Spectral Sparsification of Undirected Graphs

Ilya Razenshteyn
ilyaraz@mit.edu

December 11, 2012

Abstract

A cut sparsifier for an undirected graph is a sparse graph, which preserves up to multiplicative $(1 \pm \varepsilon)$ factor the values of all cuts. It was shown in [BK96] how to construct cut sparsifiers with $O(n \log n/\varepsilon^2)$ edges in near-linear time. In [ST11] Spielman and Teng considered much stronger notion: spectral sparsification. A spectral sparsifier is a sparse graph, which (almost) preserves the Laplacian of the original graph.

This report (which was written as a final project for the MIT course "Advanced Algorithms") surveys two constructions of spectral sparsifiers.

The first construction is by Spielman and Srivastava [SS11]. They show how to build spectral sparsifiers with $O(n \log n/\varepsilon^2)$ edges in near-linear time. The second construction is by Batson, Spielman and Srivastava [BSS09]. While having fewer edges ($O(n/\varepsilon^2)$ instead of $O(n \log n/\varepsilon^2)$), these sparsifiers require much more (but still polynomial) time to construct.

1 Introduction

One recurring topic in Graph Theory is how to approximate a given graph with much sparser one while preserving certain properties. A very useful and clean notion of *cut sparsification* was introduced in [BK96] by Benczúr and Karger. They proved the following result.

Theorem 1 ([BK96]). *Let $\varepsilon > 0$ be some sufficiently small parameter. For every undirected weighted graph G there exists another undirected weighted graph $\tilde{G}$ with the same set of vertices and only $O(n \log n/\varepsilon^2)$ edges such that every cut value in $\tilde{G}$ is within $(1 \pm \varepsilon)$ of the corresponding value in G. Moreover, $\tilde{G}$ can be found in time $O(m \log^3 n)$ with high probability.*

Two obvious applications of Theorem 1 are the following:

- by sparsifying graph and then running the algorithm of Goldberg and Rao [GR98] we can find $(1+\varepsilon)$-approximations to the maximum (s,t)-flow and the minimum

(s,t)-cut in time $O(m \log^3 n + n^{3/2} \log n \log(nU)/\varepsilon^3)$, where U is an upper bound on the capacities in our network;

- by sparsifying graph and then running the algorithm of Klein, Stein and Tardos [KST90] we can find an $O(\log n)$-approximation to the sparsest cut in time $O(n^2 \log^3 n)$.

The proof of Theorem 1 goes along the following lines. The first naive idea is to employ uniform random sampling (in case of unweighted graphs). It turns out that one can prove that it works, provided that the minimum cut is large enough. To overcome this problem with small cuts one has to sample edges with probabilities inversely proportional to their *strong connectivities* (for the precise definition see [BK96], but informally speaking, strong connectivities show how well connected the ends of edges are).

In [ST11] Spielman and Teng considered a singificantly stronger notion of sparsification: the so called *spectral sparsification*. To define it we need a notion of *graph Laplacian*.

Definition 1. *Let $G = (V, E, w)$ be an undirected weighted graph. Then the following quadratic form is called* Laplacian *of G:*

$$\mathcal{L}_G(x) = \sum_{(u,v) \in E} w(u,v)(x_u - x_v)^2,$$

for $x \in \mathbb{R}^V$.

Definition 2. *We say that $\tilde{G}$ is a spectral sparsifier of G, if for every $x \in \mathbb{R}^n$*

$$\mathcal{L}_{\tilde{G}}(x) \in (1 \pm \varepsilon)\mathcal{L}_G(x).$$

To see why the notion of spectral sparsification is stronger than that of cut sparsification it is sufficient to observe that $\mathcal{L}_G(\mathbf{1}_S)$ is exactly equal to the cut value of $(S, \bar{S})$. But we require much more: to preserve $\mathcal{L}_G(x)$ for every x, not only for the indicators.

We will survey two constructions of spectral sparsifiers. The first is from [SS11], which gives sparsifiers with $O(n \log n/\varepsilon^2)$ edges matching the bound of Theorem 1. Moreover, these sparsifiers can be found in near-linear time. The second construction is from [BSS09]: it gives sparsifiers with $O(n/\varepsilon^2)$ edges, but the algorithm for building them is much slower (nevertheless, even the existance of near-linear-sized spectral sparsifiers is higly non-trivial).

Let us mention two applications of spectral sparsification.

First, since spectral sparsifiers approximately preserve eigenvalues of Laplacians, one can look at sparsifiers of the complete graph K_n as expander graphs [HLW06]. Technically speaking, this analogy is quite vague, because expanders are constant-degree regular graphs, on the other hand, even the best construction of spectral sparsifiers from [BSS09], despite giving a constant *average* degree, does not provide graphs with constant maximum degree. Nevetheless, it is proved in [BSS09] that spectral sparsifiers of K_n share several properties with constant-degree expanders.

Second, spectral sparsification can be used to prove the existence of good *approximate John decompositions* (see [Bal97] for an excellent introduction to Convex Geometry and [Nao11] for a great exposition of geometric applications of [BSS09]).

Third, spectral sparsification can be used for dimension reduction in ℓ_1-spaces. The classic theorem of Johnson–Lindenstrauss [DG03] says that if we have n vectors in $\mathbb{R}^d$, then we can map them into $\mathbb{R}^{O(\log n/\varepsilon^2)}$ so that to $(1\pm\varepsilon)$-preserve their pairwise Euclidean distances. This theorem has immense number of applications (in particular, we will use it to find spectral sparsifiers quickly). It would be great to extend it to ℓ_1 ($\|x\|_1 := \sum_i |x_i|$), but in the series of papers [BC05], [LN04], [ACNN11], [Reg11] it was proved that one needs dimension to be essentially linear in n in this case. Nevertheless, in [Tal90] it was proved that one can reduce dimension to $O(n\log n/\varepsilon^2)$. Using spectral sparsification one can sharpen this bound to $O(n/\varepsilon^2)$ (see [Nao11] for the exposition).

1.1 Organization of the report

Our report is organized as follows. First, in Section 2 we review the definitions and the results we will need further. Then, in Section 3 we show how to reduce the problem of Laplacian sparsification to the following clean linear-algebraic problem.

Problem 1 (Sparsification of a decomposition of identity). *Let $q_1, q_2, \dots, q_m$ be the set of vectors from $\mathbb{R}^n$ such that*

$$\sum_{i=1}^{m} q_i q_i^t = I. \tag{1}$$

Let $\varepsilon > 0$ be some parameter. We want to sparsify (1): find $N \ll m$ and the numbers $\alpha_1, \alpha_2, \dots, \alpha_m \geq 0$ such that there are at most N non-zero α's and all the eigenvalues of $\sum_i \alpha_i q_i q_i^t$ are within $[1-\varepsilon; 1+\varepsilon]$.

What is the smallest $N = N(n,\varepsilon)$ can we guarantee?

In Section 4 we show the solution for this problem from [SS11] that is based on sampling (it turns out that one should sample q_i's with probabilities proportional to $\|q_i\|_2^2$) and gives $N = O(n\log n/\varepsilon^2)$, in Section 5 we show the "deterministic" approach of [BSS09] that gives $N = O(n/\varepsilon^2)$.

For clarity we assume that we want to sparsify an unweighted connected graph.

1.2 Notation

- e_i stands for the i-th basis vector in $\mathbb{R}^n$;
- I stands for the identity matrix;
- I_V stands for the operator that maps vectors from V to themselves, and maps everything from $V^\perp$ to zero;
- $\lambda_1(A), \lambda_2(A), \dots, \lambda_n(A)$ stand for the eigenvalues of A (in no particular order);

- M stands for the incidence matrix of our initial graph: this is $m \times n$ matrix with rows corresponding to edges. A row that corresponds to an edge (v, w) is equal to $e_v - e_w$.

2 Preliminaries

In this section we introduce several definitions and results in linear algebra we will need further.

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix.

Theorem 2 (Eigendecomposition). *All the eigenvalues of A are real and there exists an orthonormal basis of $\mathbb{R}^n$ that consists of A's eigenvectors.*

We can write

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^t,$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A, and $u_1, u_2, \ldots, u_n$ are the corresponding eigenvectors that form an orthonormal basis of $\mathbb{R}^n$. This decomposition is called eigendecomposition *of A.*

We can use the notion of eigendecomposition to introduce a pseudoinverse of A.

Definition 3 (Pseudoinverse). *A* pseudoinverse *of A is the following matrix:*

$$A^+ := \sum_{i : \lambda_i \neq 0} \frac{1}{\lambda_i} u_i u_i^t.$$

Clearly, if A is non-degenerate (equivalently, $\lambda_i \neq 0$ for every i), then $A^+ = A^{-1}$. On the other hand, for every symmetric A

$$AA^+ = A^+ A = I_{\mathrm{im}(A)},$$

where $I_{\mathrm{im}(A)}$ is a linear operator that maps vectors from $\ker(A)$ to zero and maps vectors from $\mathrm{im}(A)$ to themselves (this definition is correct, because we can consider a decomposition $\mathbb{R}^n = \ker(A) \oplus \mathrm{im}(A)$).

If a symmetric matrix is positive semi-definite (equivalently, all the eigenvalues are non-negative), then we can consider its square root.

Definition 4 (Square Root of a PSD Matrix).

$$A^{1/2} := \sum_i \sqrt{\lambda_i} u_i u_i^t.$$

Clearly, $A^{1/2} A^{1/2} = A$.

During the presentation of spectral sparsification algorithms we will use the following matrix norm crucially.

Definition 5 (Spectral Norm).

$$\|A\|_2 := \max_x \frac{\|Ax\|_2}{\|x\|_2}.$$

For symmetric matrices spectral norm is just the maximum absolute value of an eigenvalue. If I is the identity matrix, then $\|A - I\|_2 \leq \varepsilon$ iff all the eigenvalues of A lie within $[1-\varepsilon; 1+\varepsilon]$.

We will need the following "law of large numbers" for matrix-valued random variables proved by Rudelson and Vershynin in [Rud96], [RV07].

Theorem 3. *Let y be an n-dimensional vector-valued random variable. Suppose that $\|y\|_2 \leq M$ with probability 1 and $\|\mathbf{E}[yy^t]\|_2 \leq 1$. Then,*

$$\mathbf{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N} y_i y_i^t - \mathbf{E}[y_i y_i^t]\right\|_2\right] \leq O\left(M\sqrt{\frac{\log N}{N}}\right), \tag{2}$$

where y_i – independent copies of y.

This theorem quantifies the speed of convergence of $\sum_{i=1}^{N} y_i y_i^t/N$ to $\mathbf{E}[yy^t]$.

We need the following peculiar identity for the so-called rank-1 updates.

Theorem 4 (Sherman-Morrison formula, [SM50]). *If A is square and invertible and $v^t A^{-1} u \neq -1$, then*

$$(A+uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^tA^{-1}}{1+v^tA^{-1}u}.$$

An update of the form "$A := A + uv^t$" is called rank-1, because the rank of uv^t is equal to one.

3 Reduction to Sparsification of a Decomposition of Identity

In this section we reduce our original problem of the Laplacian sparsification to Problem 1.

If G is an undirected unweighted graph, then, clearly

$$\mathcal{L}_G(x) = x^t L x,$$

where L_{vv} equals to the degree of v, $L_{vw} = -1$, if $(v,w) \in E$, and $L_{vw} = 0$, otherwise.

We leave the following Lemma as an exercise.

Lemma 1. *We have $\mathbf{1} \in \ker(L)$. Moreover, $\ker(L) = \langle \mathbf{1} \rangle$ iff G is connected. In this case $\mathrm{im}(L) = \langle \mathbf{1} \rangle^{\perp}$*

We have the following natural rank-1 decomposition of L:

$$L = \sum_{(v,w)\in E} (e_v - e_w)(e_v - e_w)^t. \tag{3}$$

Now let us understand what does it mean for a graph $\tilde{G}$ to approximate G in the sense of Laplacian. Let us denote $\tilde{L}$ the Laplacian of G. We want that for every $x \in \mathbb{R}^n$

$$x^t \tilde{L} x \in (1 \pm \varepsilon) x^t L x. \tag{4}$$

Clearly, if $x \in \ker(L)$, then there is nothing to worry about, since both sides of (4) are equal to zero. So, it is only left to ensure that (4) holds for $x \in \mathrm{im}(L)$. In this case we can write $x = (L^+)^{1/2} y$ for $y = L^{1/2} x \in \mathrm{im}(L)$. Thus, we want that for every $y \in \mathrm{im}(L)$

$$y^t (L^+)^{1/2} \tilde{L} (L^+)^{1/2} y \in (1 \pm \varepsilon) y^t y.$$

It is equivalent to

$$\|(L^+)^{1/2} \tilde{L} (L^+)^{1/2} - I_{\mathrm{im}(L)}\|_2 \leq \varepsilon.$$

From the definitions of pseudoinverse and the square root of a PSD matrix we have

$$I_{\mathrm{im}(L)} = (L^+)^{1/2} L (L^+)^{1/2} = \sum_{(v,w) \in E} (L^+)^{1/2} (e_v - e_w)(e_v - e_w)^t (L^+)^{1/2}. \tag{5}$$

Since

$$\tilde{L} = \sum_{(v,w) \in E} \gamma(v,w)(e_v - e_w)(e_v - e_w)^t,$$

where $\gamma(v,w)$ is a weight we assign to the edge (v,w) in $\tilde{G}$, we have

$$(L^+)^{1/2} \tilde{L} (L^+)^{1/2} = \sum_{(v,w) \in E} \gamma(v,w) (L^+)^{1/2} (e_v - e_w)(e_v - e_w)^t (L^+)^{1/2}. \tag{6}$$

Comparing (5) and (6) we can observe that we are facing an instance of Problem 1: first, we pass from the whole $\mathbb{R}^n$ to a subspace $\langle \mathbf{1} \rangle^\perp$, then we observe that (5) is a decomposition of the identity in this subspace with $q_{vw} = (L^+)^{1/2}(e_v - e_w)$, and (6) is exactly a sparsification of this decomposition.

4 Sparsification by Sampling

4.1 Existential Argument

In this section we present the argument from [SS11], which shows the existence solution for Problem 1 with $N = O(n \log n / \varepsilon^2)$. Suppose we want to sparsify a decomposition of indentity with independent sampling. What probabilities for the terms $q_i q_i^t$ should we choose? Let us denote them by $p_1, p_2, \ldots, p_m$. Let us assume that we make N independent samples $i_1, i_2, \ldots, i_m$ from $[m]$ with probabilities p_i and form a sum

$$X = \frac{1}{N} \sum_{j=1}^{N} \frac{q_{i_j} q_{i_j}^t}{p_{i_j}}.$$

Clearly, $\mathbf{E}[X] = I$. Now let us use Theorem 3 to understand what we should take N equal to. We have

$$M = \max_i \frac{\|q_i\|_2}{\sqrt{p_i}}.$$

In order to make M as small as possible we should take p_i proportional to $\|q_i\|_2^2$. Thus, we have

$$M = \sqrt{\sum_{i=1}^m \|q_i\|_2^2} = \sqrt{\sum_{i=1}^m \mathrm{tr}(q_i q_i^t)} = \sqrt{\mathrm{tr}\left(\sum_{i=1}^m q_i q_i^t\right)} = \sqrt{n}.$$

Now we just invoke Theorem 3 and get $N = O(n \log n/\varepsilon^2)$ immediately.

Remark: suprisingly enough, it turns out that for the graph Laplacian case $\|q_{vw}\|_2^2$ is equal to the effective resistance of an edge (v, w). This intuition is heavily used in several recent developments. Futhermore, there is a strong similarity between this sparsification procedure and that of from [BK96]: effective resistances reflect edge connectivities.

4.2 Near-linear Time Algorithm (a High-Level Idea)

Despite it is proved in the previous section that there exists a sparsifier of size $O(n \log n/\varepsilon^2)$, we still do not have a fast algorithm for finding one. Indeed, in order to sample edges we must know $\|q_{vw}\|_2^2$ for every edge $(v, w) \in E$. Let us recall that

$$\|q_{vw}\|_2^2 = \|(L^+)^{1/2}(e_v - e_w)\|_2^2 = (e_v - e_w)^t L^+ (e_v - e_w).$$

Unfortunately, we do not know how to compute L^+ quickly, but luckily it is sufficient for us to compute $\|q_{vw}\|_2^2$'s approximately (for instance, any $O(1)$-approximation is sufficient). Let us observe that $L = M^t M$, where M is the incidence matrix of the graph ($m \times n$ matrix, where rows correspond to edges, a row that corresponds to (v, w) is equal to $e_v - e_w$). Now let us rewrite

$$\|q_{vw}\|_2^2 = (e_v - e_w)^t L^+ (e_v - e_w) = \|ML^+(e_v - e_w)\|_2^2,$$

so we are interested in pairwise ℓ_2-distances between the columns of ML^+. One could compute the rows of ML^+ by multiplying m rows of M by L^+. Amazingly, there exists a way to do it in near-linear time approximately (see [KMP10] for more details).

Theorem 5 ([KMP10]). *There is an algorithm that given $x \in \mathbb{R}^n$ computes $y \in \mathbb{R}^n$ such that*

$$\|y - L^+ x\|_L \le \varepsilon \|L^+ x\|_L,$$

where $\|z\|_L := \sqrt{z^t L z}$, in time $\tilde{O}(m)$.

Unfortunately, this is still not sufficient for our purposes, because there are m rows and we get $\tilde{O}(m^2)$ time in total.

Here comes into the play the celebrated result of Johnson and Lindenstrauss.

Theorem 6 ([DG03]). *Let $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ be arbitrary vectors. If Q is an $O(\log n/\varepsilon^2)\times d$ matrix with (properly normalized) i.i.d. Gaussian entries, then*

$$\mathbf{Pr}[\ \forall i,j\ \|Q(x_i - x_j)\|_2 \in (1\pm\varepsilon)\|x_i - x_j\|_2] \geq 0.99.$$

So, it is sufficient to compute a matrix QML^+. We can compute QM in time $O(m\log n)$, and then compute QML^+ by invoking the solver from [KMP10] $O(\log n)$ times, thus getting the overall $\tilde{O}(m)$ running time.

Remark: many more details has to be filled here. For instance, what accuracy do we need, when invoking the solver from [KMP10]? For complete treatment see [SS11].

5 Iterative Sparsification

To improve the bound $O(n\log n/\varepsilon^2)$ one should switch from (independent) edge sampling to something else. The reason is that in Theorem 3 the factor $\sqrt{\log N}$ is essential: consider the case, when y is a random basis vector. Then, clearly, we need at least $\Omega(n\log n)$ samples to choose each vector at least once. Thus, no matter what probabilities we are using for independent edge sampling, we can not overcome the bound $O(n\log n)$. So, to get the optimal $O(n/\varepsilon^2)$ bound one has to use "deterministic" methods.

We will use the iterative approach: on each step we keep track of a current matrix A and perform a rank-1 update of it — set $A := A + \alpha q_j q_j^t$ for some carefully chosen $\alpha \geq 0$ and $j \in [m]$. Our goal is that the eigenvalues of A are within a multiplicative factor $(1\pm\varepsilon)$ of each other.

Since $\mathrm{tr}(A + vv^t) = \mathrm{tr}(A) + \|v\|_2^2$, the eigenvalues of A increase by $\|v\|_2^2/n$ "on average". If all the eigenvalues always increase by the same amount, we are done: after some time all of them will be close to each other (in multiplicative sense). Unfortunately, we are not guaranteed to have "uniform" increases.

Here comes the brilliant idea of Batson, Spielman and Srivastava [BSS09]. They consider the following "barrier" potential functions, which will allow us to control the pace of eigenvalues' growth.

Definition 6. *Let A be a square matrix. Then*

$$\begin{aligned} L_x(A) &= \mathrm{tr}(A - xI)^{-1} = \sum_i \frac{1}{\lambda_i(A) - x} \\ U_x(A) &= \mathrm{tr}(xI - A)^{-1} = \sum_i \frac{1}{x - \lambda_i(A)} \end{aligned}$$

The idea of [BSS09] is to maintain two barriers l and u such that $L_l(A)$ and $U_u(A)$ are small, and all the eigenvalues are within $(l; u)$, which will allow us to conclude that we can perform a rank-1 update to A and increase l and u by certain amounts.

More formally, let $\delta_L, \delta_U, \varepsilon_L, \varepsilon_U, l_0, u_0$ be some parameters, moreover, $\delta_L, \delta_U, \varepsilon_L, \varepsilon_U$ are positive. We build a sequence of matrices $A_0, A_1, \ldots, A_N$ such that

- $A_0 = 0$;
- $A_i = A_{i-1} + \alpha q_j q_j^t$ for some $\alpha > 0$ and $j \in [m]$;
- $L_{l_i}(A_i) \leq L_{l_{i-1}}(A_{i-1}) \leq \varepsilon_L$, where $l_i = l_0 + i \cdot \delta_L$;
- $U_{u_i}(A_i) \leq U_{u_{i-1}}(A_{i-1}) \leq \varepsilon_U$, where $u_i = u_0 + i \cdot \delta_U$;
- all the eigenvalues of A_i lie within $(l_i; u_i)$.

We will show that for these invariants to hold it is sufficient for the following relations to hold.

- $\varepsilon_L = -n/l_0$, $\varepsilon_U = n/u_0$;
- $1/\delta_U + \varepsilon_U \leq 1/\delta_L - \varepsilon_L$;
- $\varepsilon_L \leq 1/\delta_L$.

One can see that we can take $\varepsilon_L = \Theta(\varepsilon)$, $\varepsilon_U = \Theta(\varepsilon)$, $\delta_L = 1$, $\delta_U = 1 + \Theta(\varepsilon)$, $l_0 = -\Theta(n/\varepsilon)$, $u_0 = \Theta(n/\varepsilon)$ and get a desired sparsifier after $O(n/\varepsilon^2)$ steps. The reason is that after N steps all the eigenvalues will be within $[l_0 + N \cdot \delta_L; r_0 + N \cdot \delta_R] = [-\Theta(n/\varepsilon) + N; \Theta(n/\varepsilon) + N(1 + \Theta(\varepsilon))]$. So, after $O(n/\varepsilon^2)$ steps all the eigenvalues are within $[N(1-\varepsilon); N(1+\varepsilon)]$. After a proper normalization of A_N we are done.

Suppose we want to set $A_i = A_{i-1} + \alpha v v^t$. As a simple consequence of Theorem 4 we get the following formulae.

$$\begin{aligned} L_{l_i}(A_i) &= L_{l_i}(A_{i-1}) - \frac{v^t (A_{i-1} - l_i I)^{-2} v}{1/\alpha + v^t (A_{i-1} - l_i I)^{-1} v} \\ U_{u_i}(A_i) &= U_{u_i}(A_{i-1}) + \frac{v^t (u_i I - A_{i-1})^{-2} v}{1/\alpha - v^t (u_i I - A_{i-1})^{-1} v} \end{aligned}$$

Using these formulae it is straightforward to prove the following Lemmas.

Lemma 2. *$L_{l_i}(A_i) \leq L_{l_{i-1}}(A_{i-1})$ iff $0 < \frac{1}{\alpha} \leq \sigma(v)$, where*

$$\sigma(v) := -v^t (A_{i-1} - l_i I)^{-1} v + \frac{v^t (A_{i-1} - l_i I)^{-2} v}{L_{l_i}(A_{i-1}) - L_{l_{i-1}}(A_{i-1})}.$$

Lemma 3. *$U_{u_i}(A_i) \leq U_{u_{i-1}}(A_{i-1})$ iff $\frac{1}{\alpha} \geq \tau(v)$, where*

$$\tau(v) := v^t (u_i I - A_{i-1})^{-1} v + \frac{v^t (u_i I - A_{i-1})^{-2} v}{U_{u_{i-1}}(A_{i-1}) - U_{u_i}(A_{i-1})}.$$

As a result we prove the following Theorem.

Theorem 7. *If $\tau(v) \leq \sigma(v)$, then there exists an $\alpha > 0$ such that if we set $A_i := A_{i-1} + \alpha v v^t$, then*

- *$L_{l_i}(A_i) \leq L_{l_{i-1}}(A_{i-1})$;*
- *$U_{u_i}(A_i) \leq U_{u_{i-1}}(A_{i-1})$;*

- *all the eigenvalues of A_i are within $(l_i; u_i)$.*

Proof. The first and the second statement are trivial corollaries of Lemma 2 and 3. Thus, it is left to argue about eigenvalues. Clearly, all the eigenvalues of A_{i-1} (and, thus, of A_i) are at least $l_{i-1} + 1/\varepsilon_L \le l_{i-1} + \delta_L = l_i$, because $L_{l_{i-1}}(A_{i-1}) \le \varepsilon_L$. Since $L_{l_i}(A_i)$ is finite, we have that all eigenvalues of A_i are strictly greater than l_i. Now let us argue about the upper bound. From Lemma 3 it follows that if α is sufficiently small, then $U_{u_i}(A_i)$ is finite (and even small). If the largest eigenvalue of A_i is at least u_i, it means that for some small α $U_{u_i}(A_i)$ blows up, thus we have a contradiction. □

Now it is left to prove that there exists $j \in [m]$ such that $\tau(q_j) \le \sigma(q_j)$. For this we prove the following theorem.

Theorem 8. *If $1/\delta_U + \varepsilon_U \le 1/\delta_L - \varepsilon_L$, then*

$$\sum_j \tau(q_j) \le \sum_j \sigma(q_j)$$

Proof. One can prove that

$$\begin{aligned}
\sum_j \tau(q_j) &= \sum_t (u_i - \lambda_t(A_{i-1}))^{-1} + \frac{\sum_t (u_i - \lambda_t(A_{i-1}))^{-2}}{\sum_t (u_{i-1} - \lambda_t(A_{i-1}))^{-1} - \sum_t (u_i - \lambda_t(A_{i-1}))^{-1}} \\
\sum_j \sigma(q_j) &= -\sum_t (\lambda_t(A_{i-1}) - l_i)^{-1} + \frac{\sum_t (\lambda_t(A_{i-1}) - l_i)^{-2}}{\sum_t (\lambda_t(A_{i-1}) - l_i)^{-1} - \sum_t (\lambda_t(A_{i-1}) - l_{i-1})^{-1}}.
\end{aligned}$$

One can fairly easy show that $\sum_j \tau(q_j) \le \varepsilon_U + 1/\delta_U$. With some work one can show that $\sum_j \sigma(q_j) \ge 1/\delta_L - \varepsilon_L$. The desired inequality follows. □

So, the final algorithm is the following: on each step we compute $\sigma(q_j)$ and $\tau(q_j)$ for every j, and choose j such that $\tau(q_j) \le \sigma(q_j)$. Then choose α such that $\tau(q_j) \le 1/\alpha \le \sigma(q_j)$, and set $A_i := A_{i-1} + \alpha q_j q_j^t$. We repeat this process until all the eigenvalues of A_i are within $(1 \pm \varepsilon)$ of each other, and then do a normalization.

6 Conclusion

We presented two construction of spectral sparsifiers. The better of two achieves the bound of $O(n/\varepsilon^2)$ edges. One challenging open problem here is to achieve the same bound for cut sparsifiers with "combinatorial" argument. The only known result in this direction is [GRV09].

7 Acknowledgements

I am very thankful to Richard Peng for enlightening discussions.

References

[ACNN11] Alexandr Andoni, Moses S. Charikar, Ofer Neiman, and Huy L. Nguyen. Near Linear Lower Bound for Dimension Reduction in L1. pages 315–323. IEEE, October 2011.

[Bal97] Keith Ball. An Elementary Introduction to Modern Convex Geometry. In *in Flavors of Geometry*, page 1–58. Univ. Press, 1997.

[BC05] Bo Brinkman and Moses Charikar. On the impossibility of dimension reduction in l_1. *J. ACM*, 52(5):766–788, 2005.

[BK96] András A. Benczúr and David R. Karger. Approximating s-t Minimum Cuts in $\tilde{O}(n^2)$ Time. In *STOC*, pages 47–55, 1996.

[BSS09] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan Sparsifiers. In *STOC*, pages 255–262, 2009.

[DG03] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.

[GR98] Andrew V. Goldberg and Satish Rao. Beyond the Flow Decomposition Barrier. *J. ACM*, 45(5):783–797, 1998.

[GRV09] Navin Goyal, Luis Rademacher, and Santosh Vempala. Expanders via random spanning trees. In *SODA*, pages 576–585, 2009.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43:439–561, 2006.

[KMP10] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching Optimality for Solving SDD Linear Systems. In *FOCS*, pages 235–244, 2010.

[KST90] Philip N. Klein, Clifford Stein, and Éva Tardos. Leighton-Rao Might Be Practical: Faster Approximation Algorithms for Concurrent Flow with Uniform Capacities. In *STOC*, pages 310–321, 1990.

[LN04] J. R. Lee and A. Naor. Embedding the diamond graph in L_p and dimension reduction in L_1. *arXiv:math/0407520*, July 2004.

[Nao11] Assaf Naor. Sparse quadratic forms and their geometric applications (after Batson, Spielman and Srivastava). *arXiv:1101.4324*, January 2011.

[Reg11] Oded Regev. Entropy-based Bounds on Dimension Reduction in L_1. *arXiv:1108.1283*, August 2011.

[Rud96] Mark Rudelson. Random vectors in the isotropic position. *arXiv:math/9608208*, August 1996.

[RV07] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4), 2007.

[SM50] Jack Sherman and Winifred J. Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, March 1950.

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.

[ST11] Daniel A. Spielman and Shang-Hua Teng. Spectral Sparsification of Graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011.

[Tal90] Michel Talagrand. Embedding Subspaces of L_1 into l_1^N. *Proceedings of the American Mathematical Society*, 108(2), February 1990.