CS-GY 6763: Lecture 12
Krylov Methods and Spectral Partitioning

NYU, Prof. Ainesh Bakshi

Power Method Formal Convergence

Theorem (Power Method Convergence, $k = 1$):

- Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ capture the gap between the first and second largest singular values.
- Initialize with a random Gaussian vector.
- After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 \leq \epsilon \quad \text{or} \quad \|\mathbf{v}_1 + \mathbf{z}^{(T)}\|_2 \leq \epsilon.$$

Power Method Formal Convergence

Theorem (Power Method Convergence, $k = 1$):

- Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ capture the gap between the first and second largest singular values.
- Initialize with a random Gaussian vector.
- After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 \leq \epsilon \quad \text{or} \quad \|\mathbf{v}_1 + \mathbf{z}^{(T)}\|_2 \leq \epsilon.$$

Note: The method won't converge if γ is very small (e.g., $\gamma = 0$):

$$\mathbf{z}^{(T)} = \frac{1}{\prod_{i=1}^{T} n_i}\left[c_1^{(0)}\sigma_1^{2T} \cdot \mathbf{v}_1 + c_2^{(0)}\sigma_2^{2T} \cdot \mathbf{v}_2 + \ldots + c_d^{(0)}\sigma_d^{2T} \cdot \mathbf{v}_d\right]$$

Power Method for Low-Rank Approximation

Low-Rank Approximation Given a matrix $\mathbf{X}$ we want to find a direction $\mathbf{v}$ such that

$$\min_{\|\mathbf{z}\|_2=1} \|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^\top\|_F^2$$

- Recall $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$

Power Method for Low-Rank Approximation

Low-Rank Approximation Given a matrix $\mathbf{X}$ we want to find a direction $\mathbf{v}$ such that

$$\min_{\|\mathbf{z}\|_2=1} \|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^\top\|_F^2$$

- Recall $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$
- The optimal solution is given by picking $\mathbf{z} = \mathbf{v}_1$ (the top singular vector of $\mathbf{X}$)

Power Method for Low-Rank Approximation

Low-Rank Approximation Given a matrix $\mathbf{X}$ we want to find a direction $\mathbf{v}$ such that

$$\min_{\|\mathbf{z}\|_2=1} \|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^\top\|_F^2$$

- Recall $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$
- The optimal solution is given by picking $\mathbf{z} = \mathbf{v}_1$ (the top singular vector of $\mathbf{X}$)
- The optimal cost is

$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

Power Method for Low-Rank Approximation

Low-Rank Approximation Given a matrix $\mathbf{X}$ we want to find a direction $\mathbf{v}$ such that

$$\min_{\|\mathbf{z}\|_2=1} \|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^\top\|_F^2$$

- Recall $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$
- The optimal solution is given by picking $\mathbf{z} = \mathbf{v}_1$ (the top singular vector of $\mathbf{X}$)
- The optimal cost is

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 &= \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 \\ &= \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{v}_1\|_2^2\end{aligned}$$

Power Method for Low-Rank Approximation

Low-Rank Approximation Given a matrix $\mathbf{X}$ we want to find a direction $\mathbf{v}$ such that

$$\min_{\|\mathbf{z}\|_2=1} \|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^\top\|_F^2$$

- Recall $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$
- The optimal solution is given by picking $\mathbf{z} = \mathbf{v}_1$ (the top singular vector of $\mathbf{X}$)
- The optimal cost is

$$\begin{aligned}
\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 &= \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 \\
&= \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{v}_1\|_2^2 \\
&= \sum_{i=1}^{d} \sigma_i^2 - \sigma_1^2 = \sum_{i=2}^{d} \sigma_i^2
\end{aligned}$$

Power Method for Low-Rank Approximation

Theorem: After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 + \epsilon\sigma_1^2.$$

Proof:

- $\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{z}^{(T)}\|_2^2$

Power Method for Low-Rank Approximation

Theorem: After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 + \epsilon\sigma_1^2.$$

Proof:

- $\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{z}^{(T)}\|_2^2$
- Write $\mathbf{z}^{(T)} = \sum_i \zeta_i \mathbf{v}_i$, so $\|\mathbf{X}\mathbf{z}^{(T)}\|_2^2 = \sum_i \sigma_i^2\zeta_i^2 \geq \sigma_1^2\zeta_1^2$

Power Method for Low-Rank Approximation

Theorem: After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 + \epsilon\sigma_1^2.$$

Proof:

- $\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{z}^{(T)}\|_2^2$
- Write $\mathbf{z}^{(T)} = \sum_i \zeta_i \mathbf{v}_i$, so $\|\mathbf{X}\mathbf{z}^{(T)}\|_2^2 = \sum_i \sigma_i^2\zeta_i^2 \geq \sigma_1^2\zeta_1^2$
- From convergence $\|\mathbf{z}^{(T)} - \mathbf{v}_1\|_2 \leq \epsilon$:

$$\epsilon^2 \geq \|\mathbf{z}^{(T)} - \mathbf{v}_1\|_2^2 = (1-\zeta_1)^2 + \sum_{i=2}^d \zeta_i^2 \geq (1-\zeta_1)^2$$

Power Method for Low-Rank Approximation

Theorem: After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 + \epsilon\sigma_1^2.$$

Proof:

- $\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{z}^{(T)}\|_2^2$
- Write $\mathbf{z}^{(T)} = \sum_i \zeta_i \mathbf{v}_i$, so $\|\mathbf{X}\mathbf{z}^{(T)}\|_2^2 = \sum_i \sigma_i^2\zeta_i^2 \geq \sigma_1^2\zeta_1^2$
- From convergence $\|\mathbf{z}^{(T)} - \mathbf{v}_1\|_2 \leq \epsilon$:

$$\epsilon^2 \geq \|\mathbf{z}^{(T)} - \mathbf{v}_1\|_2^2 = (1-\zeta_1)^2 + \sum_{i=2}^d \zeta_i^2 \geq (1-\zeta_1)^2$$

- Therefore $\zeta_1 \geq (1-\epsilon)$ and:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq \|\mathbf{X}\|_F^2 - \sigma_1^2(1-\epsilon)^2 \leq \sum_{i=2}^d \sigma_i^2 + \epsilon\sigma_1^2.$$

Souped-Up Power Method for LRA

Theorem: After $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}^{(T)}(\mathbf{z}^{(T)})^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2.$$

Power Method – No Gap Dependence

Theorem (Gapless Power Method Convergence):

- Initialize with a random Gaussian vector.

Power Method – No Gap Dependence

Theorem (Gapless Power Method Convergence):

- Initialize with a random Gaussian vector.
- After $T = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2.$$

Power Method – No Gap Dependence

Theorem (Gapless Power Method Convergence):

- Initialize with a random Gaussian vector.
- After $T = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ steps, with high probability:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2.$$

Intuition: For a good low-rank approximation, we don't need to converge exactly to $\mathbf{v}_1$ when $\sigma_1 \approx \sigma_2$. Any linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$ suffices.

Generalizations to Larger k

Block Power Method (aka Simultaneous / Subspace / Orthogonal Iteration):

- Choose $\mathbf{G} \in \mathbb{R}^{d \times k}$ randomly. Set $\mathbf{Z}^{(0)} = \text{orth}(\mathbf{G})$.

Generalizations to Larger k

Block Power Method (aka Simultaneous / Subspace / Orthogonal Iteration):

- Choose $\mathbf{G} \in \mathbb{R}^{d \times k}$ randomly. Set $\mathbf{Z}^{(0)} = \text{orth}(\mathbf{G})$.
- For $i = 1, \ldots, T$:
  - $\mathbf{Z}^{(i)} = \mathbf{X}^T(\mathbf{X}\mathbf{Z}^{(i-1)})$
  - $\mathbf{Z}^{(i)} = \text{orth}(\mathbf{Z}^{(i)})$
- Return $\mathbf{Z}^{(T)}$

Generalizations to Larger k

Block Power Method (aka Simultaneous / Subspace / Orthogonal Iteration):

- Choose $\mathbf{G} \in \mathbb{R}^{d \times k}$ randomly. Set $\mathbf{Z}^{(0)} = \text{orth}(\mathbf{G})$.
- For $i = 1, \ldots, T$:
  - $\mathbf{Z}^{(i)} = \mathbf{X}^T(\mathbf{X}\mathbf{Z}^{(i-1)})$
  - $\mathbf{Z}^{(i)} = \text{orth}(\mathbf{Z}^{(i)})$
- Return $\mathbf{Z}^{(T)}$

Guarantee: After $T = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ iterations:

$$\|\mathbf{X} - \mathbf{X}\mathbf{Z}^{(T)}(\mathbf{Z}^{(T)})^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2.$$

Runtime: $O(\text{nnz}(\mathbf{X}) \cdot k \cdot T)$.

Krylov Methods

Possible to "accelerate" these methods.

Convergence Guarantee: $T = O\left(\frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ iterations to obtain a nearly optimal low-rank approximation:

$$\|\mathbf{X} - \mathbf{X}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{V_k}\mathbf{V_k}^T\|_F^2.$$

Krylov Subspace Methods

For a normalizing constant c, power method returns:

$$\mathbf{z}^{(q)} = c \cdot \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$$

Along the way we computed:

$$\mathcal{K}_q = \left[\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right) \cdot \mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)^2 \cdot \mathbf{g}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}\right]$$

$\mathcal{K}$ is called the Krylov subspace of degree q.

Krylov Subspace Methods

For a normalizing constant c, power method returns:

$$\mathbf{z}^{(q)} = c \cdot \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$$

Along the way we computed:

$$\mathcal{K}_q = \left[\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right) \cdot \mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)^2 \cdot \mathbf{g}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}\right]$$

$\mathcal{K}$ is called the Krylov subspace of degree q.

Idea behind Krylov methods: Don't throw away everything before $\left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$.

Krylov Subspace Methods

Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method:

- Let $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be an orthonormal span for the vectors in $\mathcal{K}$.
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Krylov Subspace Methods

Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method:

- Let $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be an orthonormal span for the vectors in $\mathcal{K}$.
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.
  - Find best vector in the Krylov subspace, instead of just using last vector.

Krylov Subspace Methods

Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method:

- Let $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be an orthonormal span for the vectors in $\mathcal{K}$.
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.
  - Find best vector in the Krylov subspace, instead of just using last vector.
  - Can be done in $O\left(ndk + dk^2\right)$ time.

Krylov Subspace Methods

Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method:

- Let $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be an orthonormal span for the vectors in $\mathcal{K}$.
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.
  - Find best vector in the Krylov subspace, instead of just using last vector.
  - Can be done in $O\left(ndk + dk^2\right)$ time.
  - What you're using when you run `svds` or `eigs` in MATLAB or Python.

Krylov Subspace Methods

What vectors lie in the Krylov Subspace?

- Recall

$$\mathcal{K}_q = \left[\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)\cdot\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)^2\cdot\mathbf{g}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q\cdot\mathbf{g}\right]$$

- Then any vector in the span can be written as $\mathcal{K}_q z$:

$$\mathcal{K}_q z = \left(z_0 I + z_1\left(\mathbf{X}^T\mathbf{X}\right) + z_2\left(\mathbf{X}^T\mathbf{X}\right)^2 + \ldots + z_q\left(\mathbf{X}^T\mathbf{X}\right)^q\right) g$$

- This is some matrix valued polynomial of degree q multipied by g:

$$\left(\mathbf{X}^T\mathbf{X}\right)^q = \mathbf{V}\mathbf{\Sigma}^{2q}\mathbf{V}^T$$

Lanczos Method Analysis

- For a degree T polynomial p, let $\mathbf{v}_p = \frac{p(\mathbf{X}^T\mathbf{X})\mathbf{g}}{\|p(\mathbf{X}^T\mathbf{X})\mathbf{g}\|_2}$.

Lanczos Method Analysis

- For a degree T polynomial p, let $\mathbf{v}_p = \frac{p(\mathbf{X}^T\mathbf{X})\mathbf{g}}{\|p(\mathbf{X}^T\mathbf{X})\mathbf{g}\|_2}$.
- We always have that $\mathbf{v}_p \in \mathcal{K}_T$, the Krylov subspace constructed with T iterations.

Lanczos Method Analysis

- For a degree T polynomial p, let $\mathbf{v}_p = \frac{p(\mathbf{X}^T\mathbf{X})\mathbf{g}}{\|p(\mathbf{X}^T\mathbf{X})\mathbf{g}\|_2}$.
- We always have that $\mathbf{v}_p \in \mathcal{K}_T$, the Krylov subspace constructed with T iterations.
- Power method computes the polynomial $p(x) = x^T$ and outputs the vector $\mathbf{v}_{x^T}$.

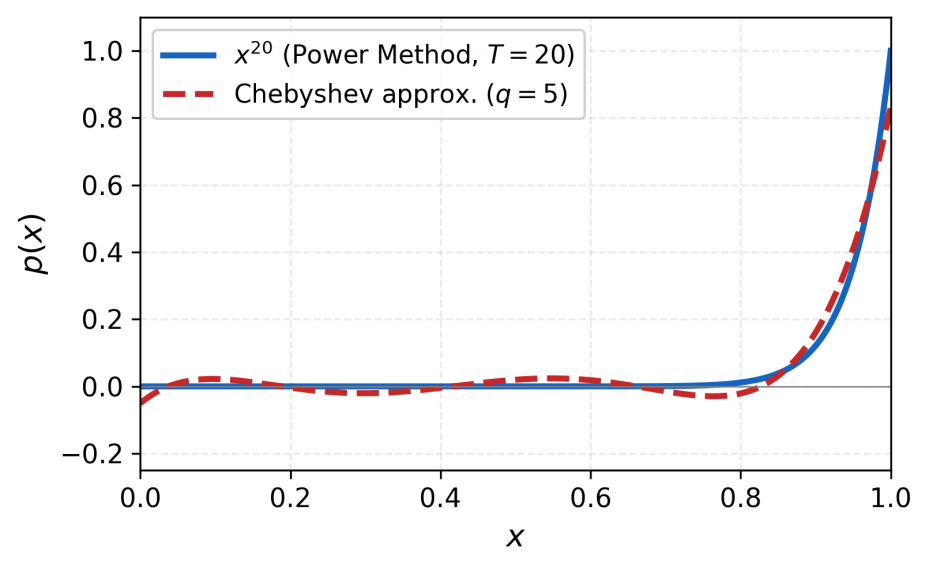
Lanczos Method Analysis

- For a degree T polynomial p, let $\mathbf{v}_p = \frac{p(\mathbf{X}^T\mathbf{X})\mathbf{g}}{\|p(\mathbf{X}^T\mathbf{X})\mathbf{g}\|_2}$.
- We always have that $\mathbf{v}_p \in \mathcal{K}_T$, the Krylov subspace constructed with T iterations.
- Power method computes the polynomial $p(x) = x^T$ and outputs the vector $\mathbf{v}_{x^T}$.
- Lanczos method can compute the best polynomial of degree T, denoted by p^*, and returns $\mathbf{v}_{p^*}$:

$$p^* = \underset{\text{degree } T \ p}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{v}_p\mathbf{v}_p^T\|_F^2.$$

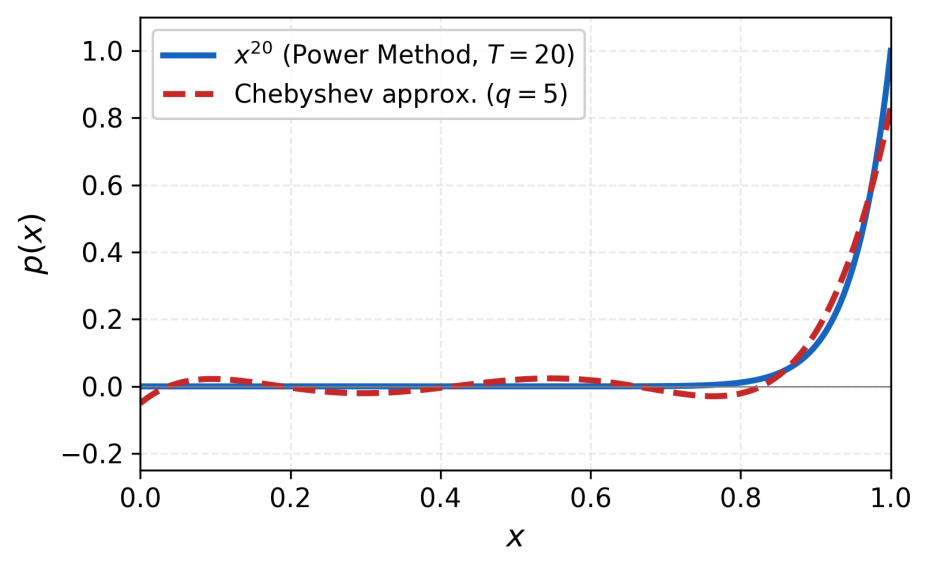
Lanczos Method Analysis

Theorem: There is a $q = O\left(\sqrt{T \log \frac{1}{\Delta}}\right)$ degree polynomial $\hat{p}$ approximating $\mathbf{x}^T$ up to error Δ on $[0, 1]$.



Lanczos Method Analysis

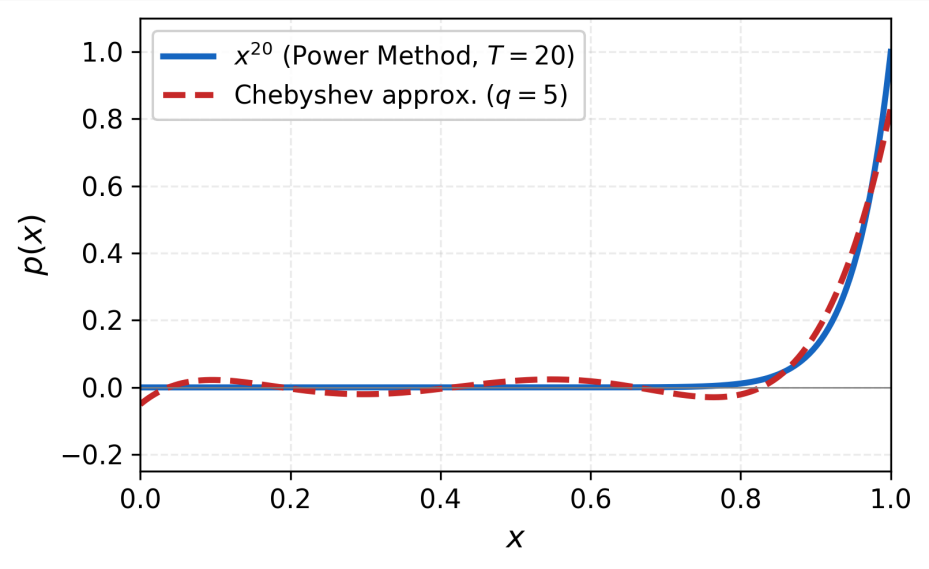
Theorem: There is a $q = O\left(\sqrt{T \log \frac{1}{\Delta}}\right)$ degree polynomial $\hat{p}$ approximating $\mathbf{x}^T$ up to error Δ on $[0, 1]$.



- Lanczos outputs the best polynomial so it is at least as good as the above polynomial
- Reduces the number of iterations from $1/\epsilon$ to $1/\sqrt{\epsilon}$!

Lanczos Method Analysis

Theorem: There is a $q = O\left(\sqrt{T \log \frac{1}{\Delta}}\right)$ degree polynomial $\hat{p}$ approximating $\mathbf{x}^T$ up to error Δ on $[0, 1]$.



- Lanczos outputs the best polynomial so it is at least as good as the above polynomial
- Reduces the number of iterations from $1/\epsilon$ to $1/\sqrt{\epsilon}$!
- Running time: $O\left(\frac{\log(d/\epsilon)}{\sqrt{\epsilon}} \cdot \text{nnz}(\mathbf{X})\right)$

Generalizations to Larger k

- Block Krylov methods
- Let $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathcal{K}_q = \left[\mathbf{G}, (\mathbf{X}^T\mathbf{X}) \cdot \mathbf{G}, (\mathbf{X}^T\mathbf{X})^2 \cdot \mathbf{G}, \ldots, (\mathbf{X}^T\mathbf{X})^q \cdot \mathbf{G}\right]$

Runtime: $O\left(\text{nnz}(\mathbf{X}) \cdot k \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ to obtain a $(1+\epsilon)$-approximate low-rank approximation.

Generalizations to Larger k

- Block Krylov methods
- Let $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathcal{K}_q = \left[\mathbf{G}, (\mathbf{X}^T\mathbf{X}) \cdot \mathbf{G}, (\mathbf{X}^T\mathbf{X})^2 \cdot \mathbf{G}, \ldots, (\mathbf{X}^T\mathbf{X})^q \cdot \mathbf{G}\right]$

Runtime: $O\left(\text{nnz}(\mathbf{X}) \cdot k \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ to obtain a $(1+\epsilon)$-approximate low-rank approximation.

Theorem [B., Clarkson, Woodruff '22]: Find a vector $\mathbf{z}$ such that

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

- **Iterations:** $O\left(\frac{\log(d/\epsilon)}{\epsilon^{1/3}}\right)$ and **Runtime:** $O\left(\text{nnz}(\mathbf{X}) \cdot \frac{\log d/\epsilon}{\epsilon^{1/3}}\right)$

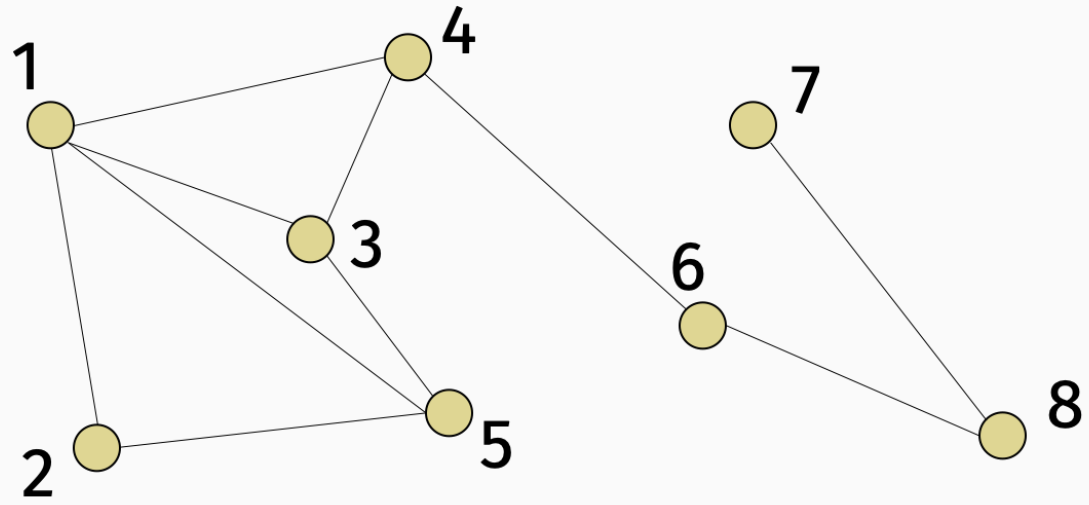
Spectral Graph Theory

Spectral Graph Theory

- **Main idea:** Understand graph data by constructing natural matrix representations, and studying that matrix's spectrum (eigenvalues/eigenvectors).

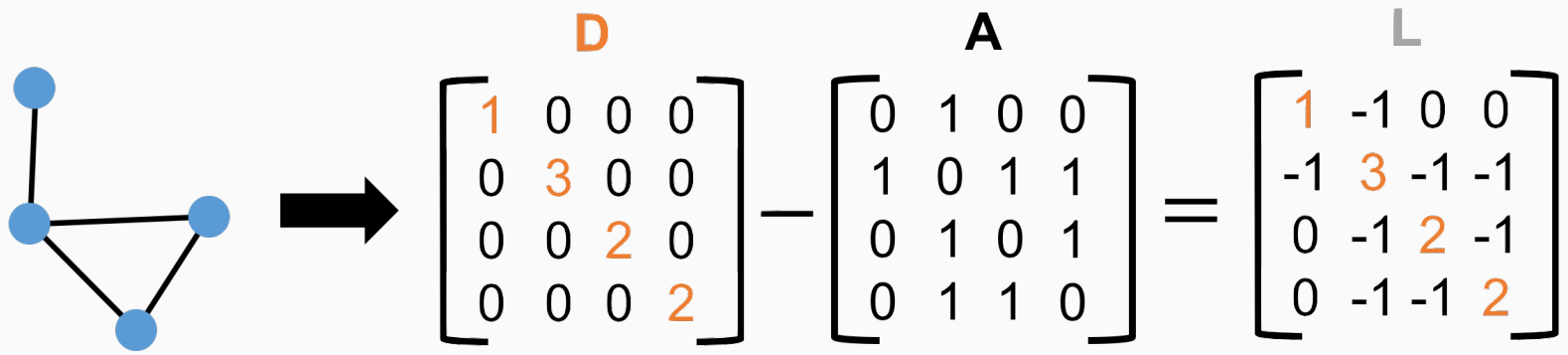
Spectral Graph Theory

- **Main idea:** Understand graph data by constructing natural matrix representations, and studying that matrix's spectrum (eigenvalues/eigenvectors).
- For now assume $G = (V, E)$ is an undirected, unweighted graph with n nodes.



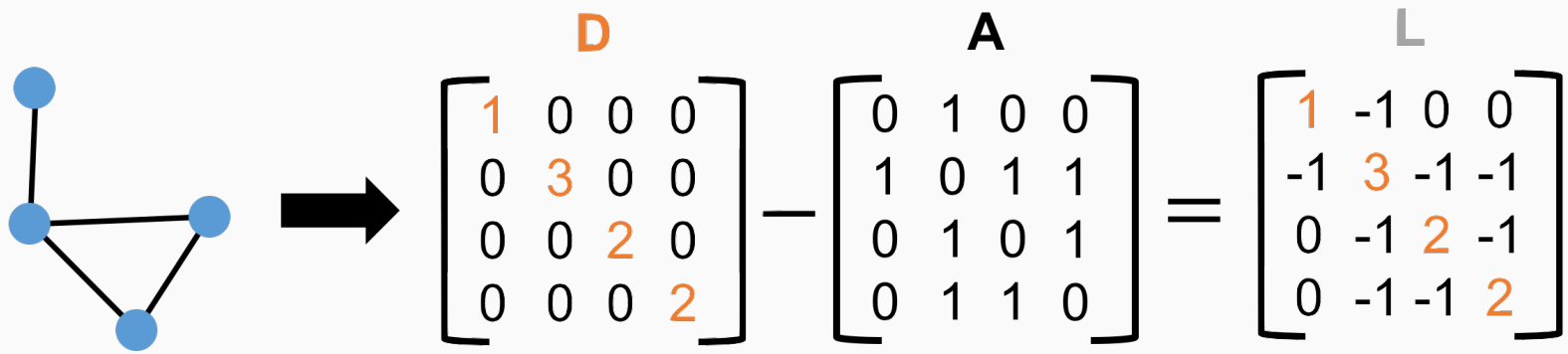
Matrix Representations of Graphs

Two most common representations: $n \times n$ adjacency matrix $\mathbf{A}$ and graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix.



Matrix Representations of Graphs

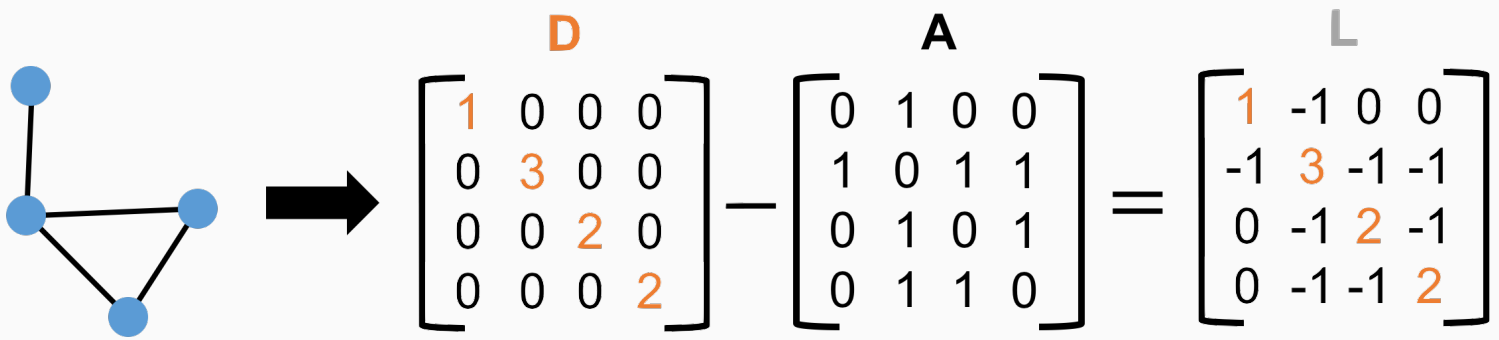
Two most common representations: $n \times n$ adjacency matrix $\mathbf{A}$ and graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix.



Also common to look at normalized versions of both of these:

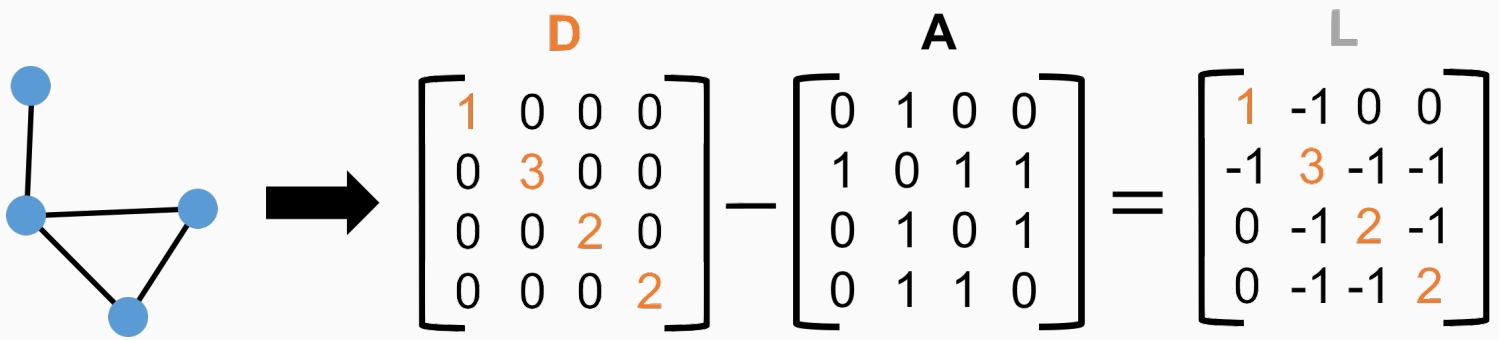
$$\bar{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \quad \text{and} \quad \bar{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

The Laplacian View



Edge-Vertex Incidence Matrix: B has a row for every edge in G. The row for edge (i, j) has a $+1$ at position i, a -1 at position j, and zeros elsewhere.

The Laplacian View



Edge-Vertex Incidence Matrix: $\mathbf{B}$ has a row for every edge in G. The row for edge (i, j) has a $+1$ at position i, a -1 at position j, and zeros elsewhere.

Fact: $\mathbf{L} = \mathbf{B}^T\mathbf{B}$ where B is the "edge-vertex incidence" matrix.

The Laplacian View

Example: For the graph above, $\mathbf{B}$ is a 4×4 matrix (4 edges, 4 vertices):

$$\mathbf{B} = \begin{array}{c} \\ (1,2) \\ (2,3) \\ (2,4) \\ (3,4) \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} +1 & -1 & 0 & 0 \\ 0 & +1 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{pmatrix} \end{array}$$

The Laplacian View

Example: For the graph above, $\mathbf{B}$ is a 4×4 matrix (4 edges, 4 vertices):

$$\mathbf{B} = \begin{array}{c} \\ (1,2) \\ (2,3) \\ (2,4) \\ (3,4) \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} +1 & -1 & 0 & 0 \\ 0 & +1 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{pmatrix} \end{array}$$

$$\mathbf{B}^T\mathbf{B} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} = \mathbf{L}$$

The Laplacian View

Edge Outer Product View: Let $\mathbf{b}_e^T$ be the row of $\mathbf{B}$ corresponding to edge $e = (i, j)$:

$$\mathbf{b}_e = \mathbf{e}_i - \mathbf{e}_j.$$

The Laplacian View

Edge Outer Product View: Let $\mathbf{b}_e^T$ be the row of $\mathbf{B}$ corresponding to edge $e = (i, j)$:

$$\mathbf{b}_e = \mathbf{e}_i - \mathbf{e}_j.$$

Write $\mathbf{B}$ as a stack of its edge rows:

$$\mathbf{B} = \begin{pmatrix} - & \mathbf{b}_{e_1}^T & - \\ - & \mathbf{b}_{e_2}^T & - \\ & \vdots & \\ - & \mathbf{b}_{e_m}^T & - \end{pmatrix}.$$

Then multiplying out $\mathbf{B}^T\mathbf{B}$ adds the contribution from each row:

$$\mathbf{L} = \mathbf{B}^T\mathbf{B} = \sum_{\ell=1}^{m} \mathbf{b}_{e_\ell}\mathbf{b}_{e_\ell}^T = \sum_{e=(i,j)\in E} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T.$$

The Laplacian View

Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

- $\mathbf{L}$ is positive semidefinite: $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x}$.

The Laplacian View

Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

- $\mathbf{L}$ is positive semidefinite: $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x}$.

- $\mathbf{L} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$ where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is $\mathbf{B}$'s SVD. Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{L}$.

The Laplacian View

Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

- $\mathbf{L}$ is positive semidefinite: $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x}$.
- $\mathbf{L} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$ where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is $\mathbf{B}$'s SVD. Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{L}$.
- For any vector $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{(i,j)\in E} (\mathbf{x}(i) - \mathbf{x}(j))^2.$$

The Laplacian View

Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

- $\mathbf{L}$ is positive semidefinite: $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for all $\mathbf{x}$.

- $\mathbf{L} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$ where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is $\mathbf{B}$'s SVD. Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{L}$.

- For any vector $\mathbf{x} \in \mathbb{R}^n$,

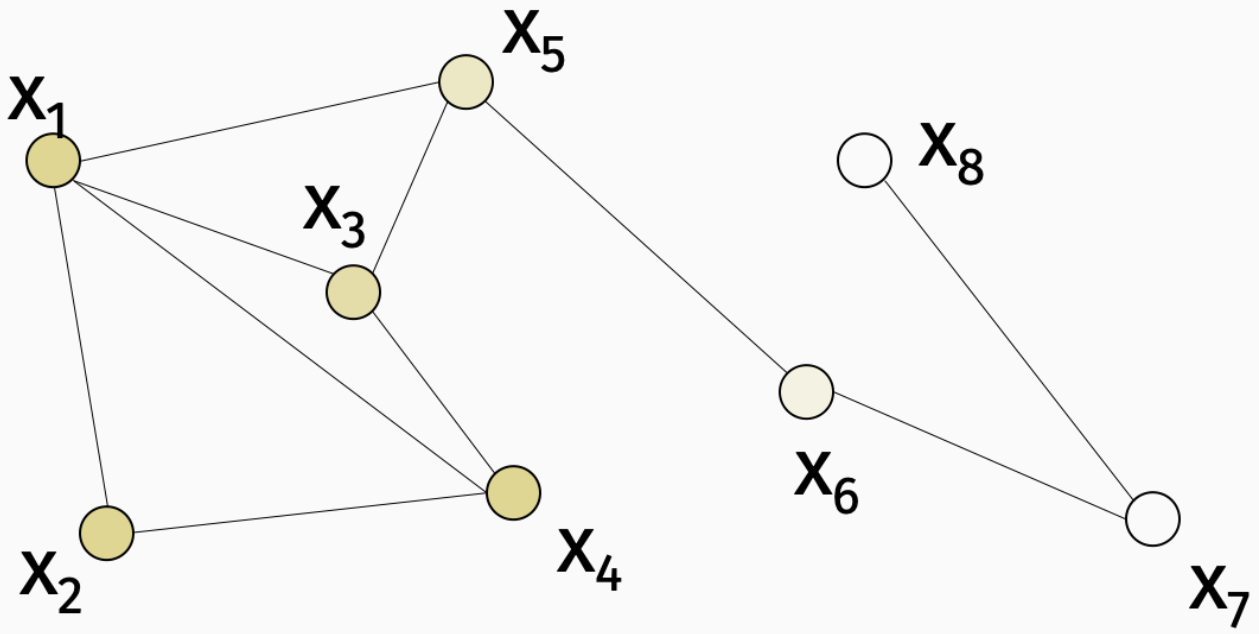
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{(i,j)\in E} (\mathbf{x}(i) - \mathbf{x}(j))^2.$$

Proof:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{e=(i,j)\in E} \mathbf{x}^T(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{x}^T$$

The Laplacian View

$\mathbf{x}^T L\mathbf{x} = \sum_{(i,j)\in E}(\mathbf{x}(i) - \mathbf{x}(j))^2$. So $\mathbf{x}^T L\mathbf{x}$ is small if $\mathbf{x}$ is a "smooth" function with respect to the graph.



Smallest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_n = \underset{\|\mathbf{v}\|=1}{\arg\min}\, \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Smallest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_n = \arg\min_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_{n-1} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Smallest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_n = \arg\min_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_{n-1} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_{n-2} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n, \mathbf{v}_{n-1}} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Smallest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_n = \arg\min_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_{n-1} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_{n-2} = \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n, \mathbf{v}_{n-1}} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\vdots$$

Smallest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$
\begin{aligned}
\mathbf{v}_n &= \arg\min_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_{n-1} &= \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_{n-2} &= \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n, \mathbf{v}_{n-1}} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
&\vdots \\
\mathbf{v}_1 &= \arg\min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_n, \ldots, \mathbf{v}_2} \mathbf{v}^T \mathbf{L} \mathbf{v}
\end{aligned}
$$

Largest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_1 = \underset{\|\mathbf{v}\|=1}{\arg\max}\, \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Largest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_1 = \arg\max_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_2 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Largest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_1 = \arg\max_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_2 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_3 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \mathbf{v}_2} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

Largest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$\mathbf{v}_1 = \arg\max_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_2 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\mathbf{v}_3 = \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \mathbf{v}_2} \mathbf{v}^T \mathbf{L} \mathbf{v}$$

$$\vdots$$

Largest Laplacian Eigenvector

Courant–Fischer min-max principle

Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be the eigenvectors of $\mathbf{L}$.

$$
\begin{aligned}
\mathbf{v}_1 &= \arg\max_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_2 &= \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_3 &= \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \mathbf{v}_2} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
&\vdots \\
\mathbf{v}_n &= \arg\max_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}} \mathbf{v}^T \mathbf{L} \mathbf{v}
\end{aligned}
$$

The Laplacian View

Another conclusion from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$:

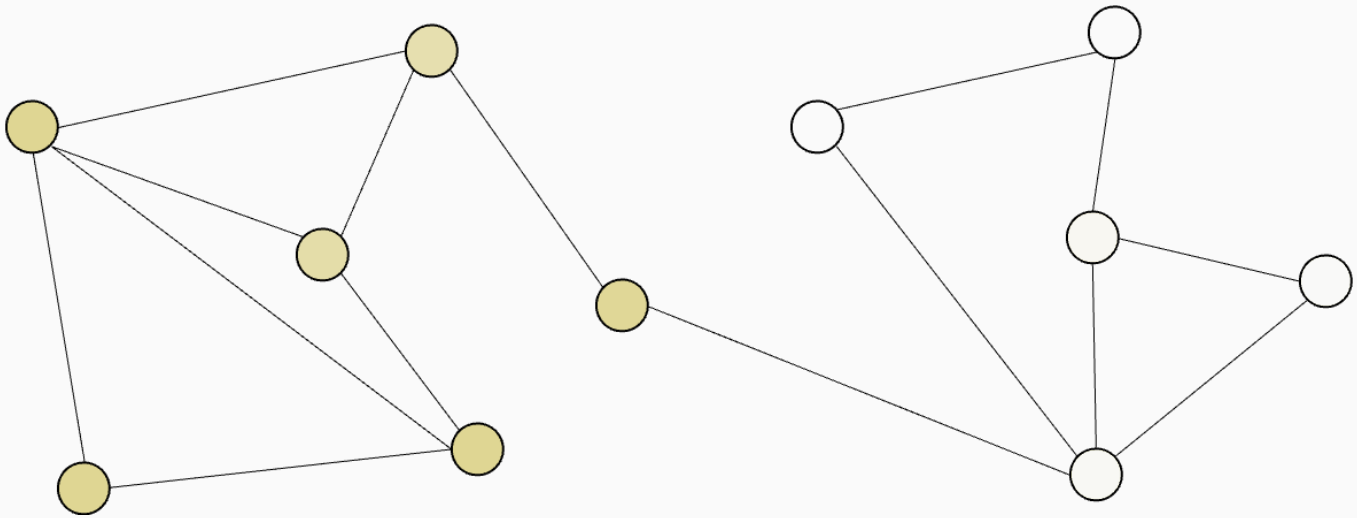
For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$ with $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T = V \setminus S$:

The Laplacian View

Another conclusion from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$:

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$ with $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T = V \setminus S$:

$$\mathbf{c}^T L\mathbf{c} = \sum_{(i,j)\in E} (\mathbf{c}(i) - \mathbf{c}(j))^2 = 4 \cdot \mathrm{cut}(S, T). \tag{1}$$



Spectral Graph Partitioning

- Introduce NP-hard graph partitioning prob. important in:
  - Understanding social networks.
  - Unsupervised machine learning (spectral clustering).
  - Graph visualization.
  - Mesh partitioning.

Spectral Graph Partitioning

- Introduce NP-hard graph partitioning prob. important in:
  - Understanding social networks.
  - Unsupervised machine learning (spectral clustering).
  - Graph visualization.
  - Mesh partitioning.
- See how this problem can be solved heuristically using Laplacian eigenvectors.

Spectral Graph Partitioning

- Introduce NP-hard graph partitioning prob. important in:
  - Understanding social networks.
  - Unsupervised machine learning (spectral clustering).
  - Graph visualization.
  - Mesh partitioning.
- See how this problem can be solved heuristically using Laplacian eigenvectors.
- Give an "average case" analysis of the method for a common random graph model.

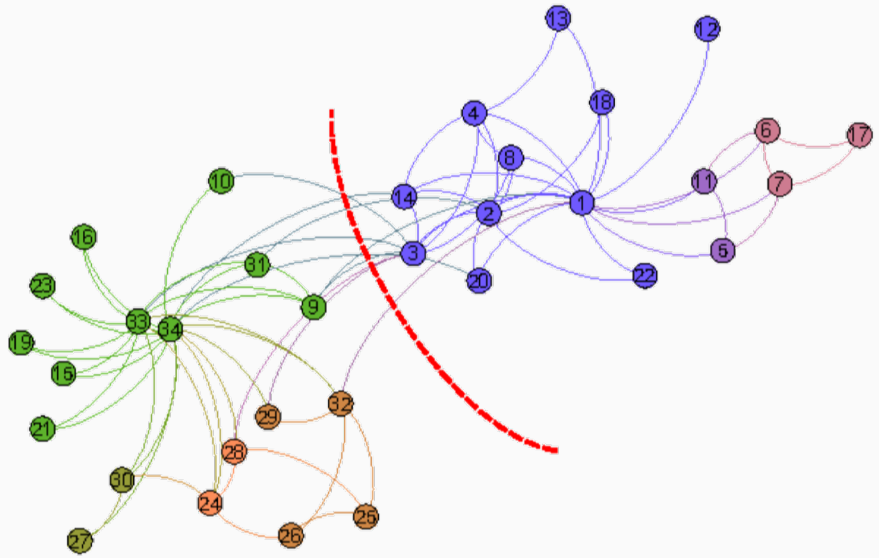
Spectral Graph Partitioning

- Introduce NP-hard graph partitioning prob. important in:
  - Understanding social networks.
  - Unsupervised machine learning (spectral clustering).
  - Graph visualization.
  - Mesh partitioning.
- See how this problem can be solved heuristically using Laplacian eigenvectors.
- Give an "average case" analysis of the method for a common random graph model.
- Use two tools: matrix concentration and eigenvector perturbation bounds.

Balanced Cut

Goal: Given a graph $G = (V, E)$, partition nodes along a cut that:

- Has few crossing edges: $|\{(u, v) \in E : u \in S, v \in T\}|$ is small.
- Separates large partitions: $|S|, |T|$ are not too small.

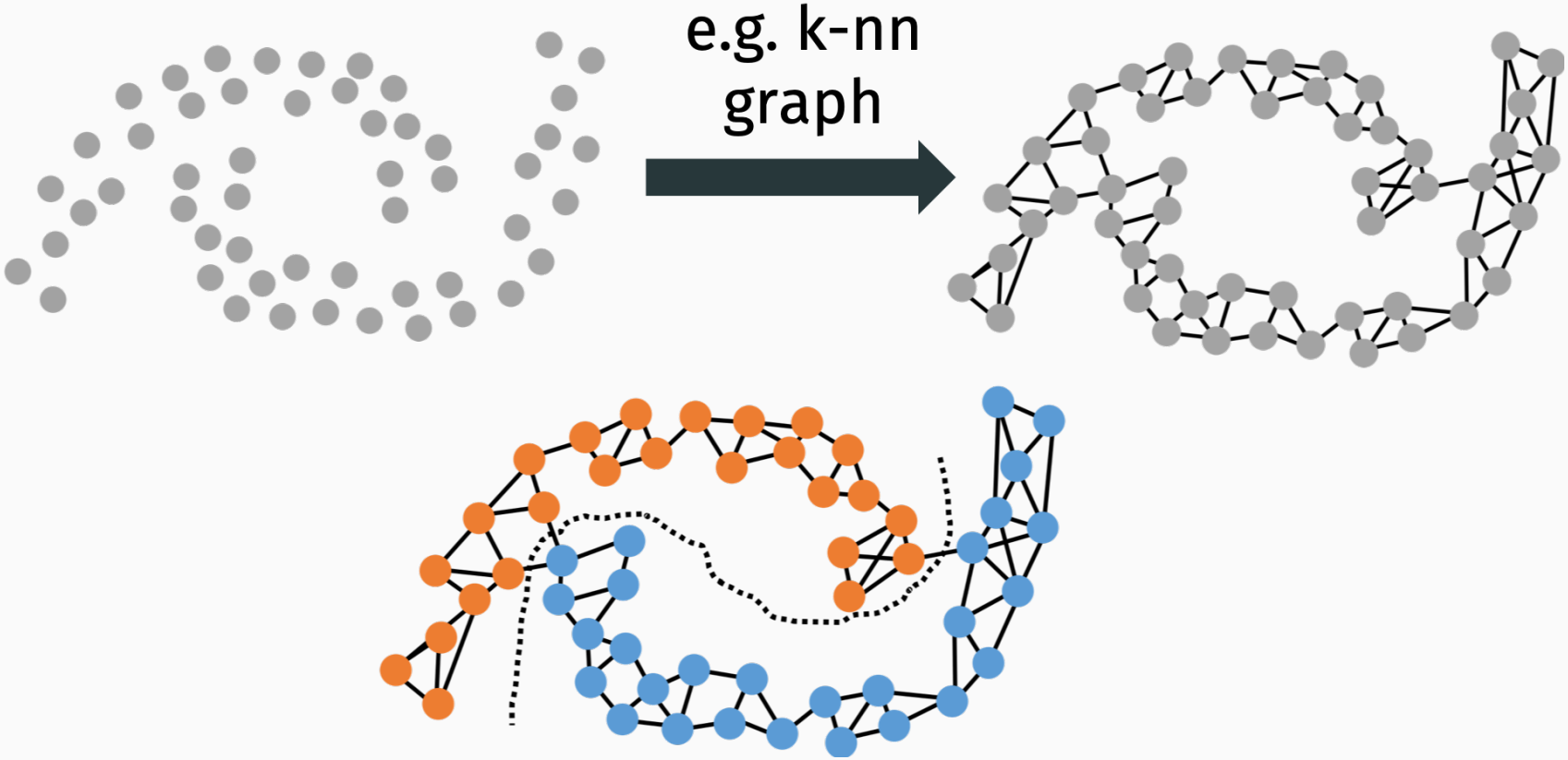


(a) Zachary Karate Club Graph

Example application: Understanding community structure in social networks.

Spectral Clustering

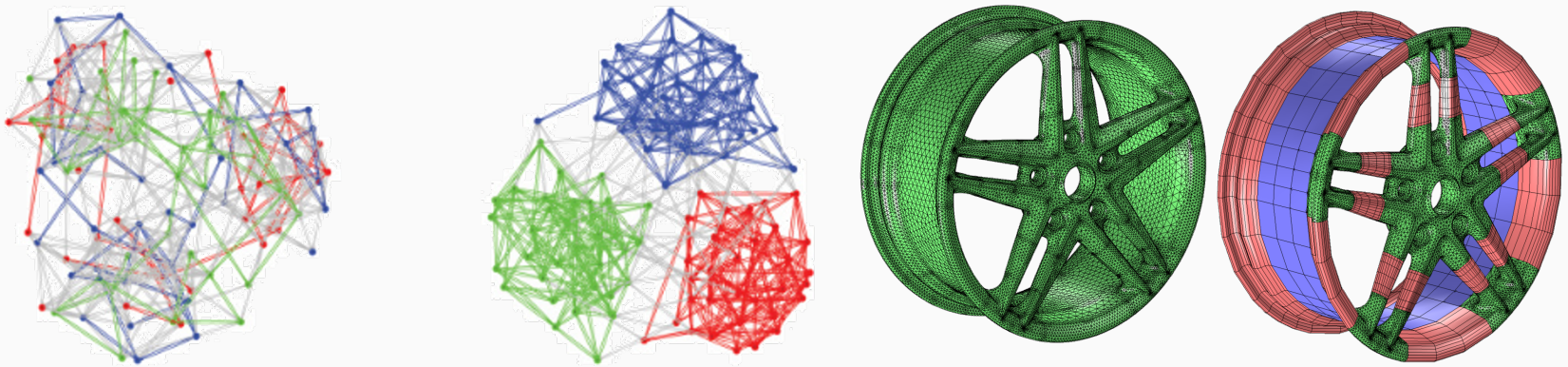
Idea: Construct synthetic graph for data that is hard to cluster.



Spectral Clustering, Laplacian Eigenmaps, Locally linear embedding, Isomap, etc.

Tons of Other Applications!

Balanced cut algorithms are also use in distributing data in graph databases, for partitioning finite element meshes in scientific computing (e.g., that arise when solving differential equations), and more.



Lots of good software packages (e.g. METIS).

Spectral Graph Partitioning

There are many ways to formalize the balanced partitioning problem.

β-Balanced Cut:

$$\min_S \text{cut}(S, V \setminus S) \quad \text{such that} \quad \min(|S|, |V \setminus S|) \geq \beta n.$$

Spectral Graph Partitioning

There are many ways to formalize the balanced partitioning problem.

β-Balanced Cut:

$$\min_S \text{cut}(S, V \setminus S) \quad \text{such that} \quad \min(|S|, |V \setminus S|) \geq \beta n.$$

Today: Why does a very simple spectral heuristic work on a natural random graph model?

Spectral Graph Partitioning

Basic spectral clustering method:

- Compute the second smallest eigenvector $\mathbf{v}_{n-1}$ of the graph Laplacian $\mathbf{L}$.

Spectral Graph Partitioning

Basic spectral clustering method:

- Compute the second smallest eigenvector $\mathbf{v}_{n-1}$ of the graph Laplacian $\mathbf{L}$.
- Interpret $\mathbf{v}_{n-1}(i)$ as a real-valued score for node i.

Spectral Graph Partitioning

Basic spectral clustering method:

- Compute the second smallest eigenvector $\mathbf{v}_{n-1}$ of the graph Laplacian $\mathbf{L}$.
- Interpret $\mathbf{v}_{n-1}(i)$ as a real-valued score for node i.
- Put nodes with positive score in T and nodes with negative score in S.

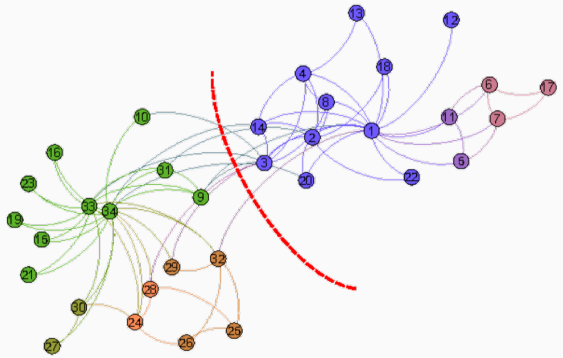
Spectral Graph Partitioning

Basic spectral clustering method:

- Compute the second smallest eigenvector $\mathbf{v}_{n-1}$ of the graph Laplacian $\mathbf{L}$.
- Interpret $\mathbf{v}_{n-1}(i)$ as a real-valued score for node i.
- Put nodes with positive score in T and nodes with negative score in S.

This is a relax-and-round algorithm in disguise.

The Laplacian View

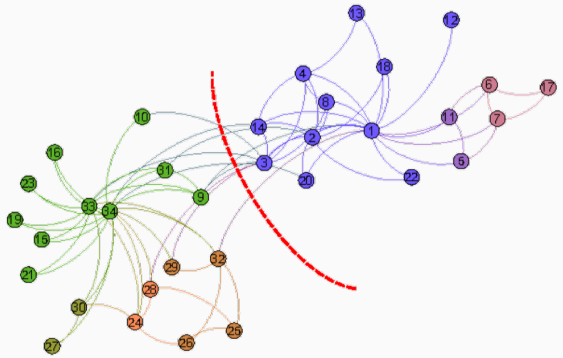


(a) Zachary Karate Club Graph

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$:

- $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$.

The Laplacian View

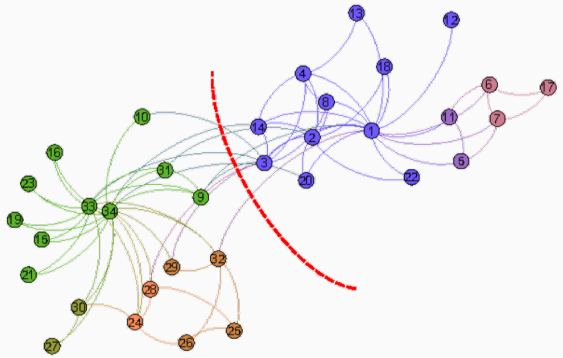


(a) Zachary Karate Club Graph

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$:

- $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$.
- $\mathbf{c}^T \mathbf{L} \mathbf{c} = 4 \cdot \text{cut}(S, T)$ measures the cut size.

The Laplacian View

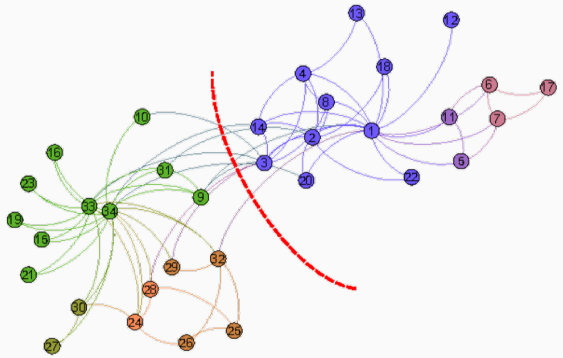


(a) Zachary Karate Club Graph

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$:

- $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$.
- $\mathbf{c}^T \mathbf{L} \mathbf{c} = 4 \cdot \text{cut}(S, T)$ measures the cut size.
- $\mathbf{c}^T \mathbf{1} = |T| - |S|$ measures the imbalance.

The Laplacian View



(a) Zachary Karate Club Graph

For a cut indicator vector $\mathbf{c} \in \{-1, 1\}^n$:

- $\mathbf{c}(i) = -1$ for $i \in S$ and $\mathbf{c}(i) = 1$ for $i \in T$.
- $\mathbf{c}^T \mathbf{L} \mathbf{c} = 4 \cdot \text{cut}(S, T)$ measures the cut size.
- $\mathbf{c}^T \mathbf{1} = |T| - |S|$ measures the imbalance.

Want small $\mathbf{c}^T \mathbf{L} \mathbf{c}$ and small $|\mathbf{c}^T \mathbf{1}|$.

Relax and Round

Scale the cut indicator so that $\|\mathbf{c}\|_2 = 1$:

$$\mathbf{c} \in \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n .$$

Relax and Round

Scale the cut indicator so that $\|\mathbf{c}\|_2 = 1$:

$$\mathbf{c} \in \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n.$$

Perfectly balanced cut problem:

$$\min_{\mathbf{c} \in \{-1/\sqrt{n}, 1/\sqrt{n}\}^n} \mathbf{c}^T \mathbf{L} \mathbf{c} \quad \text{such that} \quad \mathbf{c}^T \mathbf{1} = 0.$$

Relax and Round

Scale the cut indicator so that $\|\mathbf{c}\|_2 = 1$:

$$\mathbf{c} \in \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n .$$

Perfectly balanced cut problem:

$$\min_{\mathbf{c} \in \{-1/\sqrt{n}, 1/\sqrt{n}\}^n} \mathbf{c}^T \mathbf{L} \mathbf{c} \quad \text{such that} \quad \mathbf{c}^T \mathbf{1} = 0.$$

Relaxation:

$$\min_{\|\mathbf{x}\|_2 = 1} \mathbf{x}^T \mathbf{L} \mathbf{x} \quad \text{such that} \quad \mathbf{x}^T \mathbf{1} = 0.$$

Relax and Round

Claim: The relaxed problem is minimized by the second smallest eigenvector $\mathbf{v}_{n-1}$ of $\mathbf{L}$.

Relax and Round

Claim: The relaxed problem is minimized by the second smallest eigenvector $\mathbf{v}_{n-1}$ of $\mathbf{L}$. **Proof:**

- The smallest eigenvector is $\mathbf{v}_n = \frac{1}{\sqrt{n}}\mathbf{1}$ with eigenvalue 0:

$$\mathbf{L}\mathbf{1} = \mathbf{0},$$

since the constant vector has no variation across any edge.

Relax and Round

Claim: The relaxed problem is minimized by the second smallest eigenvector $\mathbf{v}_{n-1}$ of $\mathbf{L}$. **Proof:**

- The smallest eigenvector is $\mathbf{v}_n = \frac{1}{\sqrt{n}}\mathbf{1}$ with eigenvalue 0:

$$\mathbf{L1} = \mathbf{0},$$

  since the constant vector has no variation across any edge.

- The constraint $\mathbf{x}^T\mathbf{1} = 0$ removes this trivial direction. So the relaxed problem asks for the unit vector, orthogonal to $\mathbf{v}_n$, with smallest value of $\mathbf{x}^T\mathbf{Lx}$.

Relax and Round

Claim: The relaxed problem is minimized by the second smallest eigenvector $\mathbf{v}_{n-1}$ of $\mathbf{L}$. **Proof:**

- The smallest eigenvector is $\mathbf{v}_n = \frac{1}{\sqrt{n}}\mathbf{1}$ with eigenvalue 0:

$$\mathbf{L1} = \mathbf{0},$$

  since the constant vector has no variation across any edge.

- The constraint $\mathbf{x}^T\mathbf{1} = 0$ removes this trivial direction. So the relaxed problem asks for the unit vector, orthogonal to $\mathbf{v}_n$, with smallest value of $\mathbf{x}^T\mathbf{Lx}$. By Courant-Fischer, this is exactly:

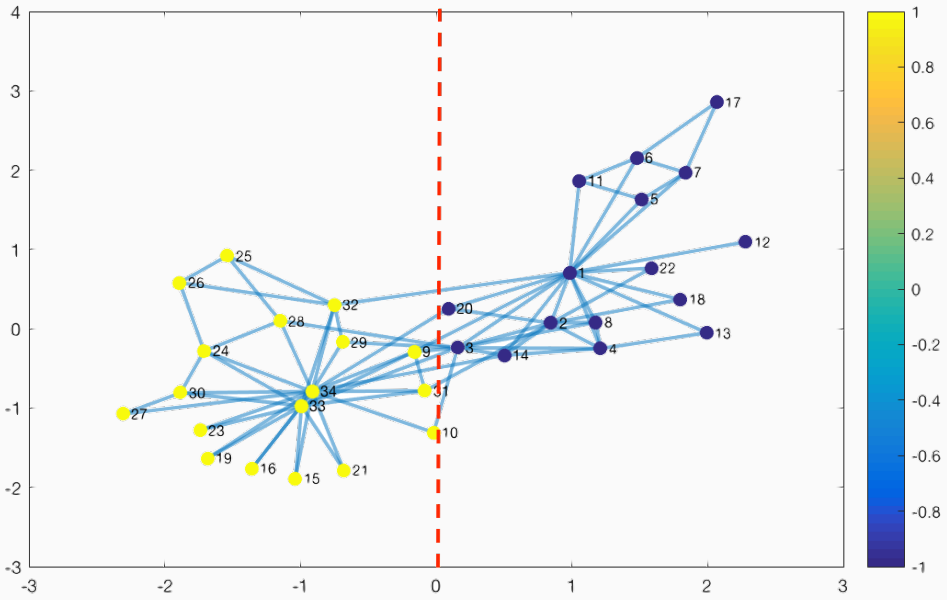
$$\mathbf{v}_{n-1} = \underset{\|\mathbf{v}\|=1,\ \mathbf{v}\perp\mathbf{v}_n}{\arg\min}\ \mathbf{v}^T\mathbf{Lv}.$$

Relax and Round

Algorithm: Relax, compute $\mathbf{v}_{n-1}$, then round by signs.

Relax and Round

Algorithm: Relax, compute $\mathbf{v}_{n-1}$, then round by signs.



Generative Models

So far: Spectral clustering is a natural heuristic for finding a small balanced cut.

Generative Models

So far: Spectral clustering is a natural heuristic for finding a small balanced cut.

- But it does not solve worst-case balanced cut.

Generative Models

So far: Spectral clustering is a natural heuristic for finding a small balanced cut.

- But it does not solve worst-case balanced cut.
- To understand why it often works, analyze it under a simple random graph model.

Generative Models

So far: Spectral clustering is a natural heuristic for finding a small balanced cut.

- But it does not solve worst-case balanced cut.
- To understand why it often works, analyze it under a simple random graph model.
- This is an average-case guarantee rather than a worst-case guarantee.

Stochastic Block Model

What random graph model should produce graphs with two communities?

Stochastic Block Model

What random graph model should produce graphs with two communities?

Nodes in the same community should be more likely to share an edge.

Stochastic Block Model

Stochastic Block Model (Planted Partition Model):

Let $G_n(p, q)$ be a distribution over graphs on n nodes, split equally into two groups B and C.

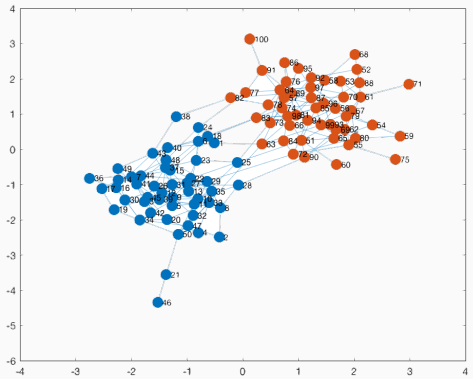
- Nodes in the **same group** are connected with probability p.

Stochastic Block Model

Stochastic Block Model (Planted Partition Model):

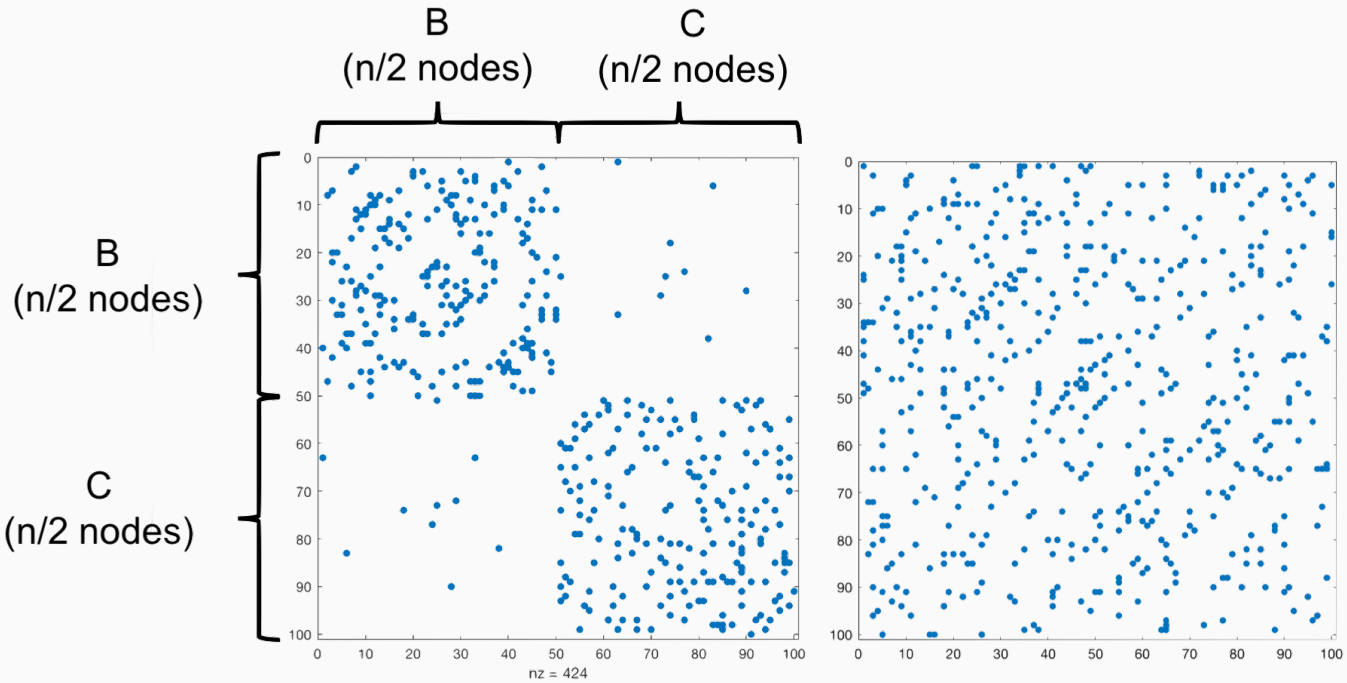
Let $G_n(p, q)$ be a distribution over graphs on n nodes, split equally into two groups B and C.

- Nodes in the **same group** are connected with probability p.
- Nodes in **different groups** are connected with probability $q < p$.



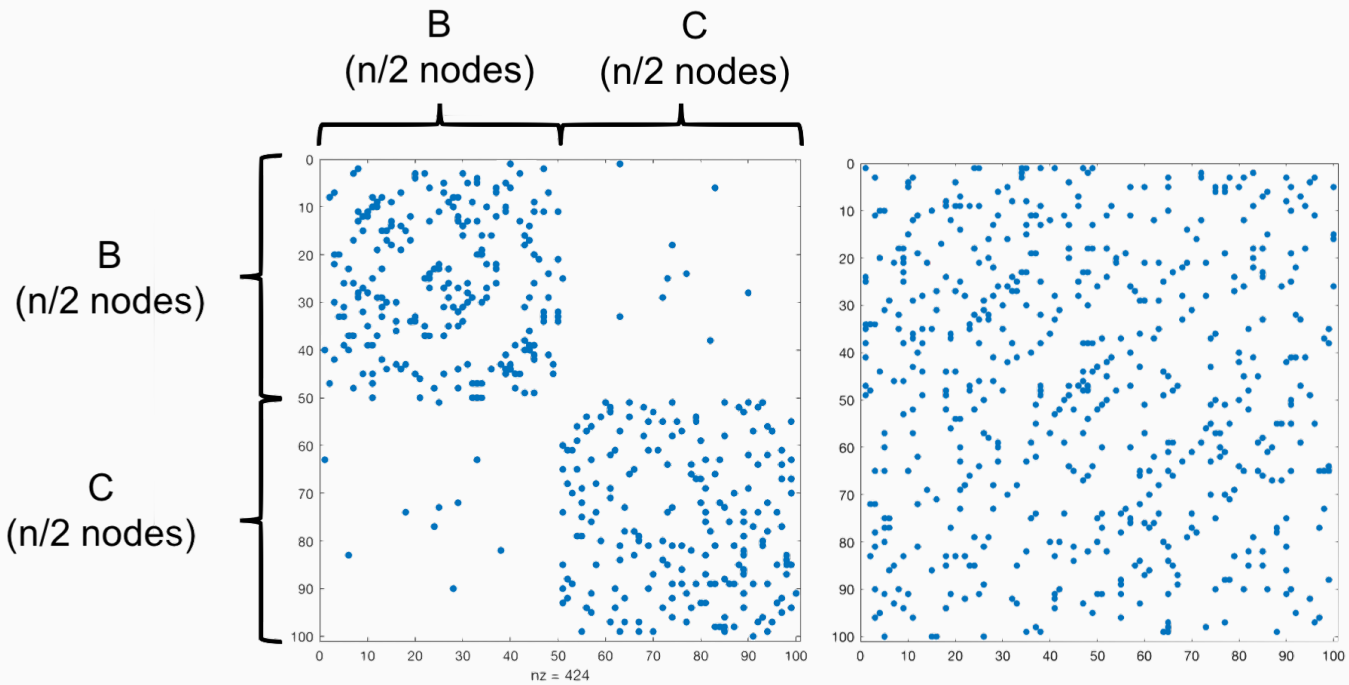
Linear Algebraic View

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of a graph drawn from $G_n(p, q)$.



Linear Algebraic View

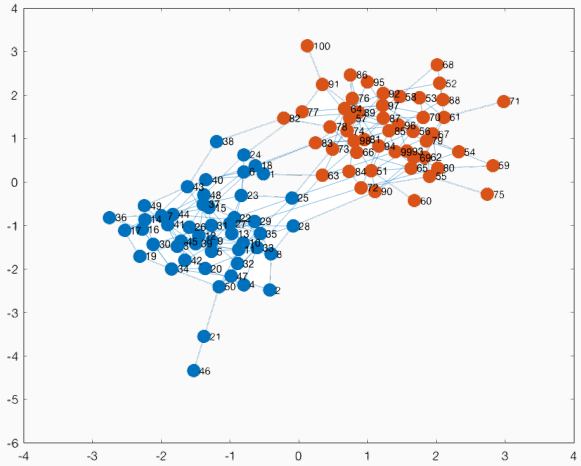
Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of a graph drawn from $G_n(p, q)$.



Note: We can order the nodes by group for analysis. In real data, the matrix would look scrambled, as on the right.

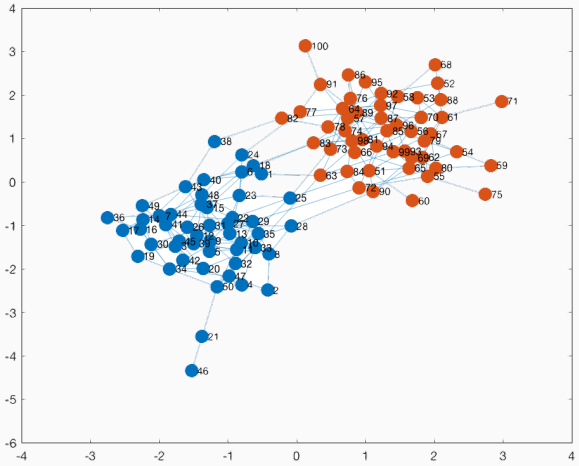
Stochastic Block Model

Goal: Recover the hidden partition B, C using the spectral method.



Stochastic Block Model

Goal: Recover the hidden partition B, C using the spectral method.

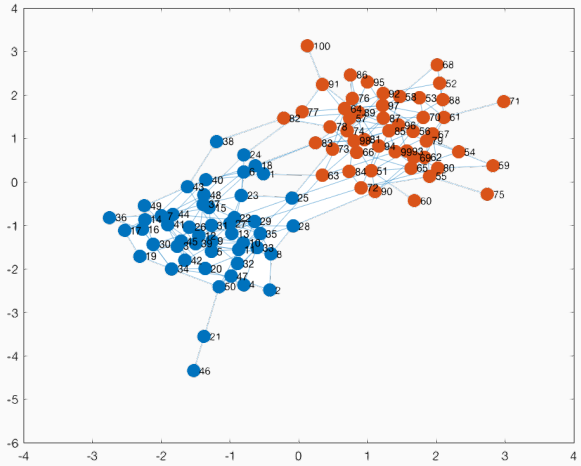


We need to understand the second smallest eigenvector of:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

Stochastic Block Model

Goal: Recover the hidden partition B, C using the spectral method.



We need to understand the second smallest eigenvector of:

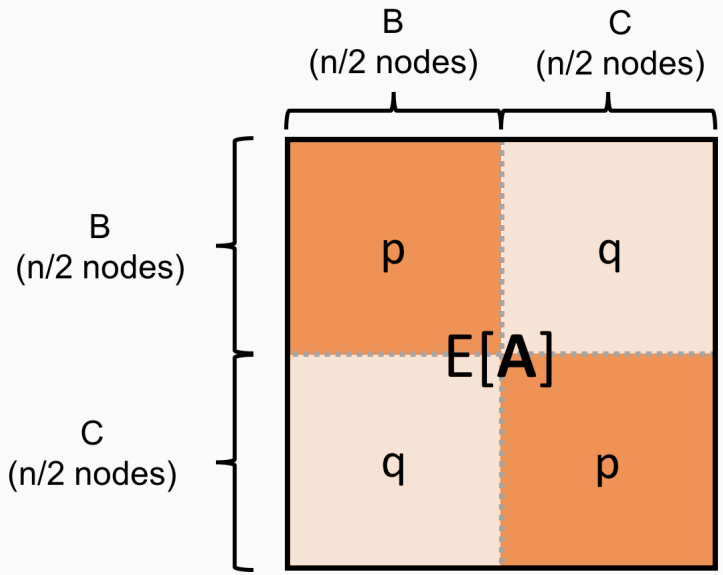
$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

We will start by studying $\mathbb{E}[\mathbf{A}]$ and $\mathbb{E}[\mathbf{L}]$.

Expected Adjacency Matrix

For $G \sim G_n(p, q)$:

$$(\mathbb{E}[\mathbf{A}])_{ij} = \begin{cases} p & \text{if } i, j \text{ are in the same group,} \\ q & \text{if } i, j \text{ are in different groups.} \end{cases}$$



Expected Laplacian

What is the expected Laplacian of $G_n(p, q)$?

Expected Laplacian

What is the expected Laplacian of $G_n(p, q)$?

For this balanced model, each node has the same expected degree:

$$\mathbb{E}[\mathbf{D}] = \frac{(p+q)n}{2} \cdot \mathbf{I}.$$

Expected Laplacian

What is the expected Laplacian of $G_n(p, q)$?

For this balanced model, each node has the same expected degree:

$$\mathbb{E}[\mathbf{D}] = \frac{(p+q)n}{2} \cdot \mathbf{I}.$$

So

$$\mathbb{E}[\mathbf{L}] = \mathbb{E}[\mathbf{D}] - \mathbb{E}[\mathbf{A}] = \frac{(p+q)n}{2}\mathbf{I} - \mathbb{E}[\mathbf{A}].$$

Expected Laplacian

What is the expected Laplacian of $G_n(p, q)$?

For this balanced model, each node has the same expected degree:

$$\mathbb{E}[\mathbf{D}] = \frac{(p+q)n}{2} \cdot \mathbf{I}.$$

So

$$\mathbb{E}[\mathbf{L}] = \mathbb{E}[\mathbf{D}] - \mathbb{E}[\mathbf{A}] = \frac{(p+q)n}{2}\mathbf{I} - \mathbb{E}[\mathbf{A}].$$

Thus $\mathbb{E}[\mathbf{L}]$ is a shifted version of $-\mathbb{E}[\mathbf{A}]$.

Expected Laplacian Eigenvectors

Claim: The eigenvectors of $\mathbb{E}[\mathbf{L}]$ are the eigenvectors of $\mathbb{E}[\mathbf{A}]$.

Expected Laplacian Eigenvectors

Claim: The eigenvectors of $\mathbb{E}[\mathbf{L}]$ are the eigenvectors of $\mathbb{E}[\mathbf{A}]$.

Proof: Let

$$c = \frac{(p+q)n}{2}, \quad \text{so} \quad \mathbb{E}[\mathbf{D}] = c\mathbf{I}.$$

Expected Laplacian Eigenvectors

Claim: The eigenvectors of $\mathbb{E}[\mathbf{L}]$ are the eigenvectors of $\mathbb{E}[\mathbf{A}]$.

Proof: Let

$$c = \frac{(p+q)n}{2}, \quad \text{so} \quad \mathbb{E}[\mathbf{D}] = c\mathbf{I}.$$

Suppose $\mathbf{v}$ is an eigenvector of $\mathbb{E}[\mathbf{L}]$:

$$\mathbb{E}[\mathbf{L}]\mathbf{v} = \lambda\mathbf{v}.$$

Expected Laplacian Eigenvectors

Claim: The eigenvectors of $\mathbb{E}[\mathbf{L}]$ are the eigenvectors of $\mathbb{E}[\mathbf{A}]$.

Proof: Let

$$c = \frac{(p+q)n}{2}, \quad \text{so} \quad \mathbb{E}[\mathbf{D}] = c\mathbf{I}.$$

Suppose $\mathbf{v}$ is an eigenvector of $\mathbb{E}[\mathbf{L}]$:

$$\mathbb{E}[\mathbf{L}]\mathbf{v} = \lambda\mathbf{v}.$$

Using $\mathbb{E}[\mathbf{L}] = \mathbb{E}[\mathbf{D}] - \mathbb{E}[\mathbf{A}] = c\mathbf{I} - \mathbb{E}[\mathbf{A}]$:

$$(c\mathbf{I} - \mathbb{E}[\mathbf{A}])\mathbf{v} = \lambda\mathbf{v}$$

$$\mathbb{E}[\mathbf{A}]\mathbf{v} = (c - \lambda)\mathbf{v}.$$

Expected Laplacian Eigenvectors

Claim: The eigenvectors of $\mathbb{E}[\mathbf{L}]$ are the eigenvectors of $\mathbb{E}[\mathbf{A}]$.

Proof: Let

$$c = \frac{(p+q)n}{2}, \quad \text{so} \quad \mathbb{E}[\mathbf{D}] = c\mathbf{I}.$$

Suppose $\mathbf{v}$ is an eigenvector of $\mathbb{E}[\mathbf{L}]$:

$$\mathbb{E}[\mathbf{L}]\mathbf{v} = \lambda\mathbf{v}.$$

Using $\mathbb{E}[\mathbf{L}] = \mathbb{E}[\mathbf{D}] - \mathbb{E}[\mathbf{A}] = c\mathbf{I} - \mathbb{E}[\mathbf{A}]$:

$$(c\mathbf{I} - \mathbb{E}[\mathbf{A}])\mathbf{v} = \lambda\mathbf{v}$$

$$\mathbb{E}[\mathbf{A}]\mathbf{v} = (c - \lambda)\mathbf{v}.$$

So $\mathbf{v}$ is also an eigenvector of $\mathbb{E}[\mathbf{A}]$. The eigenvalues change, but the eigenvectors do not.

Expected Adjacency Spectrum

Because $\mathbb{E}[\mathbf{L}]$ and $\mathbb{E}[\mathbf{A}]$ have the same eigenvectors, it suffices to understand $\mathbb{E}[\mathbf{A}]$.

Claim: $\mathbf{1}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

Expected Adjacency Spectrum

Because $\mathbb{E}[\mathbf{L}]$ and $\mathbb{E}[\mathbf{A}]$ have the same eigenvectors, it suffices to understand $\mathbb{E}[\mathbf{A}]$.

Claim: $\mathbf{1}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

$$\left[\begin{array}{ccc|ccc} p & \cdots & p & q & \cdots & q \\ \hline q & \cdots & q & p & \cdots & p \end{array}\right] \begin{bmatrix} \mathbf{1}_B \\ \mathbf{1}_C \end{bmatrix} = \begin{bmatrix} \left(\frac{pn}{2} + \frac{qn}{2}\right) \mathbf{1}_B \\ \left(\frac{qn}{2} + \frac{pn}{2}\right) \mathbf{1}_C \end{bmatrix}$$

$$= \frac{(p+q)n}{2} \begin{bmatrix} \mathbf{1}_B \\ \mathbf{1}_C \end{bmatrix}$$

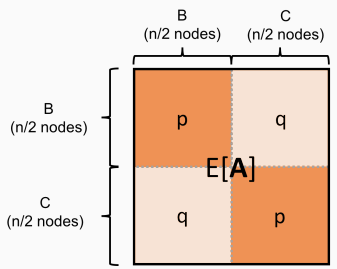
Expected Adjacency Spectrum

Because $\mathbb{E}[\mathbf{L}]$ and $\mathbb{E}[\mathbf{A}]$ have the same eigenvectors, it suffices to understand $\mathbb{E}[\mathbf{A}]$.

Claim: $\mathbf{1}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

$$\left[\begin{array}{ccc|ccc} p & \cdots & p & q & \cdots & q \\ \hline q & \cdots & q & p & \cdots & p \end{array}\right] \begin{bmatrix} \mathbf{1}_B \\ \mathbf{1}_C \end{bmatrix} = \begin{bmatrix} \left(\frac{pn}{2} + \frac{qn}{2}\right)\mathbf{1}_B \\ \left(\frac{qn}{2} + \frac{pn}{2}\right)\mathbf{1}_C \end{bmatrix}$$

$$= \frac{(p+q)n}{2} \begin{bmatrix} \mathbf{1}_B \\ \mathbf{1}_C \end{bmatrix}$$



$$\mathbb{E}[\mathbf{A}]\mathbf{1} = \frac{(p+q)n}{2}\mathbf{1}.$$

So $\mathbf{1}$ is an eigenvector with eigenvalue $\frac{(p+q)n}{2}$.

Expected Adjacency Spectrum

Now let $\chi_{B,C} = \begin{bmatrix} \mathbf{1}_B \\ -\mathbf{1}_C \end{bmatrix}$ be the perfect cut vector.

Claim: $\chi_{B,C}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

Expected Adjacency Spectrum

Now let $\chi_{B,C} = \begin{bmatrix} \mathbf{1}_B \\ -\mathbf{1}_C \end{bmatrix}$ be the perfect cut vector.

Claim: $\chi_{B,C}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

$$\left[\begin{array}{ccc|ccc} p & \cdots & p & q & \cdots & q \\ \hline q & \cdots & q & p & \cdots & p \end{array}\right] \begin{bmatrix} \mathbf{1}_B \\ -\mathbf{1}_C \end{bmatrix} = \begin{bmatrix} \frac{(p-q)n}{2}\mathbf{1}_B \\ -\frac{(p-q)n}{2}\mathbf{1}_C \end{bmatrix}$$

$$= \frac{(p-q)n}{2}\chi_{B,C}.$$

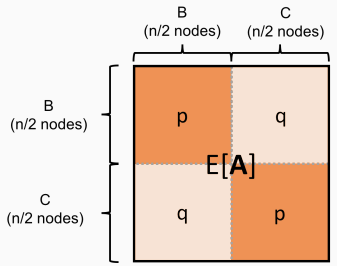
Expected Adjacency Spectrum

Now let $\chi_{B,C} = \begin{bmatrix} \mathbf{1}_B \\ -\mathbf{1}_C \end{bmatrix}$ be the perfect cut vector.

Claim: $\chi_{B,C}$ is an eigenvector of $\mathbb{E}[\mathbf{A}]$.

$$\left[\begin{array}{ccc|ccc} p & \cdots & p & q & \cdots & q \\ \hline q & \cdots & q & p & \cdots & p \end{array}\right] \begin{bmatrix} \mathbf{1}_B \\ -\mathbf{1}_C \end{bmatrix} = \begin{bmatrix} \frac{(p-q)n}{2}\mathbf{1}_B \\ -\frac{(p-q)n}{2}\mathbf{1}_C \end{bmatrix}$$

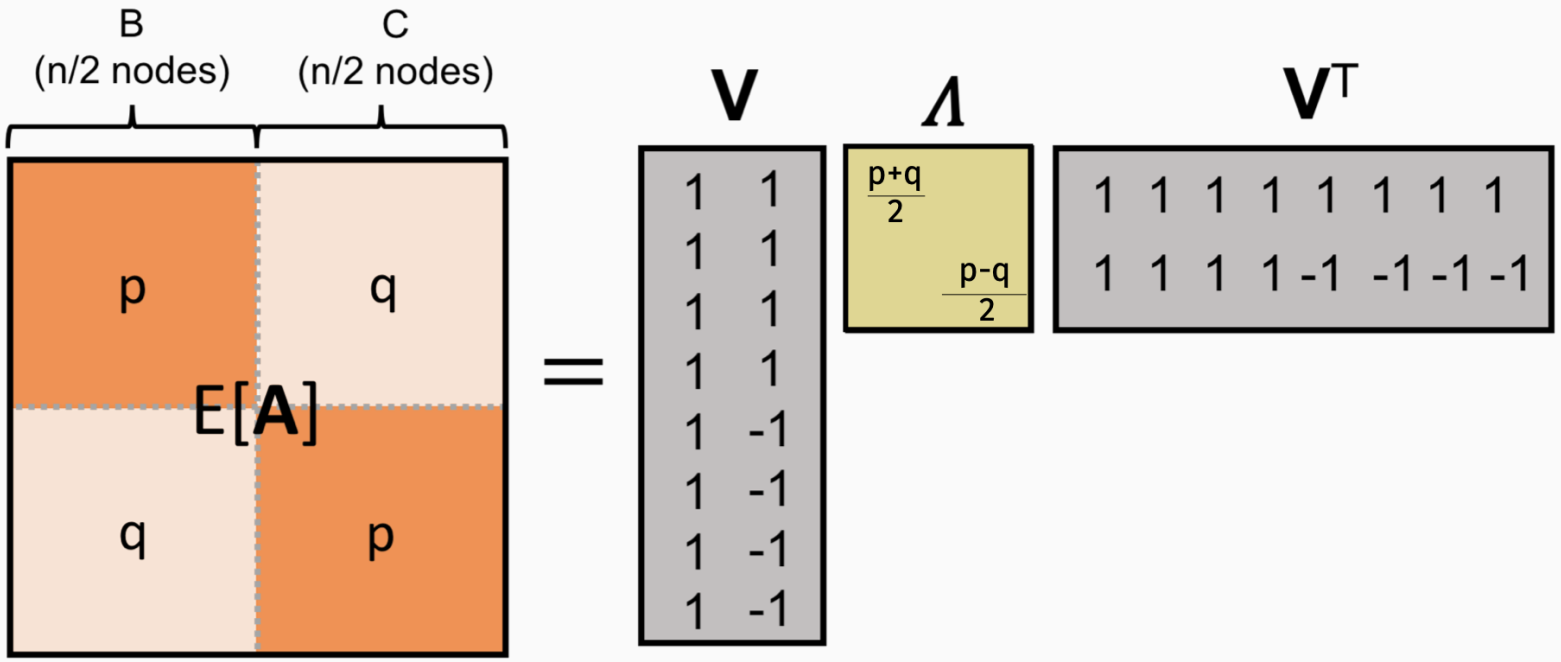
$$= \frac{(p-q)n}{2}\chi_{B,C}.$$



$$\mathbb{E}[\mathbf{A}]\chi_{B,C} = \frac{(p-q)n}{2}\chi_{B,C}.$$

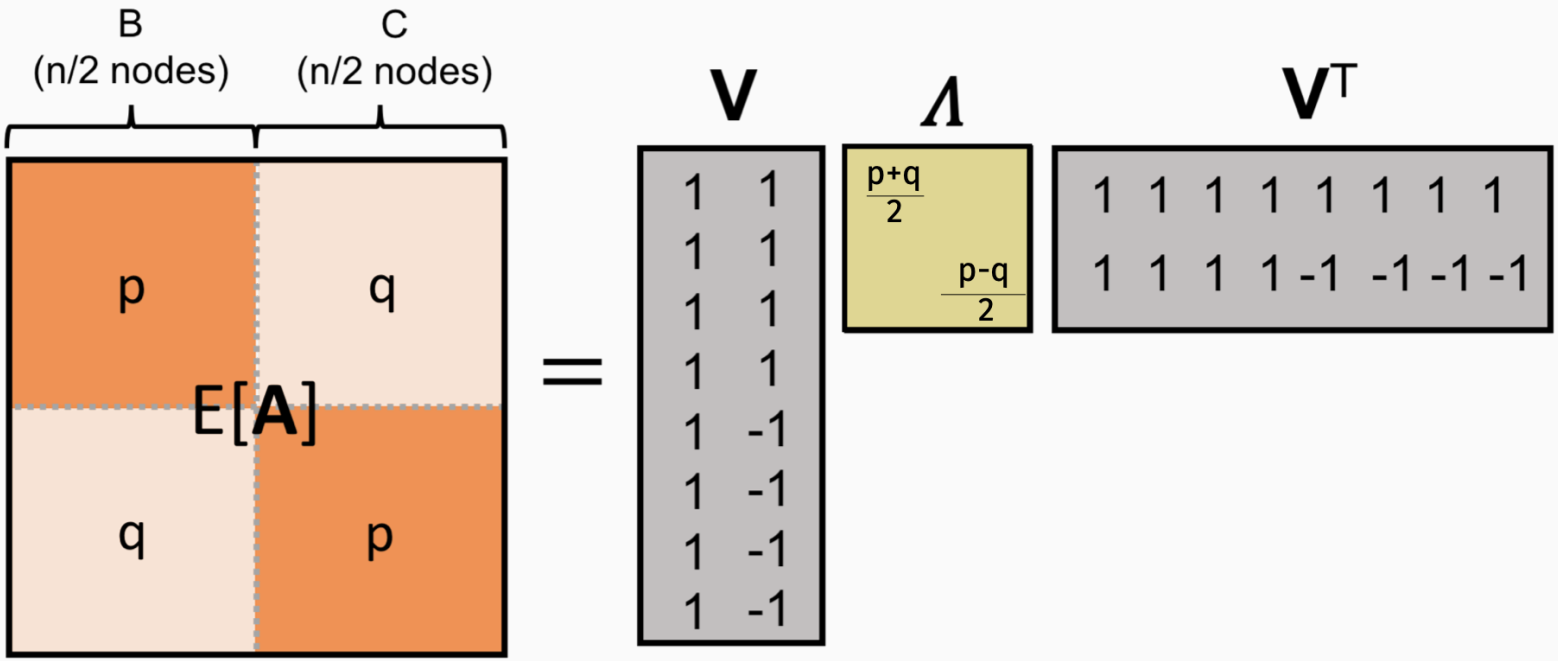
The signs separate the two communities.

Expected Adjacency Spectrum



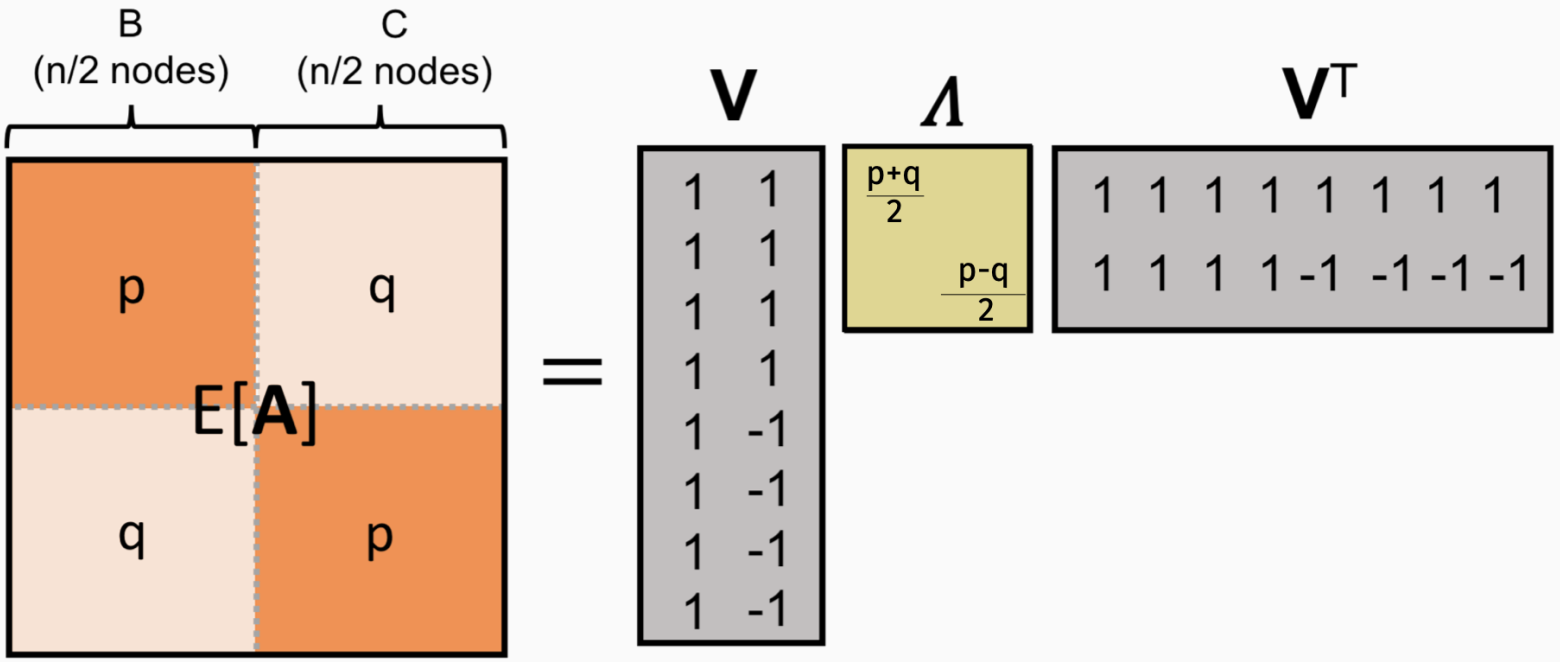
- $\bar{\mathbf{v}}_1 \sim \mathbf{1}$ with eigenvalue $\lambda_1 = \frac{(p+q)n}{2}$.

Expected Adjacency Spectrum



- $\bar{\mathbf{v}}_1 \sim \mathbf{1}$ with eigenvalue $\lambda_1 = \frac{(p+q)n}{2}$.
- $\bar{\mathbf{v}}_2 \sim \chi_{B,C}$ with eigenvalue $\lambda_2 = \frac{(p-q)n}{2}$.

Expected Adjacency Spectrum



- $\bar{\mathbf{v}}_1 \sim \mathbf{1}$ with eigenvalue $\lambda_1 = \frac{(p+q)n}{2}$.
- $\bar{\mathbf{v}}_2 \sim \chi_{B,C}$ with eigenvalue $\lambda_2 = \frac{(p-q)n}{2}$.

If we compute $\bar{\mathbf{v}}_2$, we exactly recover the communities B and C.

Expected Laplacian Spectrum

Upshot: The second smallest eigenvector of $\mathbb{E}[\mathbf{L}]$ is the community indicator $\chi_{B,C}$.

Expected Laplacian Spectrum

Upshot: The second smallest eigenvector of $\mathbb{E}[\mathbf{L}]$ is the community indicator $\chi_{B,C}$.

If the random graph exactly equaled its expectation, spectral partitioning would recover the communities perfectly.

Expected Laplacian Spectrum

Upshot: The second smallest eigenvector of $\mathbb{E}[\mathbf{L}]$ is the community indicator $\chi_{B,C}$.

If the random graph exactly equaled its expectation, spectral partitioning would recover the communities perfectly.

How do we show that $\mathbf{A}$ is close to $\mathbb{E}[\mathbf{A}]$?

Matrix concentration.

Matrix Concentration

Alon, Krivelevich, Vu, 2002:

Matrix Concentration Inequality

If $p \geq O\left(\frac{\log^4 n}{n}\right)$, then with high probability

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Matrix Concentration

Alon, Krivelevich, Vu, 2002:

Matrix Concentration Inequality

If $p \geq O\left(\frac{\log^4 n}{n}\right)$, then with high probability

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Recall:

$$\|\mathbf{M}\|_2 = \max_{\|\mathbf{z}\|_2=1} \|\mathbf{M}\mathbf{z}\|_2.$$

Matrix Concentration

Alon, Krivelevich, Vu, 2002:

Matrix Concentration Inequality

If $p \geq O\left(\frac{\log^4 n}{n}\right)$, then with high probability

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Recall:

$$\|\mathbf{M}\|_2 = \max_{\|\mathbf{z}\|_2=1} \|\mathbf{M}\mathbf{z}\|_2.$$

This says every direction sees roughly the same adjacency action as in expectation.

Eigenvector Perturbation

We know $\mathbf{A}$ is close to $\mathbb{E}[\mathbf{A}]$ in spectral norm.

Eigenvector Perturbation

We know $\mathbf{A}$ is close to $\mathbb{E}[\mathbf{A}]$ in spectral norm.

What we need is that their second eigenvectors are close.

Eigenvector Perturbation

We know $\mathbf{A}$ is close to $\mathbb{E}[\mathbf{A}]$ in spectral norm.

What we need is that their second eigenvectors are close.

Davis-Kahan Eigenvector Perturbation Theorem

Let $\mathbf{A}, \bar{\mathbf{A}}$ be symmetric and suppose $\|\mathbf{A} - \bar{\mathbf{A}}\|_2 \leq \eta$. Then:

$$\sin\theta(\mathbf{v}_i, \bar{\mathbf{v}}_i) \leq \frac{\eta}{\min_{j \neq i} |\lambda_i - \lambda_j|},$$

where the λ_j are eigenvalues of $\bar{\mathbf{A}}$.

Eigenvector Perturbation

We know $\mathbf{A}$ is close to $\mathbb{E}[\mathbf{A}]$ in spectral norm.

What we need is that their second eigenvectors are close.

Davis-Kahan Eigenvector Perturbation Theorem

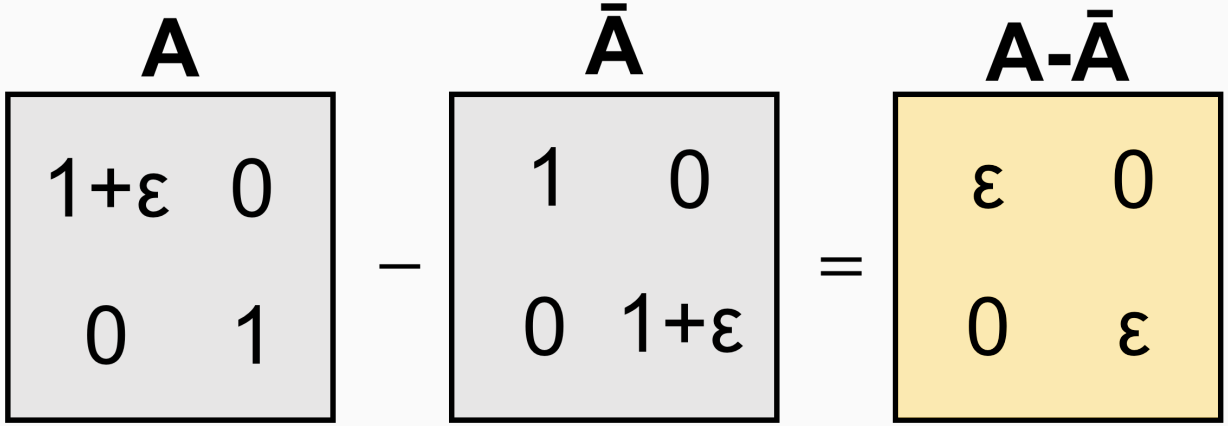
Let $\mathbf{A}, \bar{\mathbf{A}}$ be symmetric and suppose $\|\mathbf{A} - \bar{\mathbf{A}}\|_2 \leq \eta$. Then:

$$\sin\theta(\mathbf{v}_i, \bar{\mathbf{v}}_i) \leq \frac{\eta}{\min_{j \neq i} |\lambda_i - \lambda_j|},$$

where the λ_j are eigenvalues of $\bar{\mathbf{A}}$.

Large eigengap means stable eigenvectors.

Eigenvector Perturbation



Eigenvectors are only defined up to sign, so closeness always means "after a possible sign flip."

Application to Stochastic Block Model

Step 1: Matrix concentration

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Application to Stochastic Block Model

Step 1: Matrix concentration

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Step 2: Eigengap of $\mathbb{E}[\mathbf{A}]$

$$\lambda_1 = \frac{(p+q)n}{2}, \qquad \lambda_2 = \frac{(p-q)n}{2}, \qquad \lambda_i = 0 \text{ for } i \geq 3.$$

Application to Stochastic Block Model

Step 1: Matrix concentration

$$\|\mathbf{A} - \mathbb{E}[\mathbf{A}]\|_2 \leq O(\sqrt{pn}).$$

Step 2: Eigengap of $\mathbb{E}[\mathbf{A}]$

$$\lambda_1 = \frac{(p+q)n}{2}, \qquad \lambda_2 = \frac{(p-q)n}{2}, \qquad \lambda_i = 0 \text{ for } i \geq 3.$$

For $\bar{\mathbf{v}}_2$, the relevant gap is:

$$\min_{j \neq 2} |\lambda_2 - \lambda_j| = \min\left(qn, \frac{(p-q)n}{2}\right).$$

Application to Stochastic Block Model

Assume the smaller gap is $\frac{(p-q)n}{2}$.

By Davis-Kahan:

$$\begin{aligned}\sin\theta(\mathbf{v}_2, \bar{\mathbf{v}}_2) &\leq \frac{O(\sqrt{pn})}{(p-q)n/2} \\ &= O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right).\end{aligned}$$

Application to Stochastic Block Model

Assume the smaller gap is $\frac{(p-q)n}{2}$.

By Davis-Kahan:

$$\begin{aligned}\sin\theta(\mathbf{v}_2, \bar{\mathbf{v}}_2) &\leq \frac{O(\sqrt{pn})}{(p-q)n/2} \\ &= O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right).\end{aligned}$$

So the computed eigenvector $\mathbf{v}_2$ is close to the ideal community vector $\bar{\mathbf{v}}_2$.

A slightly sharper analysis can remove the extra qn gap condition.

Application to Stochastic Block Model

From the angle bound, one can show:

$$\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right).$$

Application to Stochastic Block Model

From the angle bound, one can show:

$$\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right).$$

Recall:

$$\bar{\mathbf{v}}_2 = \frac{1}{\sqrt{n}}\boldsymbol{\chi}_{B,C}.$$

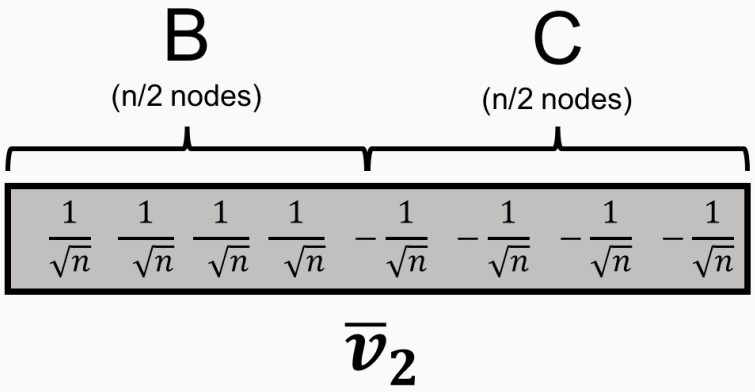
Application to Stochastic Block Model

From the angle bound, one can show:

$$\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right).$$

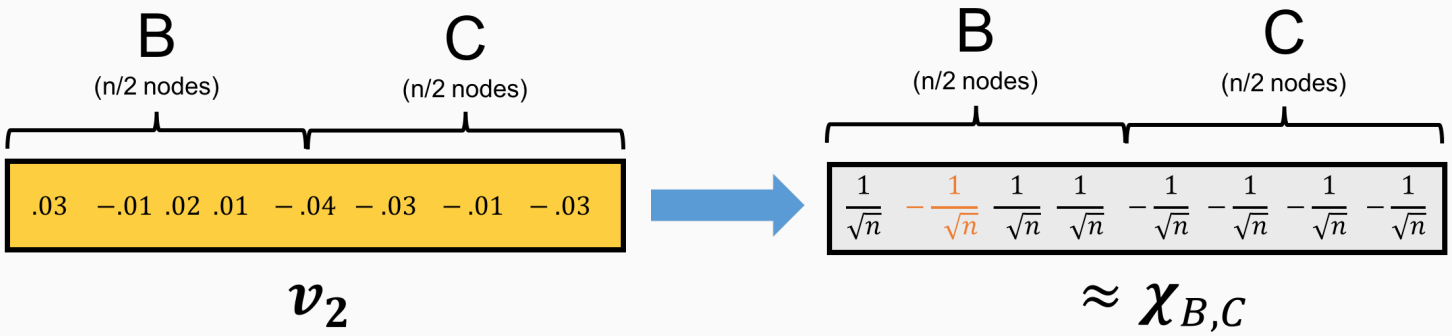
Recall:

$$\bar{\mathbf{v}}_2 = \frac{1}{\sqrt{n}}\chi_{B,C}.$$



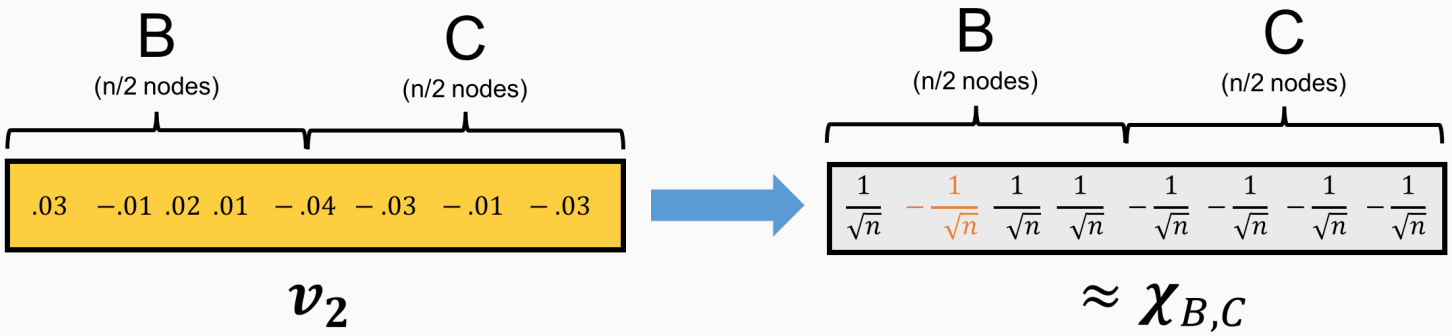
Application to Stochastic Block Model

We partition by taking signs of $\mathbf{v}_2$.



Application to Stochastic Block Model

We partition by taking signs of $\mathbf{v}_2$.



Every node where $v_2(i)$ and $\bar{v}_2(i)$ differ in sign contributes at least $\frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.

Application to Stochastic Block Model

Main argument:

- Each sign mistake contributes at least $\frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.

Application to Stochastic Block Model

Main argument:

- Each sign mistake contributes at least $\frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.
- We know $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$.

Application to Stochastic Block Model

Main argument:

- Each sign mistake contributes at least $\frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.
- We know $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$.
- Therefore the number of sign mistakes is at most:

$$O\left(\frac{p}{(p-q)^2}\right).$$

Application to Stochastic Block Model

Upshot: Spectral partitioning correctly classifies all but

$$O\left(\frac{p}{(p-q)^2}\right)$$

nodes.

Application to Stochastic Block Model

Upshot: Spectral partitioning correctly classifies all but

$$O\left(\frac{p}{(p-q)^2}\right)$$

nodes.

Hard case: Suppose $q = .8p$. Then:

$$\frac{p}{(p-q)^2} = \frac{25}{p}.$$

Application to Stochastic Block Model

Upshot: Spectral partitioning correctly classifies all but

$$O\left(\frac{p}{(p-q)^2}\right)$$

nodes.

Hard case: Suppose $q = .8p$. Then:

$$\frac{p}{(p-q)^2} = \frac{25}{p}.$$

If $p = 250/n$, then the method correctly classifies roughly 90% of nodes.