Advanced Algorithms

Jingcheng Liu

Recap

<u>Previous lecture</u>:
<u>Random walks on undirected graphs</u>

- Fundamental theorem of Markov chains
- Spectral analysis
- Examples of random walks in algorithms:
  - Finding bipartite matching
    - Return time
  - Pagerank
    - Can be simulated by random walks
    - Mixing time
  - 2-SAT
    - Hitting time

<u>What next?</u>
<u>Random walks on undirected graphs</u>

- Mixing time
- More examples: random sampling

<u>Electrical networks</u>

- Electrical flows
- Effective resistance
- Hitting time and commute time of random walks

Theorem. For any finite, connected, non-bipartite graph,

$p_t \to \vec{\pi} = \frac{\vec{d}}{2m}$ as $t \to \infty$ regardless of p_0.

Mixing Time

From the fundamental theorem of Markov chain, we know that $p_t \to \vec{\pi} = \frac{\vec{d}}{2m}$ as $t \to \infty$ regardless of p_0

We would like to understand how fast it converges to π

Recall how we measure closeness: $d_{TV}(p_t, \pi) = \frac{1}{2}\|p_t - \pi\|_1 = \frac{1}{2}\sum_{i=1}^{n}|p_t(i) - \pi(i)|$

Definition. The **ϵ-mixing time** of the random walk is defined as the smallest t such that

$$\|p_t - \pi\|_1 \leq \epsilon \quad \forall\, p_0.$$

We will bound the mixing time using the spectral gap, defined as $\lambda = \min\{1 - \alpha_2, 1 - |\alpha_n|\}$

Mixing Time by Spectral Gap

Theorem. The ϵ-mixing time is upper bounded by $\frac{1}{\lambda}\log\left(\frac{n}{\epsilon}\right)$, where $\lambda = \min\{1-\alpha_2, 1-|\alpha_n|\}$.

For simplicity we give the proof only for d-regular graphs:

Let $v_1, v_2, \ldots, v_n$ be an orthonormal basis of A. Then $p_0 = c_1 v_1 + c_2 v_2 + \ldots + c_n v_n$, and

$$p_t = W^t p_0 = c_1 \alpha_1^t v_1 + c_2 \alpha_2^t v_2 + \cdots + c_n \alpha_n^t v_n$$

$$\|x\|_2 := \sqrt{x^\top x}$$

By Cauchy-Schwarz, $\|p_t - \pi\|_1 \le \sqrt{n}\|p_t - \pi\|_2$

$$\|p_t - \pi\|_2^2 = \|c_2\alpha_2^t v_2 + \cdots + c_n \alpha_n^t v_n\|_2^2 = c_2^2\alpha_2^{2t}\|v_2\|_2^2 + \cdots + c_n^2\alpha_n^{2t}\|v_n\|_2^2$$

$$= c_2^2\alpha_2^{2t} + \cdots + c_n^2\alpha_n^{2t} \le (1-\lambda)^{2t}(c_2^2 + \cdots + c_n^2)$$

Note that p_0 is a distribution, $\|p_0\|_2^2 = \sum_i p_0(i)^2 \le \sum_i p_0(i) = \|p_0\|_1 = 1$

So $\|p_t - \pi\|_2^2 \le (1-\lambda)^{2t} \Rightarrow \|p_t - \pi\|_1 \le \sqrt{n}(1-\lambda)^t \le \sqrt{n}\, e^{-\lambda t}$

When the spectral gap is a constant (i.e. $\lambda = \Omega(1)$), then the random walk converges in $O\left(\log\frac{n}{\epsilon}\right)$ steps.

When the graph is regular with $\lambda = \Omega(1)$, we can sample an almost uniform vertex in $O(\log n)$ steps.

Aside: for general graphs, $v_1, v_2, \ldots, v_n$ is an an orthonormal basis of $\mathcal{A}$, which incurs an extra $\sqrt{n}$ factor

Mixing Time for Lazy Random Walks

Theorem. The ϵ-mixing time is upper bounded by $\frac{1}{\lambda}\log\left(\frac{n}{\epsilon}\right)$, where λ is the spectral gap.

In lazy random walks, the spectral gap is simply $\frac{\lambda_2}{2}$, where λ_2 is the second eigenvalue of $\mathcal{L}$.

From Cheeger's inequality, we know that $\lambda_2 \geq \frac{\varphi(G)^2}{2}$.

Theorem. The ϵ-mixing time is of lazy random walks is upper bounded by $\frac{2}{\varphi(G)^2}\log\left(\frac{n}{\epsilon}\right)$.

This implies that lazy random walks mix fast in an **expander graph**, a very important result.

$\varphi(G) \approx$ constant

Further applications: Random Sampling

We have seen algorithmic questions that concerns finding a solution, deciding if a solution exists, finding an optimal solution etc

There is an entire area that concerns on a very different task: sampling a solution according to certain distributions

One of the most important applications for random walks is in designing fast sampling algorithms

More often than not, the main question concerns the mixing time of these random walks

For examples,

- Card shuffling
- Sampling a graph coloring
- Sampling a perfect matching in a bipartite graph
    - Approximating 0-1 permanent
- Sampling a spanning tree
    - Generating a maze for fun
- Approximate counting/inference

Card Shuffling

Say we have a deck of 52 cards. How do you get a random permutation using simple operations?

Let's say the simple operation is to choose a random card and put it at the top of the deck.

1. Does it converge to the uniform distribution of all permutations?
2. How many steps are enough to get an almost uniform distribution?

These questions can be understood as questions about random walks on the big "state" graph.

Then the first question is about **stationary distribution**, and the second question is about **mixing time**.

A famous result is 7 "riffle" shuffling will get an almost uniform permutation

- "Trailing the dovetail shuffle to its lair", by Dave Bayer and Persi Diaconis

Cut-off phenomenon

Graph coloring

Given an undirected graph with max. degree Δ and k colors

Goal: generate a k-coloring uniformly at random

This is presumably harder than deciding if there is a k-coloring

Nevertheless, the following random walk has a stationary distribution uniform over all k-colorings:

- Start with any k-coloring σ
- Pick a vertex v and a color c uniformly at random, recolor v with c if it is legal; otherwise do nothing;

This Markov chain is irreducible provided that $k \geq \Delta + 2$, and aperiodic

Conjecture: If $k \geq \Delta + 2$, the above random walk mixes in $\mathrm{poly}(n)$.

We will see a coupling argument assuming $k \geq 2\Delta + 1$

This is known as the Metropolis chain

Other chains: Glauber dynamics, Wang–Swendsen–Kotecký chain, …

Random Combinatorial Objects

We can design a Markov chain to generate a random combinatorial object efficiently.

Another simple example is the **basis-exchange walk** algorithm to generate a random spanning tree.

Sampling algorithms are known for many combinatorial objects (e.g. colorings, perfect matchings, discrepancy minimization)

It is usually easy to construct a Markov chain so that the limiting distribution is uniform

But it is much more difficult to prove that the mixing time is fast

There are books that just focus on mixing time:

- *Markov Chains and Mixing Times*, by Levin and Peres
- *Counting and Markov Chains*, by Jerrum

Many methods are developed, including coupling, conductance, second eigenvalue, etc

Cheeger's Inequality in Markov chains

It is interesting to see how Cheeger's inequality can be used.

When we want to bound $\phi(G)$, say in constructing expander graphs,

we can come up with algebraic constructions and bound λ_2 instead

When we want to bound λ_2, say in bounding the mixing time,

we can analyze combinatorial problems and bound $\phi(G)$ instead

An alternative perspective like this is exactly what makes it so powerful

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

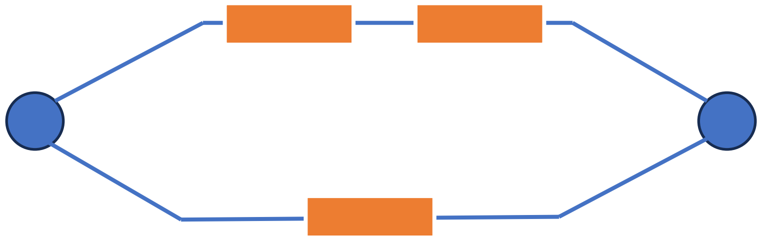
An electrical network is an undirected graph where every edge is a resistor of resistance r_e.

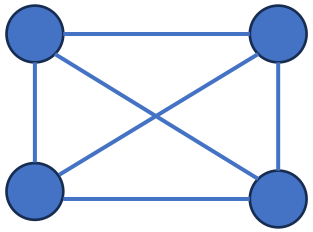
The electrical flows on this network are governed by two laws:

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 $i_{uv} = w_{uv}\big(\phi(u) - \phi(v)\big)$

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

Then the Laplacian can also be written as:

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

Then $\vec{b} = L\vec{\phi} = BWB^T\vec{\phi} = B\vec{\imath}$, 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, $\vec{\imath}$ 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

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

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

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

Proof:

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

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

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

Thompson's Principle

Theorem. $R_{\text{eff}}(s,t) \le \mathcal{E}(\vec{g})$ 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 $\overrightarrow{r'}$

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

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

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

3. Cover time: 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,

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

Proof: Firstly,

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

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 $\text{cover}(G) \le 2e^3 m \cdot R(G) \cdot \ln n + \frac{1}{n^2} n^3$

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