# 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 <u>one step of a random walk</u> 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 <u>one step of a random walk</u> 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 $\boldsymbol{W}^\top \boldsymbol{x} = \boldsymbol{x}$.

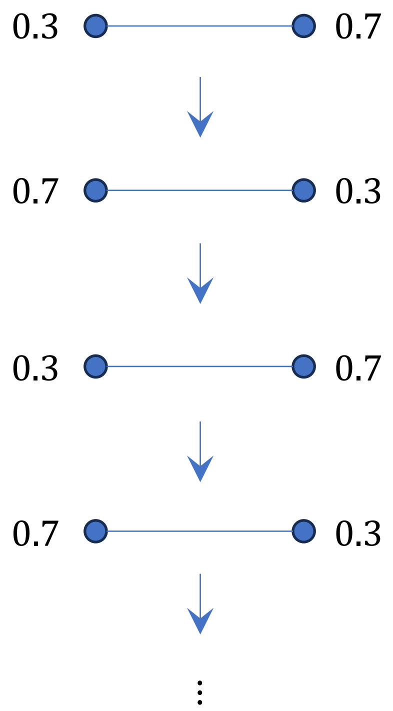
- Does a stationary distribution exist?

**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: $\lim_{t \to \infty} \left( \boldsymbol{I} - \frac{1}{2} \boldsymbol{N} \right)^t \boldsymbol{x} = \frac{1}{n} \mathbf{1}$

# 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: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\boldsymbol{1}$

Eigenvalue range for a <u>connected</u> $d$-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.

- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ 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: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\boldsymbol{1}$

Eigenvalue range for a connected $d$-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$$
$$= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ : 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 $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$$

$$= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

$$= \underbrace{\langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \boldsymbol{v}_1}_{1 - \frac{\lambda_1}{2} = 1} + \sum_{i=2}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i$$

# 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: $\lim_{t\to\infty} \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} = \frac{1}{n}\boldsymbol{1}$

Eigenvalue range for a connected $d$-regular graphs:

- $\boldsymbol{N} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{A} : [0, 2]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ is an eigenvector with eigenvalue $\lambda_1 = 0$.
  - $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$.
- $\boldsymbol{I} - \frac{1}{2}\boldsymbol{N} : [0, 1]$
  - $\boldsymbol{v}_1 = \frac{1}{\sqrt{n}}\boldsymbol{1}$ is an eigenvector with eigenvalue $1 - \frac{\lambda_1}{2} = 1$.
  - All other eigenvalues are within the range $\left[0, 1 - \frac{\lambda_2}{2}\right]$.

$$
\begin{aligned}
&\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x} \\
&= \sum_{i=1}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i \\
&= \langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \boldsymbol{v}_1 + \sum_{i=2}^{n} \left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i \\
&= \frac{1}{n}\boldsymbol{1} + \sum_{i=2}^{n} \underbrace{\left(1 - \frac{\lambda_i}{2}\right)^t}_{\to 0 \text{ as } t \to \infty} \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle \boldsymbol{v}_i
\end{aligned}
$$

$\sum_{i=1}^n x_i = 1$

# 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** $\tau_{\text{mix}}(G, \epsilon)$ 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 $\epsilon$.

# 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:** $d_{\text{TV}}\left(\frac{1}{n}\mathbf{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right) \leq \ \epsilon.$

# 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:** $\tau_{\mathrm{mix}}(G,\epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right)$

- Initial distribution: $\boldsymbol{x}$
- After $t$ steps of lazy random walk: $\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}$
- **Goal:** $\underbrace{d_{\mathrm{TV}}\left(\frac{1}{n}\mathbf{1},\ \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right)}_{} \leq \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\mathbf{1} - \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_1 \leq \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_2$$

Relation between 1-norm and 2-norm for vectors in $\mathbb{R}^n$:
$\|\boldsymbol{v}\|_2 \leq \|\boldsymbol{v}\|_1 \leq \sqrt{n}\|\boldsymbol{v}\|_2$

# 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:** $\underbrace{d_{\text{TV}}\left(\frac{1}{n}\boldsymbol{1},\ \left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right)}_{} \le\ \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\boldsymbol{1}-\left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x},\boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_1 \le \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x},\boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_2$$

$$= \frac{\sqrt{n}}{2}\cdot\sqrt{\sum_{i=2}^{n}\left(1-\frac{\lambda_i}{2}\right)^{2t}\langle \boldsymbol{x},\boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1-\frac{\lambda_2}{2}\right)^t\sqrt{\sum_{i=2}^{n}\langle \boldsymbol{x},\boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1-\frac{\lambda_2}{2}\right)^t\|\boldsymbol{x}\|_2$$

$\forall\, i \ge 2, \qquad 1-\frac{\lambda_i}{2} \le 1-\frac{\lambda_2}{2}$

$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n}\langle \boldsymbol{x},\boldsymbol{v}_i\rangle^2}$

# 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:** $d_{\text{TV}}\left(\frac{1}{n}\mathbf{1}, \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right) \le \epsilon.$

$$\frac{1}{2}\left\|\frac{1}{n}\mathbf{1} - \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{N}\right)^t \boldsymbol{x}\right\|_1 = \frac{1}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_1 \le \frac{\sqrt{n}}{2}\left\|\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^t \langle \boldsymbol{x}, \boldsymbol{v}_i\rangle \boldsymbol{v}_i\right\|_2$$

$$= \frac{\sqrt{n}}{2}\cdot\sqrt{\sum_{i=2}^{n}\left(1 - \frac{\lambda_i}{2}\right)^{2t}\langle \boldsymbol{x}, \boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1 - \frac{\lambda_2}{2}\right)^t\sqrt{\sum_{i=2}^{n}\langle \boldsymbol{x}, \boldsymbol{v}_i\rangle^2} \le \frac{\sqrt{n}}{2}\cdot\left(1 - \frac{\lambda_2}{2}\right)^t \|\boldsymbol{x}\|_2 \le \frac{\sqrt{n}}{2}\cdot\left(1 - \frac{\lambda_2}{2}\right)^t$$

$\|\boldsymbol{x}\|_2 \le \|\boldsymbol{x}\|_1 = 1$

# 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:** $\tau_{\text{mix}}(G, \epsilon) \in O\left(\frac{\log\frac{n}{\epsilon}}{\lambda_2}\right) \subseteq O\left(\frac{\log\frac{n}{\epsilon}}{\Phi(G)^2}\right)$

From now on, we return to **general graphs**, where the claim still holds, with the uniform distribution replaced by the degree distribution: $\boldsymbol{x}(u) = \frac{\deg(u)}{2|E|}$.

**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 $\tau_{\mathrm{mix}}(G, \epsilon)$ 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

↓

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 $\geq 1 - n^{-1}$

# 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 $\overbrace{\tau_{\mathrm{mix}}(G, \epsilon)}^{\in O\left(\frac{\log n}{\Phi(G)^2}\right) \subseteq O(|E|^2 \log n)}$ 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 <u>nearly uniformly at random</u>.

The total variation distance between the output distribution and the uniform distribution over all spanning trees is at most $\epsilon$.

# 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 $T' = T + e - e'$.

# 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.