Graph Sparsification

Matthew Begué

Norbert Wiener Center
Department of Mathematics
University of Maryland, College Park

Weighted Graphs

- We will only consider undirected, weighted graphs represented $G = G(V, E, \omega)$.
- V is the vertex set of size $N < \infty$.
- E is the edge set, $E = \{(u, v) : u, v \in V \text{ and } u \sim v\}$.
- Each edge is assigned a weight $\omega_{u,v} > 0$.
- For any $x \in V$, the *degree of* x, d_x, is the sum of weights of edges originating from x.

$$d_x = \sum_{y \in V} \omega_{x,y}.$$

Norbert Wiener Center
for Harmonic Analysis and Applications

Graph Laplacian

- For a finite graph, the Laplacian can be represented as a matrix. Let D denote the $N \times N$ *degree matrix*, $D = \mathrm{diag}(d_x)$. Let W denote the $N \times N$ *weighted adjacency matrix*,

$$W(i,j) = \begin{cases} \omega_{x_i,x_j}, & \text{if } x_i \sim x_j \\ 0, & \text{otherwise.} \end{cases}$$

Then the unweighted graph Laplacian can be written as

$$L = D - W.$$

Equivalently,

$$L(i,j) = \begin{cases} d_{x_i} & \text{if } i = j \\ -\omega_{x_i,x_j} & \text{if } x_i \sim x_j \\ 0 & \text{otherwise.} \end{cases}$$

Norbert Wiener Center
for Harmonic Analysis and Applications

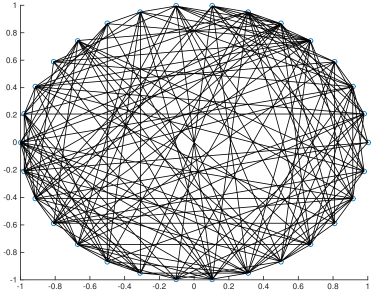
Spectrum of the Laplacian

- L is a real symmetric matrix and therefore has nonnegative eigenvalues $\{\lambda_k\}_{k=0}^{N-1}$ with associated orthonormal eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$.
- If G is finite and connected, then we have

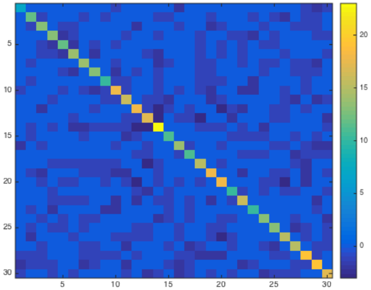
$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{N-1}.$$

- The spectrum of the Laplacian, $\sigma(L)$, is fixed but one's choice of eigenvectors $\{\varphi_k\}_{k=0}^{N-1}$ can vary.
- Since L is Hermetian ($L = L^*$), then we can choose the eigenbasis $\{\varphi_k\}_{k=0}^{N-1}$ to be entirely real-valued.
- Easy to show that $\varphi_0 \equiv 1/\sqrt{N}$.

A weighted graph and its Laplacian



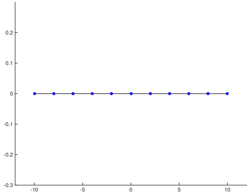
(a) A random graph on 30 vertices



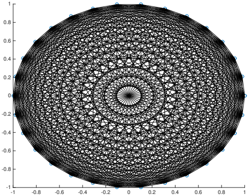
(b) Laplacian of the graph

Problems arise with graphs with many edges

- A connected graph on N vertices can have as few as $N-1$ edges,



- and can have as many as $\frac{N(N-1)}{2}$ edges



κ-approximation

- Goal is to construct a subgraph, $H = (V, \tilde{E}, \tilde{\omega})$, to be a κ-approximation of G.
- H is a *κ-approximation* of G if there exist $B \geq A > 0$ with $B/A \leq \kappa$ such that for all $x \in \mathbb{R}^N$, we have

$$A \cdot x^\top L_G x \leq x^\top L_H x \leq B \cdot x^\top L_G x.$$

- We write

$$A \cdot L_G \preceq L_G \preceq B \cdot L_G.$$

- This means for any $i = 0, 1, ..., N-1$,

$$A \leq \frac{\lambda_i^{(H)}}{\lambda_i^{(G)}} \leq B$$

- Subgraph H has the same vertex set, V, as G. But we change the edge set and the weights of those edges. In particular $\tilde{E} \subseteq$

κ-approximation

- Goal is to construct a subgraph, $H = (V, \tilde{E}, \tilde{\omega})$, to be a κ-approximation of G.
- H is a *κ-approximation* of G if there exist $B \geq A > 0$ with $B/A \leq \kappa$ such that for all $x \in \mathbb{R}^N$, we have

$$A \cdot x^\top L_G x \leq x^\top L_H x \leq B \cdot x^\top L_G x.$$

- We write

$$A \cdot L_G \preceq L_G \preceq B \cdot L_G.$$

- This means for any $i = 0, 1, ..., N-1$,

$$A \leq \frac{\lambda_i^{(H)}}{\lambda_i^{(G)}} \leq B$$

- Subgraph H has the same vertex set, V, as G. But we change the edge set and the weights of those edges. In particular $\tilde{E} \subseteq E$

κ-approximation

- Goal is to construct a subgraph, $H = (V, \tilde{E}, \tilde{\omega})$, to be a κ-approximation of G.
- H is a *κ-approximation* of G if there exist $B \geq A > 0$ with $B/A \leq \kappa$ such that for all $x \in \mathbb{R}^N$, we have

$$A \cdot x^\top L_G x \leq x^\top L_H x \leq B \cdot x^\top L_G x.$$

- We write

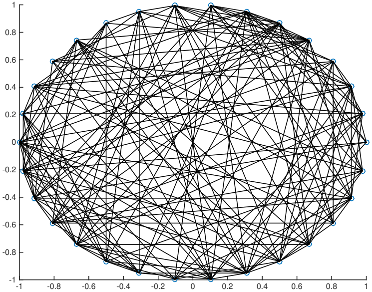
$$A \cdot L_G \preceq L_G \preceq B \cdot L_G.$$

- This means for any $i = 0, 1, ..., N-1$,

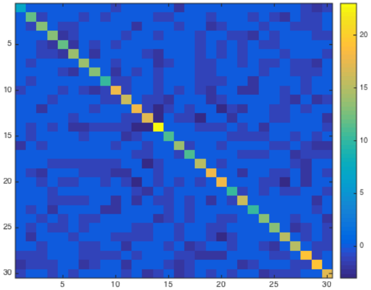
$$A \leq \frac{\lambda_i^{(H)}}{\lambda_i^{(G)}} \leq B$$

- Subgraph H has the same vertex set, V, as G. But we change the edge set and the weights of those edges. In particular $\tilde{E} \subseteq E$.

Big Picture

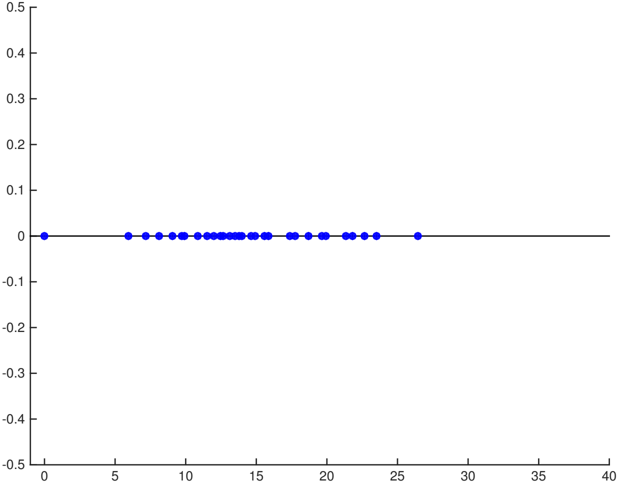


(a) A random graph on 30 vertices



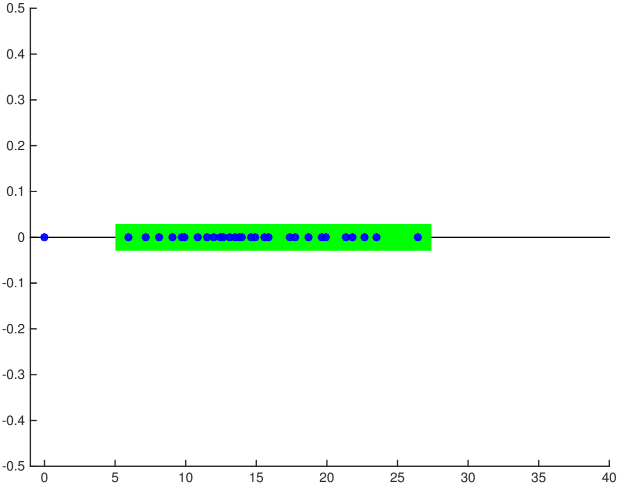
(b) Laplacian of the graph

Eigenvalues of L_G



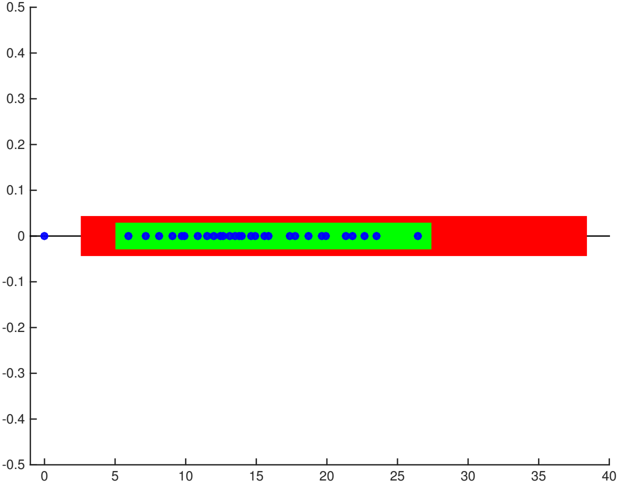
Wiener Center
nalysis and Applications

Eigenvalues of L_G



Wiener Center
nalysis and Applications

Eigenvalues of L_H will be contained in red region



Spielman's result

Theorem

Let G be an undirected weighted graph on N vertices and let $d > 1$. There exists a weighted subgraph H with at most $d \cdot N$ edges satisfying

$$\left(1 - 1/\sqrt{d}\right)^2 L_G \preceq L_H \preceq \left(1 + 1/\sqrt{d}\right)^2 L_G.$$

Hence, H is a $\kappa = \left(\frac{1+1/\sqrt{d}}{1-1/\sqrt{d}}\right)^2$-approximation of G.

- Joshua Batson, Daniel Spielman, Nikhil Srivastava, *Twice-Ramanujan Sparsifiers*, SIAM Review (2014) **56**, no. 2, pp. 315-334.

Method

- The Laplacian matrix, L_G, can be written as a sum of rank-1 outer products

$$L_G = \sum_{(u,v)\in E} \omega_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top.$$

- The sparsified graph, H, will have Laplcian

$$L_H = \sum_{(u,v)\in E} \tilde{\omega}_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top$$

  where at most $d \cdot N$ of the $\tilde{\omega} \neq 0$.

- The weights $\tilde{\omega}_{u,v}$ that are nonzero will be chosen so that the eigenvalues “play nice.”

Method

- The Laplacian matrix, L_G, can be written as a sum of rank-1 outer products

$$L_G = \sum_{(u,v)\in E} \omega_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top.$$

- The sparsified graph, H, will have Laplcian

$$L_H = \sum_{(u,v)\in E} \tilde{\omega}_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top$$

where at most $d \cdot N$ of the $\tilde{\omega} \neq 0$.

- The weights $\tilde{\omega}_{u,v}$ that are nonzero will be chosen so that the eigenvalues “play nice.”

Norbert Wiener Center for Harmonic Analysis and Applications

Method

- The Laplacian matrix, L_G, can be written as a sum of rank-1 outer products

$$L_G = \sum_{(u,v)\in E} \omega_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top.$$

- The sparsified graph, H, will have Laplcian

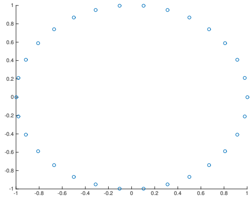
$$L_H = \sum_{(u,v)\in E} \tilde{\omega}_{u,v}(\chi_u - \chi_v)(\chi_u - \chi_v)^\top$$

  where at most $d \cdot N$ of the $\tilde{\omega} \neq 0$.
- The weights $\tilde{\omega}_{u,v}$ that are nonzero will be chosen so that the eigenvalues "play nice."

Sketch of algorithm

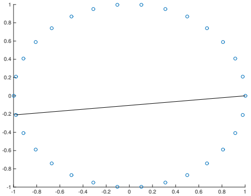
- Starting from $A_0 = \mathbf{0}$, we will construct A_n by adding a weighted outer product
$$A_n = A_{n-1} + s_n v_n v_n^\top$$
where $s_n > 0$ and $v_n = \chi_u - \chi_v$ for some $(u, v) \in E$.
- The algorithm requires the selection of four positive constants, $\epsilon_U, \epsilon_L, \delta_U, \delta_L$.

Step 0

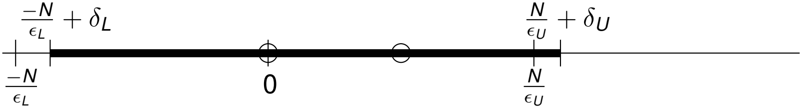


$$A_0 = \mathbf{0}$$

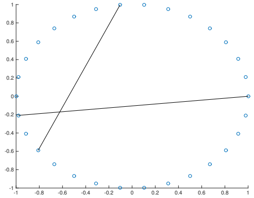
Step 1



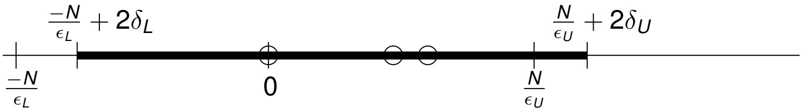
$$A_1 = A_0 + s_1 \mathbf{v}_1 \mathbf{v}_1^\top$$



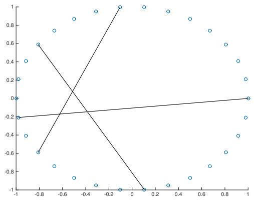
Step 2



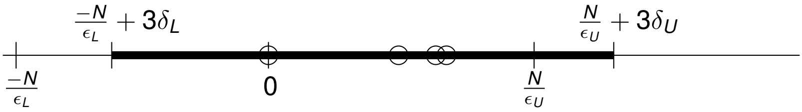
$$A_2 = A_1 + s_2 \mathbf{v}_2 \mathbf{v}_2^\top$$



Step 3

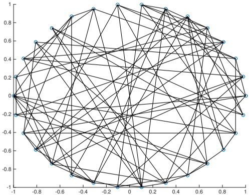


$$A_3 = A_2 + s_3 \mathbf{v}_3 \mathbf{v}_3^\top$$

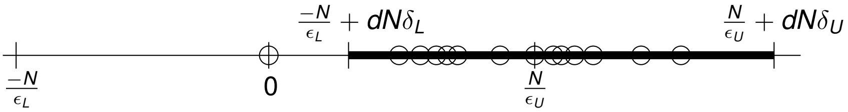


Norbert Wiener Center for Harmonic Analysis and Applications

Step $d \cdot N$



$$L_H = A_{dN} + s_{dN}\mathbf{v}_{dN}\mathbf{v}_{dN}^\top$$



Logistics

- For all $d \cdot N$ iterations of the algorithm, there will always exist at least one admisible edge v_i and scalar $s_i > 0$ provided that

$$0 \leq 1/\delta_U + \epsilon_U \leq 1/\delta_L - \epsilon_L.$$

- Spielman gives constants

$$\delta_L = 1, \quad \delta_U = \frac{\sqrt{d_1}}{\sqrt{d} - 1}, \quad \epsilon_L = \frac{1}{\sqrt{d}}, \quad \epsilon_U = \frac{\sqrt{d} - 1}{d + \sqrt{d}},$$

that make the resulting graph H a κ-approximation of G for

$$\kappa = \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}} \right)^2 .$$

Theorem (B.)

This is the smallest value of κ that guarantees the Spielman algorithm will produce a κ-approximation of G with only $d \cdot N$ edges.

Logistics

- For all $d \cdot N$ iterations of the algorithm, there will always exist at least one admisible edge v_i and scalar $s_i > 0$ provided that

$$0 \leq 1/\delta_U + \epsilon_U \leq 1/\delta_L - \epsilon_L.$$

- Spielman gives constants

$$\delta_L = 1, \quad \delta_U = \frac{\sqrt{d_1}}{\sqrt{d} - 1}, \quad \epsilon_L = \frac{1}{\sqrt{d}}, \quad \epsilon_U = \frac{\sqrt{d} - 1}{d + \sqrt{d}},$$

that make the resulting graph H a κ-approximation of G for

$$\kappa = \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}} \right)^2.$$

Theorem (B.)

This is the smallest value of κ that guarantees the Spielman algorithm will produce a κ-approximation of G with only $d \cdot N$ edges.

Logistics

- For all $d \cdot N$ iterations of the algorithm, there will always exist at least one admisible edge v_i and scalar $s_i > 0$ provided that

$$0 \leq 1/\delta_U + \epsilon_U \leq 1/\delta_L - \epsilon_L.$$

- Spielman gives constants

$$\delta_L = 1, \quad \delta_U = \frac{\sqrt{d_1}}{\sqrt{d} - 1}, \quad \epsilon_L = \frac{1}{\sqrt{d}}, \quad \epsilon_U = \frac{\sqrt{d} - 1}{d + \sqrt{d}},$$

that make the resulting graph H a κ-approximation of G for

$$\kappa = \left(\frac{1 + 1/\sqrt{d}}{1 - 1/\sqrt{d}} \right)^2.$$

Theorem (B.)

This is the smallest value of κ that guarantees the Spielman algorithm will produce a κ-approximation of G with only $d \cdot N$ edges.

Strengths/Weaknesses

Strengths

- This is a completely deterministic algorithm.
- There exists a method of sparsification via *graph effective resistances* which produces a κ-approximation of *G with high probability*.
- Daniel A. Spielman and Nikhil Srivastava, *Graph sparsification by effective resistances*, SIAM Journal on Computing, (2011)**40** no. 6, pp. 1913-1926.

Weaknesses

- Very computationally expensive. Each step requires inverting 3 $N \times N$ matrices.
- No information on the eigenvectors of L_H
  - Wildly different in numerical experiments
  - Want to preserve eigenvectors to Fourier transform on sparsified graph.

Strengths/Weaknesses

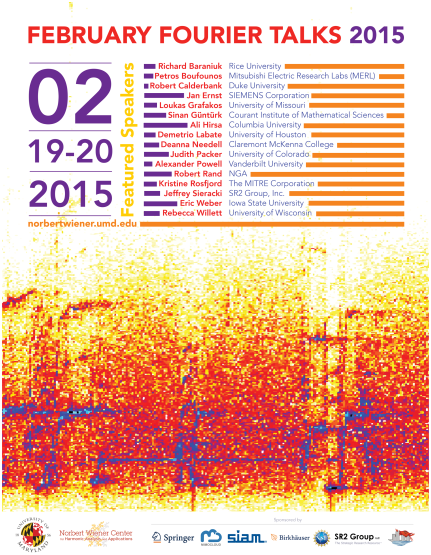
Strengths

- This is a completely deterministic algorithm.
- There exists a method of sparsification via *graph effective resistances* which produces a κ-approximation of *G with high probability*.
- Daniel A. Spielman and Nikhil Srivastava, *Graph sparsification by effective resistances*, SIAM Journal on Computing, (2011)**40** no. 6, pp. 1913-1926.

Weaknesses

- Very computationally expensive. Each step requires inverting 3 $N \times N$ matrices.
- No information on the eigenvectors of L_H
  - Wildly different in numerical experiments
  - Want to preserve eigenvectors to Fourier transform on sparsified graph.

Register for FFT today!



Norbert Wiener Center
for Harmonic Analysis and Applications