CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

Lecture 1:

Expanders – robustness against failures

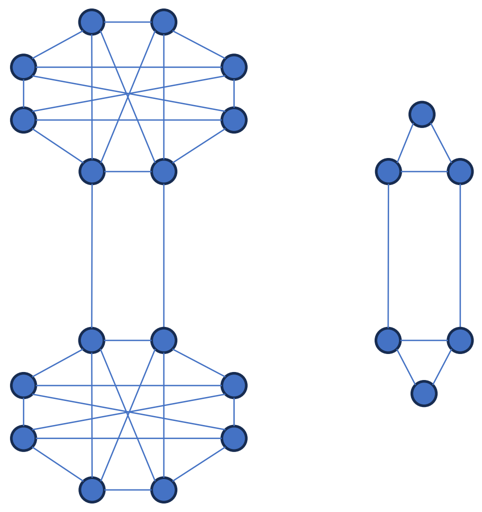
Measuring connectivity

- **Edge Connectivity**
  - Minimum number of <u>edges</u> whose removal disconnects the graph.
  - Captures global robustness against <u>edge failures</u>.

- **Vertex Connectivity**
  - Minimum number of <u>vertices</u> whose removal disconnects the graph.
  - Captures global robustness against <u>vertex failures</u>.

Measuring connectivity

- **Edge Connectivity**
  - Minimum number of edges whose removal disconnects the graph.
  - Captures global robustness against edge failures.
- **Vertex Connectivity**
  - Minimum number of vertices whose removal disconnects the graph.
  - Captures global robustness against vertex failures.



Key limitations of Edge & Vertex Connectivity:
- Ignore the size of the affected region.
- Larger cities → more roads

Note: We focus on **simple unweighted graphs**, but the definitions also extend to weighted graphs with multi-edges and self-loops.

Conductance

Consider a graph $G = (V, E)$.

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

It measures the size of a region.

Note: We focus on **simple unweighted graphs**, but the definitions also extend to weighted graphs with multi-edges and self-loops.

Conductance

Consider a graph $G = (V, E)$.

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

Conductance of a cut $(S, V \setminus S)$:

- $\Phi(S) = \frac{|E(S, V\setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V\setminus S)\}}$, where $E(A, B) = \{\, \{u, v\} \in E \mid u \in A \text{ and } v \in B \,\}$.

It measures the ratio between the size of a cut and the size of the region it separates.

Note: We focus on **simple unweighted graphs**, but the definitions also extend to weighted graphs with multi-edges and self-loops.

Conductance

Consider a graph $G = (V, E)$.

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

Conductance of a cut $(S, V \setminus S)$:

- $\Phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}$, where $E(A, B) = \{ \{u, v\} \in E \mid u \in A \text{ and } v \in B \}$.

It measures the ratio between the size of a cut and the size of the region it separates.

Some facts:

- If $A \cap B = \emptyset$, then $\mathrm{vol}(A \cup B) = \mathrm{vol}(A) + \mathrm{vol}(B)$.
- $0 \leq \Phi(S) \leq 1$.

Note: We focus on **simple unweighted graphs**, but the definitions also extend to weighted graphs with multi-edges and self-loops.

Conductance

Consider a graph $G = (V, E)$.

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

Conductance of a cut $(S, V \setminus S)$:

- $\Phi(S) = \dfrac{|E(S, V\setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V\setminus S)\}}$, where $E(A, B) = \{\, \{u, v\} \in E \mid u \in A \text{ and } v \in B \,\}$.

Conductance of a graph G:

- $\Phi(G) = \min_{S \subseteq V \,:\, S \neq V \text{ and } S \neq \emptyset} \Phi(S)$.

The minimum conductance over all cuts in the graph.

Intuition:

- What is the weakest cut in the network?
- Which region is easiest to separate relative to its size?

Note: We focus on **simple unweighted graphs**, but the definitions also extend to weighted graphs with multi-edges and self-loops.

Conductance

Consider a graph $G = (V, E)$.

Some facts:
- $0 \leq \Phi(G) \leq 1$.
- If $G = (V, E)$ is connected, then $\Phi(G) \geq 1/|E|$.

Volume of a vertex set S:
- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.

Conductance of a cut $(S, V \setminus S)$:
- $\Phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}$, where $E(A, B) = \{\, \{u, v\} \in E \mid u \in A \text{ and } v \in B \,\}$.

Conductance of a graph G:
- $\Phi(G) = \min_{S \subseteq V \,:\, S \neq V \text{ and } S \neq \emptyset} \Phi(S)$.

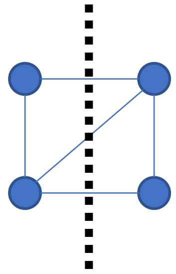
The minimum conductance over all cuts in the graph.

Intuition:
- What is the weakest cut in the network?
- Which region is easiest to separate relative to its size?

Conductance

Volume of a vertex set S:

- $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.



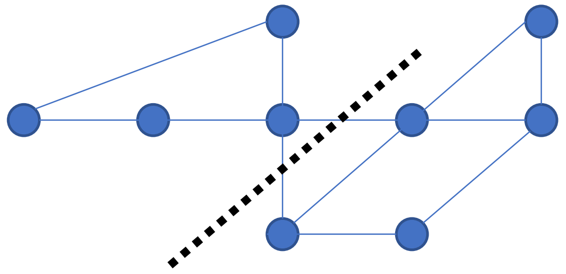
Conductance $= \frac{3}{\min\{2+3,\, 2+3\}} = \frac{3}{5} = 0.6$

Conductance of a cut $(S, V \setminus S)$:

- $\Phi(S) = \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S),\, \mathrm{vol}(V \setminus S)\}}$, where $E(A, B) = \{\, \{u, v\} \in E \mid u \in A \text{ and } v \in B \,\}$.

Conductance of a graph G:

- $\Phi(G) = \min_{S \subseteq V \,:\, S \neq V \text{ and } S \neq \emptyset} \Phi(S)$.



Conductance $= \frac{2}{\min\{4+2+2+2,\, 4+3+3+2+2\}} = \frac{2}{10} = 0.2$

Expanders

- A graph G is a $\boldsymbol{\phi}$**-expander** $\leftrightarrow \Phi(G) \geq \phi$.

Key takeaways:

- Conductance provides a size-sensitive notion of expansion.
- It better captures robustness and connectivity in real networks.

Comparing with vertex & edge connectivity

Expanders

Two $(n/2)$-cliques connected by an edge

- A graph G is a $\boldsymbol{\phi}$**-expander** $\leftrightarrow \Phi(G) \geq \phi$.

Graph	Conductance
Dumbbells	$\Theta(1/n^2)$
Paths	$\Theta(1/n)$
Cycles	$\Theta(1/n)$
$(\sqrt{n} \times \sqrt{n})$-grids	$\Theta(1/\sqrt{n})$
Hypercubes	$\Theta(1/\log n)$
Stars	$\Theta(1)$
Complete graphs / cliques	$\Theta(1)$

Key takeaways:

- Conductance provides a size-sensitive notion of expansion.
- It better captures robustness and connectivity in real networks.

Comparing with vertex & edge connectivity

Expanders

Two $(n/2)$-cliques connected by an edge

- A graph G is a $\boldsymbol{\phi}$**-expander** $\leftrightarrow \Phi(G) \geq \phi$.

Graph	Conductance
Dumbbells	$\Theta(1/n^2)$
Paths	$\Theta(1/n)$
Cycles	$\Theta(1/n)$
$(\sqrt{n} \times \sqrt{n})$-grids	$\Theta(1/\sqrt{n})$
Hypercubes	$\Theta(1/\log n)$
Stars	$\Theta(1)$
Complete graphs / cliques	$\Theta(1)$

Exercises!

Key takeaways:
- Conductance provides a size-sensitive notion of expansion.
- It better captures robustness and connectivity in real networks.

Comparing with vertex & edge connectivity

Robustness against edge deletions

- Suppose we remove a subset of edges $D \subseteq E$ from a ϕ-expander $G = (V, E)$.
  - The resulting graph is $G' = (V, E \setminus D)$.
  - We say that a connected component S of G' is **small** if $\mathrm{vol}_G(S) \le \frac{\mathrm{vol}_G(V)}{2}$.

Note: There can be at most one component that is not small.

Robustness against edge deletions

- Suppose we remove a subset of edges $D \subseteq E$ from a ϕ-expander $G = (V, E)$.
  - The resulting graph is $G' = (V, E \setminus D)$.
  - We say that a connected component S of G' is **small** if $\mathrm{vol}_G(S) \leq \frac{\mathrm{vol}_G(V)}{2}$.

- **Claim:** The total volume of small connected components of G' is $O\left(\frac{|D|}{\phi}\right)$.

In other words, deleting the edges $D \subseteq E$ can only disconnect a small subset of volume $O\left(\frac{|D|}{\phi}\right)$.

- Therefore, expanders are **robust against edge deletions**.

Useful in fault-tolerant distributed computing

Robustness against edge deletions

- Suppose we remove a subset of edges $D \subseteq E$ from a ϕ-expander $G = (V, E)$.
  - The resulting graph is $G' = (V, E \setminus D)$.
  - We say that a connected component S of G' is **small** if $\mathrm{vol}_G(S) \le \frac{\mathrm{vol}_G(V)}{2}$.

- **Claim:** The total volume of small connected components of G' is $O\left(\frac{|D|}{\phi}\right)$.

Proof of the claim – Case 1: There is a connected component that is not small.

- Let X be the union of all small connected components.

$$\boxed{\mathrm{vol}_G(X) \le \frac{\mathrm{vol}_G(V)}{2}} \longrightarrow \boxed{\phi \le \frac{|E(X,\ V \setminus X)|}{\mathrm{vol}_G(X)} \le \frac{|D|}{\mathrm{vol}_G(X)}} \longrightarrow \boxed{\mathrm{vol}_G(X) \le \frac{|D|}{\phi}}$$

Robustness against edge deletions

Proof of the claim – Case 2: All connected components are small.

- We show that there is $X =$ union of some small connected components such that:
  - $\frac{\mathrm{vol}_G(V)}{3} \leq \mathrm{vol}_G(X) \leq \frac{2\mathrm{vol}_G(V)}{3}$

Robustness against edge deletions

Proof of the claim – Case 2: All connected components are small.

- We show that there is X = union of <u>some</u> small connected components such that:
  - $\frac{\mathrm{vol}_G(V)}{3} \le \mathrm{vol}_G(X) \le \frac{2\mathrm{vol}_G(V)}{3}$

$$\phi \le \frac{|E(X,\ V \setminus X)|}{\min\{\mathrm{vol}_G(X), \mathrm{vol}_G(V \setminus X)\}} \le \frac{|D|}{\mathrm{vol}_G(V)/3}$$

$$\min\{\mathrm{vol}_G(X), \mathrm{vol}_G(V \setminus X)\} \ge \frac{\mathrm{vol}_G(V)}{3}$$

The total volume of small connected components $= \mathrm{vol}_G(V) \le \frac{3|D|}{\phi}$

Robustness against edge deletions

Proof of the claim – Case 2: All connected components are small.

- We show that there is $X =$ union of <u>some</u> small connected components such that:
  - $\frac{\mathrm{vol}_G(V)}{3} \leq \mathrm{vol}_G(X) \leq \frac{2\mathrm{vol}_G(V)}{3}$

Selection of $\boldsymbol{X}$:

- If there is a small connected component S with $\mathrm{vol}_G(S) \geq \frac{\mathrm{vol}_G(V)}{3}$:
  - We may select $X \leftarrow S$.
- Otherwise:
  - Initialize $X \leftarrow \emptyset$.
  - While ($\mathrm{vol}_G(X) < \frac{\mathrm{vol}_G(V)}{3}$)
    - $X \leftarrow$ union of X and some small connected component.

Robustness against edge deletions

- Suppose we remove a subset of edges $D \subseteq E$ from a ϕ-expander $G = (V, E)$.
  - The resulting graph is $G' = (V, E \setminus D)$.
  - We say that a connected component S of G' is **small** if $\mathrm{vol}_G(S) \le \frac{\mathrm{vol}_G(V)}{2}$.

- **Claim:** The total volume of small connected components of G' is $O\left(\frac{|D|}{\phi}\right)$.

Next: We will demonstrate an application of this result to sublinear–time algorithms.

Fast connectivity check under edge failures

- **Input:**
  - A ϕ-expander $G = (V, E)$.
  - A subset of edges $D \subseteq E$.
  - Two vertices s and t.
- **Goal:**
  - Decide whether s and t are connected in $G' = (V, E \setminus D)$.

  - Of course, the problem can be solved in linear time using BFS.
  - Here, however, our goal is to solve it in **sublinear time**.

Fast connectivity check under edge failures

- **Input:**
  - A ϕ-expander $G = (V, E)$.
  - A subset of edges $D \subseteq E$.
  - Two vertices s and t.
- **Goal:**
  - Decide whether s and t are connected in $G' = (V, E \setminus D)$.
- **Claim:**
  - The problem can be solved in $O\left(\frac{|D|}{\phi}\right)$ time.

Fast connectivity check under edge failures

It suffices to do BFS to explore up to $O\left(\frac{|D|}{\phi}\right)$ volume from both s and t.

The volume is sufficient for us to decide whether s and t belong to small connected components.

- **Input:**
  - A ϕ-expander $G = (V, E)$.
  - A subset of edges $D \subseteq E$.
  - Two vertices s and t.
- **Goal:**
  - Decide whether s and t are connected in $G' = (V, E \setminus D)$.
- **Claim:**
  - The problem can be solved in $O\left(\frac{|D|}{\phi}\right)$ time.

Fast connectivity check under edge failures

- **Input:**
  - A ϕ-expander $G = (V, E)$.
  - A subset of edges $D \subseteq E$.
  - Two vertices s and t.
- **Goal:**
  - Decide whether s and t are connected in $G' = (V, E \setminus D)$.
- **Claim:**
  - The problem can be solved in $O\left(\frac{|D|}{\phi}\right)$ time.

It suffices to do BFS to explore up to $O\left(\frac{|D|}{\phi}\right)$ volume from both s and t.

The volume is sufficient for us to decide whether s and t belong to small connected components.

If both s and t are not in small connected components, then the algorithm returns **YES**.

Otherwise, at least one of s and t belongs to a small connected component, and the volume is sufficient for us to search the entire component to decide whether s and t are connected.

Expanders have small diameter

- **Claim:** The diameter of any n-vertex ϕ-expander is $O\left(\frac{\log n}{\phi}\right)$.

Maximum shortest-path distance between any two vertices

Intuition:
- In a well-connected network:
  - Everything is only a few steps away.

Algorithmic Advantages:
- Quick broadcast and aggregation:
  - Messages reach the whole network rapidly.

Expanders have small diameter

- **Claim:** The diameter of any n-vertex ϕ-expander is $O\left(\frac{\log n}{\phi}\right)$.

We prove the claim using a **ball-growing** argument:

- $B(v,r) =$ the ball of radius r around v.
- It suffices to show that:
  - There exists $r^* \in O\left(\frac{\log n}{\phi}\right)$ such that $\mathrm{vol}\big(B(v,r^*)\big) > \frac{\mathrm{vol}(V)}{2}$ for every vertex v.

$\downarrow$

$B(u,r^*) \cap B(v,r^*) \neq \emptyset$ for any u and v.

$\downarrow$

Graph diameter $\leq 2r^* \in O\left(\frac{\log n}{\phi}\right)$.

Expanders have small diameter

- **Claim:** The <u>diameter</u> of any n-vertex ϕ-expander is $O\left(\frac{\log n}{\phi}\right)$.

We prove the claim using a **ball-growing** argument:
- $B(v,r) =$ the ball of radius r around v.

If $\mathrm{vol}\big(B(v,r)\big) \le \frac{\mathrm{vol}(V)}{2}$, then $\mathrm{vol}\big(B(v,r+1)\big) \ge (1+\phi)\mathrm{vol}\big(B(v,r)\big)$.

$\mathrm{vol}\big(B(v,r+1)\big)$
$\ge \mathrm{vol}\big(B(v,r)\big) + E\big(B(v,r), V \setminus B(v,r)\big)$
$\ge (1+\phi)\mathrm{vol}\big(B(v,r)\big)$

Expanders have small diameter

- **Claim:** The diameter of any n-vertex ϕ-expander is $O\left(\frac{\log n}{\phi}\right)$.

We prove the claim using a **ball-growing** argument:
- $B(v,r)$ = the ball of radius r around v.

$$\boxed{\text{If } \mathrm{vol}\big(B(v,r)\big) \le \frac{\mathrm{vol}(V)}{2}, \text{ then } \mathrm{vol}\big(B(v,r+1)\big) \ge (1+\phi)\mathrm{vol}\big(B(v,r)\big).}$$

$$\boxed{\text{If } \mathrm{vol}\big(B(v,r)\big) \le \frac{\mathrm{vol}(V)}{2}, \text{ then } \mathrm{vol}\big(B(v,r)\big) > e^{\frac{r\phi}{2}}.}$$

$$\mathrm{vol}\big(B(v,r)\big) \ge (1+\phi)^r \mathrm{vol}\big(B(v,0)\big) \ge (1+\phi)^r$$
$$> e^{\frac{r^2\phi}{r\phi+r}} \ge e^{\frac{r^2\phi}{2r}} = e^{\frac{r\phi}{2}}$$

$$\left(1+\frac{x}{y}\right)^y > e^{\frac{xy}{x+y}}$$

https://www.lkozma.net/inequalities_cheat_sheet/ineq.pdf

Expanders have small diameter

- **Claim:** The <u>diameter</u> of any n-vertex ϕ-expander is $O\left(\frac{\log n}{\phi}\right)$.

We prove the claim using a **ball-growing** argument:

- $B(v, r) =$ the ball of radius r around v.

If $\mathrm{vol}(B(v,r)) \le \frac{\mathrm{vol}(V)}{2}$, then $\mathrm{vol}(B(v,r+1)) \ge (1+\phi)\mathrm{vol}(B(v,r))$.

$\downarrow$

If $\mathrm{vol}(B(v,r)) \le \frac{\mathrm{vol}(V)}{2}$, then $\mathrm{vol}(B(v,r)) > e^{\frac{r\phi}{2}}$.

$\rightarrow \quad e^{\frac{r\phi}{2}} < \frac{\mathrm{vol}(V)}{2} \quad \rightarrow \quad r < \frac{2\ln\frac{\mathrm{vol}(V)}{2}}{\phi}$

$\downarrow$

There exists $r^* \in O\left(\frac{\log n}{\phi}\right)$ such that $\mathrm{vol}(B(v,r^*)) > \frac{\mathrm{vol}(V)}{2}$.

Construction of expanders

- **Stars** and **cliques** already have very good conductance:
  - $\Phi(G) \in \Omega(1)$

- However, they are undesirable in that they have <u>high-degree vertices</u>.

Construction of expanders

- **Stars** and **cliques** already have very good conductance:
  - $\Phi(G) \in \Omega(1)$
- However, they are undesirable in that they have <u>high-degree vertices</u>.
- Can we simultaneously achieve both of the following?
  - $\Phi(G) \in \Omega(1)$
  - Maximum degree $\in O(1)$
- **Hypercubes** nearly achieve this goal, up to a factor of $O(\log n)$.

Randomized construction 1

- Erdős-Renyi random graph $\mathcal{G}(n,p)$:
  - An n-vertex graph $G = (V, E)$ such that:
    - for each pair $\{u, v\}$ of vertices in V, $\{u, v\} \in E$ with probability p independently.

Randomized construction 1

- Erdős-Renyi random graph $\mathcal{G}(n,p)$:
  - An n-vertex graph $G = (V, E)$ such that:
    - for each pair $\{u,v\}$ of vertices in V, $\{u,v\} \in E$ with probability p independently.

- There is a choice of sampling probability $p \in \Theta\left(\frac{\log n}{n}\right)$ such that:
  - With probability $1 - 1/\mathrm{poly}(n)$,
    - The maximum degree of G is $O(\log n)$.
    - $\Phi(G) \in \Omega(1)$.

Intuitively, this means that <u>almost all</u> graphs are good expanders!

Randomized construction 1

- Erdős-Renyi random graph $\mathcal{G}(n, p)$:
  - An n-vertex graph $G = (V, E)$ such that:
    - for each pair $\{u, v\}$ of vertices in V, $\{u, v\} \in E$ with probability p independently.

- There is a choice of sampling probability $p \in \Theta\left(\frac{\log n}{n}\right)$ such that:
  - With probability $1 - 1/\mathrm{poly}(n)$,
    - The maximum degree of G is $O(\log n)$.
    - $\Phi(G) \in \Omega(1)$.

Intuitively, this means that <u>almost all</u> graphs are good expanders!

This is an **exercise**:

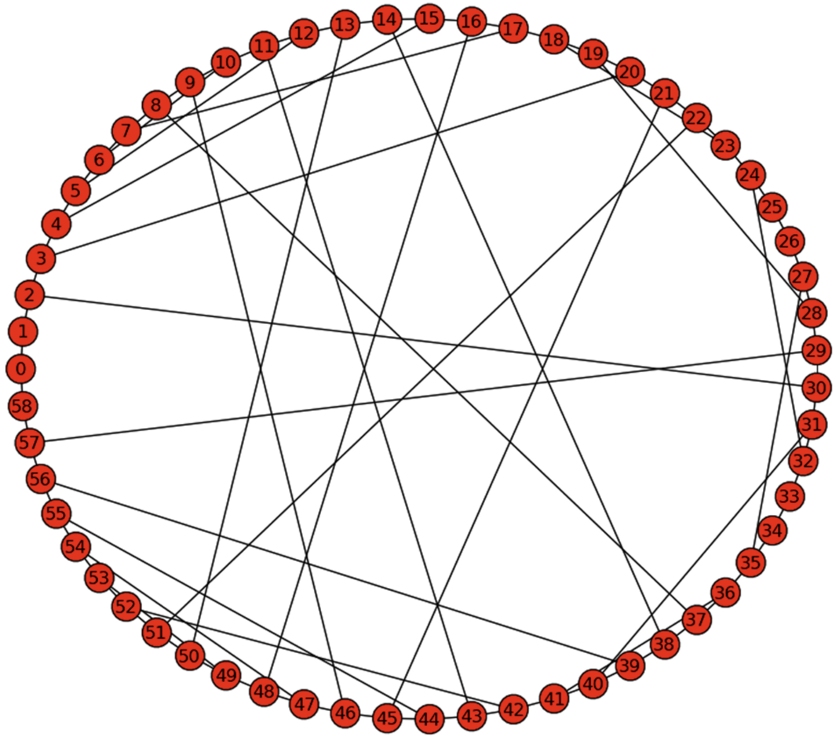
- Apply a Chernoff bound for every cut.
- Sum up the error probability.

Randomized construction 2

- Generate d random perfect matching over n vertices.
- Take the union of all edges in the matchings.
- There exists a constant d such that the resulting graph G satisfies $\Phi(G) \in \Omega(1)$ with probability $1 - 1/\mathrm{poly}(n)$.

https://lucatrevisan.github.io/teaching/expanders2016/lecture19.pdf

Deterministic construction



- Let p be a prime.
- Define the graph $G_p = \left(V_p, E_p\right)$ as follows.
  - $V_p = \{0, 1, \dots, p-1\}$
  - Each vertex $a \in V_p \setminus \{0\}$ is connected to:
    - $a - 1 \bmod p$
    - $a + 1 \bmod p$
    - a^{-1} (multiplicative inverse: the unique element $a^{-1} \in V_p$ such that $a \cdot a^{-1} \bmod p = 1$)
  - The vertex 0 is connected to 1, to $p-1$, and has a self-loop.
- $\Phi\left(G_p\right) \in \Omega(1)$
- The degree of every vertex in G_p equals 3.

https://lucatrevisan.github.io/teaching/expanders2016/lecture16.pdf

Variations

- Given a vertex set $S \subseteq V$, define:
  - $\partial_{\text{in}}(S)$ = the set of vertices in S adjacent to $V \setminus S$.
  - $\partial_{\text{out}}(S)$ = the set of vertices in $V \setminus S$ adjacent to S.

- **Edge expansion:**
  - $h(G) = \min\limits_{0<|S|\leq\frac{n}{2}} \frac{|E(V, V\setminus S)|}{|S|}$
- **Vertex expansion:**
  - $h_{\text{in}}(G) = \min\limits_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{in}}(S)|}{|S|}$
  - $h_{\text{out}}(G) = \min\limits_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{out}}(S)|}{|S|}$

Variations

Claim: If the maximum degree Δ of G is $O(1)$, then $\Phi(G)$, $h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are within a constant factor of each other.

- Given a vertex set $S \subseteq V$, define:
  - $\partial_{\text{in}}(S) =$ the set of vertices in S adjacent to $V \setminus S$.
  - $\partial_{\text{out}}(S) =$ the set of vertices in $V \setminus S$ adjacent to S.

- **Edge expansion:**
  - $h(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|E(V, V\setminus S)|}{|S|}$

- **Vertex expansion:**
  - $h_{\text{in}}(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{in}}(S)|}{|S|}$
  - $h_{\text{out}}(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{out}}(S)|}{|S|}$

Variations

Claim: If the maximum degree Δ of G is $O(1)$, then $\Phi(G)$, $h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are within a constant factor of each other.

- Given a vertex set $S \subseteq V$, define:
  - $\partial_{\text{in}}(S) =$ the set of vertices in S adjacent to $V \setminus S$.
  - $\partial_{\text{out}}(S) =$ the set of vertices in $V \setminus S$ adjacent to S.
- **Edge expansion:**
  - $h(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|E(S,V\setminus S)|}{|S|}$
- **Vertex expansion:**
  - $h_{\text{in}}(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{in}}(S)|}{|S|}$
  - $h_{\text{out}}(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|\partial_{\text{out}}(S)|}{|S|}$

What about $\Phi(G)$?

$h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are within a constant factor of each other.

- $|\partial_{\text{in}}(S)| \leq |E(V, V \setminus S)| \leq \Delta|\partial_{\text{in}}(S)|$
- $|\partial_{\text{out}}(S)| \leq |E(V, V \setminus S)| \leq \Delta|\partial_{\text{out}}(S)|$

Variations

Claim: If the maximum degree Δ of G is $O(1)$, then $\Phi(G)$, $h(G)$, $h_{\mathrm{in}}(G)$, $h_{\mathrm{out}}(G)$ are within a constant factor of each other.

- It remains to show that the two parameters are within an $O(1)$-factor:
  - $h(G) = \min_{0<|S|\le\frac{n}{2}} \frac{|E(S,V\setminus S)|}{|S|}$
  - $\Phi(G) = \min_{S\subseteq V\,:\,S\neq V \text{ and } S\neq\emptyset} \frac{|E(S,V\setminus S)|}{\min\{\mathrm{vol}(S),\mathrm{vol}(V\setminus S)\}} = \min_{0<|S|\le\frac{n}{2}} \frac{|E(S,V\setminus S)|}{\min\{\mathrm{vol}(S),\mathrm{vol}(V\setminus S)\}}$

Variations

Claim: If the maximum degree Δ of G is $O(1)$, then $\Phi(G)$, $h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are within a constant factor of each other.

- It remains to show that the two parameters are within an $O(1)$-factor:
  - $h(G) = \min_{0<|S|\le\frac{n}{2}} \frac{|E(S, V\setminus S)|}{|S|}$
  - $\Phi(G) = \min_{S\subseteq V\,:\,S\neq V \text{ and } S\neq\emptyset} \frac{|E(S, V\setminus S)|}{\min\{\text{vol}(S), \text{vol}(V\setminus S)\}} = \min_{0<|S|\le\frac{n}{2}} \frac{|E(S, V\setminus S)|}{\min\{\text{vol}(S), \text{vol}(V\setminus S)\}}$
- To do so, it suffices to show that for any $0 < |S| \le \frac{n}{2}$, the two parameters are within an $O(1)$-factor:
  - $|S|$
  - $\min\{\text{vol}(S), \text{vol}(V \setminus S)\}$

Variations

Claim: If the maximum degree Δ of G is $O(1)$, then $\Phi(G)$, $h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are within a constant factor of each other.

- It remains to show that the two parameters are within an $O(1)$-factor:
  - $h(G) = \min_{0<|S|\leq\frac{n}{2}} \frac{|E(S, V\setminus S)|}{|S|}$
  - $\Phi(G) = \min_{S\subseteq V\,:\,S\neq V \text{ and } S\neq\emptyset} \frac{|E(S, V\setminus S)|}{\min\{\text{vol}(S), \text{vol}(V\setminus S)\}} = \min_{0<|S|\leq\frac{n}{2}} \frac{|E(S, V\setminus S)|}{\min\{\text{vol}(S), \text{vol}(V\setminus S)\}}$
- To do so, it suffices to show that for any $0 < |S| \leq \frac{n}{2}$, the two parameters are within an $O(1)$-factor:
  - $|S|$
  - $\min\{\text{vol}(S), \text{vol}(V \setminus S)\}$
- Indeed, $|S| \leq \min\{|S|, |V \setminus S|\} \leq \min\{\text{vol}(S), \text{vol}(V \setminus S)\} \leq \text{vol}(S) \leq \Delta|S|$.

- This step requires that there is no isolated vertex.
- If there is an isolated vertex, then all the parameters $\Phi(G)$, $h(G)$, $h_{\text{in}}(G)$, $h_{\text{out}}(G)$ are zero.

Vertex expansion vs. conductance

- We have shown the following results for ϕ-expanders:
  - Robustness against <u>edge</u> deletions.
  - Small diameter.

- **Exercise:** Extend the above results to <u>graphs with high vertex expansion</u>:
  - Robustness against <u>vertex</u> deletions.
  - Small diameter.

Vertex expansion vs. conductance

- Vertex expansion and conductance can be very different!

There is an n-vertex graph G with
- $\Phi(G) \in \Omega(1)$
- $h_{\text{in}}(G) \in O\left(\frac{1}{n}\right)$
- $h_{\text{out}}(G) \in O\left(\frac{1}{n}\right)$

There is an n-vertex graph G with
- $\Phi(G) \in O\left(\frac{1}{n}\right)$
- $h_{\text{in}}(G) \in \Omega(1)$
- $h_{\text{out}}(G) \in \Omega(1)$

Vertex expansion vs. conductance

- Vertex expansion and conductance can be very different!

There is an n-vertex graph G with
- $\Phi(G) \in \Omega(1)$
- $h_{\text{in}}(G) \in O\left(\frac{1}{n}\right)$
- $h_{\text{out}}(G) \in O\left(\frac{1}{n}\right)$

Star graphs

There is an n-vertex graph G with
- $\Phi(G) \in O\left(\frac{1}{n}\right)$
- $h_{\text{in}}(G) \in \Omega(1)$
- $h_{\text{out}}(G) \in \Omega(1)$

Two $(n/2)$-cliques with a perfect matching between them

Outlook

- So far, we mostly talk about nice properties of expanders, focusing on the robustness against failures.

- **Next:**
  - How to compute the conductance of a graph efficiently? ← We will do this first.
  - What are the applications of expanders in algorithm design?

References

- **Main reference:**
  - Lecture 1.2 of https://sites.google.com/site/thsaranurak/teaching/Expander

- **Additional/optional reading:**
  - More about expander graphs:
    - https://en.wikipedia.org/wiki/Expander_graph
  - An application in distributed computing:
    - Ghosh, Bhaskar, et al. "Tight analyses of two local load balancing algorithms." SIAM Journal on Computing 29.1 (1999): 29-64.
    - https://epubs.siam.org/doi/10.1137/S0097539795292208

CS5275 – The Algorithm Designer's Toolkit (S2 AY2025/26)

Lecture 2:

Expanders – approximating conductance

Computing conductance

- **Racall:**
  - Conductance of a graph measures how well-connected it is.
  - Graphs with high conductance are robust against failures.

- It is **NP-hard** to compute the conductance <u>exactly</u>.

Computing conductance

- **Racall:**
  - Conductance of a graph measures how well-connected it is.
  - Graphs with high conductance are robust against failures.

- It is **NP-hard** to compute the conductance exactly.

- **Next:** $O(\log n)$-approximation in polynomial time. The Leighton–Rao algorithm

Computing an estimate ϕ such that $\Phi(G) \leq \phi \leq \alpha \cdot \Phi(G)$ for some $\alpha \in O(\log n)$.

Tool 1: LP relaxation

- A **general strategy** for **approximation algorithms** for an NP-hard problem:
  - Relax it to a problem that can be solved in polynomial time.
  - Apply an efficient algorithm to solve the relaxed problem.
  - Transform the solution back to a valid solution of the original problem.

Tool 1: LP relaxation

- A **general strategy** for **approximation algorithms** for an NP-hard problem:
  - Relax it to a <u>problem</u> that can be solved in polynomial time.

    In our case, linear programming (LP).
  - Apply an efficient algorithm to solve the relaxed problem.
  - <u>Transform</u> the solution back to a valid solution of the original problem.

    This step incurs some approximation error.

- Maximizing/minimizing a linear objective function.
  - Subject to linear constraints (equalities and inequalities).
- Variables take real values.

Metric space

- A **metric space** is an ordered pair (X, d), where
  - X is a set.
  - $d: X \times X \to \mathbb{R}_{\geq 0}$ is a **distance function (metric)** on X meeting the conditions:
    - $d(x, x) = 0$
    - $d(x, y) = d(y, x)$ (symmetry)
    - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Note: In mathematics, the definition of a metric space includes an additional requirement:

- if $x \neq y$, then $d(x, y) \neq 0$.

Without it, it is known as a pseudometric or semimetric space.

In computer science, it is more common to simply drop this requirement from the definition of metric spaces.

Metric space

- A **metric space** is an ordered pair (X, d), where
  - X is a set.
  - $d: X \times X \to \mathbb{R}_{\geq 0}$ is a **distance function (metric)** on X meeting the conditions:
    - $d(x, x) = 0$
    - $d(x, y) = d(y, x)$ (symmetry)
    - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Connections to LP:
- They are linear constraints.

Metric space

- A **metric space** is an ordered pair (X, d), where
  - X is a set.
  - $d: X \times X \to \mathbb{R}_{\geq 0}$ is a **distance function (metric)** on X meeting the conditions:
    - $d(x, x) = 0$
    - $d(x, y) = d(y, x)$ (symmetry)
    - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Closure under addition:
- If d_1 and d_2 are metrics on the same set X, then:
  - $d_1 + d_2$ is also a metric on X.

Closure under scalar multiplication:
- If d is a metric on X and $\alpha \geq 0$ is a scalar, then:
  - $\alpha \cdot d$ is also a metric on X.

Line metric

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define
$$d(x, y) = |f(x) - f(y)|$$

ℓ_1 metric

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define
$$d(x, y) = |f(x) - f(y)|$$

- **ℓ_1 metric:**
  - Given a function $f\colon X \to \mathbb{R}^k$, define
$$d(x, y) = \|f(x) - f(y)\|_1 = \sum_{i=1}^{k} |f_i(x) - f_i(y)|$$

ℓ_1 metric

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define
$$d(x, y) = |f(x) - f(y)|$$

- **ℓ_1 metric:**
  - Given a function $f\colon X \to \mathbb{R}^k$, define
$$d(x, y) = \|f(x) - f(y)\|_1 = \sum_{i=1}^{k} |f_i(x) - f_i(y)|$$

Observation: A k-dimensional ℓ_1 metric is the sum of k line metrics.

Cut metric

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define
$$d(x,y) = |f(x) - f(y)|$$

- **Cut metric:**
  - Given a subset $S \subseteq X$, define
$$d(x,y) = \begin{cases} 1 & \text{if } \big((x \in S) \wedge (y \in X \setminus S)\big) \vee \big((x \in X \setminus S) \wedge (y \in S)\big) \\ 0 & \text{otherwise} \end{cases}$$

Cut metric

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define
$$d(x, y) = |f(x) - f(y)|$$

- **Cut metric:**
  - Given a subset $S \subseteq X$, define
$$d(x, y) = \begin{cases} 1 & \text{if } \big((x \in S) \wedge (y \in X \setminus S)\big) \vee \big((x \in X \setminus S) \wedge (y \in S)\big) \\ 0 & \text{otherwise} \end{cases}$$

Observation: Cut metric is a special case of line metric, by setting $f(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$

Cut metric

Observation: A line metric is a linear combination of $|X| - 1$ cut metrics.

Order $X = \{x_1, \dots, x_n\}$ so that $f(x_1) \leq \cdots \leq f(x_n)$.

- Set $\alpha_i = f(x_{i+1}) - f(x_i)$
- Set d_i to be a cut metric with $S = \{x_1, \dots, x_i\}$.
- Then $d = \sum_{i=1}^{n-1} \alpha_i d_i$.

- **Line metric:**
  - Given a function $f\colon X \to \mathbb{R}$, define

$$d(x, y) = |f(x) - f(y)|$$

- **Cut metric:**
  - Given a subset $S \subseteq X$, define

$$d(x, y) = \begin{cases} 1 & \text{if } \big((x \in S) \wedge (y \in X \setminus S)\big) \vee \big((x \in X \setminus S) \wedge (y \in S)\big) \\ 0 & \text{otherwise} \end{cases}$$

Observation: Cut metric is a special case of line metric, by setting $f(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$

Cut metric

Observation: A line metric is a linear combination of $|X| - 1$ cut metrics.

$+$

Observation: A k-dimensional ℓ_1 metric is the sum of k line metrics.

$\downarrow$

Observation: A k-dimensional ℓ_1 metric is a linear combination of $k(|X| - 1)$ cut metrics.

Shortest path metric

- **Shortest path metric:**
  - Given a graph $G = (X, E)$, define

$$d(x, y) = \text{shortest path distance between } x \text{ and } y$$

Tool 2: metric embedding

- **Motivation:** Computing the distance in some metrics is more expensive than in other metrics.

Compare cut metric with shortest path metric.

Tool 2: metric embedding

- **Motivation:** Computing the distance in some metrics is more expensive than in other metrics.

Embed a complicated metric into a simpler one with **small distortion**.

Greatly reduce the cost of distance computation, at the expense of only a **small approximation error**.

Tool 2: metric embedding

- **Motivation:** Computing the distance in some metrics is more expensive than in other metrics.

Embed a complicated metric into a simpler one with **small distortion**.

↓

Greatly reduce the cost of distance computation, at the expense of only a **small approximation error**.

Next: Bourgain's theorem allows us to embed any n-point metric into an $O(\log^2 n)$-dimensional ℓ_1 metric space with $O(\log n)$ distortion.

A linear combination of $O(n \log^2 n)$ **cut metrics**!

Tool 2: metric embedding

- **Bourgain's theorem for metric embeddings:**
  - Given any n-point metric space (X, d),
    - there is an embedding $f\colon X \to \mathbb{R}^{O(\log^2 n)}$ such that for any $x, y \in X$,
      - $\|f(x) - f(y)\|_1 \le d(x,y) \in O(\log n) \cdot \|f(x) - f(y)\|_1$

If $d(x,y)$ is given for all $x, y \in X$, then the mapping f can be computed in (randomized) **polynomial time**.

A quick application

- **Bourgain's theorem for metric embeddings:**
  - Given any n-point metric space (X, d),
    - there is an embedding $f\colon X \to \mathbb{R}^{O(\log^2 n)}$ such that for any $x, y \in X$,
      - $\|f(x) - f(y)\|_1 \leq d(x, y) \in O(\log n) \cdot \|f(x) - f(y)\|_1$

Apply Bourgain's theorem to the shortest path metric.

We can label each vertex with a vector of $O(\log^2 n)$ values so that:

- Given any two vertices u and v:
  - We can obtain an $O(\log n)$-approximation of $\mathrm{dist}(u, v)$ in **only** $O(\log^2 n)$ time by reading the labels of u and v.

Generalized conductance

with positive edge weights

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

$$\Phi(G) = \min_{S\,:\,S \neq V \text{ and } S \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V \setminus S)\}}$$

$$\Phi(G, H) = \min_{S\,:\,E_H(S, V \setminus S) \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{|E_H(S, V \setminus S)|}$$

Here we slightly abuse the notation to write:

- $|E_G(S, V \setminus S)| = \sum_{e \in E_G(S, V \setminus S)} w_G(e)$
- $|E_H(S, V \setminus S)| = \sum_{e \in E_H(S, V \setminus S)} w_H(e)$
- $\mathrm{vol}_G(S) = \sum_{v \in S} \sum_{e\,:\,v \in e} w_G(e)$

Generalized conductance

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

$$\Phi(G) = \min_{S\,:\,S \neq V \text{ and } S \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V \setminus S)\}} \qquad \Phi(G, H) = \min_{S\,:\,E_H(S, V \setminus S) \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{|E_H(S, V \setminus S)|}$$

Claim: Given G, there is a choice of H such that $\Phi(G)$ and $\Phi(G, H)$ are within a constant factor.

$\downarrow$

$O(\log n)$-approximation of $\Phi(G, H)$ $\longrightarrow$ $O(\log n)$-approximation of $\Phi(G)$

Generalized conductance

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

$$\Phi(G) = \min_{S\,:\,S\neq V \text{ and } S\neq\emptyset} \frac{|E_G(S, V\setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V\setminus S)\}} \qquad \Phi(G,H) = \min_{S\,:\,E_H(S,V\setminus S)\neq\emptyset} \frac{|E_G(S, V\setminus S)|}{|E_H(S, V\setminus S)|}$$

Claim: Given G, there is a choice of H such that $\Phi(G)$ and $\Phi(G,H)$ are within a constant factor.

$$\frac{|E_G(S, V\setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V\setminus S)\}} \xleftrightarrow{\text{within a constant factor}} \frac{|E_G(S, V\setminus S)|}{\left(\frac{\mathrm{vol}_G(S)\cdot\mathrm{vol}_G(V\setminus S)}{\mathrm{vol}_G(V)}\right)}$$

$$\frac{\mathrm{vol}_G(S)\cdot\mathrm{vol}_G(V\setminus S)}{\mathrm{vol}_G(V)} = \min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V\setminus S)\}\cdot \overbrace{\frac{\max\{\mathrm{vol}_G(S), \mathrm{vol}_G(V\setminus S)\}}{\mathrm{vol}_G(V)}}^{\in\,[0.5,1]}$$

Generalized conductance

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

$$\Phi(G) = \min_{S\,:\,S \neq V \text{ and } S \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V \setminus S)\}} \qquad \Phi(G,H) = \min_{S\,:\,E_H(S, V\setminus S) \neq \emptyset} \frac{|E_G(S, V \setminus S)|}{|E_H(S, V \setminus S)|}$$

Claim: Given G, there is a choice of H such that $\Phi(G)$ and $\Phi(G,H)$ are within a constant factor.

$$\frac{|E_G(S, V \setminus S)|}{\min\{\mathrm{vol}_G(S), \mathrm{vol}_G(V \setminus S)\}} \xleftrightarrow{\text{within a constant factor}} \frac{|E_G(S, V \setminus S)|}{\left(\frac{\mathrm{vol}_G(S) \cdot \mathrm{vol}_G(V \setminus S)}{\mathrm{vol}_G(V)}\right)} \xleftrightarrow{w_H(u,v) = \frac{\deg_G(u) \cdot \deg_G(v)}{\mathrm{vol}_G(V)}} \frac{|E_G(S, V \setminus S)|}{|E_H(S, V \setminus S)|}$$

Connection to cut metric

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

- Let d_S be a cut metric for the cut S of V.

$$\Phi(G,H) = \min_{S\,:\,E_H(S,V\setminus S)\neq\emptyset} \frac{|E_G(S,V\setminus S)|}{|E_H(S,V\setminus S)|}$$

equivalent

$$\Phi(G,H) = \min_{S\,:\,\sum_{u,v} w_H(u,v)\cdot d_S(u,v)\neq 0} \frac{\sum_{u,v} w_G(u,v)\cdot d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot d_S(u,v)}$$

Connection to cut metric

- Let $G = (V, E_G)$ and $H = (V, E_H)$ be weighted graphs on the same vertex set V.

- Let d_S be a cut metric for the cut S of V.

$$\Phi(G,H) = \min_{S\,:\,E_H(S,V\setminus S)\neq\emptyset} \frac{|E_G(S,V\setminus S)|}{|E_H(S,V\setminus S)|}$$

equivalent

$$\Phi(G,H) = \min_{S\,:\,\sum_{u,v} w_H(u,v)\cdot d_S(u,v)\neq 0} \frac{\sum_{u,v} w_G(u,v)\cdot d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot d_S(u,v)}$$

equivalent

$\Phi(G,H)$

Minimize:
- $\sum_{u,v} w_G(u,v)\cdot d(u,v)$

Subject to:
- $\sum_{u,v} w_H(u,v)\cdot d(u,v) = 1$
- d is a cut metric scaled by a real value.

LP relaxation

Leighton–Rao

$\mathrm{LR}(G, H)$

Minimize:

- $\sum_{u,v} w_G(u,v) \cdot d(u,v)$

Subject to:

- $\sum_{u,v} w_H(u,v) \cdot d(u,v) = 1$
- d is a **metric**, i.e.,
  - $d(x,x) = 0\ \forall x$
  - $d(x,y) = d(y,x)\ \forall x,y$
  - $d(x,z) \leq d(x,y) + d(y,z)\ \forall x,y,z$

$\Phi(G, H)$

Minimize:

- $\sum_{u,v} w_G(u,v) \cdot d(u,v)$

Subject to:

- $\sum_{u,v} w_H(u,v) \cdot d(u,v) = 1$
- d is a cut metric scaled by a real value.

LP relaxation $\longrightarrow$

LP relaxation

This is a <u>relaxation</u> in the sense that:

Any cut metric scaled by a real value is a metric. → $\mathrm{LR}(G,H)$ considers a wider range of search space than $\Phi(G,H)$. → $\mathrm{LR}(G,H) \le \Phi(G,H)$

$\Phi(G,H)$

Minimize:
- $\sum_{u,v} w_G(u,v) \cdot d(u,v)$

Subject to:
- $\sum_{u,v} w_H(u,v) \cdot d(u,v) = 1$
- d is a cut metric scaled by a real value.

LP relaxation →

$\mathrm{LR}(G,H)$

Minimize:
- $\sum_{u,v} w_G(u,v) \cdot d(u,v)$

Subject to:
- $\sum_{u,v} w_H(u,v) \cdot d(u,v) = 1$
- d is a **metric**, i.e.,
  - $d(x,x) = 0\ \forall x$
  - $d(x,y) = d(y,x)\ \forall x,y$
  - $d(x,z) \le d(x,y) + d(y,z)\ \forall x,y,z$

Approximation ratio

An $O(\log n)$-approximation of conductance can be computed in polynomial time by computing $\mathrm{LR}(G, H)$ using a linear programming solver.

Claim: $\mathrm{LR}(G, H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$

$\mathrm{LR}(G, H) \leq \Phi(G, H)$

$\Phi(G, H)$

Minimize:
- $\sum_{u,v} w_G(u, v) \cdot d(u, v)$

Subject to:
- $\sum_{u,v} w_H(u, v) \cdot d(u, v) = 1$
- d is a cut metric scaled by a real value.

LP relaxation $\longrightarrow$

$\mathrm{LR}(G, H)$

Minimize:
- $\sum_{u,v} w_G(u, v) \cdot d(u, v)$

Subject to:
- $\sum_{u,v} w_H(u, v) \cdot d(u, v) = 1$
- d is a **metric**, i.e.,
  - $d(x, x) = 0\ \forall x$
  - $d(x, y) = d(y, x)\ \forall x, y$
  - $d(x, z) \leq d(x, y) + d(y, z)\ \forall x, y, z$

Proof of the claim

Claim: $\mathrm{LR}(G,H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$

$\mathrm{LR}(G,H)$

Minimize:

- $\sum_{u,v} w_G(u,v) \cdot d(u,v)$

Subject to:

- $\sum_{u,v} w_H(u,v) \cdot d(u,v) = 1$
- d is a **metric**, i.e.,
  - $d(x,x) = 0 \ \forall x$
  - $d(x,y) = d(y,x) \ \forall x,y$
  - $d(x,z) \leq d(x,y) + d(y,z) \ \forall x,y,z$

$$\mathrm{LR}(G,H) = \frac{\sum_{u,v} w_G(u,v) \cdot d_{\mathrm{OPT}}(u,v)}{\sum_{u,v} w_H(u,v) \cdot d_{\mathrm{OPT}}(u,v)}$$

d_{OPT} = an optimal metric of

Proof of the claim

Claim: $\mathrm{LR}(G, H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$

Apply **Bourgain's theorem** to d_{OPT}.

There is an $O(\log^2 n)$-dimensional ℓ_1 metric d^* such that:
$d^*(u, v) \leq d_{\mathrm{OPT}}(u, v) \in O(\log n) \cdot d^*(u, v)$

$$\mathrm{LR}(G, H) = \frac{\sum_{u,v} w_G(u, v) \cdot d_{\mathrm{OPT}}(u, v)}{\sum_{u,v} w_H(u, v) \cdot d_{\mathrm{OPT}}(u, v)}$$

Proof of the claim

Recall: d^* is a linear combination of $O(n \log^2 n)$ cut metrics.

There is a collection C of $O(n \log^2 n)$ cuts and coefficients $\{\alpha_S \mid S \in C\}$ such that:

$$d^* = \sum_{S \in C} \alpha_S \, d_S,$$

where d_S is the cut metric of the cut S.

Apply **Bourgain's theorem** to d_{OPT}.

There is an $O(\log^2 n)$-dimensional ℓ_1 metric d^* such that:

$$d^*(u,v) \leq d_{\mathrm{OPT}}(u,v) \in O(\log n) \cdot d^*(u,v)$$

$$\mathrm{LR}(G,H) = \frac{\sum_{u,v} w_G(u,v) \cdot d_{\mathrm{OPT}}(u,v)}{\sum_{u,v} w_H(u,v) \cdot d_{\mathrm{OPT}}(u,v)}$$

Proof of the claim

Recall: d^* is a linear combination of $O(n \log^2 n)$ cut metrics.

Apply **Bourgain's theorem** to d_{OPT}.

There is an $O(\log^2 n)$-dimensional ℓ_1 metric d^* such that:
$d^*(u,v) \leq d_{\mathrm{OPT}}(u,v) \in O(\log n) \cdot d^*(u,v)$

There is a collection C of $O(n \log^2 n)$ cuts and coefficients $\{\alpha_S \mid S \in C\}$ such that:

$$d^* = \sum_{S \in C} \alpha_S \, d_S,$$

where d_S is the cut metric of the cut S.

$$\sum_{S \in C} \alpha_S \, d_S(u,v) \leq d_{\mathrm{OPT}}(u,v) \in O(\log n) \cdot \sum_{S \in C} \alpha_S \, d_S(u,v)$$

$$\mathrm{LR}(G,H) = \frac{\sum_{u,v} w_G(u,v) \cdot d_{\mathrm{OPT}}(u,v)}{\sum_{u,v} w_H(u,v) \cdot d_{\mathrm{OPT}}(u,v)} \in \Omega\left(\frac{1}{\log n}\right) \cdot \frac{\sum_{u,v} w_G(u,v) \cdot \sum_{S \in C} \alpha_S \, d_S(u,v)}{\sum_{u,v} w_H(u,v) \cdot \sum_{S \in C} \alpha_S \, d_S(u,v)}$$

Proof of the claim

Claim: $\mathrm{LR}(G,H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$

Proof idea:

- Among all $O(n\log^2 n)$ cuts $S \in C$:
  - Choose the sparsest one.
  - $\frac{\sum_{u,v} w_G(u,v)\cdot\sum_{S\in C}\alpha_S d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot\sum_{S\in C}\alpha_S d_S(u,v)} \geq$ sparsity of the cut $\geq \Phi(G,H)$

$$\sum_{S\in C}\alpha_S\, d_S(u,v) \leq d_{\mathrm{OPT}}(u,v) \in O(\log n)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)$$

$$\mathrm{LR}(G,H) = \frac{\sum_{u,v} w_G(u,v)\cdot d_{\mathrm{OPT}}(u,v)}{\sum_{u,v} w_H(u,v)\cdot d_{\mathrm{OPT}}(u,v)} \in \Omega\left(\frac{1}{\log n}\right)\cdot\frac{\sum_{u,v} w_G(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)} \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$$

Proof of the claim

Claim: $\mathrm{LR}(G,H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$

$$\frac{\sum_{u,v} w_G(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)} = \frac{\sum_{S\in C}\alpha_S\cdot\sum_{u,v} w_G(u,v) d_S(u,v)}{\sum_{S\in C}\alpha_S\cdot\sum_{u,v} w_H(u,v) d_S(u,v)} \geq \min_{S\in C}\frac{\sum_{u,v} w_G(u,v)\cdot d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot d_S(u,v)} \geq \Phi(G,H)$$

$$\frac{\sum_i a_i}{\sum_i b_i} \geq \min_i \frac{a_i}{b_i}$$

$$\mathrm{LR}(G,H) = \frac{\sum_{u,v} w_G(u,v)\cdot d_{\mathrm{OPT}}(u,v)}{\sum_{u,v} w_H(u,v)\cdot d_{\mathrm{OPT}}(u,v)} \in \Omega\left(\frac{1}{\log n}\right)\cdot\frac{\sum_{u,v} w_G(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)}{\sum_{u,v} w_H(u,v)\cdot\sum_{S\in C}\alpha_S\, d_S(u,v)} \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$$

LP rounding

- Using a linear programming solver, we can obtain:
  - $\mathrm{LR}(G, H)$ $\longrightarrow$ $O(\log n)$-approximation to the conductance $\Phi(G)$
  - d_{OPT}

Exercise:
Can we use it to obtain a cut $S \subseteq V$ whose conductance is $O(\Phi(G) \cdot \log n)$?

Summary

- Computing conductance can be viewed as minimizing a **linear objective function** subject to **linear constraints**, where the search space consists of all **cut metrics** scaled by real values.

- **Tool 1:**
  - By relaxing the search space from cut metrics to **all metrics**, we obtain a **linear programming** formulation, whose optimal solution can be computed in polynomial time.

- **Tool 2:**
  - By **Bourgain's theorem**, any metric can be approximated within an $O(\log n)$ factor by a linear combination of cut metrics. Consequently, the LP solution yields an $O(\log n)$-approximation to the conductance.

Outlook

- **Next:** Proof of Bourgain's theorem.

- **Natural question:** Is it possible to do better than $O(\log n)$-approximation?

Outlook

- **Next:** Proof of Bourgain's theorem.

- **Natural question:** Is it possible to do better than $O(\log n)$-approximation?

- The bound $\mathrm{LR}(G, H) \in \Omega\left(\frac{\Phi(G,H)}{\log n}\right)$ is <u>already tight</u>, so we will need a different approach.

- By considering some special class of metrics that admits better embedding to ℓ_1 metrics, the approximation ratio can be improved to $O\left(\sqrt{\log n}\log\log n\right)$.
  - This requires the use of semidefinite programming (SDP).

https://dl.acm.org/doi/pdf/10.1145/1060590.1060673 Not covered in this course

References

- **Main reference:**
  - Lecture 2.1 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapter 10 of https://lucatrevisan.github.io/books/expanders-2016.pdf

- **Additional/optional reading:**
  - Polylogarithmic approximations of conductance can also be obtained via cut-matching games.

    - Rohit Khandekar, Satish Rao, and Umesh Vazirani. “Graph partitioning using single commodity flows.” J. ACM, 56(4)
      - https://dl.acm.org/doi/abs/10.1145/1538902.1538903
    - Rohit Khandekar, Subhash Khot, Lorenzo Orecchia, and Nisheeth K Vishnoi. “On a cut-matching game for the sparsest cut problem.” Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2007-177, 6(7):12, 2007.
      - https://www2.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-177.pdf

CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

Lecture 3:

Expanders – metric embedding

Our goal: Proving Bourgain's theorem

Bourgain's theorem:

- Given any n-point metric space (X, d),
    - there is an embedding $f \colon X \to \mathbb{R}^{O(\log^2 n)}$ such that for any $x, y \in X$,
      $\|f(x) - f(y)\|_1 \leq d(x, y) \in O(\log n) \cdot \|f(x) - f(y)\|_1$

Algorithm

Bourgain's theorem:
- Given any n-point metric space (X, d),
  - there is an embedding $f\colon X \to \mathbb{R}^{O(\log^2 n)}$ such that for any $x, y \in X$,
$$\|f(x) - f(y)\|_1 \leq d(x, y) \in O(\log n) \cdot \|f(x) - f(y)\|_1$$

Algorithm:
- Randomly sample $O(\log^2 n)$ subsets:
$$A_{i,j} \subseteq X \quad | \quad 1 \leq i \leq c \log n\,, \quad 1 \leq j \leq \log n$$
- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d\left(x, A_{i,j}\right) = \min_{y \in A_{i,j}} d(x, y)$.

$$\|f(x) - f(y)\|_1 = \sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right|$$

- For simplicity, assume $\log n$ is an integer.
- c is some constant.

Correctness

Bourgain's theorem:

- Given any n-point metric space (X, d),
  - there is an embedding $f\colon X \to \mathbb{R}^{O(\log^2 n)}$ such that for any $x, y \in X$,

$$\|f(x) - f(y)\|_1 \le d(x, y) \in O(\log n) \cdot \|f(x) - f(y)\|_1$$

Algorithm:

- Randomly sample $O(\log^2 n)$ subsets:

$$A_{i,j} \subseteq X \quad | \quad 1 \le i \le c \log n, \quad 1 \le j \le \log n$$

- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d\left(x, A_{i,j}\right) = \min_{y \in A_{i,j}} d(x, y)$.

$$\|f(x) - f(y)\|_1 = \sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right|$$

Correctness:

With high probability, we have:

- **Part 1:** $\sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right| \le d(x, y) \cdot c \log^2 n$
- **Part 2:** $\sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right| \in d(x, y) \cdot \Omega(\log n)$

With a scaling $f_{i,j}(x) \leftarrow f_{i,j}(x) \cdot \frac{1}{c \log^2 n}$,

this proves Bourgain's theorem.

Correctness: Part 1

$$\left|f_{i,j}(x) - f_{i,j}(y)\right| = \left|d(x, A_{i,j}) - d(y, A_{i,j})\right|$$

Without loss of generality, assume $d(x, A_{i,j}) \geq d(y, A_{i,j})$

$$= d(x, A_{i,j}) - d(y, A_{i,j})$$

Triangle inequality

$$\leq \left(d(x, y) + d(y, A_{i,j})\right) - d(y, A_{i,j})$$

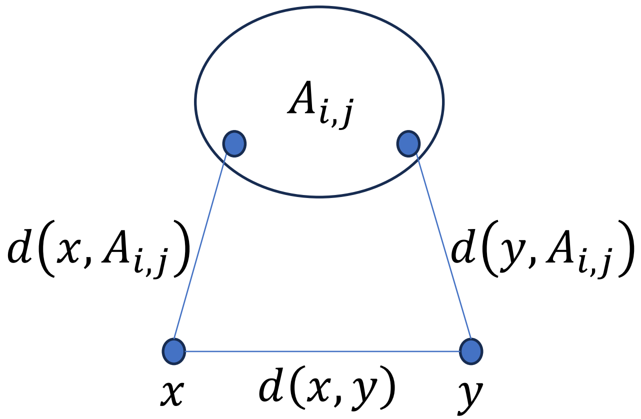
$$= d(x, y)$$

Algorithm:

- Randomly sample $O(\log^2 n)$ subsets:

$$A_{i,j} \subseteq X \quad | \quad 1 \leq i \leq c \log n, \quad 1 \leq j \leq \log n$$

- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d(x, A_{i,j}) = \min_{y \in A_{i,j}} d(x, y)$.



Correctness:

With high probability, we have:

- **Part 1:** $\sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right| \leq d(x, y) \cdot c \log^2 n$
- **Part 2:** $\sum_{i,j} \left|f_{i,j}(x) - f_{i,j}(y)\right| \in d(x, y) \cdot \Omega(\log n)$

Correctness: Part 2

Algorithm:

- Randomly sample $O(\log^2 n)$ subsets:
$$A_{i,j} \subseteq X \quad | \quad 1 \le i \le c \log n\,, \quad 1 \le j \le \log n$$
- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d\left(x, A_{i,j}\right) = \min_{y \in A_{i,j}} d(x, y)$.

Correctness:

With high probability, we have:

- **Part 1:** $\sum_{i,j} \left| f_{i,j}(x) - f_{i,j}(y) \right| \le d(x, y) \cdot c \log^2 n$
- **Part 2:** $\sum_{i,j} \left| f_{i,j}(x) - f_{i,j}(y) \right| \in d(x, y) \cdot \Omega(\log n)$

Next

Main tool: Chernoff bound

- Consider the following setting.
  - $X_1, X_2, \dots, X_n$ are independent random variables taking values in $\{0, 1\}$.
  - $X = \sum_{i=1}^{n} X_i$.
  - $\mu = \mathbb{E}[X]$.

Main tool: Chernoff bound

- Consider the following setting.
  - $X_1, X_2, \ldots, X_n$ are independent random variables taking values in $\{0, 1\}$.
  - $X = \sum_{i=1}^{n} X_i$.
  - $\mu = \mathbb{E}[X]$.

Chernoff bounds:

- $\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}$, for $1 \geq \delta \geq 0$.
- $\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2+\delta}}$, for $\delta \geq 0$.
- $\Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}$, for $1 \geq \delta \geq 0$.

https://en.wikipedia.org/wiki/Chernoff_bound

Main tool: Chernoff bound

- Consider the following setting.
  - $X_1, X_2, \dots, X_n$ are independent random variables taking values in $\{0, 1\}$.
  - $X = \sum_{i=1}^{n} X_i$.
  - $\mu = \mathbb{E}[X]$.

Chernoff bounds:

- $\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}$, for $1 \geq \delta \geq 0$.
- $\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2+\delta}}$, for $\delta \geq 0$.
- $\Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}$, for $1 \geq \delta \geq 0$.

https://en.wikipedia.org/wiki/Chernoff_bound

Key takeaway:

- X is within a constant factor of $\mathbb{E}[X]$ with a probability of $1 - e^{-\Omega(\mathbb{E}[X])}$.
- In particular, $\mathbb{E}[X] \in \Omega(c \log n)$ guarantees a success probability of $1 - n^{-\Omega(c)}$.

Correctness: Part 2

- $B(v, r) =$ the ball of radius r around v.
- $B(v, < r) =$ the open ball of radius r around v.

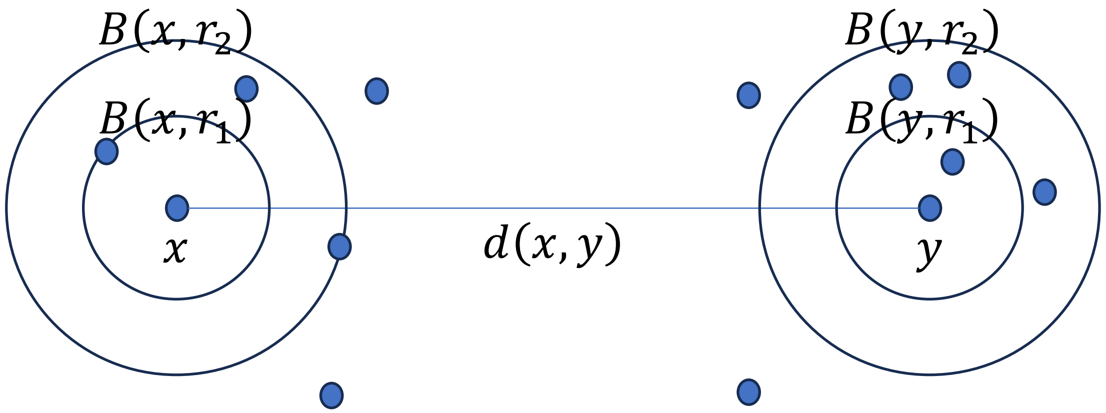
Correctness: Part 2

- $B(v, r)$ = the ball of radius r around v.
- $B(v, < r)$ = the <u>open</u> ball of radius r around v.

Consider two points x and y in X.

r_j = the minimum of:

- Smallest number r such that $\min\{|B(x,r)|, |B(y,r)|\} \geq 2^j$
- $\frac{d(x,y)}{3}$



Correctness: Part 2

- $B(v, r)$ = the ball of radius r around v.
- $B(v, < r)$ = the open ball of radius r around v.

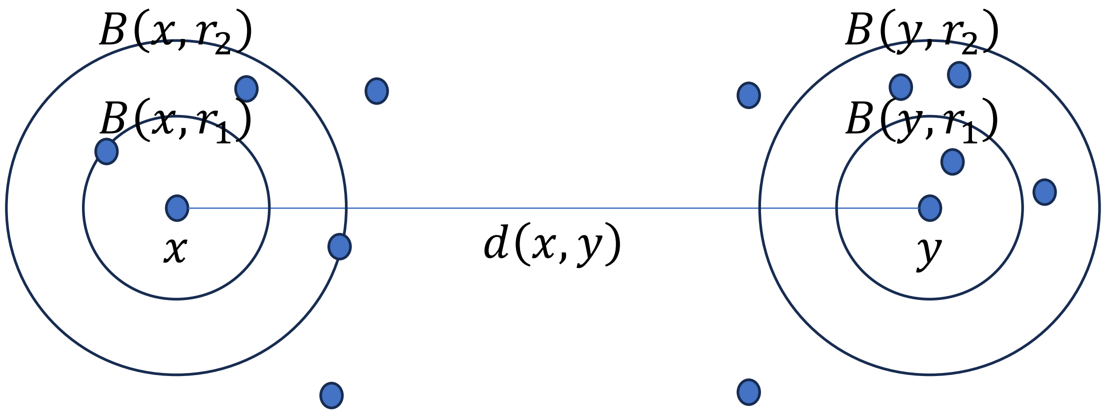
Consider two points x and y in X.

r_j = the minimum of:

- Smallest number r such that $\min\{|B(x,r)|, |B(y,r)|\} \geq 2^j$
- $\frac{d(x,y)}{3}$

Observation 1:
$|B(x, < r_j)| < 2^j$ or $|B(y, < r_j)| < 2^j$

Without loss of generality, we assume this.



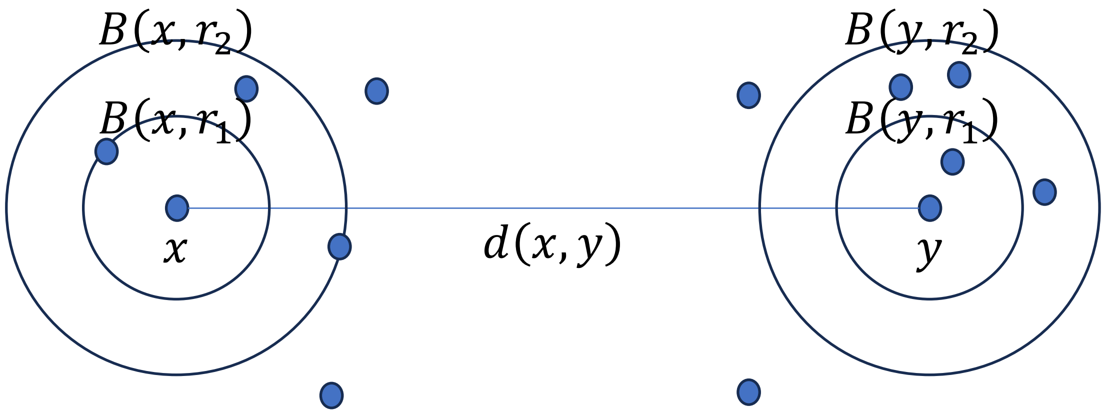
Correctness: Part 2

- $B(v, r)$ = the ball of radius r around v.
- $B(v, < r)$ = the <u>open</u> ball of radius r around v.

Consider two points x and y in X.

r_j = the minimum of:

- Smallest number r such that $\min\{|B(x,r)|, |B(y,r)|\} \geq 2^j$
- $\frac{d(x,y)}{3}$



Observation 1:
$|B(x, < r_j)| < 2^j$ or $|B(y, < r_j)| < 2^j$

Without loss of generality, we assume this.

Observation 2:
If $r_{j-1} < \frac{d(x,y)}{3}$, then
$|B(x, r_{j-1})| \geq 2^{j-1}$ and $|B(y, r_{j-1})| \geq 2^{j-1}$

Correctness: Part 2

- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d\left(x, A_{i,j}\right) = \min_{y \in A_{i,j}} d(x, y)$.

- $B(v, r)$ = the ball of radius r around v.
- $B(v, < r)$ = the open ball of radius r around v.

Observation 1:
$\left|B\left(x, < r_j\right)\right| < 2^j$ or $\left|B\left(y, < r_j\right)\right| < 2^j$

Without loss of generality, we assume this.

Observation 2:
If $r_{j-1} < \frac{d(x,y)}{3}$, then
$\left|B\left(x, r_{j-1}\right)\right| \geq 2^{j-1}$ and $\left|B\left(y, r_{j-1}\right)\right| \geq 2^{j-1}$

Suppose $r_{j-1} < \frac{d(x,y)}{3}$.
Let $\mathcal{E}_{i,j}$ be the event that:
- $B\left(x, < r_j\right) \cap A_{i,j} = \emptyset$
- $B\left(y, r_{j-1}\right) \cap A_{i,j} \neq \emptyset$

Correctness: Part 2

- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d(x, A_{i,j}) = \min_{y \in A_{i,j}} d(x,y)$.

- $B(v,r) =$ the ball of radius r around v.
- $B(v, < r) =$ the open ball of radius r around v.

For $n \geq 2$, we have: $\left(1 - \frac{1}{n}\right)^n \geq \left(1 - \frac{1}{2}\right)^2 = \frac{1}{4}$

Observation 1:
$|B(x, < r_j)| < 2^j$ or $|B(y, < r_j)| < 2^j$

Without loss of generality, we assume this.

Observation 2:
If $r_{j-1} < \frac{d(x,y)}{3}$, then
$|B(x, r_{j-1})| \geq 2^{j-1}$ and $|B(y, r_{j-1})| \geq 2^{j-1}$

Suppose $r_{j-1} < \frac{d(x,y)}{3}$.
Let $\mathcal{E}_{i,j}$ be the event that:

- $B(x, < r_j) \cap A_{i,j} = \emptyset$
  - Probability $= \left(1 - \frac{1}{2^j}\right)^{|B(x, < r_j)|} > \left(1 - \frac{1}{2^j}\right)^{2^j} \geq \frac{1}{4}$
- $B(y, r_{j-1}) \cap A_{i,j} \neq \emptyset$
  - Probability $= 1 - \left(1 - \frac{1}{2^j}\right)^{|B(y, r_{j-1})|}$

$\geq 1 - \left(1 - \frac{1}{2^j}\right)^{2^{j-1}} \geq 1 - e^{-1/2} \geq \frac{1}{3}$

Claim 1: $\Pr[\mathcal{E}_{i,j}] > \frac{1}{12}$

$1 + x \leq e^x$

Correctness: Part 2

- Each point in X joins $A_{i,j}$ with probability 2^{-j} independently.
- Set $f_{i,j}(x) = d\left(x, A_{i,j}\right) = \min_{y \in A_{i,j}} d(x,y)$.

- $B(v,r)$ = the ball of radius r around v.
- $B(v,<r)$ = the open ball of radius r around v.

Observation 1:
$\left|B\left(x,<r_j\right)\right| < 2^j$ or $\left|B\left(y,<r_j\right)\right| < 2^j$

Without loss of generality, we assume this.

Observation 2:
If $r_{j-1} < \frac{d(x,y)}{3}$, then
$\left|B\left(x,r_{j-1}\right)\right| \geq 2^{j-1}$ and $\left|B\left(y,r_{j-1}\right)\right| \geq 2^{j-1}$

Suppose $r_{j-1} < \frac{d(x,y)}{3}$.
Let $\mathcal{E}_{i,j}$ be the event that:
- $B\left(x,<r_j\right) \cap A_{i,j} = \emptyset$
- $B\left(y,r_{j-1}\right) \cap A_{i,j} \neq \emptyset$

Claim 1: $\Pr\left[\mathcal{E}_{i,j}\right] > \frac{1}{12}$
Claim 2: If $\mathcal{E}_{i,j}$ occurs, then:
- $\left|f_{i,j}(x) - f_{i,j}(y)\right| \geq r_j - r_{j-1}$

Correctness: Part 2

- Consider the following setting.
  - $X_i \in \{0,1\}$ is the indicator random variable for the event $\mathcal{E}_{i,j}$.
  - $X = \sum_{i=1}^{c \log n} X_i$.
  - $\mu = \mathbb{E}[X] > \frac{c \log n}{12}$.

Claim 1: $\Pr[\mathcal{E}_{i,j}] > \frac{1}{12}$
Claim 2: If $\mathcal{E}_{i,j}$ occurs, then:
- $|f_{i,j}(x) - f_{i,j}(y)| \geq r_j - r_{j-1}$

Correctness: Part 2

- Consider the following setting.
  - $X_i \in \{0,1\}$ is the indicator random variable for the event $\mathcal{E}_{i,j}$.
  - $X = \sum_{i=1}^{c \log n} X_i$.
  - $\mu = \mathbb{E}[X] > \frac{c \log n}{12}$.

$$\Pr\left[X \leq \frac{c \log n}{24}\right] \leq \Pr\left[X \leq \frac{\mu}{2}\right] \leq e^{-\frac{\left(\frac{1}{2}\right)^2 \mu}{2}} \in n^{-\Omega(c)}$$

Chernoff bound:

- $\Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}$, for $1 \geq \delta \geq 0$.

Claim 1: $\Pr\left[\mathcal{E}_{i,j}\right] > \frac{1}{12}$

Claim 2: If $\mathcal{E}_{i,j}$ occurs, then:

- $\left|f_{i,j}(x) - f_{i,j}(y)\right| \geq r_j - r_{j-1}$

Correctness: Part 2

- Consider the following setting.
  - $X_i \in \{0,1\}$ is the indicator random variable for the event $\mathcal{E}_{i,j}$.
  - $X = \sum_{i=1}^{c \log n} X_i$.
  - $\mu = \mathbb{E}[X] > \frac{c \log n}{12}$.

$$\Pr\left[X \le \frac{c \log n}{24}\right] \le \Pr\left[X \le \frac{\mu}{2}\right] \le e^{-\frac{\left(\frac{1}{2}\right)^2 \mu}{2}} \in n^{-\Omega(c)}$$

With probability $1 - n^{-\Omega(c)}$, we have:

$$\sum_i \left|f_{i,j}(x) - f_{i,j}(y)\right| \ge \left(r_j - r_{j-1}\right) \cdot \frac{c \log n}{24}$$

Chernoff bound:

- $\Pr[X \le (1-\delta)\mu] \le e^{-\frac{\delta^2 \mu}{2}}$, for $1 \ge \delta \ge 0$.

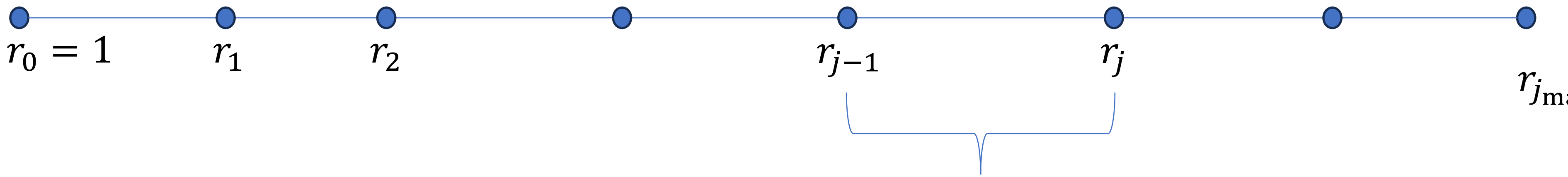
Claim 1: $\Pr\left[\mathcal{E}_{i,j}\right] > \frac{1}{12}$

Claim 2: If $\mathcal{E}_{i,j}$ occurs, then:

- $\left|f_{i,j}(x) - f_{i,j}(y)\right| \ge r_j - r_{j-1}$

Correctness: Part 2

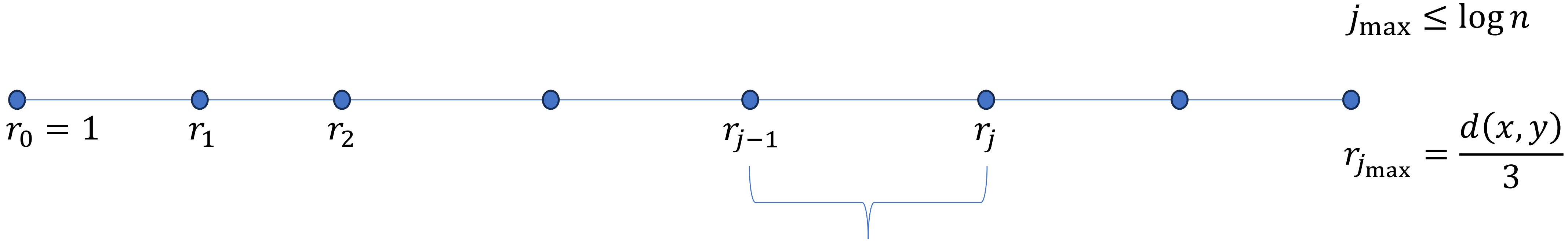
$j_{\max} \leq \log n$



$r_{j_{\max}} = \frac{d(x,y)}{3}$

With probability $1 - n^{-\Omega(c)}$, we have: $\sum_i |f_{i,j}(x) - f_{i,j}(y)| \geq (r_j - r_{j-1}) \cdot \frac{c \log n}{24}$

Correctness: Part 2

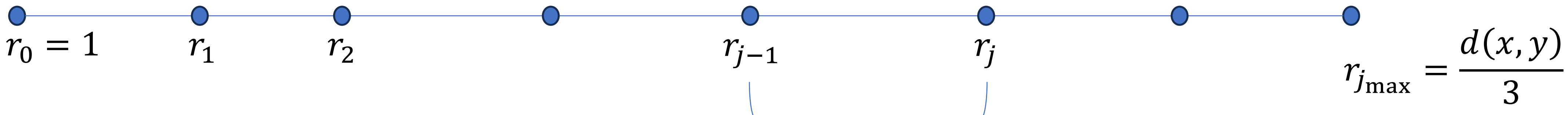


With probability $1 - n^{-\Omega(c)}$, we have: $\sum_i \left| f_{i,j}(x) - f_{i,j}(y) \right| \geq \left(r_j - r_{j-1} \right) \cdot \frac{c \log n}{24}$

With probability $1 - j_{\max} \cdot n^{-\Omega(c)}$, we have: $\sum_{i,j} \left| f_{i,j}(x) - f_{i,j}(y) \right| \geq \frac{d(x,y)}{3} \cdot \frac{c \log n}{24}$

Correctness: Part 2

$j_{\max} \le \log n$



With probability $1 - n^{-\Omega(c)}$, we have: $\sum_i \left| f_{i,j}(x) - f_{i,j}(y) \right| \ge (r_j - r_{j-1}) \cdot \frac{c \log n}{24}$

With probability $1 - j_{\max} \cdot n^{-\Omega(c)}$, we have: $\sum_{i,j} \left| f_{i,j}(x) - f_{i,j}(y) \right| \ge \frac{d(x,y)}{3} \cdot \frac{c \log n}{24}$

With probability $1 - \binom{n}{2} \cdot j_{\max} \cdot n^{-\Omega(c)}$, we have: $\sum_{i,j} \left| f_{i,j}(x) - f_{i,j}(y) \right| \ge \frac{d(x,y)}{3} \cdot \frac{c \log n}{24}$ for all x and y.

Can make this $1 - \frac{1}{\mathrm{poly}(n)}$ with an arbitrarily large exponent by selecting c to be a sufficiently large constant.

This finishes the proof of correctness.

Summary

Main tool: Chernoff bound.

- $O(\log^2 n)$ subsets are needed because:
  - Need to try all $O(\log n)$ sampling probabilities: $1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{n}$.
  - For each of them, need to repeat for $O(\log n)$ times to get a "with high probability" guarantee.

Summary

Main tool: Chernoff bound.

- $O(\log^2 n)$ subsets are needed because:
  - Need to try all $O(\log n)$ sampling probabilities: $1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{n}$.
  - For each of them, need to repeat for $O(\log n)$ times to get a "with high probability" guarantee.

If we are happy with an "in expectation" guarantee, then this is not needed!

$$\mathbb{E}[\|f(x) - f(y)\|_1] \leq d(x, y) \in O(\log n) \cdot \mathbb{E}[\|f(x) - f(y)\|_1]$$

Outlook

- We have finished the proof for $O(\log n)$-approximation of conductance.

This allows us to obtain a cut of conductance $O(\Phi(G) \cdot \log n)$ in polynomial time.

- **Next:** A completely different approach to conductance approximation via <u>spectral graph theory</u>.

This allows us to obtain a cut of conductance $O\left(\sqrt{\Phi(G)}\right)$ in polynomial time.

This is better whenever $\Phi(G) \in \omega(1/\log^2 n)$.

References

- **Main reference:**
  - Lecture 2.2 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapter 11 of https://lucatrevisan.github.io/books/expanders-2016.pdf

CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

Lecture 4:

Expanders – spectral graph theory

Matrix representation of graphs

- Let $G = (V, E)$ be a graph, with $V = \{1,2, \ldots, n\}$.

The adjacency matrix $\boldsymbol{A}$ is an $(n \times n)$-matrix:

$$\boldsymbol{A}[i,j] = \begin{cases} 0, & \{i,j\} \notin E \\ 1, & \{i,j\} \in E \end{cases}$$

The degree matrix $\boldsymbol{D}$ is an $(n \times n)$-matrix:

$$\boldsymbol{D}[i,j] = \begin{cases} \deg(i), & i = j \\ 0, & i \neq j \end{cases}$$

Matrix representation of graphs

- Let $G = (V, E)$ be a graph, with $V = \{1,2,\dots,n\}$.

The adjacency matrix $\boldsymbol{A}$ is an $(n \times n)$-matrix:

$$\boldsymbol{A}[i,j] = \begin{cases} 0, & \{i,j\} \notin E \\ 1, & \{i,j\} \in E \end{cases}$$

The degree matrix $\boldsymbol{D}$ is an $(n \times n)$-matrix:

$$\boldsymbol{D}[i,j] = \begin{cases} \deg(i), & i = j \\ 0, & i \neq j \end{cases}$$

The Laplacian matrix: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$

The normalized Laplacian matrix: $\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$

Connection to cuts

- Let $G = (V, E)$ be a graph, with $V = \{1,2, \ldots, n\}$.

The <u>adjacency matrix</u> $\boldsymbol{A}$ is an $(n \times n)$-matrix:

$$\boldsymbol{A}[i,j] = \begin{cases} 0, & \{i,j\} \notin E \\ 1, & \{i,j\} \in E \end{cases}$$

The <u>degree matrix</u> $\boldsymbol{D}$ is an $(n \times n)$-matrix:

$$\boldsymbol{D}[i,j] = \begin{cases} \deg(i)\,, & i = j \\ 0, & i \neq j \end{cases}$$

The <u>Laplacian matrix</u>: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$

The <u>normalized Laplacian matrix</u>: $\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$

Let $\boldsymbol{x} \in \mathbb{R}^n$ be a column vector.

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

If $\boldsymbol{x} = \boldsymbol{1}_S$ is the indicator vector for a cut $S \subseteq V$, then $\boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} = |E(S, V \setminus S)|$ is the size of the cut.

Regular graphs

- Let $G = (V, E)$ be a d-regular graph, with $V = \{1,2,\dots,n\}$.

The adjacency matrix $\boldsymbol{A}$ is an $(n \times n)$-matrix:

$$\boldsymbol{A}[i,j] = \begin{cases} 0, & \{i,j\} \notin E \\ 1, & \{i,j\} \in E \end{cases}$$

The degree matrix $\boldsymbol{D}$ is an $(n \times n)$-matrix:

$$\boldsymbol{D}[i,j] = d\boldsymbol{I} = \begin{cases} d, & i = j \\ 0, & i \neq j \end{cases}$$

The Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = d\boldsymbol{I} - \boldsymbol{A}$$

The normalized Laplacian matrix:

$$\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$$

For simplicity, we restrict our attention to regular graphs, noting that all results extend to general graphs.

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

$$\boldsymbol{x}^\top \boldsymbol{N}\boldsymbol{x} = \frac{1}{d}\sum_{\{u,v\}\in E} (x_u - x_v)^2$$

Regular graphs

- Let $G = (V, E)$ be a d-regular graph, with $V = \{1,2,\dots,n\}$.

The adjacency matrix $\boldsymbol{A}$ is an $(n \times n)$-matrix:

$$\boldsymbol{A}[i,j] = \begin{cases} 0, & \{i,j\} \notin E \\ 1, & \{i,j\} \in E \end{cases}$$

The degree matrix $\boldsymbol{D}$ is an $(n \times n)$-matrix:

$$\boldsymbol{D}[i,j] = d\boldsymbol{I} = \begin{cases} d, & i = j \\ 0, & i \neq j \end{cases}$$

The Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = d\boldsymbol{I} - \boldsymbol{A}$$

The normalized Laplacian matrix:

$$\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$$

For simplicity, we restrict our attention to regular graphs, noting that all results extend to general graphs.

All four types of matrices $\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{L}, \boldsymbol{N}$ are symmetric.
We start by reviewing some basic facts about such matrices.

Symmetric real matrices

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.

$\boldsymbol{M}[i,j] \in \mathbb{R}$ $\boldsymbol{M}[i,j] = \boldsymbol{M}[j,i]$

Symmetric real matrices

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.

Theorem: There exist
- $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$ (they are called eigenvalues)
- Orthonormal vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n \in \mathbb{R}^n$ (they are called eigenvectors)

such that:

$$\boldsymbol{M} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

Which implies: (for all i)

$$\boldsymbol{M}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$$

Symmetric real matrices

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.

Theorem: There exist
- $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$ (they are called eigenvalues)
- Orthonormal vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n \in \mathbb{R}^n$ (they are called eigenvectors)

such that:

$$\boldsymbol{M} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

Which implies: (for all i)

$$\boldsymbol{M}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$$

If we change the axes of the Euclidean space $\mathbb{R}^n$ to the eigenvectors $v_1, \dots, v_n$, then the matrix $\boldsymbol{M}$ becomes diagonal, with $\lambda_1, \dots, \lambda_n$ on the diagonal.

In other words, symmetric matrices stretch space independently along orthogonal directions.

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

Related to cuts

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
    - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
    - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

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

$$\boldsymbol{x}^\top \boldsymbol{N}\boldsymbol{x} = \frac{1}{d} \sum_{\{u,v\} \in E} (x_u - x_v)^2$$

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

Theorem:
$$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

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

$$\boldsymbol{x}^\top \boldsymbol{N}\boldsymbol{x} = \frac{1}{d} \sum_{\{u,v\} \in E} (x_u - x_v)^2$$

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Theorem:

$$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Proof $(\geq)$:

Consider $\mathcal{V} = \operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$

- Write $\boldsymbol{x} \in \mathcal{V}$ as $\boldsymbol{x} = \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i$.
- $R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} \alpha_i^2 \lambda_i}{\sum_{i=1}^{k} \alpha_i^2}$.
- $\max_{\boldsymbol{x} \in V \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x}) = \lambda_k$.
- $\lambda_k \geq \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in V \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$.

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Theorem:

$$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Proof $(\leq)$**:**

Consider any k-dimensional space $\mathcal{V}$.

- $\exists \boldsymbol{x} \in \mathcal{V} \cap \mathrm{span}\{\boldsymbol{v}_k, \ldots, \boldsymbol{v}_n\} \setminus \{\boldsymbol{0}\}$.
  - $\boldsymbol{x} = \sum_{i=k}^{n} \alpha_i \boldsymbol{v}_i$.
  - $R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\sum_{i=k}^{n} \alpha_i^2 \lambda_i}{\sum_{i=k}^{n} \alpha_i^2} \geq \lambda_k$.
- $\lambda_k \leq \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$.

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\top}\boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^{\top}\boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

Theorem:

$$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

$$\lambda_1 = \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Moreover, any minimizer $\boldsymbol{x}$ is a corresponding eigenvector.

Consider $-\boldsymbol{M}$.

$$\lambda_n = \max_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Moreover, any maximizer $\boldsymbol{x}$ is a corresponding eigenvector.

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Theorem:

$$\lambda_k = \min_{k\text{-dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

$$\lambda_1 = \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

A natural extension

$$\lambda_2 = \min_{\boldsymbol{x} \neq \boldsymbol{0},\ \boldsymbol{x} \perp \boldsymbol{v}_1} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Consider $-\boldsymbol{M}$.

$$\lambda_n = \max_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Variational characterizations

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{M}$:

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Theorem:
$$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

$$\lambda_1 = \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Consider $-\boldsymbol{M}$.

$$\lambda_n = \max_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

A natural extension

Theorem: We omit the proof.
$$\lambda_k = \min_{\boldsymbol{x} \ne \boldsymbol{0},\ \boldsymbol{x} \perp \text{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}\}} R_{\boldsymbol{M}}(\boldsymbol{x})$$

$$\lambda_2 = \min_{\boldsymbol{x} \ne \boldsymbol{0},\ \boldsymbol{x} \perp \boldsymbol{v}_1} R_{\boldsymbol{M}}(\boldsymbol{x})$$

Spectral graph theory

- The eigenvalues of $\boldsymbol{A}$ (adjacency), $\boldsymbol{L}$ (Laplacian), and $\boldsymbol{N}$ (normalized Laplacian) reveal information about:
  - Average density of cuts.
  - Bipartiteness.
  - Chromatic number.
  - Conductance.
  - Hamiltonicity.
  - Size of a maximum independent set.
  - Size of a maximum matching.
  - Toughness of a graph.
  - Number of connected components.

https://adga-workshop.org/2025/tijn.pdf

Application 1: Spectral embedding

Data

Graph

Embedding the vertices to $\mathbb{R}^k$ by taking the eigenvectors corresponding to the first k non-zero eigenvalues.

We will later see that, regardless of the graph, $\mathbf{1}$ is an eigenvector of $\boldsymbol{N}$ (as well as $\boldsymbol{L}$) with eigenvalue zero. This eigenvector therefore carries no informative content and should be ignored.

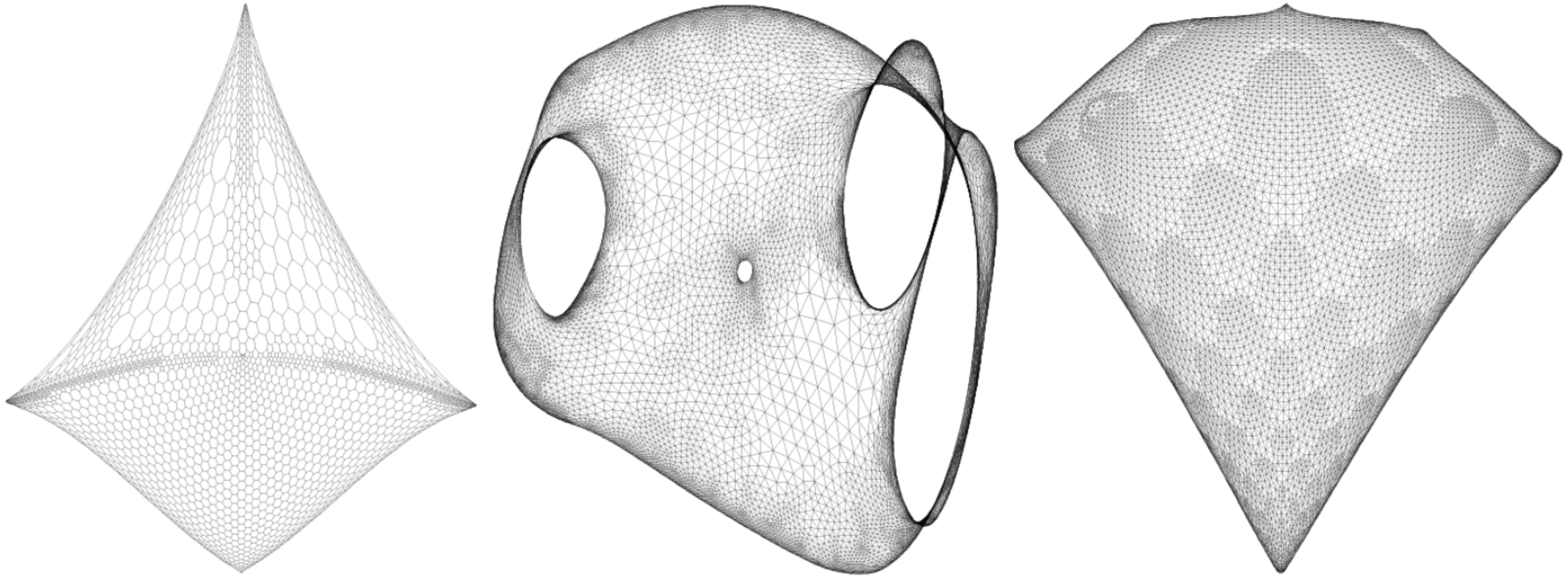
Application 1: Spectral embedding

Data

Graph

Embedding the vertices to $\mathbb{R}^k$ by taking the eigenvectors corresponding to the first k non-zero eigenvalues.

- Useful for: Minimizing $\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \sum_{\{u,v\}\in E}(x_u - x_v)^2$ leads to natural drawings.
  - **Visualization**



https://www.mathe2.uni-bayreuth.de/axel/papers/koren:on_spectral_graph_drawing.pdf

Application 1: Spectral embedding

Data

Graph

Embedding the vertices to $\mathbb{R}^k$ by taking the eigenvectors corresponding to the first k non-zero eigenvalues.

- Useful for:
  - Visualization
  - **Clustering + dimension reduction** https://en.wikipedia.org/wiki/Spectral_clustering

    - Spectral embedding turns high-dimensional data into points in a low-dimensional space where clusters become visible.
    - Eigenvectors can capture many features of the data that many traditional clustering methods fail to detect.

Already many theoretical evidences

The eigenvector equation

Application 2: Network analysis

$$\boxed{\boldsymbol{Ax} = \lambda \boldsymbol{x}}$$

- Intuitively, an eigenvector $\boldsymbol{x}$ captures the influence of nodes in a network:

Connections to influential nodes enhance your own influence.

$$\boxed{x_v = \frac{1}{\lambda} \sum_{u \in N(v)} x_u}$$

The eigenvector equation

Application 2: Network analysis

$$\boxed{\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}}$$

$$\boxed{x_v = \frac{1}{\lambda} \sum_{u \in N(v)} x_u}$$

- Intuitively, an eigenvector $\boldsymbol{x}$ captures the influence of nodes in a network:

Connections to influential nodes enhance your own influence.

Some applications:

- Eigenvector centrality is the unique measure satisfying certain natural axioms for a **ranking system**.
- In **neuroscience**, the eigenvector centrality of a neuron in a model neural network has been found to correlate with its relative firing rate.
- Eigenvector centrality and related concepts have been used to model **opinion influence in sociology and economics**.
- **Google's PageRank** is based on a variant of Eigenvector centrality.

https://en.wikipedia.org/wiki/Eigenvector_centrality

The smallest eigenvalue λ_1

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_1 = 0$, with $\mathbf{1}$ being a corresponding eigenvector.

The smallest eigenvalue λ_1

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

$\lambda_1 \geq 0$

Theorem:
- $\lambda_1 = 0$, with $\mathbf{1}$ being a corresponding eigenvector.

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2} \geq 0$$

The smallest eigenvalue λ_1

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_1 = 0$, with $\mathbf{1}$ being a corresponding eigenvector.

$\lambda_1 \geq 0$

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2} \geq 0$$

$$\lambda_1 = \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x}) \leq R_{\boldsymbol{N}}(\mathbf{1}) = \frac{\sum_{\{u,v\} \in E} (1-1)^2}{d \sum_{v \in V} 1^2} = 0$$

$\lambda_1 \leq 0$

The eigenvalue λ_k

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_k = 0$ if and only if G has at least k connected components.

The eigenvalue λ_k

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$

$$R_N(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2} = 0$$

if and only if

For every connected component S,

$$x_u = x_v \quad \forall u, v \in S$$

Theorem:

- $\lambda_k = 0$ if and only if G has at least k connected components.

The eigenvalue λ_k

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

$$R_N(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2} = 0$$

if and only if

For every connected component S,
$x_u = x_v \quad \forall u, v \in S$

Theorem:

- $\lambda_k = 0$ if and only if G has at least k connected components.

G has at least k connected components $S_1, \ldots, S_k$.

Set $\mathcal{V} = \operatorname{span}\{\mathbf{1}_{S_1}, \ldots, \mathbf{1}_{S_k}\}$.

- $\forall \boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\},\ R_N(\boldsymbol{x}) = \mathbf{0}$

$\lambda_k = \min_{k-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_N(\boldsymbol{x})$

$\lambda_k = 0$

The eigenvalue λ_k

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2} = 0$$

if and only if

For every connected component S,
$x_u = x_v \quad \forall u, v \in S$

Theorem:
- $\lambda_k = 0$ if and only if G has at least k connected components.

$k =$ dimension of $\mathcal{V} \leq$ the number of connected components.

$\lambda_k = 0$

There is a k-dimensional subspace $\mathcal{V}$ of $\mathbb{R}^n$ such that:
- $\forall \boldsymbol{x} \in \mathcal{V} \setminus \{\boldsymbol{0}\},\ R_{\boldsymbol{N}}(\boldsymbol{x}) = \boldsymbol{0}$

The value of $\boldsymbol{x}$ is constant within each connected component.

The largest eigenvalue λ_n

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_n \leq 2.$
- $\lambda_n = 2$ if and only if G has a bipartite connected component.

$$\lambda_n = \max_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$$

The largest eigenvalue λ_n

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$

Theorem:
- $\lambda_n \le 2$.
- $\lambda_n = 2$ if and only if G has a bipartite connected component.

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\sum_{\{u,v\}\in E}(x_u - x_v)^2}{d\sum_{v\in V} {x_v}^2}$$

$$= \frac{2\sum_{\{u,v\}\in E}(x_u^2 + x_v^2) - \sum_{\{u,v\}\in E}(x_u + x_v)^2}{d\sum_{v\in V} {x_v}^2}$$

$$= 2 - \frac{\sum_{\{u,v\}\in E}(x_u + x_v)^2}{d\sum_{v\in V} {x_v}^2}$$

$$\lambda_n = \max_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\boldsymbol{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x}) = 2 - \min_{\boldsymbol{x}\in\mathbb{R}^n\setminus\{\boldsymbol{0}\}} \frac{\sum_{\{u,v\}\in E}(x_u + x_v)^2}{d\sum_{v\in V} {x_v}^2} \le 2$$

The largest eigenvalue λ_n

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_n \leq 2$.
- $\lambda_n = 2$ if and only if G has a bipartite connected component.

$$\lambda_n = 2 - \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} \frac{\sum_{\{u,v\} \in E} (x_u + x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

- Suppose G has a bipartite connected component S with bipartition $S = A \cup B$.
- Setting $\boldsymbol{x} = \mathbf{1}_A - \mathbf{1}_B$ makes $\frac{\sum_{\{u,v\} \in E} (x_u + x_v)^2}{d \sum_{v \in V} {x_v}^2} = 0$.
- Therefore, $\lambda_n = 2$.

The largest eigenvalue λ_n

- Consider the normalized Laplacian $\boldsymbol{N}$ of a graph $G = (V, E)$.
  - Eigenvalues: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Theorem:
- $\lambda_n \leq 2$.
- $\lambda_n = 2$ if and only if G has a bipartite connected component.

$$\lambda_n = 2 - \min_{\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}} \frac{\sum_{\{u,v\} \in E} (x_u + x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

- Suppose $\lambda_n = 2$.
- There exists $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ such that $\frac{\sum_{\{u,v\} \in E} (x_u + x_v)^2}{d \sum_{v \in V} {x_v}^2} = 0$.
- Define:
  - $A = \{v \in V \mid x_v > 0\} \neq \emptyset$
  - $B = \{v \in V \mid x_v < 0\} \neq \emptyset$
- $A \cup B$ is a union of bipartite connected components.

Outlook

- **Next:** We will show that:

$$\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$$

Cheeger's inequality

where λ_2 is the second eigenvalue of the normalized Laplacian $\boldsymbol{N}$ of G.

Moreover, given the eigenvector $\boldsymbol{v}_2$, we can obtain a cut of conductance at most $\sqrt{2\lambda_2}$ in polynomial time.

Outlook

- **Next:** We will show that:

$$\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$$

Cheeger's inequality

where λ_2 is the second eigenvalue of the normalized Laplacian $\boldsymbol{N}$ of G.

Moreover, given the eigenvector $\boldsymbol{v}_2$, we can obtain a cut of conductance at most $\sqrt{2\lambda_2}$ in polynomial time.

- **Next:** Efficient eigenvector computation.

Useful in many applications, both in practice and in theory.

References

- **Main reference:**
  - Lecture 4.1 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapters 1, 2, and 3 of https://lucatrevisan.github.io/books/expanders-2016.pdf

CS5275 – The Algorithm Designer's Toolkit (S2 AY2025/26)

Lecture 5:

Expanders – Cheeger inequality

λ_2 and conductance

- **Recall:** $\lambda_2 = 0$ if and only if G has at least 2 connected components.

The second eigenvalue of the normalized Laplacian $\boldsymbol{N}$ of G.

λ_2 and conductance

- **Recall:** $\lambda_2 = 0$ if and only if G has at least 2 connected components.

  The second eigenvalue of the normalized Laplacian $\boldsymbol{N}$ of G.

- Intuitively, there should a robust version of this fact:
  - λ_2 is small if and only if G has a sparse cut.

λ_2 and conductance

- **Recall:** $\lambda_2 = 0$ if and only if G has at least 2 connected components.

  The second eigenvalue of the normalized Laplacian $\boldsymbol{N}$ of G.

- Intuitively, there should a robust version of this fact:
  - λ_2 is small if and only if G has a sparse cut.

- We will prove this: **Cheeger inequality:** $\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

Let $(S, V \setminus S)$ be a sparsest cut: $\Phi(S) = \Phi(G)$.

$$R_{\boldsymbol{N}}(\mathbf{1}_S) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(S)}$$

$$R_{\boldsymbol{N}}(\mathbf{1}_{V \setminus S}) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(V \setminus S)}$$

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

Part 1: $\frac{\lambda_2}{2} \le \Phi(G)$

Let $(S, V \setminus S)$ be a sparsest cut: $\Phi(S) = \Phi(G)$.

$$R_{\boldsymbol{N}}(\mathbf{1}_S) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(S)}$$

$$R_{\boldsymbol{N}}(\mathbf{1}_{V\setminus S}) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(V \setminus S)}$$

$$\lambda_2 = \min_{2\text{-dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$$

$$\le \max_{\boldsymbol{z} \in \mathrm{span}(\mathbf{1}_S, \mathbf{1}_{V\setminus S}) \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{z})$$

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

Recall: $\lambda_2 = \min_{2\text{-dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

Let $(S, V \setminus S)$ be a sparsest cut: $\Phi(S) = \Phi(G)$.

$$R_{\boldsymbol{N}}(\mathbf{1}_S) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(S)}$$

$$R_{\boldsymbol{N}}(\mathbf{1}_{V \setminus S}) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(V \setminus S)}$$

$$\lambda_2 = \min_{2-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$$

$$\leq \max_{\boldsymbol{z} \in \mathrm{span}(\mathbf{1}_S, \mathbf{1}_{V \setminus S}) \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{z})$$

$$\leq 2 \max\{R_{\boldsymbol{N}}(\mathbf{1}_S), R_{\boldsymbol{N}}(\mathbf{1}_{V \setminus S})\}$$

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

Recall: $\lambda_2 = \min_{2-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \mathrm{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\mathbf{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \le \Phi(G)$

Let $(S, V \setminus S)$ be a sparsest cut: $\Phi(S) = \Phi(G)$.

$$R_{\boldsymbol{N}}(\mathbf{1}_S) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(S)}$$

$$R_{\boldsymbol{N}}(\mathbf{1}_{V\setminus S}) = \frac{|E(S, V \setminus S)|}{\mathrm{vol}(V \setminus S)}$$

$$\begin{aligned}
\lambda_2 &= \min_{2-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x}) \\
&\le \max_{\boldsymbol{z} \in \mathrm{span}(\mathbf{1}_S, \mathbf{1}_{V\setminus S}) \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{z}) \\
&\le 2 \max\left\{R_{\boldsymbol{N}}(\mathbf{1}_S), R_{\boldsymbol{N}}\left(\mathbf{1}_{V\setminus S}\right)\right\} \\
&= 2 \cdot \frac{|E(S, V \setminus S)|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}} \\
&= 2 \cdot \Phi(S) \\
&= 2 \cdot \Phi(G)
\end{aligned}$$

Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{N}$:

$$R_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \frac{\sum_{\{u,v\} \in E} (x_u - x_v)^2}{d \sum_{v \in V} {x_v}^2}$$

Recall: $\lambda_2 = \min_{2-\text{dim space } \mathcal{V}} \max_{\boldsymbol{x} \in \mathcal{V} \setminus \{\mathbf{0}\}} R_{\boldsymbol{N}}(\boldsymbol{x})$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \mathrm{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\mathbf{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \le 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

Proof of the lemma:

- It suffices to prove it for the case of $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

$$R_{\boldsymbol{M}}(\alpha\boldsymbol{x} + \beta\boldsymbol{y}) \leq 2\max\{R_{\boldsymbol{M}}(\alpha\boldsymbol{x}), R_{\boldsymbol{M}}(\beta\boldsymbol{y})\} = 2\max\{R_{\boldsymbol{M}}(\boldsymbol{x}), R_{\boldsymbol{M}}(\boldsymbol{y})\}$$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \mathrm{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\boldsymbol{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2\max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

Proof of the lemma:

- It suffices to prove it for the case of $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

$\boldsymbol{N}$ is a real symmetric $(n \times n)$-matrix:

- Eigenvalues: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

We write:

- $\boldsymbol{x} = \sum_{i=1}^{n} a_i \boldsymbol{v}_i$
- $\boldsymbol{y} = \sum_{i=1}^{n} b_i \boldsymbol{v}_i$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \operatorname{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\boldsymbol{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

$$R_{\boldsymbol{N}}(\boldsymbol{z}) = \frac{\boldsymbol{z}^\top \boldsymbol{N} \boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{z}} = \frac{\sum_{i=1}^n \lambda_i (a_i + b_i)^2}{\|\boldsymbol{x} + \boldsymbol{y}\|^2}$$

Proof of the lemma:

- It suffices to prove it for the case of $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

$\boldsymbol{N}$ is a real symmetric $(n \times n)$-matrix:

- Eigenvalues: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

We write:

- $\boldsymbol{x} = \sum_{i=1}^n a_i \boldsymbol{v}_i$
- $\boldsymbol{y} = \sum_{i=1}^n b_i \boldsymbol{v}_i$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \operatorname{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\boldsymbol{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

$$R_{\boldsymbol{N}}(\boldsymbol{z}) = \frac{\boldsymbol{z}^\top \boldsymbol{N} \boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{z}}$$
$$= \frac{\sum_{i=1}^{n} \lambda_i (a_i + b_i)^2}{\|\boldsymbol{x} + \boldsymbol{y}\|^2}$$
$$\leq \frac{2 \sum_{i=1}^{n} \lambda_i \left(a_i^2 + b_i^2\right)}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2}$$

Proof of the lemma:

- It suffices to prove it for the case of $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

$\boldsymbol{N}$ is a real symmetric $(n \times n)$-matrix:

- Eigenvalues: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

$\boldsymbol{x} \perp \boldsymbol{y} \implies \|\boldsymbol{x} + \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2$

$(a_i + b_i)^2 \leq 2a_i^2 + 2b_i^2$

We write:

- $\boldsymbol{x} = \sum_{i=1}^{n} a_i \boldsymbol{v}_i$
- $\boldsymbol{y} = \sum_{i=1}^{n} b_i \boldsymbol{v}_i$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \operatorname{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\boldsymbol{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 1: $\frac{\lambda_2}{2} \leq \Phi(G)$

Proof of the lemma:

- It suffices to prove it for the case of $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

$\boldsymbol{N}$ is a real symmetric $(n \times n)$-matrix:

- Eigenvalues: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

We write:

- $\boldsymbol{x} = \sum_{i=1}^{n} a_i \boldsymbol{v}_i$
- $\boldsymbol{y} = \sum_{i=1}^{n} b_i \boldsymbol{v}_i$

$$
\begin{aligned}
R_{\boldsymbol{N}}(\boldsymbol{z}) &= \frac{\boldsymbol{z}^\top \boldsymbol{N} \boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{z}} \\
&= \frac{\sum_{i=1}^{n} \lambda_i (a_i + b_i)^2}{\|\boldsymbol{x} + \boldsymbol{y}\|^2} \\
&\leq \frac{2 \sum_{i=1}^{n} \lambda_i \left(a_i^2 + b_i^2\right)}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2} \\
&= \frac{2(\boldsymbol{x}^\top \boldsymbol{N} \boldsymbol{x} + \boldsymbol{y}^\top \boldsymbol{N} \boldsymbol{y})}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2} \\
&\leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}
\end{aligned}
$$

Lemma:

- Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two orthogonal vectors.
- Let $\boldsymbol{z} \in \operatorname{span}(\boldsymbol{x}, \boldsymbol{y}) \setminus \{\boldsymbol{0}\}$.

$$R_{\boldsymbol{N}}(\boldsymbol{z}) \leq 2 \max\{R_{\boldsymbol{N}}(\boldsymbol{x}), R_{\boldsymbol{N}}(\boldsymbol{y})\}$$

Part 2: $\Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:

- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Part 2: $\Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:

- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

$\downarrow$

Setting $\boldsymbol{x} = \boldsymbol{v}_2$ yields a cut with conductance at most $\sqrt{2\lambda_2}$.

$\downarrow$

$$\Phi(G) \leq \sqrt{2\lambda_2}$$

Part 2: $\Phi(G) \le \sqrt{2\lambda_2}$

Lemma:
- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Proof of the lemma:
- $\boldsymbol{x}' = \boldsymbol{x} - m\mathbf{1}$
  - m is the median of $\boldsymbol{x}$.
- $\boldsymbol{x}' = \boldsymbol{x}^+ - \boldsymbol{x}^-$
  - $x_v^+ = \begin{cases} x_v' & \text{if} \quad x_v' > 0 \\ 0 & \text{otherwise} \end{cases}$
  - $x_v^- = \begin{cases} -x_v' & \text{if} \quad x_v' < 0 \\ 0 & \text{otherwise} \end{cases}$

At least half of the
vertices have zero values.

Part 2: $\Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:
- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_N(\boldsymbol{x})}$.

Claim 1:
$R_N(\boldsymbol{x}) \geq R_N(\boldsymbol{x}') \geq \min\{R_N(\boldsymbol{x}^+), R_N(\boldsymbol{x}^-)\}$

Proof of the lemma:
- $\boldsymbol{x}' = \boldsymbol{x} - m\mathbf{1}$
  - m is the median of $\boldsymbol{x}$.
- $\boldsymbol{x}' = \boldsymbol{x}^+ - \boldsymbol{x}^-$
  - $x_v^+ = \begin{cases} x_v' & \text{if} \quad x_v' > 0 \\ 0 & \text{otherwise} \end{cases}$
  - $x_v^- = \begin{cases} -x_v' & \text{if} \quad x_v' < 0 \\ 0 & \text{otherwise} \end{cases}$

At least half of the
vertices have zero values.

Part 2: $\Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:

- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Proof of the lemma:

- $\boldsymbol{x}' = \boldsymbol{x} - m\mathbf{1}$
  - m is the median of $\boldsymbol{x}$.
- $\boldsymbol{x}' = \boldsymbol{x}^+ - \boldsymbol{x}^-$
  - $x_v^+ = \begin{cases} x_v' & \text{if} \quad x_v' > 0 \\ 0 & \text{otherwise} \end{cases}$
  - $x_v^- = \begin{cases} -x_v' & \text{if} \quad x_v' < 0 \\ 0 & \text{otherwise} \end{cases}$

At least half of the vertices have zero values.

Claim 1:

$R_{\boldsymbol{N}}(\boldsymbol{x}) \geq R_{\boldsymbol{N}}(\boldsymbol{x}') \geq \min\{R_{\boldsymbol{N}}(\boldsymbol{x}^+), R_{\boldsymbol{N}}(\boldsymbol{x}^-)\}$

Claim 2:

Let $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. There exists a threshold $t > 0$ such that $S_t = \{v : y_v \geq t\}$ satisfies:

$$\frac{|E(S_t, V \setminus S_t)|}{d|S_t|} \leq \sqrt{2R_{\boldsymbol{N}}(\boldsymbol{y})}$$

Part 2: $\Phi(G) \le \sqrt{2\lambda_2}$

Lemma:

- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

This yields a cut with conductance at most $\sqrt{2\min\{R_{\boldsymbol{N}}(\boldsymbol{x}^+), R_{\boldsymbol{N}}(\boldsymbol{x}^-)\}} \le \sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Proof of the lemma:

- $\boldsymbol{x}' = \boldsymbol{x} - m\mathbf{1}$
  - m is the median of $\boldsymbol{x}$.
- $\boldsymbol{x}' = \boldsymbol{x}^+ - \boldsymbol{x}^-$
  - $x_v^+ = \begin{cases} x_v' & \text{if} \quad x_v' > 0 \\ 0 & \text{otherwise} \end{cases}$
  - $x_v^- = \begin{cases} -x_v' & \text{if} \quad x_v' < 0 \\ 0 & \text{otherwise} \end{cases}$

Claim 1:

$R_{\boldsymbol{N}}(\boldsymbol{x}) \ge R_{\boldsymbol{N}}(\boldsymbol{x}') \ge \min\{R_{\boldsymbol{N}}(\boldsymbol{x}^+), R_{\boldsymbol{N}}(\boldsymbol{x}^-)\}$

Claim 2:

Let $\boldsymbol{y} \in \mathbb{R}^n_{\ge 0}$. There exists a threshold $t > 0$ such that $S_t = \{v : y_v \ge t\}$ satisfies:

$$\frac{|E(S_t, V \setminus S_t)|}{d|S_t|} \le \sqrt{2R_{\boldsymbol{N}}(\boldsymbol{y})}$$

At least half of the vertices have zero values.

Setting $\boldsymbol{y}$ to be $\boldsymbol{x}^+$ or $\boldsymbol{x}^-$

$$\Phi(S_t) = \frac{|E(S_t, V \setminus S_t)|}{d\min\{|S_t|, |V \setminus S_t|\}} = \frac{|E(S_t, V \setminus S_t)|}{d|S_t|} \le \sqrt{2R_{\boldsymbol{N}}(\boldsymbol{y})}$$

Proof of Claim 1

Lemma:
- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Claim 1:
$R_{\boldsymbol{N}}(\boldsymbol{x}) \geq R_{\boldsymbol{N}}(\boldsymbol{x}') \geq \min\{R_{\boldsymbol{N}}(\boldsymbol{x}^{+}), R_{\boldsymbol{N}}(\boldsymbol{x}^{-})\}$

- $\boldsymbol{x}' = \boldsymbol{x} - m\mathbf{1}$
  - m is the median of $\boldsymbol{x}$.

$$R_{\boldsymbol{N}}(\boldsymbol{x}') = \frac{(\boldsymbol{x}-m\mathbf{1})^{\top}\boldsymbol{N}(\boldsymbol{x}-m\mathbf{1})}{\|\boldsymbol{x}-m\mathbf{1}\|^2} = \frac{\boldsymbol{x}^{\top}\boldsymbol{N}\boldsymbol{x}}{\|\boldsymbol{x}-m\mathbf{1}\|^2} = \frac{\boldsymbol{x}^{\top}\boldsymbol{N}\boldsymbol{x}}{\|\boldsymbol{x}\|^2+m^2} \leq \frac{\boldsymbol{x}^{\top}\boldsymbol{N}\boldsymbol{x}}{\|\boldsymbol{x}\|^2} = R_{\boldsymbol{N}}(\boldsymbol{x})$$

$\mathbf{1}$ is an eigenvector of $\boldsymbol{N}$ with a zero eigenvalue.

$\boldsymbol{x} \perp \mathbf{1}$

Proof of Claim 1

Lemma:

- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_N(\boldsymbol{x})}$.

Claim 1:
$R_N(\boldsymbol{x}) \geq R_N(\boldsymbol{x}') \geq \min\{R_N(\boldsymbol{x}^+), R_N(\boldsymbol{x}^-)\}$

- $\boldsymbol{x}' = \boldsymbol{x}^+ - \boldsymbol{x}^-$
  - $x_v^+ = \begin{cases} x_v' & \text{if} \quad x_v' > 0 \\ 0 & \text{otherwise} \end{cases}$
  - $x_v^- = \begin{cases} -x_v' & \text{if} \quad x_v' < 0 \\ 0 & \text{otherwise} \end{cases}$

$$R_N(\boldsymbol{x}') = \frac{\sum_{\{u,v\}\in E}(x_u - x_v)^2}{d\|\boldsymbol{x}'\|^2} \geq \frac{\sum_{\{u,v\}\in E}\left((x_u^+ - x_v^+)^2 + (x_u^- - x_v^-)^2\right)}{d(\|\boldsymbol{x}^+\|^2 + \|\boldsymbol{x}^-\|^2)} \geq \min\{R_N(\boldsymbol{x}^+), R_N(\boldsymbol{x}^-)\}$$

$\|\boldsymbol{x}'\|^2 = \|\boldsymbol{x}^+\|^2 + \|\boldsymbol{x}^-\|^2$
$(x_u - x_v)^2 \geq (x_u^+ - x_v^+)^2 + (x_u^- - x_v^-)^2$

Proof of Claim 2

- Assume $\max_v y_v = 1$. Multiplication by a scalar does not affect the Rayleigh quotient of a vector.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

We will prove:
$$\frac{\mathbb{E}[|E(S_t, V \setminus S_t)|]}{d\mathbb{E}[|S_t|]} \leq \sqrt{2R_N(\boldsymbol{y})}$$

Claim 2:
Let $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. There exists a threshold $t > 0$ such that $S_t = \{v : y_v \geq t\}$ satisfies:
$$\frac{|E(S_t, V \setminus S_t)|}{d|S_t|} \leq \sqrt{2R_N(\boldsymbol{y})}$$

Sub-claim 1: $d\mathbb{E}[|S_t|] = d\sum_v y_v^2$

Proof of Claim 2

Sub-claim 2: $\mathbb{E}[|E(S_t, V \setminus S_t)|] = \sqrt{2\left(\sum_{\{u,v\}\in E}(y_u - y_v)^2\right)\left(d\sum_v y_v^2\right)}$

- Assume $\max_v y_v = 1$.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

$$\frac{\mathbb{E}[|E(S_t, V \setminus S_t)|]}{d\mathbb{E}[|S_t|]} \leq \sqrt{2 \cdot \frac{\sum_{\{u,v\}\in E}(y_u - y_v)^2}{d\sum_{v\in V} {y_v}^2}} = \sqrt{2R_N(\boldsymbol{y})}$$

We will prove:

$$\frac{\mathbb{E}[|E(S_t, V \setminus S_t)|]}{d\mathbb{E}[|S_t|]} \leq \sqrt{2R_N(\boldsymbol{y})}$$

Sub-claim 1: $d\mathbb{E}[|S_t|] = d\sum_v y_v^2$

Proof of Claim 2

Sub-claim 2: $\mathbb{E}[|E(S_t, V \setminus S_t)|] = \sqrt{2\left(\sum_{\{u,v\}\in E}(y_u - y_v)^2\right)\left(d\sum_v y_v^2\right)}$

- Assume $\max\limits_v y_v = 1$.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

Proof of Sub-claim 1:
For each vertex v, it is added to S_t when $t^2 \in [0, y_v^2]$, which happens with probability y_v^2.

Linearity of expectations:

$$\mathbb{E}[|S_t|] = \sum_v y_v^2$$

Proof of Claim 2

Sub-claim 1: $d\mathbb{E}[|S_t|] = d\sum_v y_v^2$

Sub-claim 2: $\mathbb{E}[|E(S_t, V \setminus S_t)|] = \sqrt{2\left(\sum_{\{u,v\}\in E}(y_u - y_v)^2\right)(d\sum_v y_v^2)}$

- Assume $\max_v y_v = 1$.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

Proof of Sub-claim 2:

For each edge $e = \{u, v\}$, it belongs to $E(S_t, V \setminus S_t)$ when

$$t^2 \in (\min\{y_u^2, y_v^2\}, \max\{y_u^2, y_v^2\}],$$

which happens with probability

$$|y_u^2 - y_v^2|.$$

Sub-claim 1: $d\mathbb{E}[|S_t|] = d\sum_v y_v^2$

Proof of Claim 2

Sub-claim 2: $\mathbb{E}[|E(S_t, V \setminus S_t)|] = \sqrt{2\left(\sum_{\{u,v\}\in E}(y_u - y_v)^2\right)\left(d\sum_v y_v^2\right)}$

- Assume $\max_v y_v = 1$.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

Proof of Sub-claim 2:

For each edge $e = \{u, v\}$, it belongs to $E(S_t, V \setminus S_t)$ when

$$t^2 \in (\min\{y_u^2, y_v^2\}, \max\{y_u^2, y_v^2\}],$$

which happens with probability

$$|y_u^2 - y_v^2|.$$

$$\mathbb{E}[|E(S_t, V \setminus S_t)|] = \sum_{\{u,v\}\in E} |y_u^2 - y_v^2|$$

$$= \sum_{\{u,v\}\in E} (y_u + y_v)|y_u - y_v|$$

Cauchy-Schwarz $$\leq \sqrt{\sum_{\{u,v\}\in E} (y_u + y_v)^2}\sqrt{\sum_{\{u,v\}\in E} (y_u - y_v)^2}$$

Sub-claim 1: $d\mathbb{E}[|S_t|] = d\sum_v y_v^2$

Proof of Claim 2

Sub-claim 2: $\mathbb{E}[|E(S_t, V\setminus S_t)|] = \sqrt{2\left(\sum_{\{u,v\}\in E}(y_u - y_v)^2\right)\left(d\sum_v y_v^2\right)}$

- Assume $\max_v y_v = 1$.
- Select t randomly in such a way that t^2 is uniformly distributed in $[0,1]$.
- Consider $S_t = \{v : y_v \geq t\}$.

Proof of Sub-claim 2:

For each edge $e = \{u, v\}$, it belongs to $E(S_t, V\setminus S_t)$ when

$$t^2 \in (\min\{y_u^2, y_v^2\}, \max\{y_u^2, y_v^2\}],$$

which happens with probability

$$|y_u^2 - y_v^2|.$$

$$\mathbb{E}[|E(S_t, V\setminus S_t)|] = \sum_{\{u,v\}\in E} |y_u^2 - y_v^2|$$

$$= \sum_{\{u,v\}\in E} (y_u + y_v)|y_u - y_v|$$

$$\leq \sqrt{\sum_{\{u,v\}\in E} (y_u + y_v)^2}\sqrt{\sum_{\{u,v\}\in E} (y_u - y_v)^2}$$

$$\sum_{\{u,v\}\in E} (y_u + y_v)^2 \leq 2\sum_{\{u,v\}\in E} (y_u^2 + y_v^2) = 2d\sum_v y_v^2$$

Cheeger vs. Leighton–Rao

- Cheeger inequality yields more practical algorithms.

Small and precise constants:

$$\frac{\lambda_2}{2} \le \Phi(G) \le \sqrt{2\lambda_2}$$

Simple and fast algorithm:

After sorting the vertices according to a given vector $\boldsymbol{x}$ orthogonal to $\mathbf{1}$, finding a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$ can be done in linear time.

Exercise!

Setting $\boldsymbol{x} = \boldsymbol{v}_2$ yields a cut with conductance at most $\sqrt{2\lambda_2} \le 2\sqrt{\Phi(G)}$.

$$\frac{\lambda_2}{2} \le \Phi(G)$$

Application: Image segmentation

A low-conductance cut

$\updownarrow$

A boundary where few high-similarity edges are cut

$\updownarrow$

A natural separation between regions

- Pixels = vertices.
- Edges = similarity between pixels.

Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation." *IEEE Transactions on pattern analysis and machine intelligence* (2000).

Application: Image segmentation

A low-conductance cut

A boundary where few high-similarity edges are cut

A natural separation between regions

- Pixels = vertices.
- Edges = similarity between pixels.

- This enables **image segmentation** by grouping similar pixels into meaningful regions.

- Applications:
  - Medical imaging: identifying organs, tumors, or tissue boundaries.
  - Robotics: scene understanding and object detection

Shi, Jianbo, and Jitendra Malik. “Normalized cuts and image segmentation.” *IEEE Transactions on pattern analysis and machine intelligence* (2000).

Application: graph partitioning

- **Expander decomposition:**
  - Iteratively finding low-conductance cuts partitions the vertices into well-connected clusters with few inter-cluster edges.

  This tool has many applications in designing algorithms for **general graphs**.

- **Detecting communities in a social network:**
  - Splitting the network into clusters of tightly-connected users.

Extension 1: multi-way cuts

- **Recall:** $\lambda_k = 0$ if and only if G has at least k connected components.

- Similar to Cheeger inequality, there is a robust version of this fact.

Extension 1: multi-way cuts

- **Recall:** $\lambda_k = 0$ if and only if G has at least k connected components.

- Similar to Cheeger inequality, there is a robust version of this fact.

$$\Phi_k(G) = \min_{\text{disjoint } S_1 \subseteq V, S_2 \subseteq V, \ldots, S_k \subseteq V} \max_{1 \leq i \leq k} \frac{|E(S_i, V \setminus S_i)|}{d|S_i|}$$

$$\frac{\lambda_k}{2} \leq \Phi_k(G) \in O(k^2) \cdot \lambda_k$$

https://doi.org/10.1145/2665063

Extension 2: bipartiteness

- **Recall:** $\lambda_n = 2$ if and only if G has a bipartite connected component.

- Similar to Cheeger inequality, there is a robust version of this fact.

Extension 2: bipartiteness

- **Recall:** $\lambda_n = 2$ if and only if G has a bipartite connected component.
- Similar to Cheeger inequality, there is a robust version of this fact.

A graph $G = (V, E)$ is ϵ-close to bipartite if one can make it bipartite by removing at most an ϵ-fraction of its edges.

Theorem: If A graph $G = (V, E)$ is ϵ-close to bipartite, then $\lambda_n \geq 2(1 - \epsilon)$.

Extension 2: bipartiteness

- **Recall:** $\lambda_n = 2$ if and only if G has a bipartite connected component.

- Similar to Cheeger inequality, there is a robust version of this fact.

A graph $G = (V, E)$ is ϵ-close to bipartite if one can make it bipartite by removing at most an ϵ-fraction of its edges.

Theorem: If A graph $G = (V, E)$ is ϵ-close to bipartite, then $\lambda_n \geq 2(1 - \epsilon)$.

Proof:
- Suppose $G = (V, E)$ is ϵ-close to a bipartite graph with the bipartition $V = A \cup B$.
- Let $\boldsymbol{y} = \mathbf{1}_A - \mathbf{1}_B$.

$$2 - \lambda_n = \min_{x \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\sum_{\{u,v\} \in E} (x_u + x_v)^2}{d \sum_{v \in V} {x_v}^2} \leq \frac{\sum_{\{u,v\} \in E} (y_u + y_v)^2}{d \sum_{v \in V} {y_v}^2} \leq \frac{4\epsilon |E|}{2|E|} = 2\epsilon$$

Extension 2: bipartiteness

- **Recall:** $\lambda_n = 2$ if and only if G has a bipartite connected component.

- Similar to Cheeger inequality, there is a robust version of this fact.

A graph $G = (V, E)$ is ϵ-close to bipartite if one can make it bipartite by removing at most an ϵ-fraction of its edges.

Theorem: If A graph $G = (V, E)$ is ϵ-close to bipartite, then $\lambda_n \geq 2(1-\epsilon)$.

Moreover, using approximate eigenvector computation, a set $D \subseteq E$ with $|D| \in O(\sqrt{\epsilon}) \cdot |E|$ such that $G' = (V, E \setminus D)$ is bipartite can be computed in polynomial time.

Outlook

- Given the eigenvector $\boldsymbol{v}_2$, we can obtain a cut of conductance at most $2\sqrt{\Phi(G)}$ in polynomial time.

- **Next:** The power method.
  - A fast algorithm for finding approximate eigenvectors.

For any constant $\epsilon > 0$, this allows us to obtain a cut of conductance at most $(1+\epsilon) \cdot 2\sqrt{\Phi(G)}$ in polynomial time.

References

- **Main reference:**
  - Lecture 4.1 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapters 4 and 5 of https://lucatrevisan.github.io/books/expanders-2016.pdf

- **Additional/optional reading:** Extensions of Cheeger inequality
  - Chapters 6–8 of https://lucatrevisan.github.io/books/expanders-2016.pdf

CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

Lecture 6:

Expanders – the power method

Recap

Cheeger inequality: $\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:
- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Setting $\boldsymbol{x} = \boldsymbol{v}_2$ yields a cut with conductance at most $\sqrt{2\lambda_2}$.

Recap

Cheeger inequality: $\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$

Lemma:
- Let $\boldsymbol{x}$ be any vector orthogonal to $\mathbf{1}$.
- There is a polynomial-time algorithm that computes a cut with conductance at most $\sqrt{2R_{\boldsymbol{N}}(\boldsymbol{x})}$.

Setting $\boldsymbol{x} = \boldsymbol{v}_2$ yields a cut with conductance at most $\sqrt{2\lambda_2}$.

Our goal: Obtaining this in polynomial time.

From $\overbrace{\text{any vector } \boldsymbol{x} \perp \mathbf{1} \text{ with } R_{\boldsymbol{N}}(\boldsymbol{x}) \leq \lambda_2 + \epsilon}$, we get a cut with conductance at most $\sqrt{2(\lambda_2 + \epsilon)}$.

Setting

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$ (orthonormal vectors)

Eigenvalue range for d-regular graphs:

- $\boldsymbol{A}$ (adjacency): $[-d, d]$ ✗
- $\boldsymbol{L} = d\boldsymbol{I} - \boldsymbol{A}$ (Laplacian): $[0, 2d]$ ✓
- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$ (normalized Laplacian): $[0, 2]$ ✓

Setting

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ (orthonormal vectors)

- **First goal:** Approximating the largest eigenvector v_n.
  - Find a vector $\boldsymbol{x}$ with $R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n$.

We will later extend this to:

- The second largest eigenvector $\boldsymbol{v}_{n-1}$.
- The smallest eigenvector $\boldsymbol{v}_1$.
- The second smallest eigenvector $\boldsymbol{v}_2$ (our main goal).

The power method

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Claim: There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n] \geq \frac{3}{16}$.

Proof sketch:
The matrix $\boldsymbol{M}^k$ has:
- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

The power method

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Claim: There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n] \geq \frac{3}{16}$.

Proof sketch:
The matrix $\boldsymbol{M}^k$ has:
- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon) \cdot \lambda_n < \lambda_{i^*+1}$.

Then $\lambda_{i^*}^k \leq (1-\epsilon)^k \cdot \lambda_n^k$

$$\underbrace{(1-\epsilon)^k}_{1/\mathrm{poly}(n)}$$

The power method

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Claim: There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n] \geq \frac{3}{16}$.

Proof sketch:
The matrix $\boldsymbol{M}^k$ has:
- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon) \cdot \lambda_n < \lambda_{i^*+1}$.

Then $\lambda_{i^*}^k \leq (1-\epsilon)^k \cdot \lambda_n^k$

$\downarrow$

$1/\text{poly}(n)$

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

$$\boldsymbol{x} = \boldsymbol{M}^k\boldsymbol{x}_0 = \underbrace{c_1\lambda_1^k\boldsymbol{v}_1 + c_2\lambda_2^k\boldsymbol{v}_2 + \cdots + c_{i^*}\lambda_{i^*}^k\boldsymbol{v}_{i^*}}_{\text{Negligibly small}} + \underbrace{c_{i^*+1}\lambda_{i^*+1}^k\boldsymbol{v}_{i^*+1} + \cdots + c_n\lambda_n^k\boldsymbol{v}_n}_{R_{\boldsymbol{M}}(\cdot) \geq (1-\epsilon)\cdot\lambda_n}$$

The power method

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Claim: There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n] \geq \frac{3}{16}$.

Proof sketch:
The matrix $\boldsymbol{M}^k$ has:

- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon) \cdot \lambda_n < \lambda_{i^*+1}$.

Then $\lambda_{i^*}^k \leq (1-\epsilon)^k \cdot \lambda_n^k$

$$\underbrace{(1-\epsilon)^k}_{1/\mathrm{poly}(n)}$$

To make this argument work, we need to start with a sufficiently large $c_n = \langle \boldsymbol{x}_0, \boldsymbol{v}_n \rangle$.

$$\boldsymbol{x}_0 = c_1 \boldsymbol{v}_1 + c_2 \boldsymbol{v}_2 + \cdots + c_i \boldsymbol{v}_i + c_{i+1} \boldsymbol{v}_{i+1} + \cdots + c_n \boldsymbol{v}_n$$

$$\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0 = \underbrace{c_1 \lambda_1^k \boldsymbol{v}_1 + c_2 \lambda_2^k \boldsymbol{v}_2 + \cdots + c_{i^*} \lambda_{i^*}^k \boldsymbol{v}_{i^*}}_{\text{Negligibly small}} + \underbrace{c_{i^*+1} \lambda_{i^*+1}^k \boldsymbol{v}_{i^*+1} + \cdots + c_n \lambda_n^k \boldsymbol{v}_n}_{R_{\boldsymbol{M}}(\cdot) \geq (1-\epsilon) \cdot \lambda_n}$$

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

To make this argument work, we need to start with a sufficiently large $c_n = \langle \boldsymbol{x_0}, \boldsymbol{v}_n \rangle$.

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v} \rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER**:**
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Proof:
- $\boldsymbol{x}_0 = (x_1, x_2, \dots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \dots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v} \rangle = \sum_{i=1}^n x_i v_i$

$$\mathbb{E}[Z] = \sum_{i=1}^{n} \mathbb{E}[x_i] v_i = 0$$

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Proof:
- $\boldsymbol{x}_0 = (x_1, x_2, \dots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \dots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v}\rangle = \sum_{i=1}^n x_i v_i$

$$\mathbb{E}[Z] = \sum_{i=1}^n \mathbb{E}[x_i] v_i = 0$$

$$\mathbb{E}[Z^2] = \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[x_i x_j] v_i v_j = \sum_{i=1}^n \mathbb{E}[x_i^2] v_i^2 = \sum_{i=1}^n v_i^2 = 1$$

$$\mathbb{E}[x_i x_j] = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$\boldsymbol{v}$ is a unit vector.

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER**:**
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Proof:
- $\boldsymbol{x}_0 = (x_1, x_2, \ldots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v}\rangle = \sum_{i=1}^n x_i v_i$

$$\mathbb{E}[Z] = \sum_{i=1}^n \mathbb{E}[x_i] v_i = 0$$

$$\mathbb{E}[Z^2] = \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[x_i x_j] v_i v_j = \sum_{i=1}^n \mathbb{E}[x_i^2] v_i^2 = \sum_{i=1}^n v_i^2 = 1$$

$$\mathbb{E}[Z^4] = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{\ell=1}^n \underbrace{\mathbb{E}[x_i x_j x_k x_\ell]}\, v_i v_j v_k v_\ell = 3 \sum_{i=1}^n \mathbb{E}[x_i^2] v_i^2 \sum_{j=1}^n \mathbb{E}[x_j^2] v_j^2 - 2 \sum_{i=1}^n \mathbb{E}[x_i^4] v_i^4$$

$i \notin \{j,k,l\} \to \mathbb{E}[x_i x_j x_k x_\ell] = 0$
$j \notin \{i,k,l\} \to \mathbb{E}[x_i x_j x_k x_\ell] = 0$
$k \notin \{i,j,l\} \to \mathbb{E}[x_i x_j x_k x_\ell] = 0$
$\ell \notin \{i,j,k\} \to \mathbb{E}[x_i x_j x_k x_\ell] = 0$

For any $i \neq j$, $\mathbb{E}[x_i^2 x_j^2]$ appears $\binom{4}{2} = 6$ times.
For any i, $\mathbb{E}[x_i^4]$ appears once.

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER**:**
Input: a matrix $\boldsymbol{M}$ and a parameter k.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Proof:

- $\boldsymbol{x}_0 = (x_1, x_2, \dots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \dots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v}\rangle = \sum_{i=1}^n x_i v_i$

$$\mathbb{E}[Z] = \sum_{i=1}^n \mathbb{E}[x_i] v_i = 0$$

$$\mathbb{E}[Z^2] = \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}[x_i x_j] v_i v_j = \sum_{i=1}^n \mathbb{E}[x_i^2] v_i^2 = \sum_{i=1}^n v_i^2 = 1$$

$$\mathbb{E}[Z^4] = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{\ell=1}^n \mathbb{E}[x_i x_j x_k x_\ell] v_i v_j v_k v_\ell = 3 \underbrace{\sum_{i=1}^n \mathbb{E}[x_i^2] v_i^2}_{=1} \underbrace{\sum_{j=1}^n \mathbb{E}[x_j^2] v_j^2}_{=1} - 2 \underbrace{\sum_{i=1}^n \mathbb{E}[x_i^4] v_i^4}_{= \sum_{i=1}^n v_i^4 \geq 0} \leq 3$$

The power method

Algorithm POWER**:**
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Proof:
- $\boldsymbol{x}_0 = (x_1, x_2, \dots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \dots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v}\rangle = \sum_{i=1}^n x_i v_i$

- $\mathbb{E}[Z] = 0$
- $\mathbb{E}[Z^2] = 1$
- $\mathbb{E}[Z^4] \leq 3$
- $\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - \mathbb{E}[Z^2]^2 < \infty$

The power method

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

Proof:
- $\boldsymbol{x}_0 = (x_1, x_2, \ldots, x_n)$
- $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$
- $Z = \langle \boldsymbol{x}_0, \boldsymbol{v}\rangle = \sum_{i=1}^n x_i v_i$

- $\mathbb{E}[Z] = 0$
- $\mathbb{E}[Z^2] = 1$
- $\mathbb{E}[Z^4] \leq 3$
- $\mathrm{Var}[Z^2] = \mathbb{E}[Z^4] - \mathbb{E}[Z^2]^2 < \infty$

$$\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] = \Pr\left[|Z| \geq \frac{1}{2}\right] = \Pr\left[Z^2 \geq \frac{1}{4}\right]$$

$$= \Pr\left[Z^2 \geq \delta \mathbb{E}[Z^2]\right] \geq (1-\delta)^2 \frac{\mathbb{E}[Z^2]^2}{\mathbb{E}[Z^4]} \geq \left(\frac{3}{4}\right)^2 \cdot \frac{1}{3} = \frac{3}{16}$$

$\delta = \frac{1}{4}$

$X = Z^2$

Paley–Zygmund inequality: $\Pr\left[X \geq \delta \mathbb{E}[X]\right] \geq (1-\delta)^2 \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$, for $X \geq 0$, $\mathrm{Var}[X] < \infty$, $\delta \in (0,1)$

https://www.lkozma.net/inequalities_cheat_sheet/ineq.pdf

The power method

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

Assume $|c_n| = |\langle \boldsymbol{x}_0, \boldsymbol{v}_n \rangle| \geq \frac{1}{2}$.

The matrix $\boldsymbol{M}^k$ has:

- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon) \cdot \lambda_n < \lambda_{i^*+1}$.

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

$\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$

$$= \frac{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k+1} \boldsymbol{x}_0}{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k} \boldsymbol{x}_0}$$

$$= \frac{\sum_{i=1}^n c_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^n c_i^2 \lambda_i^{2k}}$$

The power method

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

Assume $|c_n| = |\langle \boldsymbol{x_0}, \boldsymbol{v}_n\rangle| \geq \frac{1}{2}$.

The matrix $\boldsymbol{M}^k$ has:

- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon)\cdot\lambda_n < \lambda_{i^*+1}$.

$$R_{\boldsymbol{M}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} = \frac{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k+1}\boldsymbol{x}_0}{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k}\boldsymbol{x}_0} = \frac{\sum_{i=1}^n c_i^2\lambda_i^{2k+1}}{\sum_{i=1}^n c_i^2\lambda_i^{2k}} \geq \frac{(1-\epsilon)\cdot\lambda_n\cdot\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}{\sum_{i=1}^{i^*} c_i^2\lambda_i^{2k} + \sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}$$

The power method

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

Assume $|c_n| = |\langle \boldsymbol{x_0}, \boldsymbol{v}_n \rangle| \geq \frac{1}{2}$.

The matrix $\boldsymbol{M}^k$ has:

- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon)\cdot\lambda_n < \lambda_{i^*+1}$.

$$\sum_{i=1}^{n} c_i^2 = \|\boldsymbol{x}_0\|^2 = n$$

$$4 \cdot c_n^2 \geq 1$$

$$\begin{aligned}
\sum_{i=1}^{i^*} c_i^2 \lambda_i^{2k} &\leq (1-\epsilon)^{2k} \cdot \lambda_n^{2k} \cdot \sum_{i=1}^{n} c_i^2 \\
&= (1-\epsilon)^{2k} \cdot n \cdot \lambda_n^{2k} \\
&\leq (1-\epsilon)^{2k} \cdot 4n \cdot c_n^2 \lambda_n^{2k} \\
&\leq (1-\epsilon)^{2k} \cdot 4n \cdot \sum_{i=i^*+1}^{n} c_i^2 \lambda_i^{2k}
\end{aligned}$$

$$\begin{aligned}
R_{\boldsymbol{M}}(\boldsymbol{x}) &= \frac{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \\
&= \frac{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k+1} \boldsymbol{x}_0}{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k} \boldsymbol{x}_0} \\
&= \frac{\sum_{i=1}^{n} c_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^{n} c_i^2 \lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n \cdot \sum_{i=i^*+1}^{n} c_i^2 \lambda_i^{2k}}{\sum_{i=1}^{i^*} c_i^2 \lambda_i^{2k} + \sum_{i=i^*+1}^{n} c_i^2 \lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n \cdot \sum_{i=i^*+1}^{n} c_i^2 \lambda_i^{2k}}{(1+(1-\epsilon)^{2k}\cdot 4n)\sum_{i=i^*+1}^{n} c_i^2 \lambda_i^{2k}}
\end{aligned}$$

The power method

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

Assume $|c_n| = |\langle \boldsymbol{x_0}, \boldsymbol{v}_n\rangle| \geq \frac{1}{2}$.

The matrix $\boldsymbol{M}^k$ has:

- Eigenvalues: $0 \leq \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$
- The same eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

Suppose $\lambda_{i^*} \leq (1-\epsilon)\cdot\lambda_n < \lambda_{i^*+1}$.

$$\begin{aligned}
R_{\boldsymbol{M}}(\boldsymbol{x}) &= \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} \\
&= \frac{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k+1}\boldsymbol{x}_0}{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k}\boldsymbol{x}_0} \\
&= \frac{\sum_{i=1}^n c_i^2\lambda_i^{2k+1}}{\sum_{i=1}^n c_i^2\lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n\cdot\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}{\sum_{i=1}^{i^*} c_i^2\lambda_i^{2k} + \sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n\cdot\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}{(1+(1-\epsilon)^{2k}\cdot 4n)\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}} \\
&= \frac{(1-\epsilon)\cdot\lambda_n}{1+(1-\epsilon)^{2k}\cdot 4n} \\
k \in O\left(\frac{\log n}{\epsilon}\right) \quad &\in \big(1-O(\epsilon)\big)\cdot\lambda_n
\end{aligned}$$

The power method

$$\boldsymbol{x}_0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_i\boldsymbol{v}_i + c_{i+1}\boldsymbol{v}_{i+1} + \cdots + c_n\boldsymbol{v}_n$$

Assume $|c_n| = |\langle \boldsymbol{x}_0, \boldsymbol{v}_n\rangle| \geq \frac{1}{2}$.

This happens with probability $\geq \frac{3}{16}$.

Claim: There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon)\cdot\lambda_n] \geq \frac{3}{16}$.

with a change of variable: $\epsilon' \in \Theta(\epsilon)$

$$\begin{aligned}
R_{\boldsymbol{M}}(\boldsymbol{x}) &= \frac{\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}} \\
&= \frac{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k+1}\boldsymbol{x}_0}{\boldsymbol{x}_0^\top \boldsymbol{M}^{2k}\boldsymbol{x}_0} \\
&= \frac{\sum_{i=1}^n c_i^2\lambda_i^{2k+1}}{\sum_{i=1}^n c_i^2\lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n\cdot\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}{\sum_{i=1}^{i^*} c_i^2\lambda_i^{2k} + \sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}} \\
&\geq \frac{(1-\epsilon)\cdot\lambda_n\cdot\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}}{(1+(1-\epsilon)^{2k}\cdot 4n)\sum_{i=i^*+1}^n c_i^2\lambda_i^{2k}} \\
&= \frac{(1-\epsilon)\cdot\lambda_n}{1+(1-\epsilon)^{2k}\cdot 4n} \\
&\in \big(1-O(\epsilon)\big)\cdot\lambda_n
\end{aligned}$$

Second largest eigenvector

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ (orthonormal vectors)

- **Done:** Find a vector $\boldsymbol{x}$ with $R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_n$.

- **Next:** The second largest eigenvector $\boldsymbol{v}_{n-1}$.

Given $\boldsymbol{v}_n$, the goal is to find a vector $\boldsymbol{x} \perp \boldsymbol{v}_n$ with $R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon) \cdot \lambda_{n-1}$.

Second largest eigenvector

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

This restricts ourselves to $\text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_{n-1}\}$.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- $\boldsymbol{x}_0' = \boldsymbol{x}_0 - \langle \boldsymbol{x}_0, \boldsymbol{v}_n \rangle \cdot \boldsymbol{v}_n$
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0'$.

Second largest eigenvector

Claim: for any unit vector $\boldsymbol{v}$, $\Pr\left[|\langle \boldsymbol{x}_0, \boldsymbol{v}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$

$$\langle \boldsymbol{x}_0', \boldsymbol{v}_{n-1}\rangle = \langle \boldsymbol{x}_0, \boldsymbol{v}_{n-1}\rangle$$

We still have:

$$\Pr\left[|\langle \boldsymbol{x}_0', \boldsymbol{v}_{n-1}\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}$$

The same analysis:

There is a choice of parameter $k \in O\left(\frac{\log n}{\epsilon}\right)$ such that $\Pr[R_{\boldsymbol{M}}(\boldsymbol{x}) \geq (1-\epsilon)\cdot\lambda_{n-1}] \geq \frac{3}{16}$.

Algorithm POWER:
Input: a matrix $\boldsymbol{M}$ and a parameter k.
- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0$.

- Pick $\boldsymbol{x}_0 \in \{-1,1\}^n$ uniformly at random.
- $\boldsymbol{x}_0' = \boldsymbol{x}_0 - \langle \boldsymbol{x}_0, \boldsymbol{v}_n\rangle \cdot \boldsymbol{v}_n$
- Return $\boldsymbol{x} = \boldsymbol{M}^k \boldsymbol{x}_0'$.

Approximating $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
  - Eigenvalues: $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$ (orthonormal vectors)

- Suppose we know an upper bound $c \geq \lambda_n$
- Consider $c\boldsymbol{I} - \boldsymbol{M}$:
  - Same eigenvectors: $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$
  - Eigenvalues: $\underbrace{c - \lambda_1 \geq c - \lambda_2}_{\text{Now we can use the same method to approximate } \boldsymbol{v}_1 \text{ and } \boldsymbol{v}_2.} \geq \cdots \geq c - \lambda_n \geq 0$

Computing a low-conductance cut

- Approximating the second largest eigenvector of $2\boldsymbol{I} - \boldsymbol{N}$:
  - We get a vector $\boldsymbol{x} \perp \mathbf{1}$ with

$$R_{2\boldsymbol{I}-\boldsymbol{N}}(\boldsymbol{x}) = 2 - R_{\boldsymbol{N}}(\boldsymbol{x}) \geq (1 - \epsilon) \cdot (2 - \lambda_2)$$

$$\big\downarrow$$

$$R_{\boldsymbol{N}}(\boldsymbol{x}) \leq 2 - (1 - \epsilon) \cdot (2 - \lambda_2) \leq \lambda_2 + 2\epsilon$$

$$\Big\downarrow \; \epsilon' = 2\epsilon$$

We get a cut with conductance at most $\sqrt{2(\lambda_2 + \epsilon')}$.

Summary

- Let $\boldsymbol{M}$ be a real symmetric $(n \times n)$-matrix.
    - Eigenvalues: $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
    - Eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$

We showed how to approximate them efficiently.

Many applications in theory and in practice.

Outlook

- We finished the discussion of these two topics:
  - Conductance approximation.
  - Spectral graph theory.

- **Next:** The <u>probabilistic aspect</u> of expanders.

This includes random walks and their applications.

References

- **Main reference:**
  - Lecture 4.2 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapter 9 of https://lucatrevisan.github.io/books/expanders-2016.pdf

- **Additional/optional reading:** The power method in distributed computing.
  - "Distributed Sparsest Cut via Eigenvalue Estimation" by Yannic Maus and Tijn de Vos https://arxiv.org/pdf/2508.19898

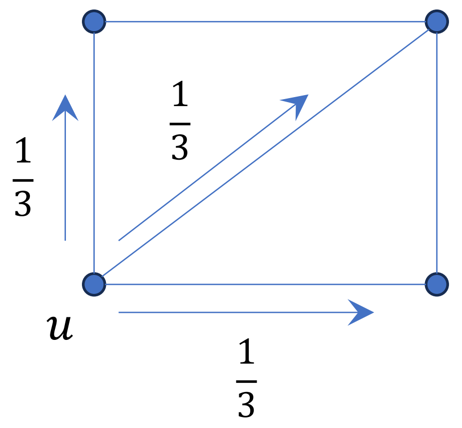
CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

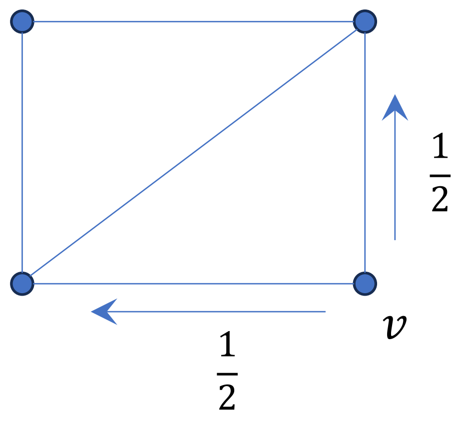
Lecture 7:

Expanders – random walks

Random walks on graphs

- Given an undirected unweighted connected graph G and a vertex u, the result of one step of a random walk from u is a uniformly random neighbor of u in G.





Random walks on graphs

- Given an undirected unweighted connected graph G and a vertex u, the result of one step of a random walk from u is a uniformly random neighbor of u in G.

- Random walk matrix: $\boldsymbol{W} = \boldsymbol{D}^{-1}\boldsymbol{A}$
  - $\boldsymbol{W}[u, v] = \begin{cases} \frac{1}{\deg(u)} & \{u, v\} \in E \\ 0 & \{u, v\} \notin E \end{cases}$

$\boldsymbol{x}$ is the current probability distribution over the vertices. → $\boldsymbol{W}^\top \boldsymbol{x}$ is the probability distribution resulting from one step of a random walk.

Stationary distribution

- A probability distribution $\boldsymbol{x}$ over the vertices is stationary if $\boldsymbol{W}^\top \boldsymbol{x} = \boldsymbol{x}$.

- Does a stationary distribution exist?

Stationary distribution

- A probability distribution $\boldsymbol{x}$ over the vertices is stationary if $\boldsymbol{W}^\top \boldsymbol{x} = \boldsymbol{x}$.

- Does a stationary distribution exist?

Yes: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$

Stationary distribution

- A probability distribution $\boldsymbol{x}$ over the vertices is stationary if $\boldsymbol{W}^\top \boldsymbol{x} = \boldsymbol{x}$.

- Does a stationary distribution exist?

Yes: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$

- Do random walks converge to a stationary distribution?

Stationary distribution

- A probability distribution $\boldsymbol{x}$ over the vertices is stationary if $\boldsymbol{W}^\top \boldsymbol{x} = \boldsymbol{x}$.

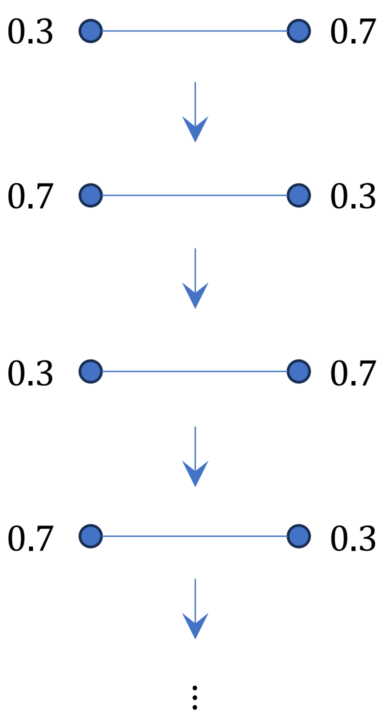
- Does a stationary distribution exist?

Yes: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$

- Do random walks converge to a stationary distribution?

No for bipartite graphs.

How to fix it?



Lazy random walks

- In one step of a **lazy** random walk:
  - With probability $\frac{1}{2}$, you stay at your current place.
  - With probability $\frac{1}{2}$, you perform a standard random walk step.

Lazy random walks

- In one step of a **lazy** random walk:
  - With probability $\frac{1}{2}$, you stay at your current place.
  - With probability $\frac{1}{2}$, you perform a standard random walk step.

- Lazy random walk matrix: $\widetilde{\boldsymbol{W}} = \frac{1}{2}(\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{A})$
  - $\widetilde{\boldsymbol{W}}[u,v] = \begin{cases} \frac{1}{2\deg(u)} & \{u,v\} \in E \\ \frac{1}{2} & u = v \\ 0 & \text{otherwise} \end{cases}$

Lazy random walks

- In one step of a **lazy** random walk:
  - With probability $\frac{1}{2}$, you stay at your current place.
  - With probability $\frac{1}{2}$, you perform a standard random walk step.

- Lazy random walk matrix: $\widetilde{\boldsymbol{W}} = \frac{1}{2}(\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{A})$
  - $\widetilde{\boldsymbol{W}}[u,v] = \begin{cases} \frac{1}{2\deg(u)} & \{u,v\} \in E \\ \frac{1}{2} & u = v \\ 0 & \text{otherwise} \end{cases}$

$\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$ is still a stationary distribution: $\widetilde{\boldsymbol{W}}^\top \boldsymbol{x} = \boldsymbol{x}$

What about convergence?

Regular graphs

- For simplicity, we restrict our discussion to d-regular graphs.

Lazy random walk matrix: $\widetilde{\boldsymbol{W}} = \frac{1}{2}(\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{A}) = \frac{1}{2}\boldsymbol{I} + \frac{1}{2d}\boldsymbol{A} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{N}$

$$\widetilde{\boldsymbol{W}}[u,v] = \begin{cases} \dfrac{1}{2\deg(u)} = \dfrac{1}{2d} & \{u,v\} \in E \\ \dfrac{1}{2} & u = v \\ 0 & \text{otherwise} \end{cases}$$

$\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$

Regular graphs

- For simplicity, we restrict our discussion to d-regular graphs.

Lazy random walk matrix: $\widetilde{\boldsymbol{W}} = \frac{1}{2}(\boldsymbol{I} + \boldsymbol{D}^{-1}\boldsymbol{A}) = \frac{1}{2}\boldsymbol{I} + \frac{1}{2d}\boldsymbol{A} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{N}$

$$\widetilde{\boldsymbol{W}}[u,v] = \begin{cases} \frac{1}{2\deg(u)} = \frac{1}{2d} & \{u,v\} \in E \\ \frac{1}{2} & u = v \\ 0 & \text{otherwise} \end{cases}$$

$\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$

$\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|} = \frac{1}{n}$ is a stationary distribution : $\widetilde{\boldsymbol{W}}^\top \boldsymbol{x} = \boldsymbol{x}$

Uniform distribution: $\boldsymbol{x} = \frac{1}{n}\mathbf{1}$

$\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{W}}^\top$

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^{n} x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\boldsymbol{1}$

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^n x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1}$

Eigenvalue range for a connected d-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.

$\lambda_2 > 0$ if and only if the graph is connected.

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^{n} x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1}$

Eigenvalue range for a <u>connected</u> d-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^n x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1}$

Eigenvalue range for a <u>connected</u> d-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \le \cdots \le \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$$

$$= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n$: orthonormal eigenvectors of $\boldsymbol{N}$

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^{n} x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1}$

Eigenvalue range for a <u>connected</u> d-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$
\begin{aligned}
&\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} \\
&= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i \\
&= \underbrace{\langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \boldsymbol{v}_1}_{1 - \frac{\lambda_1}{2} = 1} + \sum_{i=2}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i
\end{aligned}
$$

Convergence

Lazy random walk on a regular graph converges to the uniform distribution.

Claim: $\forall \boldsymbol{x} \in \mathbb{R}^n$ with $\sum_{i=1}^n x_i = 1$, we have: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1}$

Eigenvalue range for a connected d-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$$

$$= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$$= \langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \boldsymbol{v}_1 + \sum_{i=2}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$$= \frac{1}{n}\mathbf{1} + \sum_{i=2}^{n} \underbrace{\left(1 - \frac{\lambda_i}{2}\right)^t}_{\to 0 \text{ as } t \to \infty} \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$\sum_{i=1}^n x_i = 1$

Convergence rate

- How fast does a lazy random walk converge to the uniform distribution?

- A standard way to measure the distance between two distributions:

Total variation distance:

$$d_{\mathrm{TV}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \sum_{v} |x_v - y_v| = \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_1$$

Convergence rate

- How fast does a lazy random walk converge to the uniform distribution?

- A standard way to measure the distance between two distributions:

Total variation distance:

$$d_{\mathrm{TV}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\sum_{v} |x_v - y_v| = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_1$$

The **mixing time** $\tau_{\mathrm{mix}}(G, \epsilon)$ is the smallest integer t such that:

- Starting from any initial distribution $\boldsymbol{x}$ over the vertex set, after t steps of lazy random walk, the total variation distance to the uniform distribution is at most ϵ.

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After t steps of lazy random walk: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $d_{\text{TV}}\left(\frac{1}{n}\boldsymbol{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right) \leq \ \epsilon.$

Convergence rate

Claim: $\tau_{\text{mix}}(G, \epsilon) \in O\left(\frac{\log \frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After t steps of lazy random walk: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $d_{\text{TV}}\left(\frac{1}{n}\mathbf{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right) \leq\ \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\mathbf{1} - \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i\right\|_1$$

Recall: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\mathbf{1} + \sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After t steps of lazy random walk: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $d_{\text{TV}}\left(\frac{1}{n}\boldsymbol{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right) \le \epsilon.$

$$\underbrace{\frac{1}{2}\left\|\frac{1}{n}\boldsymbol{1} - \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1}_{} = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_1 \le \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_2$$

Relation between 1-norm and 2-norm for vectors in $\mathbb{R}^n$:
$\|\boldsymbol{v}\|_2 \le \|\boldsymbol{v}\|_1 \le \sqrt{n}\|\boldsymbol{v}\|_2$

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After t steps of lazy random walk: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $\underbrace{d_{\text{TV}}\left(\frac{1}{n}\mathbf{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right)} \leq\ \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\mathbf{1} - \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_1 \leq \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_2$$

$$= \frac{\sqrt{n}}{2} \cdot \sqrt{\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^{2t} \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle^2} \leq \frac{\sqrt{n}}{2} \cdot \left(1 - \frac{\lambda_2}{2}\right)^t \sqrt{\sum_{i=2}^{n}\langle \boldsymbol{x}, \boldsymbol{v}_i\rangle^2} \leq \frac{\sqrt{n}}{2} \cdot \left(1 - \frac{\lambda_2}{2}\right)^t \|\boldsymbol{x}\|_2$$

$\forall\ i \geq 2, \qquad 1 - \frac{\lambda_i}{2} \leq 1 - \frac{\lambda_2}{2}$

$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n}\langle \boldsymbol{x}, \boldsymbol{v}_i\rangle^2}$

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After t steps of lazy random walk: $\left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $\underbrace{d_{\text{TV}}\left(\frac{1}{n}\mathbf{1},\ \left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right)} \le\ \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\mathbf{1}-\left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{N}\right)^t\boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t\langle\boldsymbol{x},\boldsymbol{v}_i\rangle\boldsymbol{v}_i\right\|_1 \le \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t\langle\boldsymbol{x},\boldsymbol{v}_i\rangle\boldsymbol{v}_i\right\|_2$$

$$= \frac{\sqrt{n}}{2}\cdot\sqrt{\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^{2t}\langle\boldsymbol{x},\boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1-\frac{\lambda_2}{2}\right)^t\sqrt{\sum_{i=2}^{n}\langle\boldsymbol{x},\boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1-\frac{\lambda_2}{2}\right)^t\|\boldsymbol{x}\|_2 \le \frac{\sqrt{n}}{2}\cdot\left(1-\frac{\lambda_2}{2}\right)^t$$

$\|\boldsymbol{x}\|_2 \le \|\boldsymbol{x}\|_1 = 1$

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

From now on, we return to **general graphs**, where the claim still holds, with the uniform distribution replaced by the degree distribution: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$.

Convergence rate

Claim: $\tau_{\mathrm{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right) \subseteq O\left(\frac{\log\frac{n}{\epsilon}}{\Phi(G)^2}\right)$

From now on, we return to **general graphs**, where the claim still holds, with the uniform distribution replaced by the degree distribution: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$.

Cheeger inequality: $\frac{\lambda_2}{2} \leq \Phi(G) \leq \sqrt{2\lambda_2}$

Convergence rate

Claim: $\tau_{\text{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right) \subseteq O\left(\frac{\log\frac{n}{\epsilon}}{\Phi(G)^2}\right)$

From now on, we return to **general graphs**, where the claim still holds, with the uniform distribution replaced by the degree distribution: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$.

Exercise: $\tau_{\text{mix}}\left(G,\frac{1}{4}\right) \in \Omega\left(\frac{1}{\Phi(G)}\right)$

Proof sketch:

- Consider a cut with conductance $\Phi(G)$.
- Start a lazy random walk at a random vertex in the smaller side of the cut.
- In t steps, the amount of probability mass transferred to the other side is at most $t \cdot \Phi(G)$.

Cover time

- How long does it take for a random walk to visit all vertices?

Cover time

- How long does it take for a random walk to visit all vertices?
  - Set $\epsilon = \frac{1}{8|E|}$.
  - Doing a random walk for $\tau_{\text{mix}}(G, \epsilon)$ steps.
  - The probability that we are at vertex v is at least $\frac{\deg(v)}{2|E|} - 2\epsilon \geq \frac{1}{4|E|}$.

Cover time

- How long does it take for a random walk to visit all vertices?
  - Set $\epsilon = \frac{1}{8|E|}$.
  - Doing a random walk for $\tau_{\text{mix}}(G, \epsilon)$ steps.
  - The probability that we are at vertex v is at least $\frac{\deg(v)}{2|E|} - 2\epsilon \geq \frac{1}{4|E|}$.

Repeating this for $4|E| \cdot 2 \ln n$ times

$\downarrow$

Vertex v is visited with probability $\geq 1 - \left(1 - \frac{1}{4|E|}\right)^{4|E| \cdot 2 \ln n} \geq 1 - n^{-2}$

$\downarrow$

Every vertex is visited with probability $\geq 1 - n^{-1}$

Cover time

- How long does it take for a random walk to visit all vertices?
  - Set $\epsilon = \frac{1}{8|E|}$.
  - Doing a random walk for $\underbrace{\tau_{\text{mix}}(G,\epsilon)}_{\in O\left(\frac{\log n}{\Phi(G)^2}\right) \subseteq O(|E|^2 \log n)}$ steps.
  - The probability that we are at vertex v is at least $\frac{\deg(v)}{2|E|} - 2\epsilon \geq \frac{1}{4|E|}$.

Repeating this for $4|E| \cdot 2 \ln n$ times ← $O(|E|^3 \log^2 n)$ steps are enough.

↓

Vertex v is visited with probability $\geq 1 - \left(1 - \frac{1}{4|E|}\right)^{4|E| \cdot 2 \ln n} \geq 1 - n^{-2}$

↓

Every vertex is visited with probability $\geq 1 - n^{-1}$

Application 1: Space-efficient algorithm

- Consider this problem:
  - Input: An n-vertex m-edge graph $G = (V, E)$ and two vertices s and t.
  - Goal: Decide whether s and t are connected.

Application 1: Space-efficient algorithm

- Consider this problem:
  - Input: An n-vertex m-edge graph $G = (V, E)$ and two vertices s and t.
  - Goal: Decide whether s and t are connected.

- **Easy:** Solvable in linear time by a BFS from s.

Application 1: Space-efficient algorithm

- Consider this problem:
  - Input: An n-vertex m-edge graph $G = (V, E)$ and two vertices s and t.
  - Goal: Decide whether s and t are connected.

- **Easy:** Solvable in linear time by a BFS from s.

Maintaining a search tree requires $O(n)$ words of $O(\log n)$ bits.

- **Difficult:** Is it possible to solve the problem in polynomial time with a space complexity that is significantly smaller than $O(n)$?

Application 1: Space-efficient algorithm

- Consider this problem:
  - Input: An n-vertex m-edge graph $G = (V, E)$ and two vertices s and t.
  - Goal: Decide whether s and t are connected.

- **Easy:** Solvable in linear time by a BFS from s.

Maintaining a search tree requires $O(n)$ words of $O(\log n)$ bits.

- **Difficult:** Is it possible to solve the problem in polynomial time with a space complexity that is significantly smaller than $O(n)$?

Yes: Just do a random walk from s for $O(|E|^3 \log^2 n)$ steps and see if t is reached.
- Storing the state of a random walk costs only one word of $O(\log n)$ bits.

Application 2: Sampling

- **Easy:** Computing a spanning tree can be done in polynomial time.

- **Difficult:** Sampling a spanning tree nearly uniformly at random.

The total variation distance between the output distribution and the uniform distribution over all spanning trees is at most ϵ.

Application 2: Sampling

- **Easy:** Computing a spanning tree can be done in polynomial time.

- **Difficult:** Sampling a spanning tree <u>nearly uniformly at random</u>.

Given a spanning tree T, Consider the following **flip** operation:

- Choose a non-tree-edge e.
- Consider the unique cycle C in $T + e$.
- Choose a tree-edge $e' \in C$.
- Return the spanning tree $T' = T + e - e'$.

Application 2: Sampling

- **Easy:** Computing a spanning tree can be done in polynomial time.

- **Difficult:** Sampling a spanning tree <u>nearly uniformly at random</u>.

Given a spanning tree T, Consider the following **flip** operation:
- Choose a non-tree-edge e.
- Consider the unique cycle C in $T + e$.
- Choose a tree-edge $e' \in C$.
- Return the spanning tree $T' = T + e - e'$.

Algorithm: Do this with random choices of e and e' for a polynomial number of steps.

Application 2: Sampling

- **Easy:** Computing a spanning tree can be done in polynomial time.

- **Difficult:** Sampling a spanning tree <u>nearly uniformly at random</u>.

Given a spanning tree T, Consider the following **flip** operation:

- Choose a non-tree-edge e.
- Consider the unique cycle C in $T + e$.
- Choose a tree-edge $e' \in C$.
- Return the spanning tree $T' = T + e - e'$.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over the set of all spanning trees $\mathcal{V}$:

- $\{T, T'\} \in \mathcal{E}$ if T' can be reached from T by a flip operation.

Application 2: Sampling

- **Easy:** Computing a spanning tree can be done in polynomial time.
- **Difficult:** Sampling a spanning tree nearly uniformly at random.

Given a spanning tree T, Consider the following **flip** operation:

- Choose a non-tree-edge e.
- Consider the unique cycle C in $T + e$.
- Choose a tree-edge $e' \in C$.
- Return the spanning tree $T' = T + e - e'$.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over the set of all spanning trees $\mathcal{V}$:

- $\{T, T'\} \in \mathcal{E}$ if T' can be reached from T by a flip operation.

While the size of $\mathcal{G}$ is exponential in n, it is a very good expander, with its mixing time polynomial in n.

To sample a spanning tree nearly uniformly at random, it suffices to simulate a random walk in $\mathcal{G}$ for a polynomial number of steps, and this can be done in polynomial time.

https://arxiv.org/pdf/2004.07220

Outlook

- We have discussed several key aspects of expanders:
  - **Connectivity:** robustness against deletions.
  - **Linear algebra:** connections to eigenvalues.
  - **Probability:** rapid mixing of random walks.

  with many tools and applications

Outlook

- We have discussed several key aspects of expanders:
  - **Connectivity:** robustness against deletions.
  - **Linear algebra:** connections to eigenvalues.
  - **Probability:** rapid mixing of random walks.

  with many tools and applications

Conductance $\in \Omega\left(\frac{1}{\operatorname{polylog} n}\right)$

equivalent

Second eigenvalue of normalized Laplacian $\in \Omega\left(\frac{1}{\operatorname{polylog} n}\right)$

equivalent

Mixing time of lazy random walk $\in O(\operatorname{polylog} n)$

Outlook

- We have discussed several key aspects of expanders:
  - **Connectivity:** robustness against deletions.
  - **Linear algebra:** connections to eigenvalues.
  - **Probability:** rapid mixing of random walks.

- **Next:**
  - We will introduce a variant of expanders with a stronger connectivity guarantee.

The guarantee applies not only to all cuts $(S, V \setminus S)$ but also to all pairs of subsets (A, B).

References

- **Main reference:**
  - Lecture 6.1 of https://sites.google.com/site/thsaranurak/teaching/Expander

- **Additional/optional reading:**
  - Mohsen Ghaffari, Fabian Kuhn, and Hsin-Hao Su. 2017. “Distributed MST and Routing in Almost Mixing Time.” In Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC). Association for Computing Machinery, New York, NY, USA, 131–140. https://doi.org/10.1145/3087801.3087827

Using random walks to achieve efficient routing in expander networks.

CS5275 – The Algorithm Designer’s Toolkit (S2 AY2025/26)

Lecture 8:

Expanders – Ramanujan graphs

Quasirandomness

- **Observation:**
  - For n-vertex d-regular graphs, the following two statements are equivalent.

For any cut $(S, V \setminus S)$, we have:
$$|E(S, V \setminus S)| \in \Theta\left(\frac{d}{n} \cdot |S||V \setminus S|\right)$$

$G = (V, E)$ is an $\Omega(1)$-expander.

Quasirandomness

- **Observation:**
  - For n-vertex d-regular graphs, the following two statements are equivalent.

For any cut $(S, V \setminus S)$, we have:

$$|E(S, V \setminus S)| \in \Theta\left(\underbrace{\frac{d}{n} \cdot |S||V \setminus S|}\right)$$

$G = (V, E)$ is an $\Omega(1)$-expander.

This is the expected number of edges in $E(S, V \setminus S)$ if G is a random n-vertex d-regular graph.

This can be seen as a **quasirandomness** property.

Quasirandomness

- **Observation:**
  - For n-vertex d-regular graphs, the following two statements are equivalent.

For any cut $(S, V \setminus S)$, we have:
$$|E(S, V \setminus S)| \in \Theta\left(\frac{d}{n} \cdot |S||V \setminus S|\right)$$

$G = (V, E)$ is an $\Omega(1)$-expander.

This is the expected number of edges in $E(S, V \setminus S)$ if G is a random n-vertex d-regular graph.

This can be seen as a **quasirandomness** property.

Question: Can we extend this to $E(S, T)$ for any two subsets $S \subseteq V$ and $T \subseteq V$?

Notation

- For simplicity, we restrict ourselves to n-vertex d-regular graphs.

$e(A, B) = |(a, b) \in A \times B : \{a, b\} \in E|$

- This is the number of edges between A and B.
- If $\{u, v\} \in E$ satisfies $\{u, v\} \subseteq A \cap B$, then $\{u, v\}$ is counted twice.

$\sigma_2 = \max\{|\lambda_2|, |\lambda_3|, \ldots, |\lambda_n|\}$,

- where $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -d$ are the eigenvalues of the adjacency matrix $\boldsymbol{A}$.

Expander mixing lemma

- For simplicity, we restrict ourselves to n-vertex d-regular graphs.

$e(A,B) = |(a,b) \in A \times B : \{a,b\} \in E|$
- This is the number of edges between A and B.
- If $\{u,v\} \in E$ satisfies $\{u,v\} \subseteq A \cap B$, then $\{u,v\}$ is <u>counted twice</u>.

$\sigma_2 = \max\{|\lambda_2|, |\lambda_3|, \dots, |\lambda_n|\}$,
- where $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -d$ are the eigenvalues of the adjacency matrix $\boldsymbol{A}$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

The number of edges between S and T is close to the expected number of edges between them in a random d-regular graph.

Proof of the lemma

- $e(S,T) = \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T$
- $|S||T| = \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T$, where $\boldsymbol{J}$ is the all-1 $(n \times n)$ matrix.

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| = \left| \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T - \frac{d}{n} \cdot \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T \right| = \left| \mathbf{1}_S^\top \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right|$$

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Proof of the lemma

- $e(S,T) = \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T$
- $|S||T| = \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T$, where $\boldsymbol{J}$ is the all-1 $(n \times n)$ matrix.

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| = \left| \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T - \frac{d}{n} \cdot \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T \right| = \left| \mathbf{1}_S^\top \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right|$$

$$\leq \|\mathbf{1}_S\| \cdot \left\| \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right\|$$

Eigenvalues

Eigenvectors	$\boldsymbol{A}$	$\frac{d}{n}\boldsymbol{J}$	$\boldsymbol{A} - \frac{d}{n}\boldsymbol{J}$
$\boldsymbol{v}_1 = \frac{1}{\sqrt{n}} \mathbf{1}$	$\lambda_1 = d$	d	0
$\boldsymbol{v}_2$	λ_2	0	λ_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{v}_n$	λ_n	0	λ_n

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Proof of the lemma

- $e(S,T) = \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T$
- $|S||T| = \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T$, where $\boldsymbol{J}$ is the all-1 $(n \times n)$ matrix.

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| = \left| \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T - \frac{d}{n} \cdot \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T \right| = \left| \mathbf{1}_S^\top \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right|$$

$$\leq \|\mathbf{1}_S\| \cdot \left\| \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right\| \leq \|\mathbf{1}_S\| \cdot \sigma_2 \cdot \|\mathbf{1}_T\|$$

$$\left\| \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right\| = \left\| \sum_{i=2}^{n} \lambda_i \langle \mathbf{1}_T, \boldsymbol{v}_i \rangle \boldsymbol{v}_i \right\|$$

$$= \sqrt{\sum_{i=2}^{n} \lambda_i^2 \langle \mathbf{1}_T, \boldsymbol{v}_i \rangle^2} \leq \sigma_2 \cdot \sqrt{\sum_{i=2}^{n} \langle \mathbf{1}_T, \boldsymbol{v}_i \rangle^2} \leq \sigma_2 \cdot \|\mathbf{1}_T\|$$

$$\sigma_2 = \max\{|\lambda_2|, |\lambda_3|, \dots, |\lambda_n|\}$$

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

	Eigenvalues		
Eigenvectors	$\boldsymbol{A}$	$\frac{d}{n}\boldsymbol{J}$	$\boldsymbol{A} - \frac{d}{n}\boldsymbol{J}$
$\boldsymbol{v}_1 = \frac{1}{\sqrt{n}} \mathbf{1}$	$\lambda_1 = d$	d	0
$\boldsymbol{v}_2$	λ_2	0	λ_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{v}_n$	λ_n	0	λ_n

Proof of the lemma

- $e(S,T) = \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T$
- $|S||T| = \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T$, where $\boldsymbol{J}$ is the all-1 $(n \times n)$ matrix.

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| = \left| \mathbf{1}_S^\top \boldsymbol{A} \mathbf{1}_T - \frac{d}{n} \cdot \mathbf{1}_S^\top \boldsymbol{J} \mathbf{1}_T \right| = \left| \mathbf{1}_S^\top \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right|$$

$$\leq \|\mathbf{1}_S\| \cdot \left\| \left(\boldsymbol{A} - \frac{d}{n} \boldsymbol{J} \right) \mathbf{1}_T \right\| \leq \|\mathbf{1}_S\| \cdot \sigma_2 \cdot \|\mathbf{1}_T\| = \sigma_2 \cdot \sqrt{|S||T|}$$

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

Expander mixing lemma:

$$\left|e(S,T) - \frac{d}{n} \cdot |S||T|\right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

Alon–Boppana bound: For every d and $\epsilon > 0$, there exists n_0 such that all graphs with $\geq n_0$ vertices have $\sigma_2 > 2\sqrt{d-1} - \epsilon$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

Alon–Boppana bound: For every d and $\epsilon > 0$, there exists n_0 such that all graphs with $\geq n_0$ vertices have $\sigma_2 > 2\sqrt{d-1} - \epsilon$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Proof of a weaker bound:

$\sigma_2 \in \Omega\left(\sqrt{d}\right)$ when $d \leq 0.99 \cdot n$

$\mathrm{tr}(\boldsymbol{A}^2) = nd$

The trace of a matrix is the <u>sum of its diagonal entries</u> and equals the <u>sum of its eigenvalues</u>.

$$\mathrm{tr}(\boldsymbol{A}^2) = \sum_{i=1}^{n} \lambda_i^2 \leq d^2 + (n-1)\sigma_2^2$$

$$\sigma_2 \geq \sqrt{\frac{nd - d^2}{n-1}} \in \Omega\left(\sqrt{d}\right)$$

Ramanujan graphs

The name comes from the Ramanujan–Petersson conjecture, which was used in a construction of some of these graphs.

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

The graphs with $\sigma_2 \leq 2\sqrt{d-1}$ are called **Ramanujan graphs**.

Alon–Boppana bound: For every d and $\epsilon > 0$, there exists n_0 such that all graphs with $\geq n_0$ vertices have $\sigma_2 > 2\sqrt{d-1} - \epsilon$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs

The name comes from the Ramanujan–Petersson conjecture, which was used in a construction of some of these graphs.

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

The graphs with $\sigma_2 \leq 2\sqrt{d-1}$ are called **Ramanujan graphs**.

Alon–Boppana bound: For every d and $\epsilon > 0$, there exists n_0 such that all graphs with $\geq n_0$ vertices have $\sigma_2 > 2\sqrt{d-1} - \epsilon$.

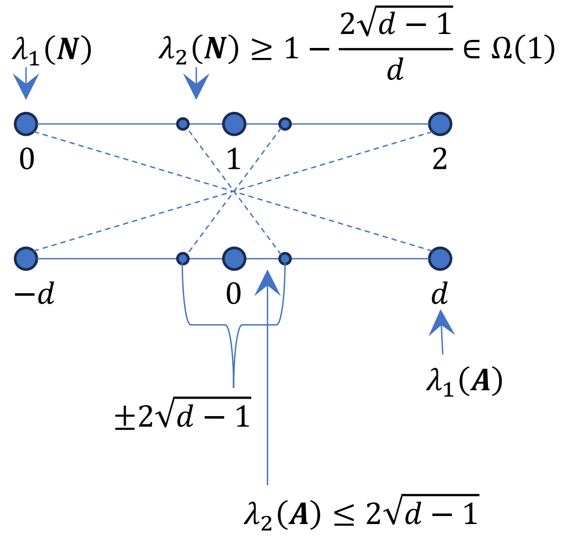
Ramanujan graphs are $\Omega(1)$-expanders.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Eigenvalues of $\boldsymbol{N}$:

Eigenvalues of $\boldsymbol{A}$:



$$\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A}$$

Ramanujan graphs

The name comes from the Ramanujan–Petersson conjecture, which was used in a construction of some of these graphs.

- How small σ_2 can be?
  - **Answer:** $2\sqrt{d-1}$

The graphs with $\sigma_2 \leq 2\sqrt{d-1}$ are called **Ramanujan graphs**.

Alon–Boppana bound: For every d and $\epsilon > 0$, there exists n_0 such that all graphs with $\geq n_0$ vertices have $\sigma_2 > 2\sqrt{d-1} - \epsilon$.

Ramanujan graphs are $\Omega(1)$-expanders.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Some $\Omega(1)$-expanders are not Ramanujan.

$\sigma_2 = d$ for any bipartite graph.

Random graphs are nearly Ramanujan

- For every d and $\epsilon > 0$, the probability that a random d-regular graph satisfies $\sigma_2 < 2\sqrt{d-1} + \epsilon$ tends to 1 as $n \to \infty$.

- Joel Friedman (Duke Mathematical Journal, 2003)

Algebraic constructions

- There is an infinite family of d-regular Ramanujan graphs, whenever $d - 1$ is a prime power.

  - Alexander Lubotzky, Ralph Phillips, and Peter Sarnak (Combinatorica, 1988)
  - Moshe Morgenstern (Journal of Combinatorial Theory, Series B, 1994).

Algebraic constructions

- There is an infinite family of d-regular Ramanujan graphs, whenever $d - 1$ is a prime power.

  - Alexander Lubotzky, Ralph Phillips, and Peter Sarnak (Combinatorica, 1988)
  - Moshe Morgenstern (Journal of Combinatorial Theory, Series B, 1994).

Open problem: Show this for all $d \geq 3$.

Algebraic constructions

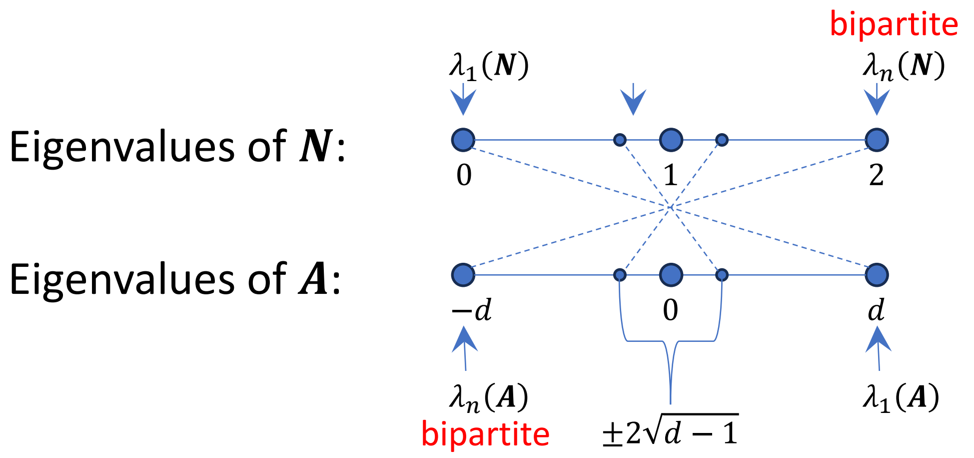
- There is an infinite family of d-regular Ramanujan graphs, whenever $d-1$ is a prime power.

> - Alexander Lubotzky, Ralph Phillips, and Peter Sarnak (Combinatorica, 1988)
> - Moshe Morgenstern (Journal of Combinatorial Theory, Series B, 1994).

Open problem: Show this for all $d \geq 3$.

Solved for <u>bipartite</u> Ramanujan graphs:

$$\max\{|\lambda_2|, |\lambda_3|, \ldots, |\lambda_{n-1}|\} \leq 2\sqrt{d-1}$$



> - Adam Marcus, Daniel Spielman and Nikhil Srivastava (Annals of Mathematics, 2015)

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S,T) > 0$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S,T) > 0$.

$$\sqrt{d|S||T|} \geq 2n$$

$$\downarrow$$

$$\frac{d}{n} \cdot |S||T| \geq 2\sqrt{d|S||T|} > \sigma_2 \cdot \sqrt{|S||T|}$$

$$\downarrow$$

$$e(S,T) \geq \frac{d}{n} \cdot |S||T| - \sigma_2 \cdot \sqrt{|S||T|} > 0$$

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $\underbrace{|S| \cdot |T| \cdot d \geq 4n^2}$, then $e(S,T) > 0$.

Exercise: The result is asymptotically the best possible.

- If you change $4n^2$ to ϵn^2 for some sufficiently small constant $\epsilon > 0$,
  - then this claim is false for every large enough d-regular graph.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S,T) > 0$.

Corollary 1: Any independent set S has size $|S| < \frac{2n}{\sqrt{d}}$.

If $|S| \geq \frac{2n}{\sqrt{d}}$, then $|S|^2 d \geq 4n^2$, so $e(S,S) > 0$.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S,T) > 0$.

Corollary 1: Any independent set S has size $|S| < \frac{2n}{\sqrt{d}}$.

Corollary 2: Chromatic number $> \frac{\sqrt{d}}{2}$.

If a proper coloring with $\leq \frac{\sqrt{d}}{2}$ colors exists, then $n < \frac{\sqrt{d}}{2} \cdot \frac{2n}{\sqrt{d}} = n$, which is impossible.

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Properties of Ramanujan graphs

- **Claim:** For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S,T) > 0$.

Corollary 1: Any independent set S has size $|S| < \frac{2n}{\sqrt{d}}$.

Corollary 2: Chromatic number $> \frac{\sqrt{d}}{2}$.

These results have been used to give lower bounds on the number of communication rounds needed to compute certain colorings and independent sets in a distributed network.

- Marijke H.L. Bodlaender, Magnús M. Halldórsson, Christian Konrad, and Fabian Kuhn. "Brief announcement: Local independent set approximation." *Proceedings of the ACM Symposium on Principles of Distributed Computing* (PODC 2016).
- Nathan Linial. "Locality in distributed graph algorithms." *SIAM Journal on computing* (1992).

Expander mixing lemma:

$$\left| e(S,T) - \frac{d}{n} \cdot |S||T| \right| \leq \sigma_2 \cdot \sqrt{|S||T|}$$

Ramanujan graphs: $\sigma_2 \leq 2\sqrt{d-1}$

Success probability amplification

- Consider a $\overbrace{\text{randomized algorithm}}^{\text{It uses } r \text{ random bits and takes } t \text{ time.}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

Success probability amplification

It uses r random bits and takes t time.

- Consider a randomized algorithm that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.
- **Standard solution:** Repeat for $x = \left\lceil \log \frac{1}{f} \right\rceil$ times.

Probability of no success is $\frac{1}{2^x} \leq f$.

Success probability amplification

It uses r random bits and takes t time.

- Consider a randomized algorithm that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.
- **Standard solution:** Repeat for $\underbrace{x = \left\lceil \log \frac{1}{f} \right\rceil}$ times.

This requires $rx = r\left\lceil \log \frac{1}{f} \right\rceil$ random bits.

Probability of no success is $\frac{1}{2^x} \leq f$.

Question: Is it possible to reduce the failure probability <u>without</u> using more than r random bits?

Success probability amplification

- Consider a $\overbrace{\text{randomized algorithm}}^{\text{It uses } r \text{ random bits and takes } t \text{ time.}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

- Set $d = \left\lceil \frac{8}{f} \right\rceil$.
- Take a d-regular Ramanujan graph with $n = 2^r$ vertices.
  - Each vertex corresponds to an r-bit string.

Success probability amplification

It uses r random bits and takes t time.

- Consider a randomized algorithm that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

- Set $d = \left\lceil \frac{8}{f} \right\rceil$.
- Take a d-regular Ramanujan graph with $n = 2^r$ vertices.
  - Each vertex corresponds to an r-bit string.
  - S = the set of vertices that lead to a successful execution of the algorithm.
    - $|S| = \frac{n}{2}$.
  - T = the set of vertices that do not have a neighbor in S.
    - $|S| \cdot |T| \cdot d < 4n^2 \rightarrow |T| < fn$.

Recall: For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $e(S, T) > 0$.

Success probability amplification

- Consider a $\underbrace{\text{randomized algorithm}}^{\text{It uses } r \text{ random bits and takes } t \text{ time.}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

- Set $d = \left\lceil \frac{8}{f} \right\rceil$.
- Take a d-regular Ramanujan graph with $n = 2^r$ vertices.
  - Each vertex corresponds to an r-bit string.
  - S = the set of vertices that lead to a successful execution of the algorithm.
    - $|S| = \frac{n}{2}$.
  - T = the set of vertices that do not have a neighbor in S.
    - $|S| \cdot |T| \cdot d < 4n^2 \rightarrow |T| < fn$.
- Sample a vertex v uniformly at random, for each neighbor u of v, run the algorithm using the r-bit string of u.
  - Successful $\leftrightarrow v \notin T$
  - This happens with probability at least $1 - f$.

Success probability amplification

- Consider a $\underbrace{\text{randomized algorithm}}$ that succeeds with probability $\frac{1}{2}$.

  It uses r random bits and takes t time.

- **Goal:** Amplify the success probability to $1 - f$.

- Set $d = \left\lceil \frac{8}{f} \right\rceil$.
- Take a d-regular Ramanujan graph with $n = 2^r$ vertices.
  - Each vertex corresponds to an r-bit string.
  - $S =$ the set of vertices that lead to a successful execution of the algorithm.
    - $|S| = \frac{n}{2}$.
  - $T =$ the set of vertices that do not have a neighbor in S.
    - $|S| \cdot |T| \cdot d < 4n^2 \rightarrow |T| < fn$.
- Sample a vertex v uniformly at random, for each neighbor u of v, run the algorithm using the r-bit string of u.
  - Successful $\leftrightarrow v \notin T$
  - This happens with probability at least $1 - f$.

	Time	Random bits
Standard	$O\left(t \log \frac{1}{f}\right)$	$O\left(r \log \frac{1}{f}\right)$
Ramanujan	$O\left(\frac{t}{f}\right)$	r

Here we omit the cost for simulating a Ramanujan graph.

Success probability amplification

It uses r random bits and takes t time.

- Consider a $\underbrace{\text{randomized algorithm}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

- Set $d = \left\lceil \frac{8}{f} \right\rceil$.
- Take a d-regular Ramanujan graph with $n = 2^r$ vertices.
  - Each vertex corresponds to an r-bit string.
  - $S =$ the set of vertices that lead to a successful execution of the algorithm.
    - $|S| = \frac{n}{2}$.
  - $T =$ the set of vertices that do not have a neighbor in S.
    - $|S| \cdot |T| \cdot d < 4n^2 \rightarrow |T| < fn$.
- Sample a vertex v uniformly at random, for each neighbor u of v, run the algorithm using the r-bit string of u.
  - Successful $\leftrightarrow v \notin T$
  - This happens with probability at least $1 - f$.

	Time	Random bits
Standard	$O\left(t \log \frac{1}{f}\right)$	$O\left(r \log \frac{1}{f}\right)$
Ramanujan	$O\left(\frac{t}{f}\right)$	r
Next	$O\left(t \log \frac{1}{f}\right)$	$O\left(r + \log \frac{1}{f}\right)$

A Chernoff Bound for random walks

- Let $G = (V, E)$ be an n-vertex d-regular graph.
- Let $(v_1, v_2, \dots, v_\ell)$ be an $(\ell - 1)$-step random walk starting from a uniformly random vertex.
- Let $f : V \to [0, 1]$ be a function.
- Let $\mu = \frac{1}{n} \sum_{v \in V} f(v)$.
- Let $\epsilon > 0$.

$$\Pr\left[\frac{1}{\ell}\sum_{i=1}^{\ell} f(v_i) \geq \mu + \epsilon + \frac{\sigma_2}{d}\right] \in e^{-\Omega(\epsilon^2 \ell)}$$

The proof is omitted.

A Chernoff Bound for random walks

It uses r random bits and takes t time.

- Consider a $\underbrace{\text{randomized algorithm}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

	Time	Random bits
Standard	$O\left(t \log \frac{1}{f}\right)$	$O\left(r \log \frac{1}{f}\right)$
Ramanujan	$O\left(\frac{t}{f}\right)$	r
Next	$O\left(t \log \frac{1}{f}\right)$	$O\left(r + \log \frac{1}{f}\right)$

- Let $G = (V, E)$ be an n-vertex d-regular graph.
- Let $(v_1, v_2, \dots, v_\ell)$ be an $(\ell - 1)$-step random walk starting from a uniformly random vertex.
- Let $f : V \to [0, 1]$ be a function.
- Let $\mu = \frac{1}{n} \sum_{v \in V} f(v)$.
- Let $\epsilon > 0$.

$$\Pr\left[\frac{1}{\ell} \sum_{i=1}^{\ell} f(v_i) \geq \mu + \epsilon + \frac{\sigma_2}{d}\right] \in e^{-\Omega(\epsilon^2 \ell)}$$

New approach:

- Take $d \in O(1)$ and $n = 2^r$.
- Construct a random walk $(v_1, v_2, \dots, v_\ell)$ with $\ell \in O\left(\log \frac{1}{f}\right)$.
- Run the algorithm with these bit strings.

A Chernoff Bound for random walks

- Consider a $\overbrace{\text{randomized algorithm}}^{\text{It uses } r \text{ random bits and takes } t \text{ time.}}$ that succeeds with probability $\frac{1}{2}$.
- **Goal:** Amplify the success probability to $1 - f$.

$n = 2^r$ $\quad$ $d \in O(1)$ is large enough so that $\frac{\sigma_2}{d} \leq \frac{1}{8}$.

- Let $G = (V, E)$ be an n-vertex d-regular graph.
- Let $(v_1, v_2, \ldots, v_\ell)$ be an $(\ell - 1)$-step random walk starting from a uniformly random vertex.
- Let $f : V \to [0, 1]$ be a function.
  - 0 : the algorithm succeeds.
  - 1 : the algorithm fails.
- Let $\mu = \frac{1}{n}\sum_{v \in V} f(v)$.
- Let $\epsilon > 0$.

$$\Pr\left[\frac{1}{\ell}\sum_{i=1}^{\ell} f(v_i) \geq \underbrace{\overset{\frac{1}{2}}{\mu} + \overset{\frac{1}{8}}{\epsilon} + \overset{\leq \frac{1}{8}}{\frac{\sigma_2}{d}}}_{\leq \frac{3}{4}}\right] \in e^{-\Omega(\epsilon^2 \ell)}$$

$\ell \in O\left(\log \frac{1}{f}\right)$ is large enough so that the error probability $e^{-\Omega(\epsilon^2 \ell)}$ is at most f.

$\frac{1}{\ell}\sum_{i=1}^{\ell} f(v_i) < 1 \to$ some of $v_1, v_2, \ldots, v_\ell$ make the algorithm successful.

Further applications

- Ramanujan graphs can be used to construct error correcting codes.

  Michael Sipser and Daniel A. Spielman.
  “Expander codes.”
  IEEE transactions on Information Theory, 2002.

- Ramanujan graphs can be used to construct cryptographic hash functions.

  Denis X. Charles, Eyal Z. Goren, and Kristin E. Lauter.
  “Cryptographic hash functions from expander graphs.”
  Journal of Cryptology, 2009.

Outlook

- **Next:**
  - An interesting application of Ramanujan graphs for $\underbrace{\text{graph algorithm design}}_{\text{for general graphs}}$.

References

- **Main reference:**
  - Lecture 5.1 of https://sites.google.com/site/thsaranurak/teaching/Expander
  - Chapter 21 of https://lucatrevisan.github.io/books/expanders-2016.pdf

- **Additional/optional reading:**
  - https://en.wikipedia.org/wiki/Expander_mixing_lemma
  - https://en.wikipedia.org/wiki/Ramanujan_graph
  - Shlomo Hoory, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications." *Bulletin of the American Mathematical Society* 43.4 (2006): 439-561.

CS5275 – The Algorithm Designer's Toolkit (S2 AY2025/26)

Lecture 9:

Expanders – vertex connectivity

Vertex connectivity

A **vertex cut** is a partition of the vertex set V into three parts:

$$V = L \cup S \cup R$$

such that $E(L, R) = \emptyset$.

The **size** of a vertex cut (L, S, R) is $|S|$.

$L \neq \emptyset, \qquad R \neq \emptyset$

Vertex connectivity

A **vertex cut** is a partition of the vertex set V into three parts:

$$V = L \cup S \cup R$$

such that $E(L, R) = \emptyset$.

The **size** of a vertex cut (L, S, R) is $|S|$.

$L \neq \emptyset, \qquad R \neq \emptyset$

The **vertex connectivity** $\kappa(G)$ of a graph is the minimum size of a vertex cut.

$n - 1$ if G is a clique.

Vertex connectivity

A set $S \subseteq V \setminus \{s, t\}$ is an $(\boldsymbol{s}, \boldsymbol{t})$**-vertex cut** if the removal of S disconnects s and t.

Equivalently, there is a vertex cut (L, S, R) with $s \in L$ and $t \in R$

The $(\boldsymbol{s}, \boldsymbol{t})$**-vertex connectivity** $\kappa_G(s, t)$ is the minimum size of an (s, t)-vertex cut.

Observation: $\kappa(G) = \min_{s \neq t} \kappa_G(s, t)$.

Computing a minimum (s, t)-vertex cut

- **Observation:** A minimum (s, t)-vertex cut can be computed in one max-flow call.

Li Chen, Rasmus Kyng, Yang Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva
"Maximum Flow and Minimum-Cost Flow in Almost-Linear Time"
Journal of the ACM, 2025

Although the algorithm is randomized, subsequent work makes it deterministic.

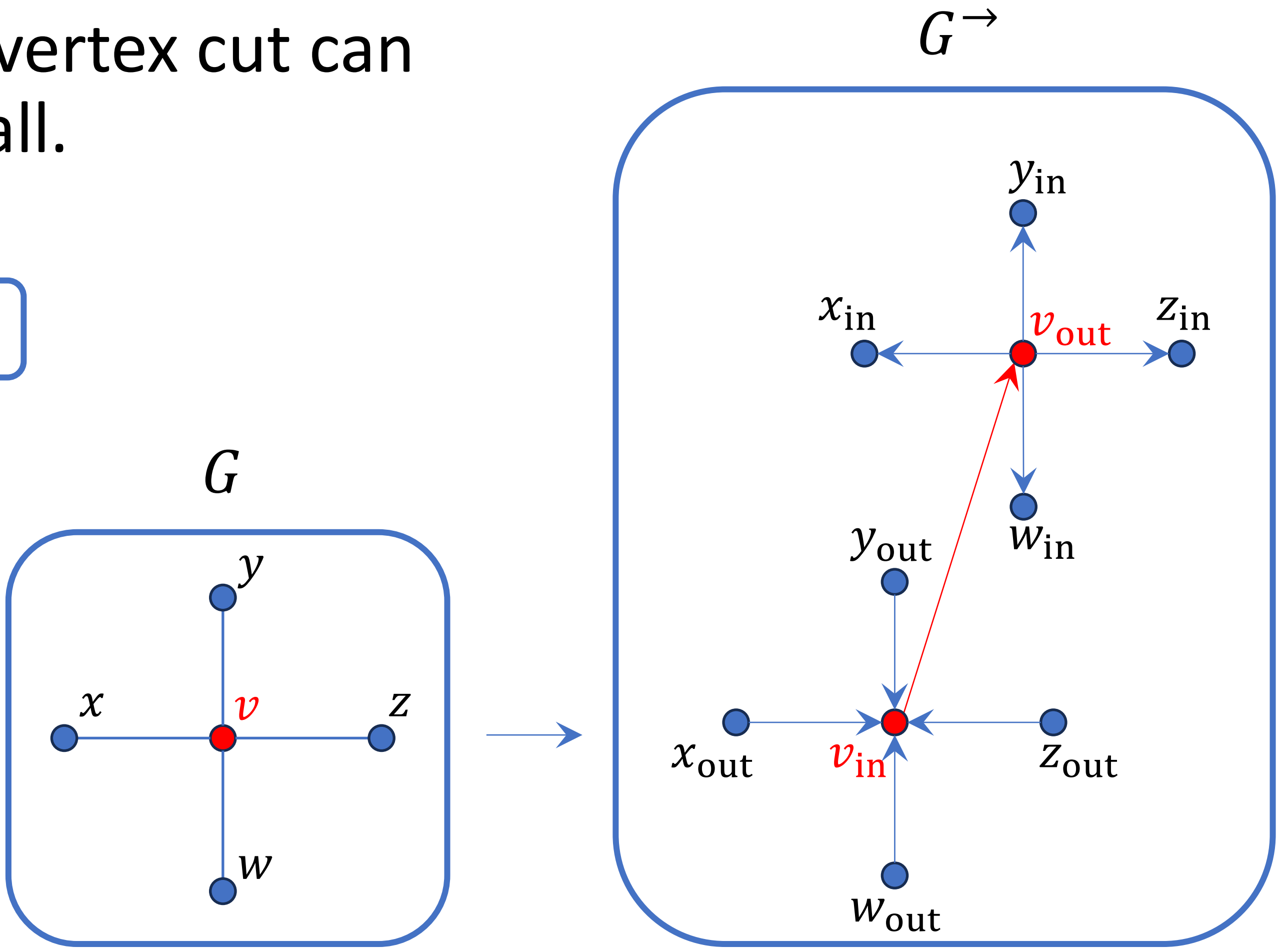
Computing a minimum (s, t)-vertex cut

- **Observation:** A minimum (s, t)-vertex cut can be computed in one max-flow call.

Maximum number of vertex-disjoint (s, t)-paths in G

$\|$

Maximum $(s_{\text{out}}, t_{\text{in}})$-flow in $G^{\rightarrow}$



Computing a minimum (s, t)-vertex cut

- **Observation:** A minimum (s, t)-vertex cut can be computed in one max-flow call.

Menger's theorem

Maximum number of vertex-disjoint (s, t)-paths in G = Minimum (s, t)-vertex cut in G

Maximum number of vertex-disjoint (s, t)-paths in G = Maximum $(s_{\text{out}}, t_{\text{in}})$-flow in $G^{\rightarrow}$

Minimum (s, t)-vertex cut in G = Minimum (s, t)-cut in $G^{\rightarrow}$

Maximum $(s_{\text{out}}, t_{\text{in}})$-flow in $G^{\rightarrow}$ = Minimum (s, t)-cut in $G^{\rightarrow}$

Max-flow min-cut theorem

Computing a minimum vertex cut

- **Slow:** Computing a $\underbrace{\text{minimum } (s,t)\text{-vertex for every pair } (s,t)}_{\text{This requires } O(n^2) \text{ max-flow calls.}}$.

Computing a minimum vertex cut

- **Slow:** Computing a $\underbrace{\text{minimum } (s,t)\text{-vertex for every pair } (s,t).}_{\text{This requires } O(n^2) \text{ max-flow calls.}}$

- **Key observation:**
  - The problem becomes easier if a minimum vertex cut is β-balanced.

A vertex cut (L, S, R) is ***β*-balanced** if $|L| \geq \beta$ and $|R| \in \Omega(n)$.

Computing a minimum vertex cut

- **Slow:** Computing a minimum (s,t)-vertex for every pair (s,t).

This requires $O(n^2)$ max-flow calls.

- **Key observation:**
  - The problem becomes easier if a minimum vertex cut is β-balanced.

A vertex cut (L,S,R) is $\boldsymbol{\beta}$**-balanced** if $|L| \geq \beta$ and $|R| \in \Omega(n)$.

For $\Omega\left(\frac{\beta}{n}\right)$ fraction of the pairs (s,t), we have: $\kappa(G) = \kappa_G(s,t)$.

By computing a minimum (s,t)-vertex cut for $O\left(\frac{n}{\beta}\right)$ random choices of (s,t) and selecting the smallest among them, we obtain a minimum vertex cut with probability at least 0.99.

Can we make it deterministic?

Derandomization via Ramanujan graphs

- We fix a β-balanced minimum vertex cut $\underbrace{(L, S, R)}_{\text{Unknown to the algorithm}}$ of $G = (V, E)$:
  - $|L| \geq \beta$
  - $|R| \in \Omega(n)$
- **Goal:** Select $s \in L$ and $t \in R$.

Derandomization via Ramanujan graphs

- We fix a β-balanced minimum vertex cut $\underbrace{(L, S, R)}_{\text{Unknown to the algorithm}}$ of $G = (V, E)$:
  - $|L| \geq \beta$
  - $|R| \in \Omega(n)$
- **Goal:** Select $s \in L$ and $t \in R$.

$\boxed{\text{Consider a } d\text{-regular Ramanujan graph } H \text{ over the same vertex set } V.}$

Recall: For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $E_H(S, T) \neq \emptyset$.

$\downarrow$

There is a choice $d \in O\left(\frac{n}{\beta}\right)$ such that:

- If $|L| \geq \beta$ and $|R| \in \Omega(n)$, then $E_H(L, R) \neq \emptyset$.

Derandomization via Ramanujan graphs

- We fix a β-balanced minimum vertex cut $\underbrace{(L, S, R)}_{\text{Unknown to the algorithm}}$ of $G = (V, E)$:
  - $|L| \geq \beta$
  - $|R| \in \Omega(n)$
- **Goal:** Select $s \in L$ and $t \in R$.

Consider a d-regular Ramanujan graph H over the same vertex set V.

Recall: For any $S, T \subseteq V$, if $|S| \cdot |T| \cdot d \geq 4n^2$, then $E_H(S, T) \neq \emptyset$.

There is a choice $d \in O\left(\frac{n}{\beta}\right)$ such that:

- If $|L| \geq \beta$ and $|R| \in \Omega(n)$, then $E_H(L, R) \neq \emptyset$.

It suffices to go over all edges $\{s, t\}$ in H.

This requires $nd \in O\left(\frac{n^2}{\beta}\right)$ max-flow calls.

The balanced case

- **Summary:** If a minimum vertex cut is β-balanced, then a minimum vertex cut can be computed using $\underbrace{O\left(\frac{n^2}{\beta}\right) \text{ max-flow calls.}}_{\text{Improvement over the naïve } O(n^2) \text{ bound.}}$

The imbalanced case

- **Summary:** If a minimum vertex cut is β-balanced, then a minimum vertex cut can be computed using $\underbrace{O\left(\frac{n^2}{\beta}\right) \text{ max-flow calls.}}$

Improvement over the naïve $O(n^2)$ bound.

- **Next:** Extend the approach to the imbalanced case.

The plan: gradually modify the graph to make it balanced.

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

The $\delta - \kappa$ gap

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

We always have: $\delta(G) \geq \kappa(G)$

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

The $\delta - \kappa$ gap

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

By symmetry, assume $|L^*| \leq |R^*|$.

$$|R^*| \geq \frac{n - |S^*|}{2} = \frac{n - \kappa(G)}{2} \geq \frac{n - \delta(G)}{2} \geq \frac{0.01 \cdot n}{2}$$

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

The $\delta - \kappa$ gap

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

By symmetry, assume $|L^*| \leq |R^*|$.

$$|R^*| \geq \frac{n - |S^*|}{2} = \frac{n - \kappa(G)}{2} \geq \frac{n - \delta(G)}{2} \geq \frac{0.01 \cdot n}{2}$$

Consider any $v \in L^*$.

$$\delta(G) \leq \deg(v) < |L^*| + |S^*| = |L^*| + \kappa(G) \quad \rightarrow \quad |L^*| > \delta(G) - \kappa(G) = \beta$$

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

Gap amplification

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

The plan:
We will show that $O(n)$ max-flow calls suffice to achieve one of the following.
- A minimum vertex cut.
- Increasing the $\delta - \kappa$ gap by 1.

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

Gap amplification

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

The plan:
We will show that $O(n)$ max-flow calls suffice to achieve <u>one of the following</u>.

- A minimum vertex cut.
- Increasing the $\delta - \kappa$ gap by 1.

After repeating this for β times, we are in the β-balanced case.

This takes $O(\beta n)$ max-flow calls.

A minimum vertex cut can be computed using $O\left(\frac{n^2}{\beta}\right)$ max-flow calls.

The algorithm works under the condition: $\delta(G) \leq 0.99 \cdot n$

Gap amplification

- We write $\delta(G)$ to denote the minimum degree of G.

Claim: Every minimum vertex cut (L^*, S^*, R^*) of $G = (V, E)$ is β-balanced for $\beta = \delta(G) - \kappa(G)$.

The plan:

We will show that $O(n)$ max-flow calls suffice to achieve <u>one of the following</u>.

- A minimum vertex cut.
- Increasing the $\delta - \kappa$ gap by 1.

Total cost: $O(n^{1.5})$ max-flow calls.

$\beta = \sqrt{n}$

After repeating this for β times, we are in the β-balanced case.

This takes $O(\beta n)$ max-flow calls.

A minimum vertex cut can be computed using $O\left(\frac{n^2}{\beta}\right)$ max-flow calls.

Single-source vertex connectivity

- **Define:** $\underbrace{\kappa_G(s) = \min_{t \neq s} \kappa_G(s,t).}_{\text{Can be computed using } n-1 \text{ max-flow calls.}}$

Single-source vertex connectivity

- **Define:** $\underbrace{\kappa_G(s) = \min_{t \neq s} \kappa_G(s,t).}_{\text{Can be computed using } n-1 \text{ max-flow calls.}}$

Claim: If $\kappa_G(s) > \kappa(G)$, then $s \in S^*$ for every minimum vertex cut (L^*, S^*, R^*) of G.

If $s \notin S^*$, then $|S^*| \geq \kappa_G(s) > \kappa(G)$, so (L^*, S^*, R^*) is not a minimum cut.

Algorithm

- Select a vertex s.
- Compute a minimum (s, t)-vertex cut for every $t \neq s$.
- **Case 1:** $\kappa_G(s) = \kappa(G)$.
  - We have already obtained a minimum vertex cut.
- **Case 2:** $\kappa_G(s) > \kappa(G)$.

Algorithm

- Select a vertex s.
- Compute a minimum (s, t)-vertex cut for every $t \neq s$.
- **Case 1:** $\kappa_G(s) = \kappa(G)$.
  - We have already obtained a minimum vertex cut.
- **Case 2:** $\kappa_G(s) > \kappa(G)$.
  - $s \in S^*$ for every minimum vertex cut (L^*, S^*, R^*) of G.
  - Consider $G' = G - s$.

(L^*, S^*, R^*) is a minimum cut of G $\longleftrightarrow$ $(L^*, S^* \setminus \{s\}, R^*)$ is a minimum cut of G'

This reduces the problem of computing a minimum cut in G to the same problem in G'.

Algorithm

Clarification: Since the algorithm cannot determine whether it is in Case 1 or Case 2, it records the cuts obtained in both cases. The smaller of the two is guaranteed to be a minimum cut.

- Select a vertex s.
- Compute a minimum (s, t)-vertex cut for every $t \neq s$.
- **Case 1:** $\kappa_G(s) = \kappa(G)$.
  - We have already obtained a minimum vertex cut.
- **Case 2:** $\kappa_G(s) > \kappa(G)$.
  - $s \in S^*$ for every minimum vertex cut (L^*, S^*, R^*) of G.
  - Consider $G' = G - s$.

(L^*, S^*, R^*) is a minimum cut of G $\longleftrightarrow$ $(L^*, S^* \setminus \{s\}, R^*)$ is a minimum cut of G'

This reduces the problem of computing a minimum cut in G to the same problem in G'.

Algorithm

Recall: We want to increase the $\delta - \kappa$ gap by 1.

- Since $\kappa(G') = \kappa(G) - 1$, we achieve the goal if $\delta(G') = \delta(G)$.
- However, it is possible that $\underbrace{\delta(G') = \delta(G) - 1 \text{ due to the removal of vertex } s}_{\text{How to fix it?}}$.

- Select a vertex s.
- Compute a minimum (s, t)-vertex cut for every $t \neq s$.
- **Case 1:** $\kappa_G(s) = \kappa(G)$.
  - We have already obtained a minimum vertex cut.
- **Case 2:** $\kappa_G(s) > \kappa(G)$.
  - $s \in S^*$ for every minimum vertex cut (L^*, S^*, R^*) of G. $\longrightarrow$ $\kappa(G') = \kappa(G) - 1$
  - Consider $G' = G - s$.

$\boxed{(L^*, S^*, R^*) \text{ is a minimum cut of } G} \longleftrightarrow \boxed{(L^*, S^* \setminus \{s\}, R^*) \text{ is a minimum cut of } G'}$

This reduces the problem of computing a minimum cut in G to the same problem in G'.

Degree restoration

- **Setting:**
  - $G = (V, E)$.
  - $G' = G - s$.
  - $F = \{v \in V : \{v, s\} \in E,\ \deg_G(v) = \delta(G)\}$

    - This is the set of vertices whose degrees drop to $\delta(G) - 1$ after the removal of s.
    - If we can restore their degrees to $\delta(G)$, then the goal is achieved.
    - The challenge is to do so in a way that does not affect the minimum vertex cut.

Degree restoration

How to see the existence of u?

- **Reason 1:** The assumption that the minimum degree of the input graph is at most $0.99 \cdot n$.
- **Reason 2:** If u does not exist, then G' is already a clique.

- **Setting:**
  - $G = (V, E)$.
  - $G' = G - s$.
  - $F = \{v \in V : \{v, s\} \in E,\ \deg_G(v) = \delta(G)\}$

- For each $v \in F$, select a vertex $u \in V$ such that $\{u, v\} \notin E$.
- Compute a minimum (u, v)-vertex cut in G'.
- **Case 1:** $\kappa_{G'}(u, v) = \kappa(G')$.
  - We have already obtained a minimum vertex cut in G'.
- **Case 2:** $\kappa_{G'}(u, v) > \kappa(G')$.

Degree restoration

$\kappa_{G'}(u,v) > \kappa(G')$

For every minimum cut (L^*, S^*, R^*) of G', we have:

- $\{u,v\} \subseteq L^* \cup S^*$ or $\{u,v\} \subseteq R^* \cup S^*$.

- **Setting:**
  - $G = (V,E)$.
  - $G' = G - s$.
  - $F = \{v \in V : \{v,s\} \in E,\ \deg_G(v) = \delta(G)\}$

- For each $v \in F$, select a vertex $u \in V$ such that $\{u,v\} \notin E$.
- Compute a minimum (u,v)-vertex cut in G'.
- **Case 1:** $\kappa_{G'}(u,v) = \kappa(G')$.
  - We have already obtained a minimum vertex cut in G'.
- **Case 2:** $\kappa_{G'}(u,v) > \kappa(G')$.
  - We have $\kappa(G') = \kappa(G' + \{u,v\})$.
  - Therefore, we may set: $G' \leftarrow G + \{u,v\}$.

This restore v's degree to $\delta(G)$

Degree restoration

- **Setting:**
  - $G = (V, E)$.
  - $G' = G - s$.
  - $F = \{v \in V : \{v, s\} \in E,\ \deg_G(v) = \delta(G)\}$

Clarification: Since the algorithm cannot determine whether it is in Case 1 or Case 2, it records the cuts obtained in both cases. The smaller of the two is guaranteed to be a minimum cut.

- For each $v \in F$, select a vertex $u \in V$ such that $\{u, v\} \notin E$.
- Compute a minimum (u, v)-vertex cut in G'.
- **Case 1:** $\kappa_{G'}(u, v) = \kappa(G')$.
  - We have already obtained a minimum vertex cut in G'.
- **Case 2:** $\kappa_{G'}(u, v) > \kappa(G')$.
  - We have $\kappa(G') = \kappa(G' + \{u, v\})$.
  - Therefore, we may set: $G' \leftarrow G + \{u, v\}$.

This restore v's degree to $\delta(G)$

References

- **Main reference:**
  - Lecture 5.2 of https://sites.google.com/site/thsaranurak/teaching/Expander

- **Additional/optional reading:**
  - Harold N. Gabow. “Using expander graphs to find vertex connectivity.” Journal of the ACM (2006) https://doi.org/10.1145/1183907.1183912

The algorithm presented in this lecture is from this paper.