Advanced Algorithms

Spectral methods and algorithms

尹一通　栗师　**刘景铖**

Recall: Cut Sparsifiers

Definition. A graph H is an ϵ-cut approximator of a graph G if for all $S \subseteq V$,

$$(1-\epsilon)\cdot w\big(\delta_G(S)\big) \leq w\big(\delta_H(S)\big) \leq (1+\epsilon)\cdot w\big(\delta_G(S)\big).$$

Theorem. [Benczur, Karger 96] Any graph G has an ϵ-cut approximator H with $O\left(\frac{n \log n}{\epsilon^2}\right)$ edges.

$A \preccurlyeq B \iff B - A \succcurlyeq 0$
$\iff \forall x \in \mathbb{R}^n, x^\top A x \le x^\top B x$

Spectral Sparsification

Definition. [Spielman, Teng] A graph H is an ϵ-spectral approximator of a graph G if

$$(1-\epsilon)L_G \preccurlyeq L_H \preccurlyeq (1+\epsilon)L_G.$$

Observation. An ϵ-spectral approximator is an ϵ-cut approximator. Converse is not always true (Path vs. cycle)

$$\chi_S^\top L_G \chi_S = \sum_{uv \in E} w_{uv}(\chi_S(u) - \chi_S(v))^2 = w(\delta_G(S))$$

Side note: this is a “physical approximation”: H and G as electrical networks generate roughly the same energy

Theorem. [Spielman, Srivastava] Any graph G has an ϵ-spectral approximator H with $O\left(\frac{n \log n}{\epsilon^2}\right)$ edges.

Theorem. [Batson,Spielman, Srivastava] Any graph G has an ϵ-spectral approximator H with $O\left(\frac{n}{\epsilon^2}\right)$ edges.

Spectral approximation and comparison

Exercise: Let G be a d-regular graph with spectral radius $\alpha = \epsilon d$, and W be its random walk matrix. Then

$$(1-\epsilon)(I-J) \preccurlyeq I - W \preccurlyeq (1+\epsilon)(I-J)$$

where J is the all 1/n matrix.

Example:

- If H is a subgraph of G, then $L_H \preccurlyeq L_G$
- If $A \preccurlyeq B$, then $\forall k, \lambda_k(A) \leq \lambda_k(B)$

Exercise:

- If $\forall k, \lambda_k(A) \leq \lambda_k(B)$, do we have $A \preccurlyeq B$?

The level set of a quadratic form $x^\top Ax \leq 1$ defines an ellipsoid
So $A \preccurlyeq B$ is asking about ellipsoid containment

Path inequality

If P is a path of length r with endpoints a and b, then

$$L_{(a,b)} \preccurlyeq r \cdot L_P$$

Proof Idea: write $L_{(a,b)}$ as a telescoping sum over the path.

Bounding λ_2 of a Path graph

Idea: We compare the path with a complete graph, and use known bounds for the complete graph to deduce a bound for the path

- $L_{K_n} = \sum_{a<b} L_{G_{a,b}}$
- Recall that $\lambda_2(L_{K_n}) = n$
- For every $a < b$, let $P_{a,b}$ be the path connecting $a, a+1, \ldots, b-1, b$
- Path inequality tells us that $L_{G_{a,b}} \preccurlyeq (b-a) L_{P_{a,b}} \preccurlyeq (b-a) L_{P_n}$
- Thus $L_{K_n} \preccurlyeq \sum_{a<b} (b-a) L_{P_n} = \frac{n(n+1)(n-1)}{6} L_{P_n}$
- Therefore $\lambda_2(L_{P_n}) \geq \frac{6}{(n+1)(n-1)}$

Spectral Sparsification: Linear Algebraic Formulation

There is a reduction from the spectral sparsification problem to a purely linear algebraic problem.

Theorem. Suppose $v_1, \ldots, v_m \in \mathbb{R}^n$ are given with $\sum_{i=1}^{m} v_i v_i^\top = I_n$.

There exist scalars $s_1, \ldots, s_m$ with at most $O\left(\frac{n \log n}{\epsilon^2}\right)$ nonzeros such that

$$(1-\epsilon) \cdot I_n \preccurlyeq \sum_{i=1}^{m} s_i v_i v_i^\top \preccurlyeq (1+\epsilon) \cdot I_n$$

$$L_G = \sum_{i \geq 2} \lambda_i u_i u_i^\top$$

$$L_G^{-\frac{1}{2}} = \sum_{i \geq 2} \frac{1}{\sqrt{\lambda_i}} u_i u_i^\top$$

Recall that in spectral approximation, we want to find L_H such that $(1-\epsilon)L_G \preccurlyeq L_H \preccurlyeq (1+\epsilon)L_G$.
Note that $L_G = \sum_e b_e b_e^\top$, and we would like to choose a subset of these vectors to form

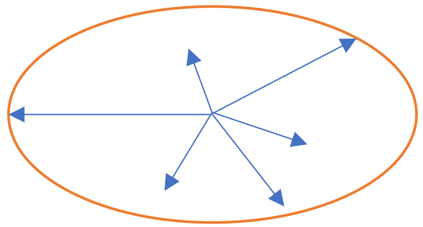
$$L_H = \sum_e s_e \, b_e b_e^\top$$

In general, this defines an ellipsoid, and spectral approximation is asking for ellipsoid containment
Notice that rescaling by a PSD matrix preserves Loewner order

So $(1-\epsilon)L_G \preccurlyeq L_H \preccurlyeq (1+\epsilon)L_G \Leftrightarrow (1-\epsilon) \cdot I_n \preccurlyeq L_G^{-\frac{1}{2}} L_H L_G^{-\frac{1}{2}} \preccurlyeq (1+\epsilon) \cdot I_n$

$$\Leftrightarrow (1-\epsilon) \cdot I_n \preccurlyeq \sum_e s_e L_G^{-\frac{1}{2}} b_e b_e^\top L_G^{-\frac{1}{2}} \preccurlyeq (1+\epsilon) \cdot I_n$$

v_e



Isotropy Condition

The condition $\sum_{i=1}^{m} v_i v_i^{\top} = I_n$ is called the "isotropy" condition.

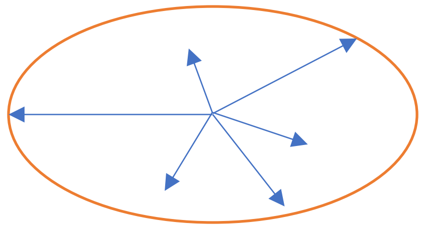
Another way to think of it is as an overcomplete basis.

- If $m = n$, then v_i forms an orthonormal basis
- In general $m \geq n$
- Quadratic form is the same in every direction for unit vectors (circle), so equally important directions
- Longer vectors are therefore more important
- Such rescaling does not seem to have a combinatorial correspondence

Recall that $v_e = L_G^{-\frac{1}{2}} b_e$

The length of a vector $\|v_e\|_2^2 = b_e^{\top} L_G^{-1} b_e$ has a physical meaning: **<u>effective resistance</u>**!

Also known as **<u>leverage score</u>** in numerical linear algebra



Intuition for an algorithm

Idea: Random Sampling (from Karger).

- Uniform sampling won't work.

- Non-uniform sampling?
  - Need to bias towards the middle edges in a dumb-bell graph
  - Intuition from **effective resistance**:
    - Higher resistance means fewer alternative paths, or electrically "important"
    - Lower resistance means more alternative paths, or electrically "redundant"

Sampling algorithm for approximating identity

- Initialization: $F \leftarrow \emptyset, \quad \vec{s} \leftarrow 0, \quad C = \frac{9n \log n}{\epsilon^2}$.

- For $1 \leq t \leq C$ do

  Sample i with probability $p_i = \frac{1}{n}\|v_i\|_2^2$, update $F \leftarrow F \cup \{i\}$ and $s_i \leftarrow s_i + \frac{1}{Cp_i}$.

- Return $\sum_{i \in F} s_i v_i v_i^\top$ as our solution.

Note that $\sum_i p_i = \frac{1}{n} \sum_i v_i^\top v_i = \frac{1}{n} \sum_i Tr(v_i v_i^\top) = \frac{1}{n} Tr(\sum_i v_i v_i^\top) = 1$

Trace trick: $Tr(AB) = Tr(BA)$

Matrix Chernoff Bound

There is an elegant generalization of Chernoff bound to the matrix setting.

Theorem. Let $X_1, \ldots, X_m$ be independent $n \times n$ real symmetric matrices with $0 \preccurlyeq X_i \preccurlyeq R \cdot I$ for some $R \in \mathbb{R}$.

Let $\mu_{\min} I \preccurlyeq \sum_{i=1}^{m} E[X_i] \preccurlyeq \mu_{\max} I$. For any $0 < \epsilon \leq 1$,

$$\Pr\left(\lambda_{\max}\left(\sum_{i=1}^{m} X_i\right) \geq (1+\epsilon)\mu_{\max}\right) \leq n e^{-\frac{\epsilon^2 \mu_{\max}}{3R}}$$

$$\Pr\left(\lambda_{\min}\left(\sum_{i=1}^{m} X_i\right) \leq (1-\epsilon)\mu_{\min}\right) \leq n e^{-\frac{\epsilon^2 \mu_{\min}}{2R}}.$$

Concentration

$C = \frac{9n \log n}{\epsilon^2}$

The random variables are $X_t = \frac{v_i v_i^T}{C\, p_i}$ with probability p_i. We apply Matrix Chernoff:

$$\mathbb{E}X_t = \sum_i p_i \frac{v_i v_i^T}{C p_i} = \frac{1}{C} I$$

This gives a multiplicative approximation with high probability

Sample C random matrices with replacement, each with probability $p_i = \frac{1}{n}\|v_i\|_2^2$ and reweighted by $\frac{1}{Cp_i}$ approximates works whp

Effective Resistance

What is the sampling probability?

Recall that $v_e = L_G^{-\frac{1}{2}} b_e$

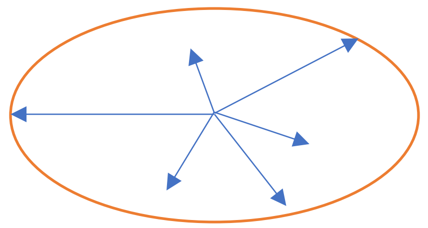
$$\|v_e\|_2^2 = b_e^\top L_G^{-1} b_e$$

In the graph case, it is possible to compute good approximations of the sampling probabilities in near-linear time. The idea is to do dimension reduction.

Discussion on sampling based sparsification

- Other ways to think about the sampling probability:
  - “leverage score” in numerical linear algebra
  - $\forall uv \in E,\ R_{\text{eff}}(u,v) \cdot w_{uv} = \Pr[e \in T]$, where T is a uniformly random spanning tree.
  - Matrix-tree theorem

- Tight example: Consider $\sum_{i=1}^{n} e_i e_i^T = I_n$
  - Pick one direction uniformly at random each time
  - By coupon collector, an extra O(log n) factor is necessary!
  - However, a greedy approach might do better: find the missing direction, then add the corresponding direction



Linear-Sized Spectral Sparsifiers

Theorem. [Batson Spielman, Srivastava] Any graph G has an ϵ-spectral approximator H with $O\left(\frac{n}{\epsilon^2}\right)$ edges.

- The proof is purely linear algebraic, and it gives a deterministic greedy algorithm to construct a sparsifier
- There are near-linear time algorithms to find a linear-sized spectral sparsifier now

- Converting to vectors seems to be the best way to look at the graph sparsification problem

- The ideas are extended to lead to a breakthrough in mathematics (Kadison-Singer problem)
- There is also an interpretation of BSS's result in the **matrix** multiplicative weight update framework
- Other sampling distribution: sample O(log(n)) random spanning trees
- Matrix concentration from strongly Rayleigh of spanning tree distribution (real-stability)

Cauchy Interlacing

What happens when you add a rank-one update to a matrix

- Eigenvalues are determined by roots of characteristic polynomials

$$p_A(x) = \det(\boldsymbol{xI} - \boldsymbol{A})$$

- $p_{A+vv^\top}(x) = \det(\boldsymbol{xI} - \boldsymbol{A} - vv^\top)$

$$= \det(\boldsymbol{xI} - \boldsymbol{A})\det(\boldsymbol{I} - (\boldsymbol{xI} - \boldsymbol{A})^{-1}vv^\top)$$

$$= \det(\boldsymbol{xI} - \boldsymbol{A})\,(\boldsymbol{I} - v^\top(\boldsymbol{xI} - \boldsymbol{A})^{-1}v)$$

$$= p_A(x)\left(\mathbf{1} - \sum_i \frac{1}{x-\lambda_i}\langle u_i, v\rangle^2\right)$$

$$(\boldsymbol{xI} - \boldsymbol{A})^{-1} = \sum_i \frac{1}{x-\lambda_i} u_i u_i^\top$$

The new eigenvalues are therefore x such that $\mathbf{1} = \sum_i \frac{1}{x-\lambda_i}\langle u_i, v\rangle^2$

$$\sum_{i=1}^{m} v_i v_i^\top = I_n$$

Adding a balanced vector

By choosing v_i uniformly at random,

$$\forall y \in \mathbb{R}^n, \mathbb{E}\langle y, v_i\rangle^2 = \frac{1}{m}\sum_{i=1}^{m} y^\top v_i v_i^\top y = \frac{1}{m} y^\top y$$

If one can add a very balanced vector that behaves like the expectation

$$p_{A+vv^\top}(x) = p_A(x)\left(\mathbf{1} - \sum_i \frac{1}{x-\lambda_i}\langle u_i, v\rangle^2\right) = p_A(x)\left(\mathbf{1} - \frac{1}{m}\sum_i \frac{1}{x-\lambda_i}\right)$$

$$\mathbb{E}\, p_{A+vv^\top}(x) = \left(1 - \frac{1}{m}\frac{\partial}{\partial x}\right) p_A(x)$$

This is not a proof because roots($\mathbb{E}p$)$\neq$ $\mathbb{E}$roots(p)

But there is a way to make this essentially happens

By a barrier argument, BSS showed that one can evenly move the eigenvectors through a greedy algorithm

Applications: Reductions between randomized nearly linear time algorithms

- Solve $\hat{L}x = b$ for $\hat{L} \approx L$
- Solve $Lx = b$ or compute $L^{\dagger}b$
- Approximate one-pair effective resistance
- Approximate all pairs effective resistances
- Spectral sparsification

Electrical networks

Electrical flows, effective resistance, hitting time and cover time

Why hitting time and cover time?

Hitting time

- Finding bipartite matching
  - Use random walk to find an augmenting cycle
  - Interested in the first return time, in expectation
- 2SAT, and more generally the Moser-Tardos algorithm
  - Can be seen as a random walk over all assignments
  - Interested in the first time of hitting a satisfying assignment, in expectation

Cover time? Imagine you want to explore the graph

Using DFS/BFS, you need time $O(|E| + |V|)$ and space $O(|V|)$

What if we use random walk instead?

Space $= O(\log n)$, expected running time = cover time $\leq O(|V||E|)$

In fact, U. Feige showed that there is an entire spectrum of time-space trade-off:

For every s there is an algorithm using space s and time $\tilde{O}\left(\frac{|V||E|}{s}\right)$ that covers all vertices w.h.p.

Electrical Flow

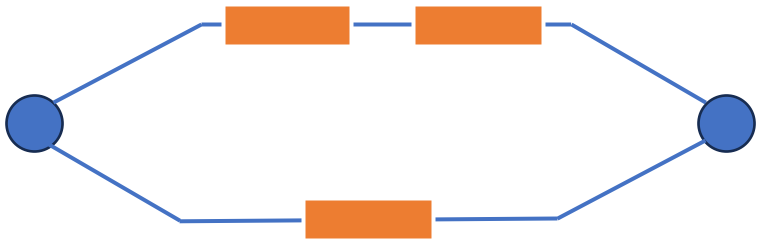
An electrical network is an undirected graph where every edge is a resistor of resistance r_e.

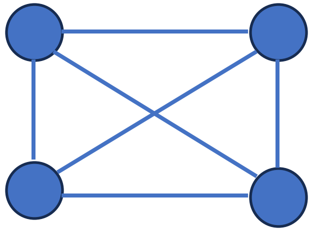
The electrical flows on this network are governed by two laws:

1) Kirchhoff's law: The sum of incoming currents is equal to the sum of outgoing currents.
2) Ohm's law: There exists a voltage vector $\phi: V \to \mathbb{R}$ such that $\phi(u) - \phi(v) = i_{uv} r_{uv}$ for all $e \in E$,

   where i_{uv} is positive in the forward direction and negative in the backward direction

Given an electrical network, how do you compute these quantities?





Not every graph is series-parallel

Matrix formulation of electrical networks

Input: graph $G = (V, E)$, resistance r_e or conductance $w_e = 1/r_e$ for $e \in E$, **demand** b_v for $v \in V$.

Output: the current/flow i_{uv} on each edge $uv \in E$, and the voltage ϕ_v on each vertex $v \in V$.

$b_v > 0$ if injecting a flow; source
$b_v < 0$ if outputting a flow; sink
$b_v = 0$ everywhere else

Ohm's law: $\phi(u) - \phi(v) = i_{uv} r_{uv} \quad \Leftrightarrow \quad i_{uv} = w_{uv}\big(\phi(u) - \phi(v)\big)$ for all $uv \in E$.

Kirchhoff's law: The sum of incoming flows is equal to the sum of outgoing flows.

$$\sum_{u:vu \in E} i_{vu} = b_v, \qquad \forall v \in V$$

Combined:

$$b_v = \sum_{u:vu \in E} i_{vu} = \sum_{u:vu \in E} w_{uv}\big(\phi(v) - \phi(u)\big) = \deg_w(v)\,\phi(v) - \sum_{u:vu \in E} w_{uv}\phi(u)$$

where $\deg_w(v) = \sum_{u:vu \in E} w_{uv}$ is a weighted degree. Specifically, if $w_{uv} = 1$, the above is simply $\vec{b} = L\vec{\phi}$

In general, we have a weighted Laplacian

Matrix formulation of electrical networks

Given resistor network, we inject 1A current into a node s, and let the current flow out of a node t

How do you compute the voltages? Solve the equations $\vec{b} = L\vec{\phi}$
Now that we have the voltages $\vec{\phi}$, by Ohm's law, the current $i_{uv} = w_{uv}(\phi(u) - \phi(v))$

Consider the incidence matrix B, we have $\vec{\imath} = WB^\top\vec{\phi}$ for a diagonal matrix W of conductances

Then the Laplacian can also be written as:

$$L = \sum_e w_e b_e b_e^\top = BWB^\top$$

Then $\vec{b} = L\vec{\phi} = BWB^\top\vec{\phi} = B\vec{\imath}$, which is exactly the law of flow conservation (Kirchhoff's law)

To relate electrical quantities to random walks, we observe that **<u>they follow the same set of equations</u>**
<u>Question</u>: is there always a solution to these equations? Are they unique?

Solution Space and Pseudo-inverse of L

L is not of full rank, so inverse doesn't exist, e.g. can't say $x = L^{-1}b$ is the unique solution.

But if G is connected (WLOG), then the nullspace of L is spanned by $\vec{1}$, and we can characterize the solutions.

Claim. If $Lx = b$, then $b \perp \vec{1}$.

Proof:

Suppose $Lx = b$, where $x = \sum_i c_i v_i$. Then $Lx = \sum_{i \geq 2} c_i \lambda_i v_i$ is orthogonal to $v_1 = \frac{1}{\sqrt{n}} \vec{1}$

This makes sense for electrical flow, because the sum of demands should be equal to zero.

Solution Space and Pseudo-inverse of L

Claim. If $b \perp \vec{1}$, then there exists x such that $Lx = b$.

Proof: Let $\vec{b} = \sum_{i=2}^{n} a_i\, v_i$. Consider $x = \sum_{i=2}^{n} \frac{a_i}{\lambda_i} v_i$. Then $Lx = \sum_{i=2}^{n} a_i\, v_i = b$.

The **pseudo-inverse** of L is defined as $L^{\dagger} := \sum_{i=2}^{n} \frac{1}{\lambda_i} v_i v_i^T$.

$L^{\dagger}$ maps any vector $b \perp \vec{1}$ to the unique vector x such that $Lx = b$ and $x \perp \vec{1}$.

So, the set of all solutions for $Lx = b$ is $\{L^{\dagger}b + c\vec{1} \mid c \in \mathbb{R}\}$, a “translation” of the solution $L^{\dagger}b$. (So, $\vec{i}$ is unique.)

In particular, if we fix the value of one node, e.g. $x_t = 0$, then there is a unique solution.

Any Laplacian system can be thought of as an electrical flow problem!

Effective resistance

The effective resistance $R_{\text{eff}}(s,t)$ between vertices s and t is defined as $\phi(s) - \phi(t)$,

where $\vec{\phi}$ satisfies $L\vec{\phi} = \vec{b}$ for a demand $\vec{b}$ sending one unit of electrical flow from s to t.

We should think of it as the resistance of the whole graph as a single big resistor.

Claim. $R_{\text{eff}}(s,t) = b_{st}^{\top} L^{\dagger} b_{st}$ where $b_{st} \in \mathbb{R}^n$ with $b_{st}(s) = 1, b_{st}(t) = -1$, and zero otherwise.

Proof: $R_{\text{eff}}(s,t) = b_{st}^{\top} \vec{\phi} = b_{st}^{\top} L^{\dagger} b_{st}$

Energy

The energy of an electrical flow is defined as

$$\mathcal{E}(\vec{i}) := \sum_{e \in E} i_e^2 \cdot r_e$$

Intuitively, if we think of the graph as a big resistor, then $\mathcal{E}(\vec{i}) = R_{\text{eff}}(s,t)$.

Claim. $\mathcal{E}(\vec{i}) = R_{\text{eff}}(s,t)$, where $\vec{i}$ is a one-unit electrical flow from s to t.

Proof:

$$\sum_{e \in E} i_e^2 \cdot r_e = \sum_{e} \frac{\big(\phi(u) - \phi(v)\big)^2}{r_e} = \phi^\top L \phi$$

where ϕ satisfies $L\phi = b_{st}$, so that $\phi = L^\dagger b_{st}$. Thus, $\mathcal{E}(\vec{i}) = b_{st}^\top L^\dagger b_{st} = R_{\text{eff}}(s,t)$

In words, the effective resistance between s and t is the energy of a one-unit electrical s-t flow.

Thompson's Principle

Theorem. $R_{\text{eff}}(s,t) \leq \mathcal{E}(\vec{g})$ where $\vec{g}$ is a one-unit s-t flow.

For simplicity we assume $r_e = 1, \forall r_e$

Proof (sketch):

Consider $\min \mathcal{E}(\vec{g}) = \min \sum_{e \in E} g_e^2$, s.t. $B\vec{g} = b_{st}$

As a convex constrained optimization problem, it is minimized when the gradient of the Lagrangian is zero:

$$\exists \phi \in \mathbb{R}^n \; s.t. B^\top \phi = \vec{g}$$

This is precisely the Ohm's law: $\vec{g}$ is a flow determined by a voltage vector ϕ

This means that $\vec{g}$ is an electrical flow

(For an elementary proof, consider $\vec{g} = \vec{\imath} + \vec{c}$, then try to show that the cross-terms are zero in the energy)

So, the one unit s-t electrical flow is the flow that minimizes the energy among all one unit s-t flow.

Rayleigh's Monotonicity Principle

Theorem. If $\overrightarrow{r'} \geq \vec{r}$, then $R_{\text{eff},\overrightarrow{r'}}(s,t) \geq R_{\text{eff},\vec{r}}(s,t)$.

Proof: Let $\vec{\imath}$ be a one-unit s-t electrical flow in the network of resistors $\vec{r}$, and $\overrightarrow{i'}$ be that of resistors $\overrightarrow{r'}$

$$R_{\text{eff},\vec{r}}(s,t) = \mathcal{E}_{\vec{r}}(\vec{\imath}) \leq \mathcal{E}_{\vec{r}}\left(\overrightarrow{i'}\right) \leq \mathcal{E}_{\overrightarrow{r'}}\left(\overrightarrow{i'}\right) = R_{\text{eff},\overrightarrow{r'}}(s,t)$$

The first inequality follows from Thompson's principle, and the second from $\overrightarrow{r'} \geq \vec{r}$ and $\mathcal{E}_{\vec{r}}(\vec{\imath}) \coloneqq \sum_{e \in E} i_e^2 \cdot r_e$

This is very intuitive, increasing the resistance of an edge could never decrease the effective resistance,

and decreasing the resistance of an edge could never increase the effective resistance.

Effective Resistances as Distances

Effective resistance is probably a better distance function to measure how close are two nodes

Especially for random walks

It is known that effective resistances satisfy the triangle inequality

Lemma. $R_{\text{eff}}(a, b) + R_{\text{eff}}(b, c) \geq R_{\text{eff}}(a, c)$ for any a, b, c

Random Walks on Undirected Graphs

We study some interesting quantities about random walks in undirected graphs.

1. <u>Hitting time:</u> $H_{u,v} \coloneqq \min\{t \geq 1 \mid X_1 = u \text{ and } X_t = v\}$ and $h_{u,v} = \mathbb{E}[H_{u,v}]$.

2. <u>Commute time</u>: $C_{u,v} \coloneqq h_{u,v} + h_{v,u}$.

3. <u>Cover time:</u> cover_v is defined as expected time to visit every vertex at least once

   if the random walk starts at v, and $\mathrm{cover}_G \coloneqq \max_{\mathrm{v}} \mathrm{cover}_v$

Commute Time

Theorem. For any two vertices s and t, $C_{s,t} = 2mR_{\text{eff}}(s,t)$, where $m = |E(G)|$

Proof:

Fix any node t, let $h_{u,t}$ be the hitting time from node u to node t, then $\forall u \neq t$

$$h_{u,t} = 1 + \frac{1}{d_u}\sum_{v\sim u} h_{v,t} \Rightarrow d_u h_{u,t} - \sum_{v\sim u} h_{v,t} = d_u$$

Consider the vector $\overrightarrow{h_{*,t}}$, it satisfies:

$$\begin{pmatrix} & D - A & \\ & & \end{pmatrix} \begin{pmatrix} h_{u,t} \\ h_{t,t} \end{pmatrix} = \begin{pmatrix} d_u \\ d_t - 2m \end{pmatrix}$$

Note that we have artificially added one row of equation on $h_{t,t}$

To ensure there is a solution, we have to make sure that the right hand side sum up to 0

(To be cont'd..)

Commute Time

Theorem. For any two vertices s and t, $C_{s,t} = 2mR_{\text{eff}}(s,t)$, where $m = |E(G)|$

Proof (cont'd):

Fix any node s, let $h_{u,s}$ be the hitting time from node u to node s, then $\forall u \neq s$

$$h_{u,s} = 1 + \frac{1}{d_u} \sum_{v \sim u} h_{v,s} \Rightarrow d_u h_{u,s} - \sum_{v \sim u} h_{v,s} = d_u$$

Consider the vector $\overrightarrow{h_{*,s}}$, it satisfies:

$$\begin{pmatrix} & & \\ & D - A & \\ & & \end{pmatrix} \begin{pmatrix} h_{s,s} \\ h_{u,s} \\ \\ h_{t,s} \end{pmatrix} = \begin{pmatrix} d_s - 2m \\ d_u \\ \\ d_t \end{pmatrix}$$

Again, we have artificially added one row of equation on $h_{s,s}$

(To be cont'd..)

Commute Time

Theorem. For any two vertices s and t, $C_{s,t} = 2mR_{\text{eff}}(s,t)$, where $m = |E(G)|$

Proof (cont'd):

$$L\left(\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}}\right) = \begin{pmatrix} d_s \\ d_u \\ \vdots \\ d_t - 2m \end{pmatrix} - \begin{pmatrix} d_s - 2m \\ d_u \\ \vdots \\ d_t \end{pmatrix} = \begin{pmatrix} 2m \\ 0 \\ \vdots \\ -2m \end{pmatrix}$$

Thus, $\frac{L\left(\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}}\right)}{2m} = b_{s,t}$

Recall that $L\phi = b_{st}$ has a solution that is unique up to translation

Let $\phi = \frac{\overrightarrow{h_{*,t}} - \overrightarrow{h_{*,s}}}{2m}$, we have

$$R_{\text{eff}}(s,t) = \phi(s) - \phi(t) = \frac{h_{s,t} - h_{s,s}}{2m} - \frac{h_{t,t} - h_{t,s}}{2m} = \frac{h_{s,t} + h_{t,s}}{2m} = \frac{C_{s,t}}{2m}$$

Cover Time

Corollary. $C_{u,v} \leq 2m$ for every edge $uv \in E$.

Proof: Notice that $R_{\text{eff}}(u,v) \leq 1$ for every edge $uv \in E$. Then it follows from $C_{u,v} = 2mR_{\text{eff}}(u,v) \leq 2m$

Theorem. The cover time of a connected graph is at most $2m(n-1)$.

Proof: Consider any spanning tree T.

Then the cover time is at most traversing the time to commute along each tree edges of T.

Approximating Cover Time by Resistance Diameter

Theorem. Let $R(G) \coloneqq \max_{u,v} R_{\text{eff}}(u,v)$ be the resistance diameter. Then,

$$m \cdot R(G) \ \leq \ \text{cover}(G) \leq \ 2e^3 m \cdot R(G) \cdot \ln n + n$$

Proof: Firstly,

$$\text{cover}(G) \geq \max\{h_{uv}, h_{vu}\} \geq \frac{C_{uv}}{2} = mR_{uv} \ ,$$

which is the lowerbound.

For the upperbound, notice that the maximum commute time from any vertex is at most $2mR(G)$

If the random walk is run for $2e^3 m \cdot R(G)$, by Markov's inequality, the probability that a vertex is not visited is at most $1/e^3$

If we repeat this $\ln n$ times, the probability that a vertex is not visited is at most $1/n^3$

By a union bound, the probability that there exists a vertex not visited is at most $1/n^2$

In such cases, we can pay for another pessimistic cover time of n^3

Combined, we have $\text{cover}(G) \leq \ 2e^3 m \cdot R(G) \cdot \ln n + \frac{1}{n^2} n^3$

Graph Connectivity

Theorem. There is an $O(n^3)$ time algorithm to solve s-t connectivity using only $O(\log n)$ space

Using random walk, the space requirement is $O(\log n)$ and expected running time is $O(|V||E|) = O(n^3)$

You may wonder, is randomness necessary for checking graph connectivity in log-space?

Definition. A sequence σ is (d, n)-universal if for every labeled connected d-regular graphs and every starting vertex s, the walk defined by σ started from s covers every vertices

Theorem. There exists (d, n)-universal sequence of length $O(n^3 d^2 \log nd)$ for undirected graphs

HINT: Cover time is at most $O(n^2 d)$ for d-regular graphs

Reingold's Theorem For undirected graphs, one can explicitly construct such a universal sequence in log-space

It is an open problem to derandomize log-space connectivity
Though likely not through "directed" universal sequences