**18.409: An Algorithmist’s Toolkit**

**Lecture 6**

Jonathan Kelner

---

**Administrivia**

- Scribe notes
  - Remember to sign up if you haven’t already
  - If you end up dropping the class but have signed up, please let me know
  - Please make sure to send LaTeX source (not just PDF)
  - Please sign OCW license form (on course website) and give to Mayank
    - If you don’t want to sign it, please let me know
- Will post first problem set soon
  - Probably after lecture Thursday, maybe a little earlier
  - Due two weeks from Thursday
- Will post hints a few days after posting the pset
- As we approach the deadline, I may respond to requests for more hints on specific problems

---

**Diameters and Eigenvalues**

- So far we’ve related eigenvalues to cuts and mixing of random walks
- They give a lot more structural info about a graph
- Here, we’ll show they let us bound the diameter of graph
- **Intuition:**
  - Well-connected graphs have big $\lambda_2$
  - Well-connected graphs have small diameter
  - So graphs with big $\lambda_2$ should have small diameter
A First Bound

- From here on, let \( \delta \) be the diameter of \( G \)
- For this lecture, we’ll assume \( G \) is \( d \)-regular
  - Simplifies arguments, but doesn’t change results
- Let \( M = A/2d + I/2 \) be the random walk matrix
- Let \( \mu_2 \) be its second largest eigenvalue and let \( \lambda = 1 - \mu_2 \)
- **Claim:** For any \( G \),
  \[ \delta \leq \frac{\ln n}{\lambda} \]

Proof:

- We’ve already done it!
- Let \( u \) and \( v \) be the vertices that are farthest from each other
- Start a random walk at \( u \) and let \( p_t(v) \) be probability of walk being at \( v \) at time \( t \)
  \[ |p_t(v) - \pi(v)| \leq (1 - \lambda)^t \sqrt{\frac{d(v)}{\min_y d(y)}} = (1 - \lambda)^t \]
- If \( p_t(v) \geq 0 \), then \( \delta \leq t \)
- So if \( |p_t(v) - \pi(v)| < 1/n \), know \( \delta \leq t \)
- If \( t = \ln n / \lambda \),
  \[ (1 - \lambda)^t = (1 - \lambda)^{\ln n / \lambda} \leq \left( \frac{1}{e} \right)^{\ln n} = \frac{1}{n} \]

What Just Happened?

- If \((u,v)^{th} \) entry of \( A^k \) is nonzero, there’s a path of length \( \leq k \) from \( u \) to \( v \)
  - Replacing \( A \) with \( M \) doesn’t change this
- Let \( e_u, e_v \) be corresponding basis vectors
- We showed
  \[ \left| e_v^T M^k e_u - 1/n \right| < 1/n \]

Can We Do Better?

- Let \( p(x) = \sum_{j=1}^{k} c_j x^j \)
- Can apply to matrix \( M \):
  \[ p(M) = \sum_{j=1}^{k} c_j M^j \]
- If \( p(M) \) has no zero entries, \( \delta \leq k \)
  - Why?
- **Claim:** Suppose \( p(1)=1 \) and \( |p(\mu_i)| < 1/n \) for all \( i \geq 2 \). Then
  \[ \delta \leq k \]

  **Proof:** Blackboard
Magic Polynomials

- I assert the existence of magic polynomials with the following properties:
- **Claim:** For any \( t \in (0, 1) \), there exists a polynomial \( p_k(t) \) such that
  - \( p_k(t) \) is of degree \( k \)
  - \( p_k(t)(1) = 1 \)
  - \( |p_k(t)(x)| \leq 2 \left( 1 + \sqrt{2t} \right)^{-k} \) for any \( x \in [0, 1 - t] \)
- They’re derived from Chebyshev polynomials
- We’ll use them again later
- I’ll put more about them on a future problem set
- See Matlab for pictures

The Better Bound

- If we set \( t = \lambda \), we get a degree \( k \) poly p s.t.
  - \( p(0) = 1 \)
  - \( |p(x)| \leq 2 \left( 1 + \sqrt{2\lambda} \right)^{-k} \) for any \( x \in [0, \mu_2] \)
- If \( k = \left( \frac{1}{1 + \sqrt{2\lambda}} \right) \ln(2n) \),
  - can show \( p(x) < 1/n \) for all \( x \in [0, \mu_2] \)
- Gives:
  - **Theorem:**
    \[ \delta \leq \left( 1 + \frac{1}{\sqrt{2\lambda}} \right) \ln(2n) \]

What Just Happened?

- Why can’t this speed up mixing of every random walk?
- Foreshadows next unit on iterative linear algebra
- Suppose you have a sym. matrix \( M \) and want its biggest eigenvector (say normalized s.t. eigenval = 1)
- Easy way to get an approximate answer: compute \( M^k x \) for a pretty big \( k \) and a random \( x \)
  - Why?
    - Big eigenvector stays, smaller ones drop out
    - Error term falls off like \((1-\lambda)^k\), so takes about \(1/\lambda\) steps to get really close
    - Need to be a little careful to keep things from getting too big
- **We just found a much faster algorithm!**
  - Assuming we know some good bound on \( \lambda \)
  - **Compute** \( p_k^{(k)}(M)x \) **instead of** \( M^k x \)
  - Converges way faster
  - Obey nice three-term recurrence
    - Depends on last two vals, not just last one
  - More in a few lectures...
Expanders

- We’ll assume throughout that all graphs are d-regular
- Expanders turn up **everywhere**
  - Routing, lower bounds, derandomization, counterexamples to many conjectures, PCPs
  - More recently in algorithms for sparsest cut, graph connectivity, and more

Definitions

- Really need to study a family of d-regular graphs \((G_n)_n\) as \(n\) goes to infinity
- **Definition:** \((G_n)_n\) is an expander family if \(\lambda_2(G_n) \geq c\) for some constant \(c\) and for all \(n\)
- You should think of \(d\) as a constant

Expanders and Cuts

- We know \(\lambda_2/2 \leq \phi(G)\)
- So \(\phi(G) \geq c/2\)
- Means that any set \(S\) of vertices of size \(\leq n/2\) has at least \((c/2)|S|\) edges leaving it
- So no small cuts
- Also know \(\Theta(1)\phi^2/d \leq \lambda_2\)
  - So \(\phi(G) \leq \) the constant \(\Theta(1)\sqrt{cd}\)
  - So could alternatively define:
    - **Definition:** \((G_n)_n\) is an expander family if \(\phi(G) \geq c'\) for some constant \(c'\) and all \(n\)

Do They Exist?

- Natural question is for what parameters expanders exist (if any)
- **Claim:** For any \(G\), \(\lambda_2 \leq d - 2\sqrt{d - 1} + o(1)\)
- Won’t prove, but note \(\lambda_2 \leq d + o(1)\) is easy
- Actually exist families of graphs (called Ramanujan graphs) meeting this bound
- A random graph is an expander
**Expanders and Randomness**

- Expanders are all over study of randomness, but we'll just study one interesting property.
- We'll use $\mu_2 = d - \lambda_2$ to simplify formulas.
- Suppose you make a graph by randomly including each edge with probability $d/n$.
- Expected number of edges between any two sets $S$ and $T$ is $d|S||T|/n$.

**Claim (Expander Mixing Lemma):**

$$|e(S, T) - d|S||T|/n| \leq \frac{\mu_2}{n} \sqrt{|S||S||T||T|}$$

$$\leq \mu_2 \sqrt{\min(|S|, |S|) \cdot \min(|T|, |T|)}$$

**Proof:** See blackboard.

**Random walks on expanders mix in a logarithmic number of steps.**

**Vertex Expansion**

- For a set of vertices $X$, let $N(X)$ be its neighbor set.

**Claim:**

$$N(X) \geq \frac{d^2 |X|}{\mu^2 + (d^2 - \mu^2)|X|/n}$$

**Proof works by plugging the following into the expander mixing lemma:**

- Set 1: $X$
- Set 2: $Y = V \setminus (N(X) \cup X)$
- $e(X, Y) = 0$

**Algebra gets a little messy**

If $X/n$ is small and $\mu = 2\sqrt{d - 1}$, gives

$$N(X) \geq \frac{d}{4} |X|$$

- Note $N(X) \leq d|X|$, so quite strong.

**Random walks on expanders mix in a logarithmic number of steps.**

**Expander Mixing Lemma:**

$$|e(S, T) - d|S||T|/n| \leq \frac{\mu_2}{n} \sqrt{|S||S||T||T|}$$

$$\leq \mu_2 \sqrt{\min(|S|, |S|) \cdot \min(|T|, |T|)}$$

**Proof:** See blackboard.

**Vertex Expansion of Smaller Sets**

- We want to show bounds of form $|N(S)| \geq \gamma |S|$
- Say “vertex expansion of $G$ is $\geq \gamma$”
- Sometimes we’ll only care about expansion of smaller sets (e.g. size less than 0.01n)

**Definition:** $G$ is an $(\alpha, \beta)$-expander if all sets $S$ of size $\leq \alpha n$ have $|N(S)| \geq \beta |S|$

- Only makes sense if $\alpha \beta < 1$
- We showed Ramanujan graphs are $(\alpha, d/4)$ expanders for some const. $\alpha$

- Some applications need expansion $> d/2$ but with smaller (constant) $\alpha$
- These exist, but can’t prove better than $d/2$ with spectral techniques

**Explicit construction:**

- Random graphs work
- 2002: Capalbo, Reingold, Vadhan, and Wigderson gave explicit construction with expansion $d-o(1)$
Many of the applications of expanders use bipartite expanders
- Just expanders that are bipartite graphs
- Easier to show exist and such (homework problem!)

**Definition:** A d-regular bipartite graph is an \((\alpha, \beta)\)-expander if every set \(S\) on the left with \(|S| \leq \alpha n\) has \(N(S) \geq \beta |S|\)
- Whenever \(\alpha \beta < 1\), there exists some \(d\) s.t.
  almost all d-regular graphs on \(n\) nodes (\(n\) sufficiently large) are \((\alpha, \beta)\) expanders

**Routing Networks (cont.)**
- Easy to show that if want to route all permutations (input) \(\rightarrow\) (output), need \(\Omega(N \log N)\) nodes
- Can we meet this bound?
- Can we quickly find the routes in a decentralized way?
- One classical network: butterfly network
- Fails pretty badly
  - E.g., \(0000x_1 x_2 x_3 x_4 \rightarrow x_1 x_2 x_3 x_4 0000\) has cong. \(n^{1/2}\)

**Application: Routing Networks**
- Useful for communications (e.g., phones), massively parallel computers, networks, etc.
- \(N\) inputs, \(N\) outputs, want to talk to each other simultaneously
- Signals need to be routed along vertex disjoint paths (can’t overload routers)
- Called “a nonblocking routing network” (exact name varies)
  - Pictures from here on taken from paper by Arora, Leighton, and Maggs
More on the Butterfly Network

- Butterfly network has \( \log(N) \) layers
- \( i \)th layer has blocks of size \( N/2^i \)
- A block splits into “up” and “down” blocks in next layer
- Each node has one edge to “up” block and one edge to “down” block in next layer
- Improved version is d-multibutterfly:
  - Each node has \( d \) edges out to “up” and \( d \) edges to “down”
  - Graph from, e.g. N to N/2 “up” block form an \((\alpha, \beta)\)-expander, \( \beta > d/2, \quad \beta = 1/2\alpha \)

Finding the Paths

- Claim: Multibutterfly can route any permutation of \( 2\alpha N \) inputs to \( 2\alpha N \) outputs
- Proof will use Halls theorem:
- Theorem: A bipartite graph \( G \) has a perfect matching iff every set \( S \) of verts on left has at least \( |S| \) verts on right
  - i.e., if \( G \) is a \((1,1)\)-expander
- Will prove claim layer by layer
  - Just talk about first layer, others identical
- At most \( \alpha N/2 \) calls need to go to top half, \( \alpha N/2 \) to bottom half
  - Let \( S = \) set to go to top half, \( T = \) top half of next layer
- Need a perfect matching between \( S \) and \( T \)
  - \( |S| \leq \alpha N \), so guaranteed by Hall’s Theorem and expansion

- d-multibutterfly:
  - Each node has \( d \) edges out to “up” and \( d \) edges to “down”
  - Graph from, e.g. N to N/2 “up” block form an \((\alpha, \beta)\)-expander, \( \beta > d/2, \quad \beta = 1/2\alpha \)
  - Only \( 2N\alpha \) inputs/outputs
  - Actual inputs/outputs connected to \( \alpha/2 \) successive multibutterfly inputs/outputs
  - Picture of a 2-multibutterfly (hard to draw bigger \( d \))

Claim: Converges in \( O(\log n) \) steps

- In real life, don’t want to have to do complicated global computation for routing—need a simple distributed algorithm
- We’ll describe it for first layer; others are similar
- Suppose \( S = \) set of nodes on left that want to match, \( R = \) all nodes on right
  - Let \( S_1 = S \)
- Algorithm:
  - Repeat until \( S_i \) is empty
    - Every node in \( S_i \) sends a “proposal” to all neighbors in \( R \)
    - Every node in \( R \) getting exactly 1 proposal accepts it
    - Every node in \( S_i \) that got \( 1 \) accepted proposal picks one arbitrarily and matches to it
    - \( S_{i+1} = \) set of unmatched remaining nodes
- Claim: Converges in \( O(\log n) \) steps
Proof of Convergence Bound

- **Lemma 1:**
  \[
  \frac{|S_{t+1}|}{|S|} \leq 2(1 - \beta/d)
  \]
  - Guarantees convergence
  - Follows from:

- **Lemma 2:** For any set $S$ on left of size $\leq \alpha n$, number of verts on right with exactly 1 neighbor in $S$ is at least $(2\beta-d)|S|
  
- **Proof:** Blackboard

- Any node on left can receive at most $d$ acceptances, from which lemma 1 follows