two summands in the definition of U_A (note that they are positive semidefinite), we get $\sum_{i=1}^m U_A \bullet Y_i$ equals

$$\begin{aligned}
U_A \bullet \sum_{i=1}^m Y_i &= U_A \bullet (M^* - X)\\
&= \frac{((u+\delta_U)I - A)^{-2}}{\Phi^u(A) - \Phi^{u+\delta_U}(A)} \bullet (M^* - X)\\
&\quad + ((u+\delta_U)I - A)^{-1} \bullet (M^* - X)\\
&\le \sum_{i=n-T+1}^{n} \lambda_i\left(\frac{((u+\delta_U)I - A)^{-2}}{\Phi^u(A) - \Phi^{u+\delta_U}(A)}\right)\\
&\quad + \sum_{i=n-T+1}^{n} \lambda_i\left(((u+\delta_U)I - A)^{-1}\right)\\
&= \frac{\sum_{i=n-T+1}^{n} \frac{1}{(u+\delta_U - \lambda_i(A))^2}}{\Phi^u(A) - \Phi^{u+\delta_U}(A)}\\
&\quad + \sum_{i=n-T+1}^{n} \frac{1}{(u+\delta_U) - \lambda_i(A)}
\end{aligned}$$

Note that the first term is at most $1/\delta_U$, since

$$\begin{aligned}
&\sum_{i=n-T+1}^{n} \frac{1}{(u+\delta_U - \lambda_i(A))^2}\\
&\le \sum_{i=n-T+1}^{n} \frac{1}{(u - \lambda_i(A))(u+\delta_U - \lambda_i(A))}\\
&= \frac{1}{\delta_U} \sum_{i=n-T+1}^{n} \left(\frac{1}{u-\lambda_i(A)} - \frac{1}{(u+\delta_U) - \lambda_i(A)}\right)\\
&= \frac{\Phi^u(A) - \Phi^{u+\delta_U}(A)}{\delta_U}
\end{aligned}$$

and the second term equals $\Phi^{u+\delta_U}(A)$. Thus $\sum_{i=1}^m U_A \bullet Y_i \le \epsilon_U + 1/\delta_U$.

For the second part, let P be the projection on $\mathrm{Im}(M^* - X)$. Since $(M^* - X)$ is non-singular on S, $PP_S = P_S$. We have,

$$\begin{aligned}
&\sum_{i=1}^m L_B \bullet ZY_iZ = L_B \bullet \sum_{i=1}^m ZY_iZ\\
&= L_B \bullet Z(M^* - X)Z = L_B \bullet P\\
&= \mathrm{tr}\left(\frac{(P_S(B-(l+\delta_L)I)P_S)^{\dagger 2}}{\Phi_{l+\delta_L}(B) - \Phi_l(B)} - (P_S(B-(l+\delta_L)I)P_S)^{\dagger}\right)\\
&= \frac{\sum_{i=1}^k (\lambda_i(B|_S) - (l+\delta_L))^{-2}}{\Phi_{l+\delta_L}(B) - \Phi_l(B)} - \sum_{i=1}^k \frac{1}{\lambda_i(B|_S) - (l+\delta_L)}\\
&\ge 1/\delta_L - \epsilon_L,
\end{aligned}$$

where the last line follows from Claim 3.6 in [3]. □

PROOF OF LEMMA 3.12. For the previous lemma, we get:

$$\begin{aligned}
\sum_{i=1}^m (U_A \bullet Y_i + \max(N,T)cost_i) &\le \frac{1}{\delta_U} + \epsilon_U + \max(N,T)\\
&\le L_B \bullet (ZY_iZ).
\end{aligned}$$

Thus for some i, $U_A \bullet Y_i + \max(N,T)cost_i \le L_B \bullet (ZY_iZ)$. Letting $t = (L_B \bullet (ZY_iZ))^{-1}$, we satisfy (3) and (4). □

3.3 Proof of Theorem 3.3

Now we are ready to prove Theorem 3.3. We assume that $M^* - X$ is non-singular on S (which we can ensure by an arbitrary small perturbation).

We start with $A^{(0)} = X$, $B^{(0)} = 0$ and all weights $w_i^{(0)} = 0$. We define parameters as follows,

$$\begin{array}{lll}
\delta_L = 1/(2\max(N,T)), & \epsilon_L = 1/(4\delta_L), & l_0 = -4k\delta_L,\\
\delta_U = 4\delta_L, & \epsilon_U = 1/(4\delta_L), & u_0 = 4T\delta_L + 1,
\end{array}$$

so as to satisfy conditions of Lemma 3.12,

$$\Phi^u(A^{(0)}) = \Phi^u(X) = \sum_{i=1}^T \frac{1}{u_0 - \lambda_{n+1-i}(X)} \le T/(u_0 - 1) = \epsilon_U,$$

$$\Phi_l(B^{(0)}) = \sum_{i=1}^k \frac{1}{0 - l_0} = -k/l_0 = \epsilon_L,$$

$$1/\delta_U + \epsilon_U + \max(N,T) = \frac{3}{2}\max(N,T) = 1/\delta_L - \epsilon_L.$$

Then we iteratively apply Lemma 3.12. At iteration q, we find an index i and a positive t such that $L_{B^{(q)}}(ZY_iZ) \ge 1/t \ge U_{A^{(q)}}(Y_i)$, $cost_i \cdot t \le 1/\max(N,T)$, and increment the weight of matrix Y_i by t: $w_i^{(q+1)} = w_i^{(q)} + t$; update $l = l + \delta_L$ and $u = u + \delta_U$. The total cost increases by at most $1/\max(N,T)$. Finally, after N iterations we obtain matrices $A^{(N)}$ and $B^{(N)}$ with

$$\lambda_{\max}(A^{(N)}) \le u_0 + N\delta_U = 2(N+T)/\max(N,T) + 1 \equiv \theta_{\max}$$

$$\lambda_{\min}(B^{(N)}|_S) \ge l_0 + N\delta_L = (N/2 - 2k)/\max(N,T) \equiv \theta_{\min}.$$

Now consider an arbitrary unit vector v. Let $v = v_S + v_{S^\perp}$, where $v_S \in S$ and $v_{S^\perp} \perp S$. Since $B^{(N)} \succeq \theta_{\min}P_S$ and $v_S \in S$,

$v_S^TA^{(N)}v_S$

$$= v_S^T(X + (P_S(M^* - X)P_S)^{1/2}B^{(N)}(P_S(M^* - X)P_S)^{1/2})v_S$$

$$\geq v_S^T(X+(P_S(M^*-X)P_S)^{1/2}\theta_{\min}P_S(P_S(M^*-X)P_S)^{1/2})v_S$$

$$= \theta_{\min}v_S^T M^* v_S + (1 - \theta_{\min})v_S^T X v_S \geq \theta_{\min}\lambda_{\min}(M^*)\|v_S\|^2.$$

On the other hand, $v_{S^\perp}^T A^{(N)} v_{S^\perp} \leq \theta_{\max}\|v_{S\perp}\|$. Thus from the triangle inequality for the norm induced by $A^{(N)}$, we get

$$(v^T A^{(N)} v)^{1/2} \geq \theta_{\min}^{1/2}\lambda_{\min}(M)^{1/2}\|v_S\| - \theta_{\max}^{1/2}\|v_{S^\perp}\|$$
$$\geq \theta_{\min}^{1/2}\lambda_{\min}(M)^{1/2} - (\theta_{\max}^{1/2} + \theta_{\min}^{1/2}\lambda_{\min}(M)^{1/2})\|v_{S^\perp}\|.$$

Since S is an eigenspace of X corresponding to k smallest eigenvalues,

$$(v^T A^{(N)} v)^{1/2} \geq (v^T X v)^{1/2} \geq (v_{S^\perp}^T X v_{S^\perp})^{1/2} \geq \lambda^{*1/2}\|v_{S^\perp}\|.$$

One of the two bounds above for $(v^T A^{(N)} v)^{1/2}$ increases and the other decreases as $\|v_{S^\perp}\|$ increases. They are equal when

$$\|v_{S^\perp}\| = \frac{\theta_{\min}^{1/2}\lambda_{\min}(M^*)^{1/2}}{\lambda^{*1/2} + \theta_{\max}^{1/2} + \theta_{\min}^{1/2}\lambda_{\min}(M^*)^{1/2}}.$$

Therefore,

$$(v^T A^{(N)} v)^{1/2} \geq \frac{\theta_{\min}^{1/2}\lambda^{*1/2}\lambda_{\min}(M^*)^{1/2}}{\lambda^{*1/2} + \theta_{\max}^{1/2} + \theta_{\min}^{1/2}\lambda_{\min}(M^*)^{1/2}}.$$

We conclude that $\lambda_{\min}(A^{(N)}) = \min_{v:\|v\|=1} v^T A^{(N)} v$ is at least

$$\frac{\theta_{\min}\lambda^*\lambda_{\min}(M^*)}{\left(\lambda^{*1/2} + \theta_{\max}^{1/2} + \theta_{\min}^{1/2}\lambda_{\min}(M^*)^{1/2}\right)^2}.$$

Plugging in the values of parameters, we get the statement of the theorem for $M = A^{(N)}$. The total cost is at most $N/\max(N,T) = \min(1, N/T)$. □

Finally, we prove Claim 3.2.

PROOF CLAIM 3.2. Let $V = \mathrm{Im}(\mathcal{L}_{G+W}) = \ker(\mathcal{L}_{G+W})^\perp$. Let $\mathcal{L}_e$ be the Laplacian of the edge e. Define

$$X = \left((\mathcal{L}_{G+W}^\dagger)^{1/2}\mathcal{L}_G(\mathcal{L}_{G+W}^\dagger)^{1/2}\right)\Big|_V,$$
$$Y_e = w_e\left((\mathcal{L}_{G+W}^\dagger)^{1/2}\mathcal{L}_e(\mathcal{L}_{G+W}^\dagger)^{1/2}\right)\Big|_V,$$
$$cost_e = w_e/\left(\sum\nolimits_{d\in E_W} w_d\right).$$

Since $\mathcal{L}_G + \sum_{e\in E_W}^m w_e\mathcal{L}_e = \mathcal{L}_{G+W}$, we have $X + \sum_{e\in E_W} Y_e = I$. By the definition of the (k, T, λ^*)-patch, $\mathrm{tr}(I - X) \leq T$ and $\lambda^* \leq \lambda^{k+1}(X)$. We apply Theorem 3.3 to matrices X, Y_e and $M^* = I$. We obtain a set of weights ρ_e — supported on at most N edges — such that

$$c_1 \min(N/T, 1)\cdot\lambda^* \leq \lambda_{\min}\left(X + \sum\nolimits_{e\in E_W}\rho_e Y_e\right)$$
$$\leq \lambda_{\max}\left(X + \sum\nolimits_{e\in E_W}\rho_e Y_e\right) \leq c_2,$$

Let $\tilde{w}_e = \rho_e w_e$. Weights $\tilde{w}_i$ define subgraph W_k with at most N edges. It follows that

$$c_1\min(N/T,1)\lambda^*\mathcal{L}_{G+W} \preceq \mathcal{L}_{G+W_k} \preceq c_2\mathcal{L}_{G+W}.$$

The total weight of edges of W_k is

$$\sum_{e\in E_W}\rho_e w_e = (\sum_{e\in E_W}\rho_e cost_e)\sum_{d\in E_W} w_d \leq \min(1, N/T)\sum_{d\in E_W} w_d.$$

□

4. CONSTRUCTING NEARLY-OPTIMAL ULTRASPARSIFIERS

We now apply our subgraph sparsification to build ultrasparsifiers. Recall that a weighted graph U is a (κ, k)-ultrasparsifier of another graph G if $U \preceq G \preceq \kappa\cdot U$ and U has only $n - 1 + k$ edges, where n is the number of vertices in U and G. The main result of this section is the following theorem.

THEOREM 4.1. *For any integer $k > 0$, every graph has an $(\frac{n}{k}\log n\,\tilde{O}(\log\log n), k)$–ultrasparsifier.*

Our basic idea to build a good ultrasparsifier U is quite simple. Without loss of generality, we can assume that G is connected and has $O(n)$ edges. Otherwise given a graph G, we can first find a linear size sparsifier using [3], for each of its connected components, and build a good ultrasparsifier for each component. Because U is only k edges away from a tree, our construction starts with good tree T. As it will be much more clear below, the quality of a tree is measured by its *stretch*, as introduced by Alon, Karp, Peleg and West [2].

Suppose T is a spanning tree of $G = (V, E, w)$. For any edge $e \in E$, let $e_1, \cdots, e_k$ be the edges on the unique path in T connecting the endpoints of e. The *stretch* of e w.r.t. T is given by $\mathrm{st}_T(e) = w(e)(\sum_{i=1}^k \frac{1}{w(e_i)})$. The stretch of the graph G with respect to T is defined by $\mathrm{st}_T(G) = \sum_{e\in E}\mathrm{st}_T(e)$. Our construction will start with a spanning tree with the lowest possible stretch. By [1], we can in polynomial time grow a spanning tree T with

$$\mathrm{st}_T(G) = O(n\log n\log\log n(\log\log\log n)^3).$$

REMARK 4.2. *For the sake of simplicity of the presentation, we will show the construction of ultrasparsifiers with $\Theta(k)$ edges. We note that by choosing the appropriate constants, the number of edges can be made exactly k.*

Let $\kappa = c_1\cdot\mathrm{st}_T(G)/k$ for a sufficiently large constant c_1. Our job is to choose $\Theta(k)$ more weighted edges $\tilde{W}$ and set $U = T + \tilde{W}$ such that $c_2\cdot U \preceq G \preceq \kappa\cdot U$, for a constant c_2. To this end, let $W = (1/(c_3\kappa))\cdot G$, for some constant c_3. Then, $G = c_3\kappa\cdot W \preceq c_3\kappa\cdot(W + T)$. Also, because $T \preceq G$, we have $T + W \preceq (1 + 1/(c_3\kappa))G \preceq c_4\cdot G$, for a constant c_4. Therefore, if we can find a $\Theta(k)$–edge subgraph $\tilde{W}$ of W such that $T + \tilde{W} \preceq \Theta(1)\cdot(T + W)$, we can then build a $n - 1 + \Theta(k)$ edge graph $U = T + \tilde{W}$ satisfying $c_2\cdot U \preceq G \preceq \kappa\cdot U$ (if we choose our constants c_i's carefully).

To apply our subgraph sparsification results to construct $\tilde{W}$, we use the following structure result of Spielman and Woo ([21]: Theorem 2.1 and Corollary 2.2).

LEMMA 4.3 (THEOREM 2.1 IN [21]).

1. *$Tr(\mathcal{L}_T^{\dagger\,1/2}\mathcal{L}_G\mathcal{L}_T^{\dagger\,1/2}) = st_T(G)$.*

2. *For every $t > 0$, the number of eigenvalues of $(\mathcal{L}_T^\dagger)^{1/2}\mathcal{L}_G(\mathcal{L}_T^\dagger)^{1/2}$ greater than t is at most $st_T(G)/t$.*

We now use Lemma 4.3 to prove the following lemma, from which Theorem 4.1 follows directly.

LEMMA 4.4. *W is a $(k, O(k), \Theta(1))$–patch for T.*

PROOF. Let $\lambda_i = \lambda_i((\mathcal{L}_{T+W}^{\dagger})^{1/2}\mathcal{L}_T(\mathcal{L}_{T+W}^{\dagger})^{1/2})$ be the i-th eigenvalue, and y_i be the corresponding eigenvector. Let $x_i = L_{T+W}^{1/2}y_i$. Then,

$$\lambda_i = \lambda_i((\mathcal{L}_{T+W}^{\dagger})^{1/2}\mathcal{L}_T(\mathcal{L}_{T+W}^{\dagger})^{1/2}) = \frac{x_i^T\mathcal{L}_T x_i}{x_i^T\mathcal{L}_T x_i + x_i^T\mathcal{L}_W x_i}$$
$$= \frac{x_i^T\mathcal{L}_T x_i}{x_i^T\mathcal{L}_T x_i + x_i^T\mathcal{L}_G x_i/(c_3\kappa)},$$

implying

$$\frac{x_i^T\mathcal{L}_G x_i}{x_i^T\mathcal{L}_T x_i} = \frac{1-\lambda_i}{\lambda_i}c_3\kappa = \left(\frac{1-\lambda_i}{\lambda_i}\right)c_3c_1\frac{\mathrm{st}_T(G)}{k} = \frac{\mathrm{st}_T(G)}{\frac{k}{c_1c_3}\frac{\lambda_i}{1-\lambda_i}}$$

It follows from the definition of λ_i that $0 \le \lambda_i < 1$. Hence, $(1-\lambda_{i-1})/\lambda_{i-1} \ge (1-\lambda_i)/\lambda_i$. By Courant—Fischer theorem and the property 2 of Lemma 4.3, we have $k \le \frac{k}{c_1c_3}\frac{\lambda_{k+1}}{1-\lambda_{k+1}}$. Therefore, $\lambda_{k+1} \ge \frac{c_1c_3}{1+c_1c_3} = \Theta(1)$. We also have,

$$\mathrm{tr}\left((\mathcal{L}_{T+W}^{\dagger})^{1/2}\mathcal{L}_W(\mathcal{L}_{T+W}^{\dagger})^{1/2}\right) \le \mathrm{tr}\left((\mathcal{L}_T^{\dagger})^{1/2}\mathcal{L}_W(\mathcal{L}_T^{\dagger})^{1/2}\right)$$
$$= \frac{1}{c_3\kappa}\,\mathrm{tr}\left((\mathcal{L}_T^{\dagger})^{1/2}\mathcal{L}_G(\mathcal{L}_T^{\dagger})^{1/2}\right) \le \frac{k}{c_3c_1\mathrm{st}_T(G)}\mathrm{st}_T(G)$$
$$= \frac{k}{c_3c_1} = \Theta(k).$$

We proved that W is a $(k, O(k), \Theta(1))$–patch for T. □

It is easy to see that the parameters of the ultrasparsifiers we obtained are optimal, up to $\tilde{O}(\log n)$.

5. MAXIMIZING ALGEBRAIC CONNECTIVITY BY ADDING FEW EDGES

In this section, we present an approximation algorithm for the following problem: given a graph $G = (V, E_{base})$, a set of candidate edges E_{cand}, and a parameter k, add at most k candidate edges to G so as to maximize its algebraic connectivity. In other words, find a subset $E \subset E_{cand}$ that maximizes $\lambda_2(\mathcal{L}_{G+E})$. The problem was introduced by Ghosh and Boyd [5], who presented a heuristic for it. It is known that the problem is NP-hard [15]. But prior to this work, no approximation algorithm was known for it.

We use two upper bounds for the cost of the combinatorial solution in order to prove an approximation guarantee: one upper bound is the SDP value, λ_{SDP}, and the other is $\lambda_{k+2}(\mathcal{L}_G)$ (see Lemma 5.1). Note that neither of these two bounds are good approximations for the value of the optimum solution by themselves. For instance, if G consists of n isolated vertices, (V, E_{cand}) is an expander, and $k < n$, then the value of the combinatorial solution is 0 but $\lambda_{SDP} \sim k/n$. But the combination of these two bounds lead to a good upper bound for the optimum solution λ_{OPT}.

For clarity and simplicity of exposition, we assume here that (V, E_{base}) and (V, E_{cand}) are bounded degree graphs with the maximum degree Δ. Our algorithm uses a natural semidefinite relaxation that was also used by Ghosh and Boyd [5]. We introduce a variable w_e (the weight of the edge e) for each candidate edge $e \in E_{cand}$; add constraints that all edge weights are between 0 and 1, and the total weight is at most k. Then we require that $\lambda_2(\mathcal{L}_G + \sum_e w_e\mathcal{L}_e) \ge \lambda_{SDP}$ (where $\mathcal{L}_e$ is the Laplacian of the edge e). We do that by adding an SDP constraint $\mathcal{L}_G + \sum_e w_e\mathcal{L}_e \succeq \lambda_{SDP}P_{(1,\dots,1)^\perp}$, where $P_{(1,\dots,1)^\perp}$ is the projection on the space orthogonal to $(1, \dots, 1)$. We get the following SDP relaxation.

$$\begin{aligned} &\text{maximize: } \lambda_{SDP},\\ &\text{subject to: } \mathcal{L}_G + \sum_{e\in E_{cand}} w_e\mathcal{L}_e \succeq \lambda_{SDP}\cdot P_{(1,\dots,1)^\perp},\\ &\qquad\sum_{e\in E_{cand}} w_e \le k,\\ &\qquad 0 \le w_e \le 1 \text{ for every } e \in E_{cand}. \end{aligned}$$

We solve the semidefinite program and obtain the solution $\{w_e\}_{e\in E_{cand}}$. The total weight of all edges is k, however, the number of edges involved, or the support of the solution could be significantly higher than k.

We use our algorithm to *sparsify* the SDP solution using Theorem 3.3. More precisely, we apply Theorem 3.3 with $X = \mathcal{L}_G/(4\Delta)$ and $Y_e = w_i\mathcal{L}_e/(4\Delta)$ restricted to the space $(1, \dots, 1)^\perp$, $N = 8k$, $T = \mathrm{tr}(\sum_e w_e\mathcal{L}_e)/(4\Delta) \le k$ and $cost_i = w_i$ (we divide $\mathcal{L}_G$ and $\mathcal{L}_e$ by 4Δ to ensure that $\lambda_{\max}(X + \sum_e Y_i) \le 1$). We get a set of weights ρ_e supported on at most $8k$ edges s.t.

$$\frac{1}{4\Delta}\lambda_2(\mathcal{L}_G + \sum_e \rho_e w_e\mathcal{L}_e) = \lambda_{\min}(X + \sum_e \rho_e Y_e)$$
$$\ge c\lambda_{k+2}(X)\lambda_{\min}(X + \sum_e Y_e) \ge c\frac{1}{(4\Delta)^2}\lambda_{k+2}(\mathcal{L}_G)\lambda_{SDP}.$$

That is, we obtain a combinatorial weighted solution $\tilde{w}_e = \rho_i w_i$ whose value is at least $c\lambda_{k+2}(\mathcal{L}_G)\lambda_{SDP}/(4\Delta)$ (if $k + 2 > n$, the value is at least $c\lambda_{SDP}$). We next show that $\lambda_{SDP} \ge \lambda_{OPT}$ and $\lambda_{k+2}(G) \ge \lambda_{OPT}$. Therefore, the value of the solution is at least $c\lambda_{OPT}^2/\Delta$.

LEMMA 5.1. *The value of the optimal solution, λ_{OPT}, is at most $\lambda_{k+2}(\mathcal{L}_G)$.*

PROOF. Consider the optimal solution E. Let $\mathcal{L}_E$ be the Laplacian of the graph formed by E. Note that $\mathrm{rank}(\mathcal{L}_E) \le |E| \le k$, therefore, $\dim\ker\mathcal{L}_E \ge n - k$. Let S be the $k+1$-dimensional space spanned by the eigenvectors of $\mathcal{L}_G$ corresponding to $\lambda_2(\mathcal{L}_G), \dots, \lambda_{k+2}(\mathcal{L}_G)$. Since $\dim S + \dim\ker E > n$, spaces S and $\ker\mathcal{L}_E$ have a non-trivial intersection. Choose a unit vector $v \in \ker S \cap \mathcal{L}_E$. We have

$$v(\mathcal{L}_G + \mathcal{L}_E)v^T \le \lambda_{k+2}(\mathcal{L}_G) + 0 = \lambda_{k+2}(\mathcal{L}_G).$$

Also v is orthogonal to the vector $(1, \dots, 1)^\perp$. Therefore,

$$\lambda_{OPT} = \lambda_2(\mathcal{L}_G + \mathcal{L}_E) \le \lambda_{k+2}(\mathcal{L}_G).$$

□

The edges in the support of $\tilde{w}_e$, $E = \{\tilde{w}_e : \tilde{w}_e \ne 0\}$, form a non-weighted combinatorial solution. Since $\lambda_{\max}(\mathcal{L}_X + \sum_e \tilde{w}_e\mathcal{L}_e) = O(\Delta)$, all weights $\tilde{w}_e$ are bounded by $O(\Delta)$, and thus the algebraic connectivity of $G + E$ is at least $c\lambda_{k+2}(\mathcal{L}_G)\lambda_{SDP}/\Delta^2$.

THEOREM 5.2. *There is a polynomial time approximation algorithm that finds a solution of value at least $c\lambda_{OPT}^2/\Delta$ supported on at most $8k$ edges with total weight at most k. If $k \ge n$ the algorithm finds a constant factor approximation.*

We present two corollaries for special instances of the problem.

Corollary 5.3. *If it is possible to make G an expander by adding k edges (and thus $\lambda_{OPT} \sim \Delta$), then the algorithm finds a constant factor approximation.*

Note that if the graph formed by candidate edges is an expander then the value of the following SDP solution $w_e = k/|E_{cand}|$ for each edge $e \in E_{cand}$ is $\Omega(k/n)$, thus $\lambda_{SDP} \geq ck/n$.

Corollary 5.4. *If the graph formed by candidate edges is an expander, then the approximation algorithm from Theorem 5.2 finds a solution of value at least $c\frac{k}{n\Delta}\lambda_{OPT}$.*

Remark 5.5. *It is possible to get rid of the dependence on Δ in Theorem 5.2 and Corollary 5.4 and obtain approximation guarantees of $c\min(\lambda_{OPT}, \lambda_{OPT}^2)$ and $\frac{ck}{n}\lambda_{OPT}$ respectively. We omit the details in this extended abstract.*

Acknowledgments

This research was conducted in part while the authors were visiting Microsoft Research New England (and the second author was a postdoctoral researcher at MSR-NE). We would like to thank Microsoft Research for their hospitality.

We are grateful to Dan Spielman for his insightful help in our work on ultrasparsifers.

6. REFERENCES

[1] I. Abraham, Y. Bartal, and O. Neiman. Nearly tight low stretch spanning trees. In *Proceedings of the 49th annual Symposium on Foundations of Computer Science,* pp. 781–790, 2008.

[2] N. Alon, R. Karp, D. Peleg, and D. West. A Graph-Theoretic Game and its Application to the k-Server Problem, In *SIAM J. Comput.* vol. 24-1, pp. 78–100, 1995.

[3] J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan Sparsifiers. In *Proceedings of the 50th annual Symposium on Foundations of Computer Science,* pp. 255–262, 2009.

[4] A. A. Benczur and D. R. Karger. Approximating $s - t$ minimum cuts in $O(n^2)$ time. In *Proceedings of the 28th annual Symposium on the Theory of Computing* pp. 47–55, 1996.

[5] A. Ghosh and S. Boyd. Growing well–connected graphs. In *Proceedings of the 45th conference on Decision and Control,* 2006.

[6] P. Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the second annual symposium on Computational geometry,* pp. 169–177, ACM, 1986.

[7] M. Elkin, Y. Emek, D. A. Spielman and S. Teng. Lower-stretch spanning trees In *Proceedings of the 37th annual ACM symposium on Theory of Computing*, pp. 494–503, 2005.

[8] G. H. Golub and C. F. Van Loan. Matrix Computations, 3rd. Edition. The Johns Hopkins University Press, Baltimore, MD, 1996.

[9] A. Ghosh. Designing Well-Connected Networks, Ph.D. Dissertation, Stanford University, 2006.

[10] A. Ghosh, S. Boyd, and A. Saberi. Minimizing Effective Resistance of a Graph. In SIAM Review, problems and techniques section, 50(1):37-66, 2008.

[11] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. In *Combinatorica,* 8(3):261–277, 1988.

[12] D. G. Luenberger. A Combined Penalty Function and Gradient Projection Method for Nonlinear Programming.

[13] G. A. Margulis. Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. In *Problems of Information Transmission,* 24(1):39–46, July 1988.

[14] L. Mirsky. A trace inequality of John von Neumann. *Monats. Math.* 79(4):303–306, 1975.

[15] D. Mosk–Aoyama. Maximum algebraic connectivity augmentation is NP-hard. *Operations Research Letters,* vol. 36(6), Nov. 2008, 677–679.

[16] D. A. Spielman. Personal Communication.

[17] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pp. 563–568, 2008. Full version available at http://arXiv.org/abs/0803.0929.

[18] D. A. Spielman, and S. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th annual ACM Symposium on Theory of Computing*, pp. 81–90, 2004.

[19] D. A. Spielman and S. Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. CoRR, abs/cs/0607105, 2008. Available at http://www.arxiv.org/abs/cs.NA/0607105.

[20] D. A. Spielman and S. Teng. Spectral sparsification of graphs. CoRR, abs/0808.4134, 2008. Available at http://arxiv.org/abs/0808.4134.

[21] D. A. Spielman and J. O. Woo. A note on preconditioning with low stretch spanning trees. http://arxiv.org/abs/0903.2816.