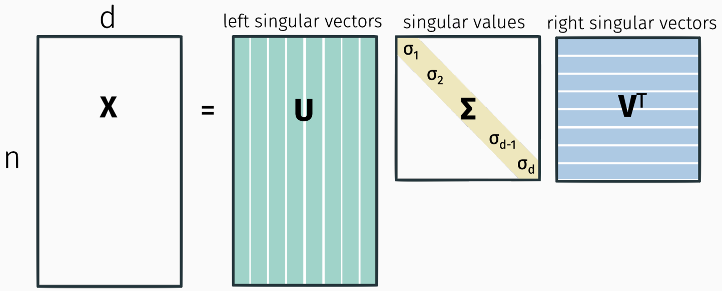
CS-GY 6763: Lecture 12
Power Method, Krylov Subspace Methods, Spectral Graph Partitioning

NYU Tandon School of Engineering, Prof. Christopher Musco

SINGULAR VALUE DECOMPOSITION

One of the most fundamental results in linear algebra.

Any matrix **X** can be written:

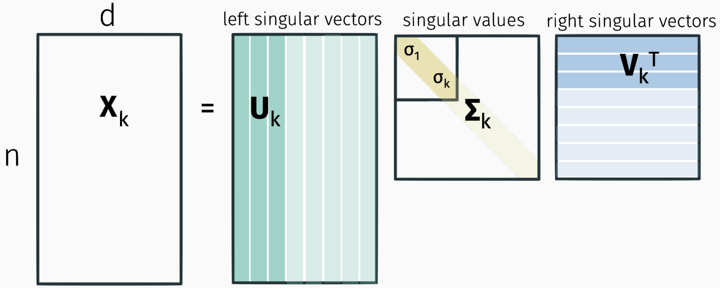


Where $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$.

Singular values are unique. Factors are not. E.g. would still get a valid SVD by multiplying both i^{th} column of **V** and **U** by -1.

PARTIAL SVD

Key result: Can find the best low-rank approximation from the singular value decomposition.



$$U_k = \underset{\text{orthogonal } Z \in \mathbb{R}^{d\times k}}{\arg\min} \|X - ZZ^T X\|_F^2$$

$$V_k = \underset{\text{orthogonal } W \in \mathbb{R}^{d\times k}}{\arg\min} \|X - XWW^T\|_F^2$$

COMPUTING THE SVD

Suffices to compute right singular vectors $\mathbf{V}$:

- Compute $\mathbf{X}^T\mathbf{X}$.
- Find eigendecomposition $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T = \mathbf{X}^T\mathbf{X}$ using e.g. QR algorithm.
- Compute $\mathbf{L} = \mathbf{X}\mathbf{V}$. Set $\sigma_i = \|\mathbf{L}_i\|_2$ and $\mathbf{U}_i = \mathbf{L}_i / \|\mathbf{L}_i\|_2$.

Total runtime $\approx$

computing the svd (faster)

How to go faster?

- Compute approximate solution.
- Only compute top k singular vectors/values.
- Iterative algorithms achieve runtime $\approx O(ndk)$ vs. $O(nd^2)$ time.
  - **Krylov subspace methods** like the Lanczos method are most commonly used in practice.
  - **Power method** is the simplest Krylov subspace method, and still works very well.

POWER METHOD

Today: What about when $k = 1$?

Goal: Find some $\mathbf{z} \approx \mathbf{v}_1$.

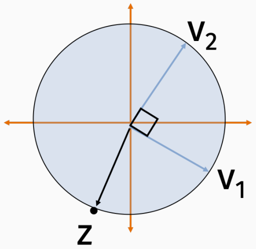
Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$ with SVD $\mathbf{U\Sigma V}^T$.

Power method:

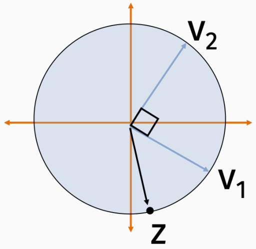
- Choose $\mathbf{z}^{(0)}$ randomly. $\mathbf{z}_0 \sim \mathcal{N}(0,1)$.
- $\mathbf{z}^{(0)} = \mathbf{z}^{(0)}/\|\mathbf{z}^{(0)}\|_2$
- For $i = 1, \ldots, T$
    - $\mathbf{z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{z}^{(i-1)})$
    - $n_i = \|\mathbf{z}^{(i)}\|_2$
    - $\mathbf{z}^{(i)} = \mathbf{z}^{(i)}/n_i$

  Return $\mathbf{z}^{(T)}$

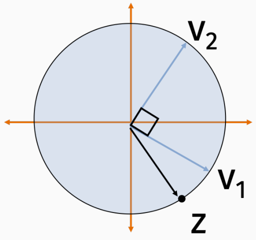
0 iterations



1 iterations



2 iterations



POWER METHOD FORMAL CONVERGENCE

Theorem (Basic Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be parameter capturing the “gap” between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, we have either:

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

Total runtime: $O\left(nd \cdot \frac{\log d/\epsilon}{\gamma}\right)$

ONE STEP ANALYSIS OF POWER METHOD

Write $\mathbf{z}^{(i)}$ in the right singular vector basis:

$$\begin{aligned}
\mathbf{z}^{(0)} &= c_1^{(0)}\mathbf{v}_1 + c_2^{(0)}\mathbf{v}_2 + \ldots + c_d^{(0)}\mathbf{v}_d \\
\mathbf{z}^{(1)} &= c_1^{(1)}\mathbf{v}_1 + c_2^{(1)}\mathbf{v}_2 + \ldots + c_d^{(1)}\mathbf{v}_d \\
&\vdots \\
\mathbf{z}^{(i)} &= c_1^{(i)}\mathbf{v}_1 + c_2^{(i)}\mathbf{v}_2 + \ldots + c_d^{(i)}\mathbf{v}_d
\end{aligned}$$

Note: $[c_1^{(i)}, \ldots, c_d^{(i)}] = \mathbf{c}^{(i)} = \mathbf{V}^T\mathbf{z}^{(i)}$.

Also: Since $\mathbf{V}$ is orthogonal and $\|\mathbf{z}^{(i)}\|_2 = 1$, $\|\mathbf{c}^{(i)}\|_2^2 = 1$.

ONE STEP ANALYSIS OF POWER METHOD

Claim: After update $\mathbf{z}^{(i)} = \frac{1}{n_i}\mathbf{X}^T\mathbf{X}\mathbf{z}^{(i-1)}$,

$$c_j^{(i)} = \frac{1}{n_i}\sigma_j^2 c_j^{(i-1)}$$

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

Equivalently: $\mathbf{c}^{(i)} = \frac{1}{n_i}\mathbf{\Sigma}^2\mathbf{c}^{(i-1)}$.

MULTI-STEP ANALYSIS OF POWER METHOD

Claim: After T updates:

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

Let $\alpha_j = \frac{1}{\prod_{i=1}^{T} n_i} c_j^{(0)}\sigma_j^{2T}$. **Goal:** Show that $\alpha_j \ll \alpha_1$ for all $j \neq 1$.

POWER METHOD FORMAL CONVERGENCE

Since $\mathbf{z}^{(T)}$ is a unit vector, $\sum_{i=1}^{d} \alpha_i^2 = 1$.

If we can prove that $\left|\frac{\alpha_j}{\alpha_1}\right| \leq \sqrt{\frac{\epsilon}{2d}}$ then we will have that $\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2^2 \leq \epsilon$.

$$\alpha_j^2 \leq \alpha_1^2 \cdot \frac{\epsilon}{2d}$$

$$1 = \alpha_1^2 + \sum_{j=2}^{d} \alpha_d^2 \leq \alpha_1^2 + \frac{\epsilon}{2}$$

$$\alpha_1^2 \geq 1 - \frac{\epsilon}{2}$$

$$|\alpha_1| \geq 1 - \frac{\epsilon}{2}$$

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2^2 = 2 - 2\langle \mathbf{v}_1, \mathbf{z}^{(T)} \rangle \leq \epsilon$$

POWER METHOD FORMAL CONVERGENCE

Want to prove that after $T = O(\log(d/\epsilon)/\gamma)$ steps, $\left|\frac{\alpha_j}{\alpha_1}\right| \leq \sqrt{\frac{\epsilon}{2d}}$ where $\alpha_j = \frac{1}{\prod_{i=1}^{T} n_i} c_j^{(0)} \sigma_j^{2T}$. Recall that $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_2}$.

Assumption: Starting coeff. on first eigenvector is not too small:

$$\left|c_1^{(0)}\right| \geq O\left(\frac{1}{\sqrt{d}}\right).$$

We will prove shortly that this holds with probability 99/100.

$$\frac{|\alpha_j|}{|\alpha_1|} = \frac{\sigma_j^{2T}}{\sigma_1^{2T}} \cdot \frac{|c_j^{(0)}|}{|c_1^{(0)}|} \leq$$

Need $T =$

STARTING COEFFICIENT ANALYSIS

Need to prove: Starting coefficient on first eigenvector is not too small. I.e., with probability 99/100,

$$\left|c_1^{(0)}\right| \geq O\left(\frac{1}{\sqrt{d}}\right).$$

Prove using Gaussian $\underline{\text{anti}}$-concentration. First use rotational invariance of Gaussian:

$$\mathbf{c}^{(0)} = \frac{\mathbf{V}^T\mathbf{z}^{(0)}}{\|\mathbf{z}^{(0)}\|_2} = \frac{\mathbf{V}^T\mathbf{z}^{(0)}}{\|\mathbf{V}^T\mathbf{z}^{(0)}\|_2} \sim \frac{\mathbf{g}}{\|\mathbf{g}\|_2},$$

where $\mathbf{g} \sim \mathcal{N}(0,1)^d$.

STARTING COEFFICIENT ANALYSIS

Need to show that with high probability, first entry of $\frac{\mathbf{g}}{\|\mathbf{g}\|_2} \geq c \cdot \frac{1}{\sqrt{d}}$.

Part 1: With super high probability (e.g. 99/100),

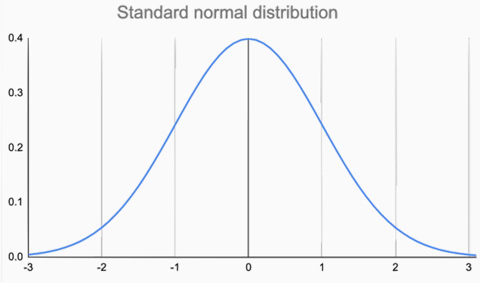
$$\|\mathbf{g}\|_2^2 \leq$$

STARTING COEFFICIENT ANALYSIS

Need to show that with high probability, the magnitude of the first entry of $\mathbf{g} \geq c$ for a constant c. Think e.g. $c = 1/100$.

Part 2: With probablility $1 - O(\alpha)$,

$$|g_1| \geq \alpha.$$



POWER METHOD FORMAL CONVERGENCE

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

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be parameter capturing the "gap" between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, we have either:

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

The method truly won't converge if γ is very small. Consider extreme case when $\gamma = 0$.

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

power method – no gap dependence

Theorem (Gapless Power Method Convergence)

If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ steps, we obtain a $\mathbf{z}$ satisfying:

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

Intuition: For a good low-rank approximation, we don't actually need to converge to $\mathbf{v}_1$ if σ_1 and σ_2 are the same or very close. Would suffice to return either $\mathbf{v}_1$ or $\mathbf{v}_2$, or some linear combination of the two.

generalizations to larger k

- Block Power Method aka Simultaneous Iteration aka Subspace Iteration aka Orthogonal Iteration

Power method:

- Choose $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathbf{Z}_0 = \mathrm{orth}(\mathbf{G})$.
- For $i = 1, \ldots, T$
    - $\mathbf{Z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{Z}^{(i-1)})$
    - $\mathbf{Z}^{(i)} = \mathrm{orth}(\mathbf{Z}^{(i)})$

  Return $\mathbf{Z}^{(T)}$

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

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

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

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.

Idea behind Krlyov 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 $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.
    - Find <u>best</u> 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.

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 returns:

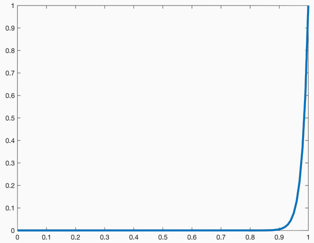
$$\mathbf{v}_{x^T}.$$

Lanczos method returns $\mathbf{v}_{p*}$ where:

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

LANCZOS METHOD ANALYSIS

Claim: 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 $[0,1]$.



$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_{p*}\mathbf{v}_{p*}^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_{\hat{p}}\mathbf{v}_{\hat{p}}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_{x^T}\mathbf{v}_{x^T}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

Runtime: $O\left(\frac{\log(d/\epsilon)}{\sqrt{\epsilon}} \cdot \text{nnz}(\mathbf{X})\right)$ vs. $O\left(\frac{\log(d/\epsilon)}{\epsilon} \cdot \text{nnz}(\mathbf{X})\right)$

GENERALIZATIONS TO LARGER k

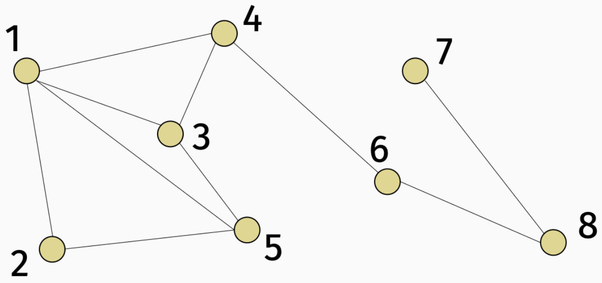
- Block Krylov methods
- Let $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathcal{K}_q = \left[\mathbf{G}, \left(\mathbf{X}^T\mathbf{X}\right) \cdot \mathbf{G}, \left(\mathbf{X}^T\mathbf{X}\right)^2 \cdot \mathbf{G}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{G}\right]$

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

BREAK

SPECTRAL GRAPH THEORY

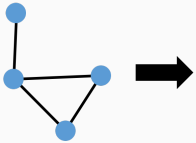
Main idea: Understand graph data by constructing natural matrix representations, and studying that matrix's spectrum (eigenvalues/eigenvectors).



For now assume $G = (V, E)$ is an undirected, unweighted graph with n nodes.

MATRIX REPRESENTATIONS OF GRAPHS

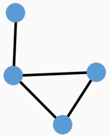
Two most common representations: $n \times n$ adjacency matrix $\mathbf{A}$ and graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix.



$$\overset{\mathbf{D}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}} - \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}} = \overset{\mathbf{L}}{\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}}$$

Also common to look at normalized versions of both of these: $\bar{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\bar{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.

THE LAPLACIAN VIEW



$$\Rightarrow \overset{\mathbf{D}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}} - \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}} = \overset{\mathbf{L}}{\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}}$$

$\mathbf{L} = \mathbf{B}^T\mathbf{B}$ where B is the signed “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.

THE LAPLACIAN VIEW

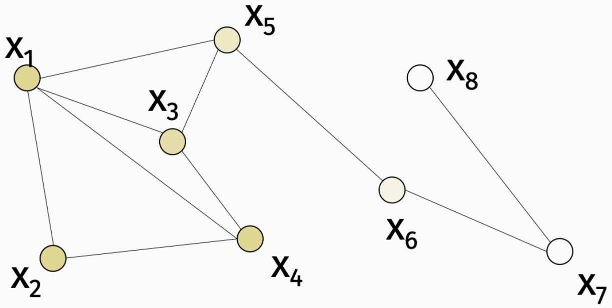
Conclusions from $\mathbf{L} = \mathbf{B}^T\mathbf{B}$

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

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

THE LAPLACIAN VIEW

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

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

LARGEST LAPLACIAN EIGENVECTOR

Courant–Fischer min-max principle

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

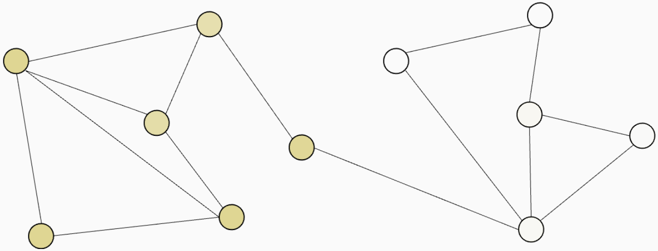
$$\begin{aligned}
\mathbf{v}_1 &= \mathop{\arg\max}_{\|\mathbf{v}\|=1} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_2 &= \mathop{\arg\max}_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1} \mathbf{v}^T \mathbf{L} \mathbf{v} \\
\mathbf{v}_3 &= \mathop{\arg\max}_{\|\mathbf{v}\|=1, \mathbf{v} \perp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

THE LAPLACIAN VIEW

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

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

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



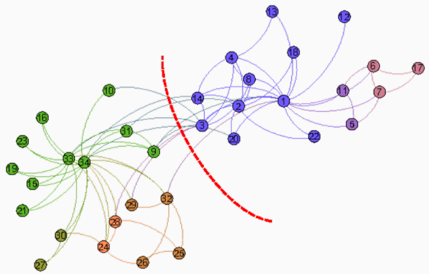
SPECTRAL GRAPH PARTITIONING

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

SOCIAL NETWORKS IN THE 1970S

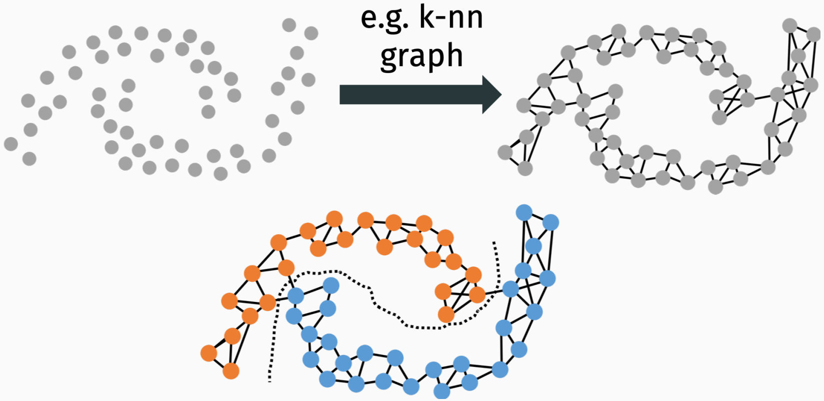
Wayne W. Zachary (1977). An Information Flow Model for Conflict and Fission in Small Groups.

"At the beginning of the study there was an incipient conflict between the club president, John A., and Mr. Hi over the price of karate lessons. Mr. Hi, who wished to raise prices, claimed the authority to set his own lesson fees, since he was the instructor. John A., who wished to stabilize prices, claimed the authority to set the lesson fees since he was the club's chief administrator. As time passed the entire club became divided over this issue, and the conflict became translated into ideological terms by most club members."

Zachary constructed a social network by hand and used a minimum cut algorithm to correctly predict who sided with who in the conflict. Beautiful paper – definitely worth checking out!

SPECTRAL CLUSTERING

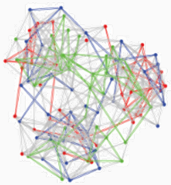
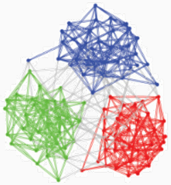
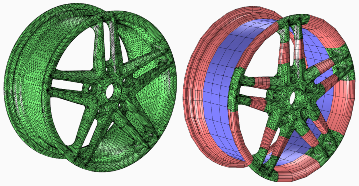
Idea: Construct synthetic graph for data that is hard to cluster.



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 way's to formalize Zachary's problem:

β-Balanced Cut:

$$\min_S \text{cut}(S, V \setminus S) \quad \text{such that} \quad \min(|S|, |V \setminus S|) \geq \beta \cdot n \text{ for } \beta \leq .5$$

Sparsest Cut:

$$\min_S \frac{\text{cut}(S, V \setminus S)}{\min(|S|, |V \setminus S|)}$$

All natural formalizations lead to NP-hard problems. Lots of interest in designing polynomial time approximation algorithms, but tend to be slow. In practice, much simpler methods based on the graph spectrum are used.

Spectral methods run no more than $O(n^3)$ time (must faster if you use iterative methods for computing eigenvectors).

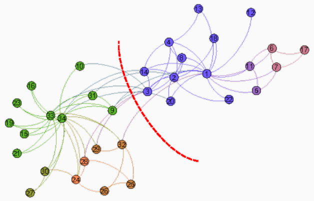
SPECTRAL GRAPH PARTITIONING

Basic spectral clustering method:

- Compute <u>second</u> smallest eigenvector of graph, $\mathbf{v}_{n-1}$.
- $\mathbf{v}_{n-1}$ has an entry for every node i in the graph.
- If the i^{th} entry is positive, put node i in T.
- Otherwise if the i^{th} entry is negative, put i in S.

This shouldn't make much sense yet! We will see that is a "relax and round" algorithm in disguise.

THE LAPLACIAN VIEW

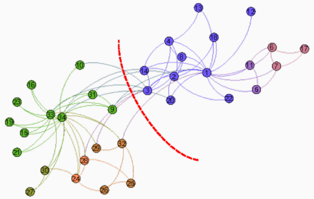

(a) Zachary Karate Club Graph

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

- $\mathbf{c}^T L \mathbf{c} = 4 \cdot cut(S, T)$.
- $\mathbf{c}^T \mathbf{1} = |T| - |S|$.

Want to minimize both $\mathbf{c}^T L \mathbf{c}$ (cut size) and $|\mathbf{c}^T \mathbf{1}|$ (imbalance).

THE LAPLACIAN VIEW



(a) Zachary Karate Club Graph

Equivalent formulation if we divide everything by $\sqrt{n}$ so that $\mathbf{c}$ has norm 1. Then $\mathbf{c} \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n$ and:

- $\mathbf{c}^T L \mathbf{c} = \frac{4}{n} \cdot cut(S, T)$.
- $\mathbf{c}^T \mathbf{1} = \frac{1}{\sqrt{n}}(|T| - |S|)$.

Want to minimize both $\mathbf{c}^T L \mathbf{c}$ (cut size) and $|\mathbf{c}^T \mathbf{1}|$ (imbalance).

RELAX AND ROUND

Perfectly balanced balanced cut problem:

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

Relaxed perfectly balanced balanced cut problem:

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

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

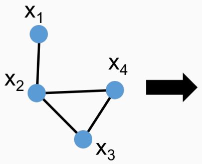
Approach: Relax, find $\mathbf{v}_{n-1}$, then round back to a vector with $-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}$ entries.

SMALLEST LAPLACIAN EIGENVECTOR

Claim: The smallest eigenvector/singular vector of any graph Laplacian **L** always equals:

$$\mathbf{v}_n = \underset{v \in \mathbb{R}^n \text{ with } \|\mathbf{v}\|=1}{\arg\min} \mathbf{v}^T L \mathbf{v} = \frac{1}{\sqrt{n}} \cdot \mathbf{1}$$

with $\mathbf{v}_n^T L \mathbf{v}_n = 0$.



$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

SECOND SMALLEST LAPLACIAN EIGENVECTOR

By Courant-Fischer, $\mathbf{v}_{n-1}$ is given by:

$$\mathbf{v}_{n-1} = \underset{\|\mathbf{v}\|=1,\ \mathbf{v}_n^T\mathbf{v}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

which is equivalent to

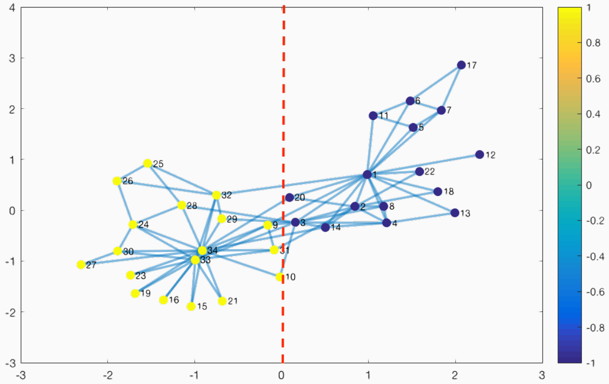
$$\mathbf{v}_{n-1} = \underset{\|\mathbf{v}\|=1,\ \mathbf{1}^T\mathbf{v}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}.$$

CUTTING WITH THE SECOND LAPLACIAN EIGENVECTOR

Final relax and round algorithm: Compute

$$\mathbf{v}_{n-1} = \underset{v \in \mathbb{R}^n \text{ with } \|\mathbf{v}\|=1,\ \mathbf{v}^T\mathbf{1}=0}{\arg\min} \mathbf{v}^T L \mathbf{v}$$

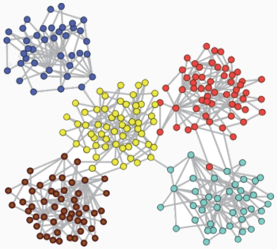
Set S to be all nodes with $\mathbf{v}_{n-1}(i) < 0$, and T to be all with $\mathbf{v}_{n-1}(i) \geq 0$. I.e. set $\mathbf{c} = \text{sign}(\mathbf{v}_{n-1})$



SPECTRAL PARTITIONING IN PRACTICE

Lots of different variants used in practice:

- Often do some sort of normalization of edge weights by degree. E.g. the Shi-Malik normalized cuts algorithm use the normalized Laplacian $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.
- Different methods for how to choose the threshold to partition the second smallest eigenvector.
- Lots of variants to split the graph into more than two parts.



SPECTRAL PARTITIONING IN PRACTICE

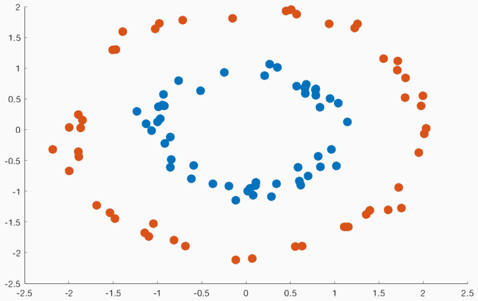
Multiway spectral partitioning:

- Compute smallest ℓ eigenvectors $\mathbf{v}_{n-1}, \ldots, \mathbf{v}_{n-\ell}$ of $\mathbf{L}$.
- Represent each node by its corresponding row in $\mathbf{V} \in \mathbb{R}^{n \times \ell}$ whose rows are $\mathbf{v}_{n-1}, \ldots \mathbf{v}_{n-\ell}$.
- Cluster these rows using k-means clustering (or really any clustering method).
- Often we choose $\ell = k$, but not necessarily.

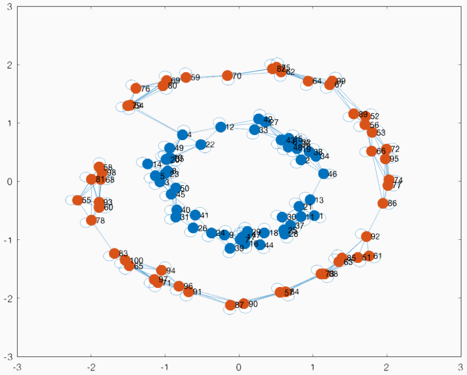
Let $\tilde{\mathbf{x}}_i \in \mathbb{R}^{\ell}$ denote this embedding for a particullar node i.

LAPLACIAN EMBEDDING

Original Data: (not linearly separable)

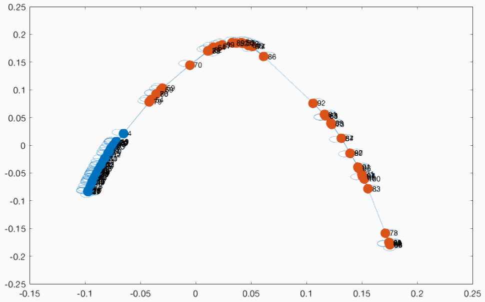


k-Nearest Neighbors Graph:



LAPLACIAN EMBEDDING

Embedding with eigenvectors v_{n-1}, v_{n-2}: (linearly separable)

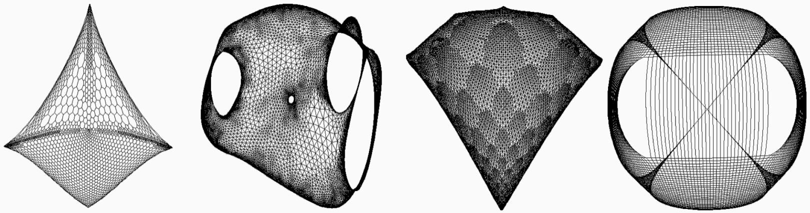


WHY DOES THIS WORK?

Formally, this choice of embedding minimizes:

$$\sum_{i,j \in E} \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2$$

I.e., we explicitly encourages nodes connected by an edge to be placed in nearby locations in the embedding.



Also useful e.g., in graph drawing.

GENERATIVE MODELS

So far: Showed that spectral clustering partitions a graph along a small cut between large pieces.

- No formal guarantee on the 'quality' of the partitioning.
- Can fail for worst case input graphs.

Common approach: Design a natural generative model that produces random but realistic inputs and analyze how the algorithm performs on inputs drawn from this model.

- Very common in algorithm design and analysis. Great way to start approaching a problem. Often our best way to understand why some algorithms "just work" in practice.
- Similar approach to Bayesian modeling in machine learning.

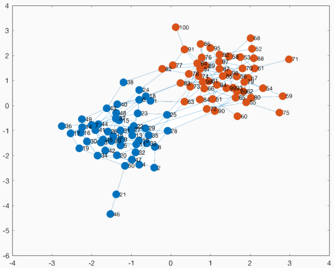
STOCHASTIC BLOCK MODEL

Ideas for a generative model for **social network graphs** that would allow us to understand partitioning?

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, each with $n/2$ nodes.

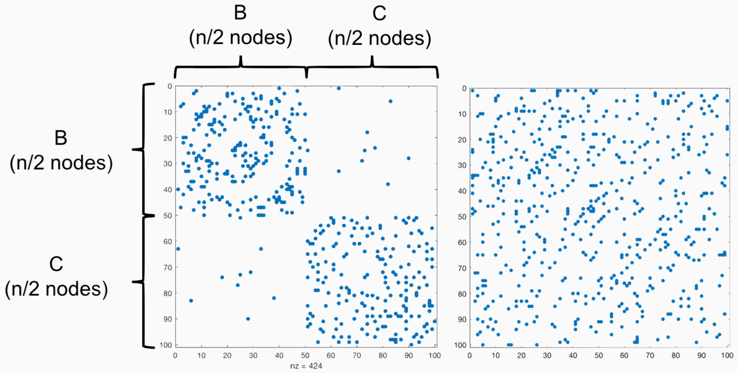
- Any two nodes in the same group are connected with probability p (including self-loops).
- Any two nodes in different groups are connected with prob. $q < p$.



LINEAR ALGEBRAIC VIEW

Let G be a stochastic block model graph drawn from $G_n(p, q)$.

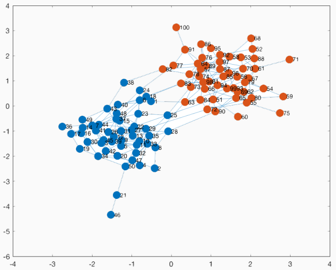
- Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ denote the adjacency matrix of G.



Note that we are arbitrarily ordering the nodes in A by group. In reality A would look “scrambled” as on the right.

STOCHASTIC BLOCK MODEL

Goal is to find the "ground truth" balanced partition B, C using our standard spectal method.

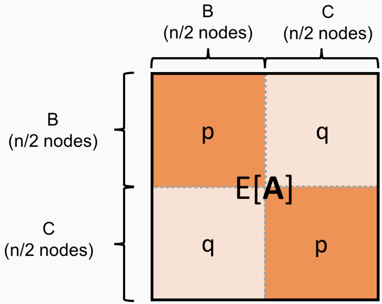


To do so, we need to understand the second smallest eigenvector of $\mathbf{L} = \mathbf{D} - \mathbf{A}$. We will start by considering the expected value of these matrices:

$$\mathbb{E}[\mathbf{L}] = \mathbb{E}[\mathbf{D}] - \mathbb{E}[\mathbf{A}].$$

EXPECTED ADJACENCY SPECTRUM

Letting G be a stochastic block model graph drawn from $G_n(p, q)$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ be its adjacency matrix. $(\mathbb{E}[\mathbf{A}])_{i,j} = p$ for i, j in same group, $(\mathbb{E}[\mathbf{A}])_{i,j} = q$ otherwise.



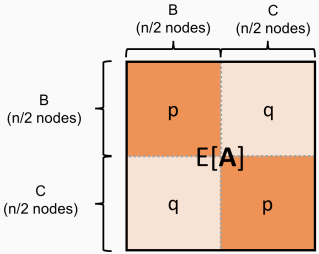
EXPECTED LAPLACIAN

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

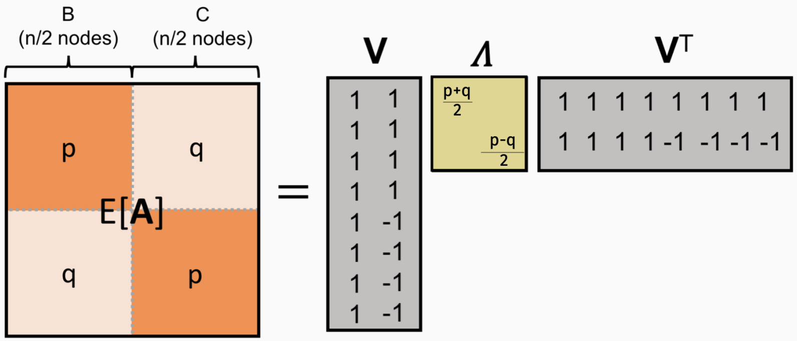
$\mathbb{E}[\mathbf{A}]$ and $\mathbb{E}[\mathbf{L}]$ have the same eigenvectors and eigenvalues are equal up to a shift/inversion. So second largest eigenvector of $\mathbb{E}[\mathbf{A}]$ is the same as the second smallest of $\mathbb{E}[\mathbf{L}]$

EXPECTED ADJACENCY SPECTRUM

Letting G be a stochastic block model graph drawn from $G_n(p,q)$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ be its adjacency matrix, what are the eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{A}]$?



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 $\bar{\mathbf{v}}_2$ then we exactly recover the communities B and C!

EXPECTED LAPLACIAN SPECTRUM

Upshot: The second smallest eigenvector of $\mathbb{E}[\mathbf{L}]$, equivalently the second largest of $\mathbb{E}[\mathbf{A}]$, is exactly $\boldsymbol{\chi}_{B,C}$ – the indicator vector for the cut between the communities.

- If the random graph G (equivilantly $\mathbf{A}$ and $\mathbf{L}$) were exactly equal to its expectation, partitioning using this eigenvector would exactly recover communities B and C.

How do we show that a matrix (e.g., $\mathbf{A}$) is close to its expectation? **Matrix concentration inequalities.**

- Analogous to scalar concentration inequalities like Markovs, Chebyshevs, Bernsteins.

MATRIX CONCENTRATION

Alon, Krivelevich, Vu, 2002:

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

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

where $\|\cdot\|_2$ is the matrix spectral norm (operator norm).

Recall that $\|\mathbf{X}\|_2 = \max_{z \in \mathbb{R}^d : \|z\|_2 = 1} \|\mathbf{X}z\|_2 = \sigma_1(\mathbf{X})$.

$\|\mathbf{A}\|_2$ is on the order of $O(p\sqrt{n})$ so another way of thinking about the right hand side is $\frac{\|\mathbf{A}\|_2}{\sqrt{p}}$. I.e. get's better with p.

EIGENVECTOR PERTURBATION

For the stochastic block model application, we want to show that the second eigenvectors of $\mathbf{A}$ and $\mathbb{E}[\mathbf{A}]$ are close. How does this relate to their difference in spectral norm?

Davis-Kahan Eigenvector Perturbation Theorem: Suppose $\mathbf{A}, \overline{\mathbf{A}} \in \mathbb{R}^{d\times d}$ are symmetric with $\|\mathbf{A} - \overline{\mathbf{A}}\|_2 \leq \epsilon$ and eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ and $\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \ldots, \bar{\mathbf{v}}_n$. Letting $\theta(\mathbf{v}_i, \bar{\mathbf{v}}_i)$ denote the angle between $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$, for all i:

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

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $\overline{\mathbf{A}}$.

We will apply with $\bar{\mathbf{A}} = \mathbb{E}[\mathbf{A}]$.

EIGENVECTOR PERTURBATION

$$\begin{array}{c}\mathbf{A}\\ \begin{bmatrix}1+\varepsilon & 0\\ 0 & 1\end{bmatrix}\end{array} - \begin{array}{c}\bar{\mathbf{A}}\\ \begin{bmatrix}1 & 0\\ 0 & 1+\varepsilon\end{bmatrix}\end{array} = \begin{array}{c}\mathbf{A}\text{-}\bar{\mathbf{A}}\\ \begin{bmatrix}\varepsilon & 0\\ 0 & \varepsilon\end{bmatrix}\end{array}$$

APPLICATION TO STOCHASTIC BLOCK MODEL

Claim 1 (Matrix Concentration): For $p \geq O\left(\frac{\log^4 n}{n}\right)$,

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

Recall: $\mathbb{E}[\mathbf{A}]$, has eigenvalues $\lambda_1 = \frac{(p+q)n}{2}$, $\lambda_2 = \frac{(p-q)n}{2}$, $\lambda_i = 0$ for $i \geq 3$.

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

Assume $\frac{(p-q)n}{2}$ will be the minimum of these two gaps.

Claim 2 (Davis-Kahan): For $p \geq O\left(\frac{\log^4 n}{n}\right)$,

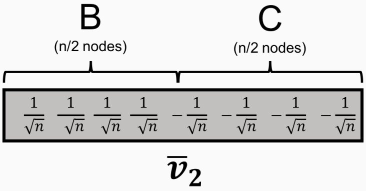
$$\sin\theta(\mathbf{v}_2, \bar{\mathbf{v}}_2) \leq \frac{O(\sqrt{pn})}{\min_{j\neq i}|\lambda_i - \lambda_j|} \leq \frac{O(\sqrt{pn})}{(p-q)n/2} = O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right)$$

(A slightly trickier analysis can remove the qn term entirely.)

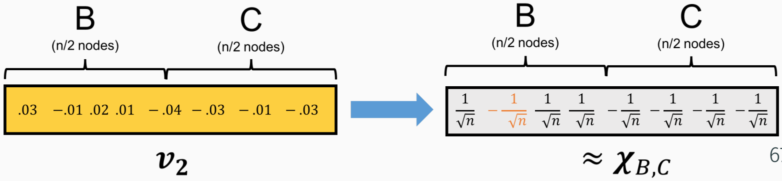
APPLICATION TO STOCHASTIC BLOCK MODEL

So far: $\sin\theta(\mathbf{v}_2, \bar{\mathbf{v}}_2) \leq O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right)$. What does this give us?

- Can show that this implies $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$ (exercise).
- $\bar{\mathbf{v}}_2$ is $\frac{1}{\sqrt{n}}\chi_{B,C}$: the community indicator vector.


B
(n/2 nodes)
C
(n/2 nodes)
$\frac{1}{\sqrt{n}}$ $\frac{1}{\sqrt{n}}$ $\frac{1}{\sqrt{n}}$ $\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$
$\bar{v}_2$

- We want to show that $\text{sign}(\mathbf{v}_2)$ and $\bar{\mathbf{v}}_2$ are close. They only differ at locations where $\mathbf{v}_2$ and $\bar{\mathbf{v}}_2$ differ in sign.


B
(n/2 nodes)
C
(n/2 nodes)
.03 −.01 .02 .01 −.04 −.03 −.01 −.03
v_2
B
(n/2 nodes)
C
(n/2 nodes)
$\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $\frac{1}{\sqrt{n}}$ $\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$ $-\frac{1}{\sqrt{n}}$
$\approx \chi_{B,C}$

APPLICATION TO STOCHASTIC BLOCK MODEL

Main argument:

- Every i where $v_2(i), \bar{v}_2(i)$ differ in sign contributes $\geq \frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.
- We know that $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$.
- So $\mathbf{v}_2$ and $\bar{\mathbf{v}}_2$ differ in sign in at most $O\left(\frac{p}{(p-q)^2}\right)$ positions.

APPLICATION TO STOCHASTIC BLOCK MODEL

Upshot: If G is a stochastic block model graph with adjacency matrix $\mathbf{A}$, if we compute its second largest eigenvector $\mathbf{v}_2$ and assign nodes to communities according to the sign pattern of this vector, we will correctly assign all but $O\left(\frac{p}{(p-q)^2}\right)$ nodes.

- **Hard case:** Suppose $q = .8p$ so $\frac{p}{(p-q)^2} = 25/p$.

If $p = c/n$, the number of mistakes is $\frac{25}{c} \cdot n$. I.e., $< 10\%$ error if average node has roughly 250 connections.

RANDOMIZED NUMERICAL LINEAR ALGEBRA

Forget about the previous problem, but still consider the matrix $\mathbf{M} = \mathbb{E}[\mathbf{A}]$.

- Dense $n \times n$ matrix.
- Computing top eigenvectors takes $\approx O(n^2/\sqrt{\epsilon})$ time.

If someone asked you to speed this up and return approximate top eigenvectors, what could you do?

RANDOMIZED NUMERICAL LINEAR ALGEBRA

Main idea: If you want to compute singular vectors, multiply two matrices, solve a regression problem, etc.:

1. Compress your matrices using a randomized method (e.g. subsampling).
2. Solve the problem on the smaller or sparser matrix.
   - $\tilde{\mathbf{A}}$ called a "sketch" or "coreset" for $\mathbf{A}$.