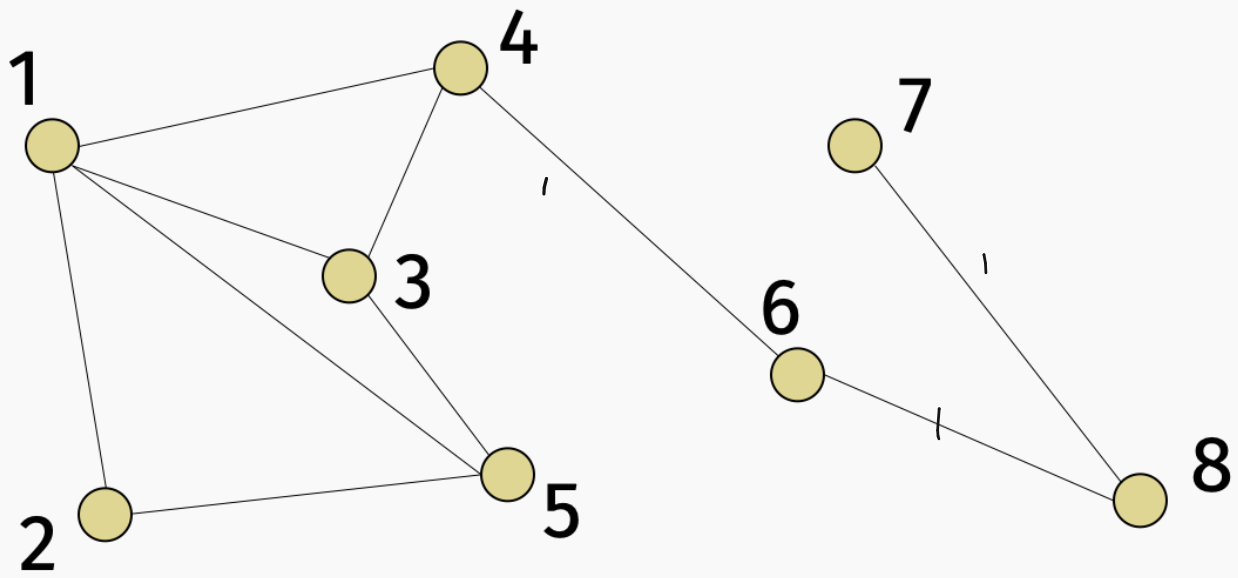
# CS-GY 6763: Lecture 11
# Spectral clustering, spectral graph theory.

NYU Tandon School of Engineering, Prof. Christopher Musco

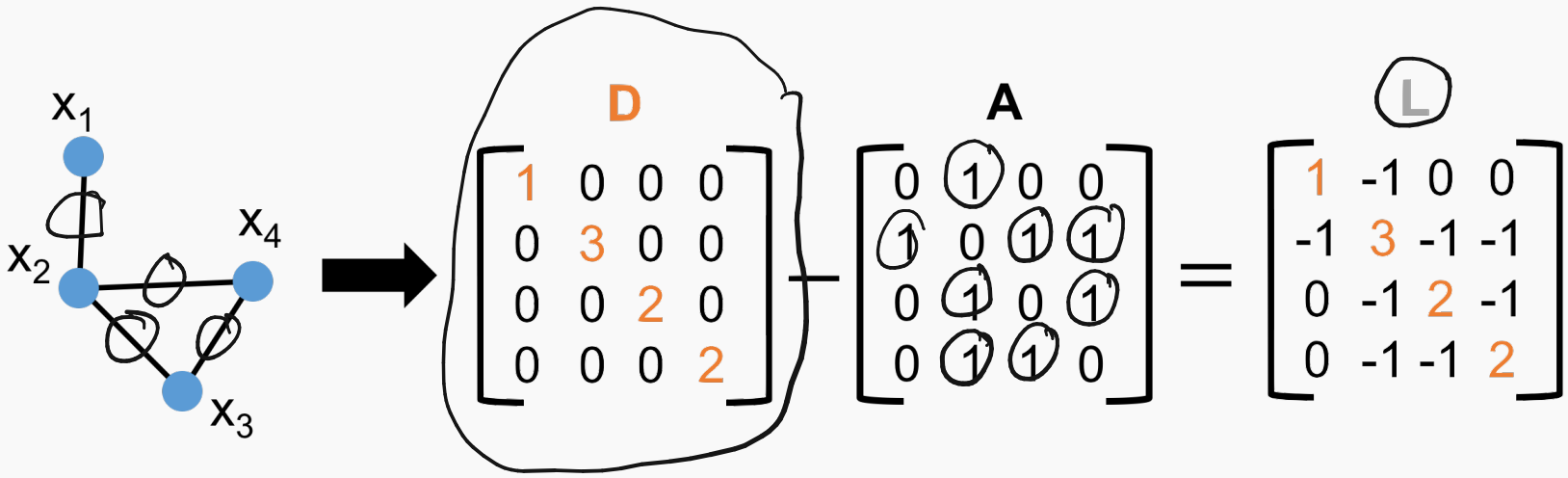
**Main idea:** Understand graph data by constructing natural matrix representations, and studying that matrix's spectrum (eigenvalues/eigenvectors).




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

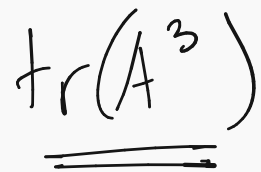
# MATRIX REPRESENTATIONS OF GRAPHS

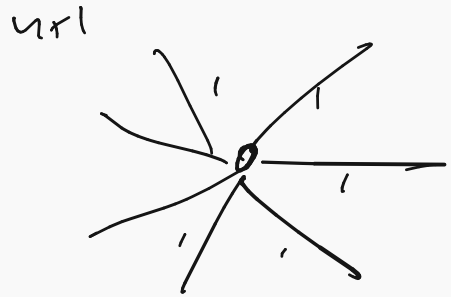
Two most common representations: $n \times n$ adjacency matrix $\mathbf{A}$ and graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix.

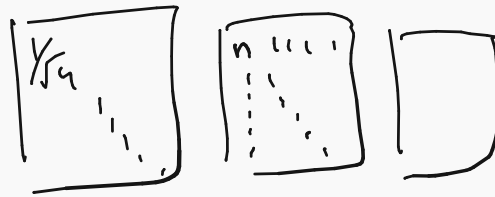



Also common to look at normalized versions of both of these: $\bar{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\bar{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.

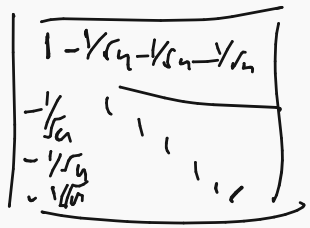
# SPECTRAL GRAPH THEORY TIDBITS



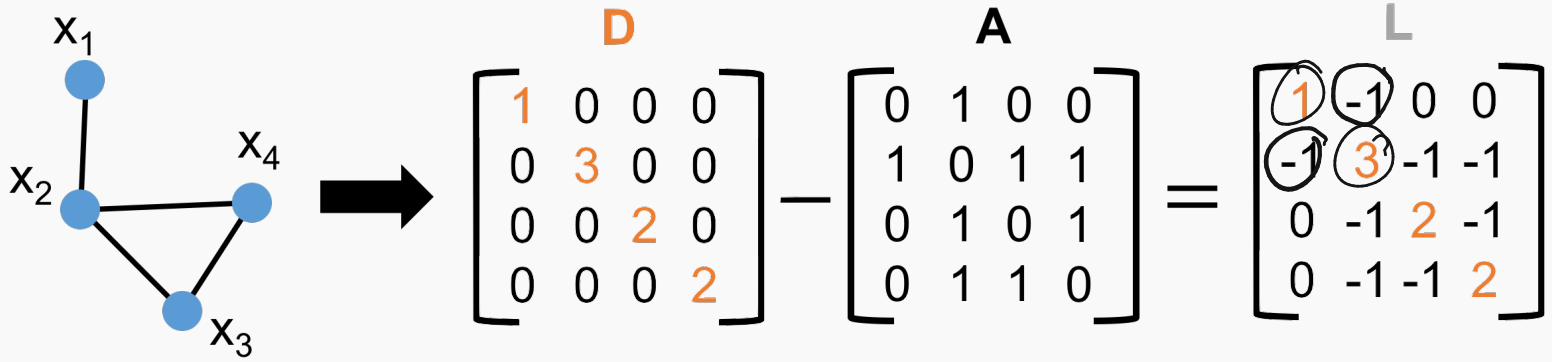




- If $\mathbf{L}$ have $k$ eigenvalues equal to 0, then $G$ has $k$ connected components.
- Sum of cubes of $\mathbf{A}$'s eigenvalues is equal to number of triangles in the graph times 6.
- Sum of eigenvalues to the power $q$ is proportional to the number of $q$ cycles.

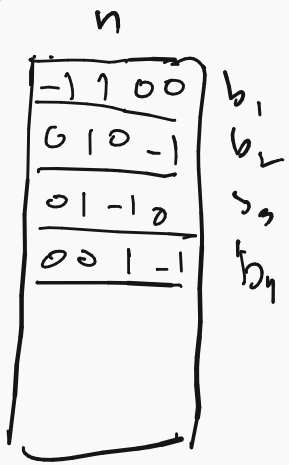


# THE LAPLACIAN VIEW




$L = B^T B$ where $B$ is the signed "edge-vertex incidence" matrix.

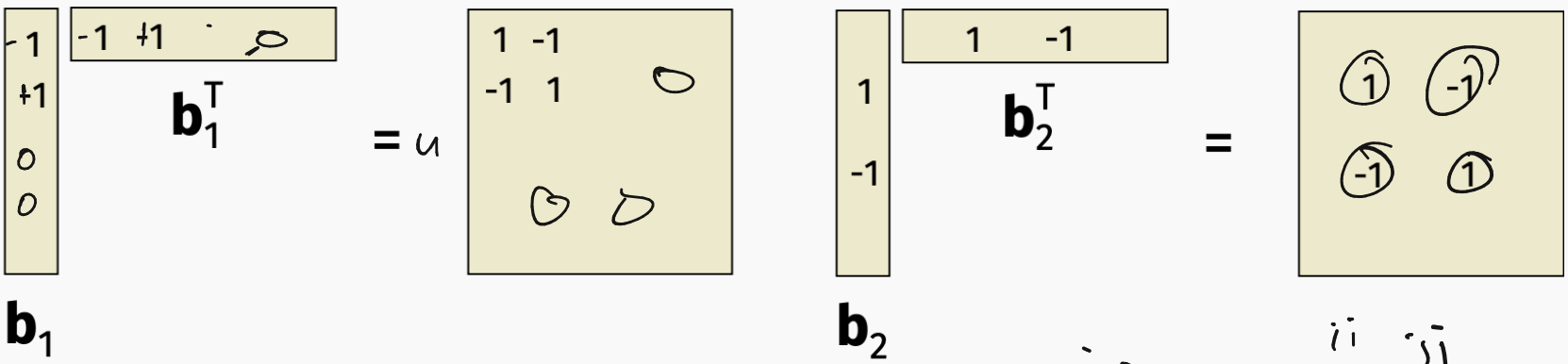
$B =$



# THE LAPLACIAN VIEW

$$L = B^T B = b_1 b_1^T + b_2 b_2^T + \ldots + b_m b_m^T,$$

where $b_i$ is the $i^{th}$ row of $B$ (each row corresponds to a single edge).



## Conclusions from $L = B^T B$

$x^T L x = x^T B^T B x = \|Bx\|_2^2 \geq 0$

- $L$ is positive semidefinite: $x^T L x \geq 0$ for all $x$.
- $L = V \Sigma^2 V^T$ where $U \Sigma V^T$ is $B$'s SVD. Columns of $V$ are eigenvectors of $L$.

$L = (U \Sigma V^T)^T U \Sigma V^T = V \Sigma \underbrace{U^T U}_{=I} \Sigma V^T = V \Sigma^2 V^T$

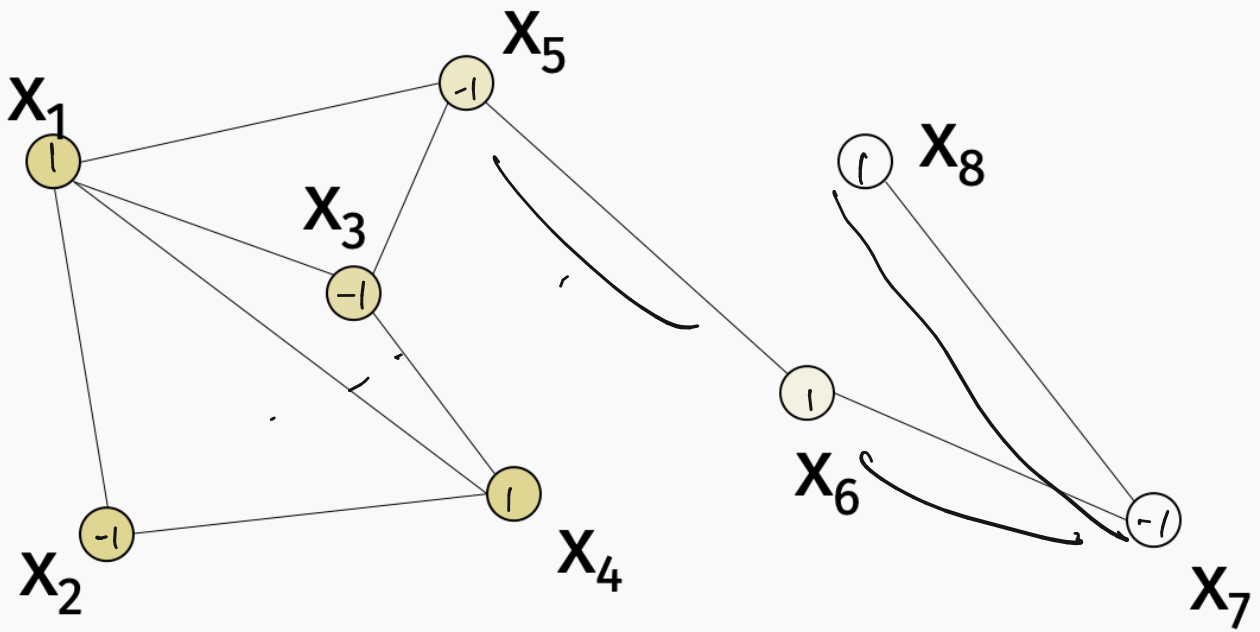
- For any vector $x \in \mathbb{R}^n$,

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

$Bx = \begin{bmatrix} x(i) - x(j) \\ \vdots \end{bmatrix}$

# THE LAPLACIAN VIEW

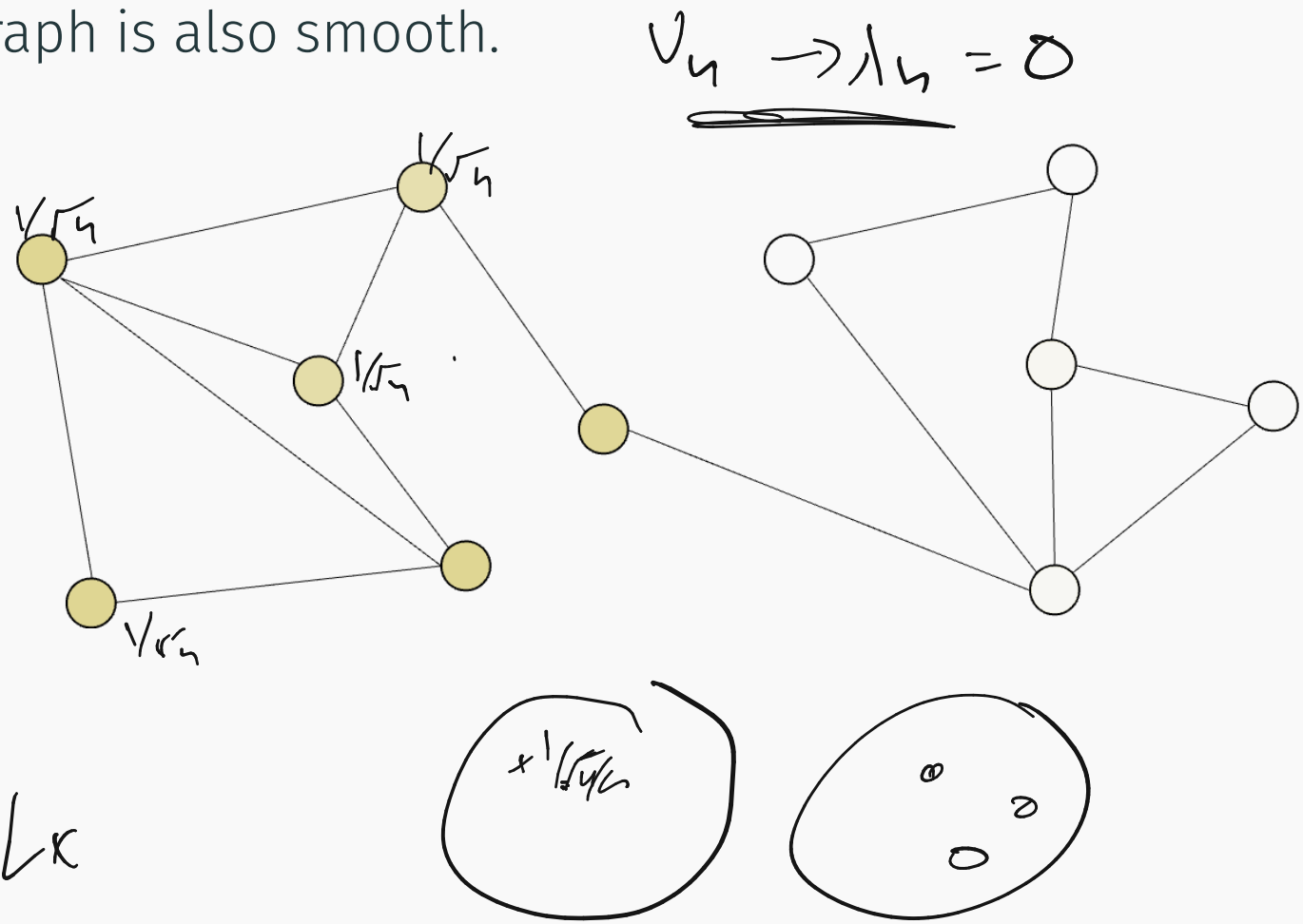
$\mathbf{x}^T L \mathbf{x} = \sum_{(i,j)\in E} (\mathbf{x}(i) - \mathbf{x}(j))^2$. So $\mathbf{x}^T L \mathbf{x}$ is small if $\mathbf{x}$ is a "smooth" function with respect to the graph.




Eigenvectors of the Laplacian with small eigenvalues correspond to smooth functions over the graph.

# ANOTHER EXAMPLE OF A SMOOTH FUNCTION

Any function that only has a large change across a small cut in the graph is also smooth.



## Courant–Fischer min-max principle

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

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

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

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

$$\vdots$$

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

## Courant–Fischer min-max principle

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

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

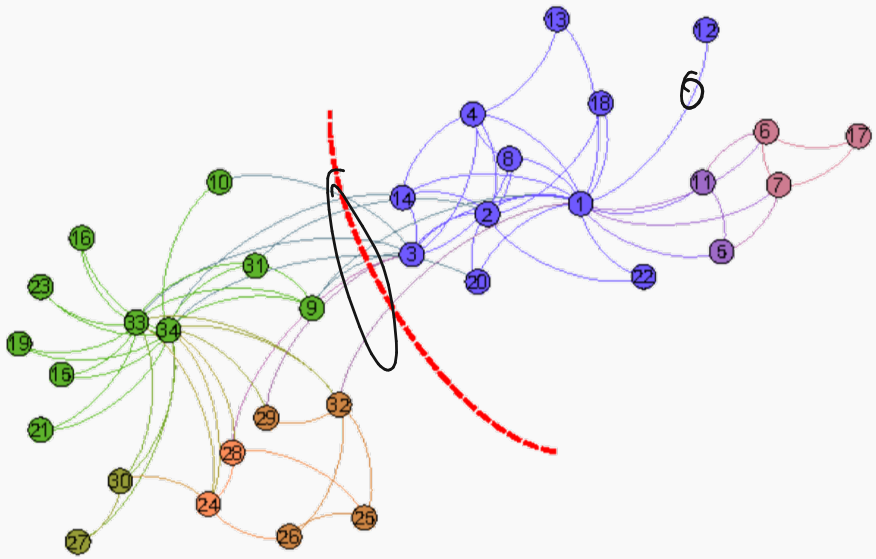
# EXAMPLE APPLICATION OF SPECTRAL GRAPH THEORY

- Study graph partitioning problem important in 1) understanding social networks 2) nonlinear clustering in unsupervised machine learning (spectral clustering). 3) Graph visualization 4) Mesh partitioning
- See how this problem can be solved heuristically using Laplacian eigenvectors.
- Give a full analysis of the method for a common random graph model.
- Use two tools: matrix concentration and eigenvector perturbation bounds.

# BALANCED CUT

**Common 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

Important in 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.

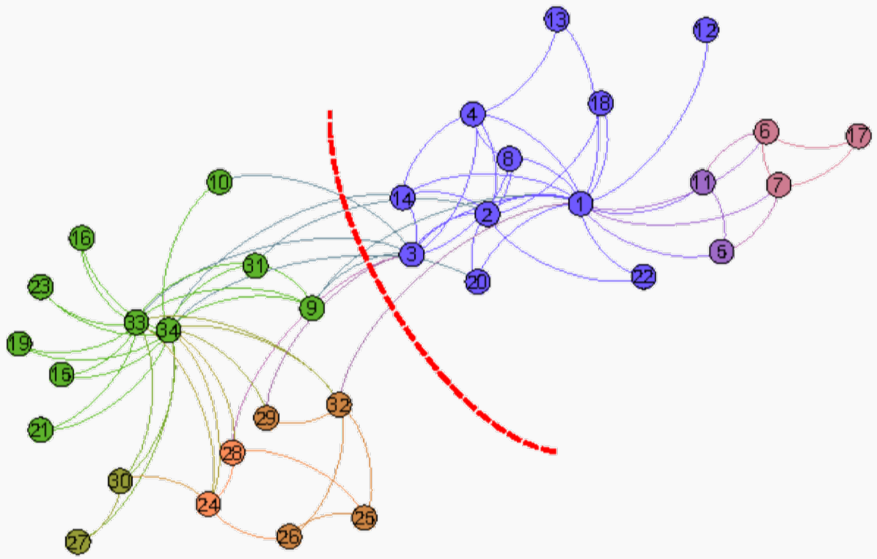
"The network captures 34 members of a karate club, documenting links between pairs of members who interacted outside the club. During the study a conflict arose between the administrator "John A" and instructor "Mr. Hi" (pseudonyms), which led to the split of the club into two. Half of the members formed a new club around Mr. Hi; members from the other part found a new instructor or gave up karate. Based on collected data Zachary correctly assigned all but one member of the club to the groups they actually joined after the split." – Wikipedia

**Beautiful paper – definitely worth checking out!**

# BALANCED CUT

**Common 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.

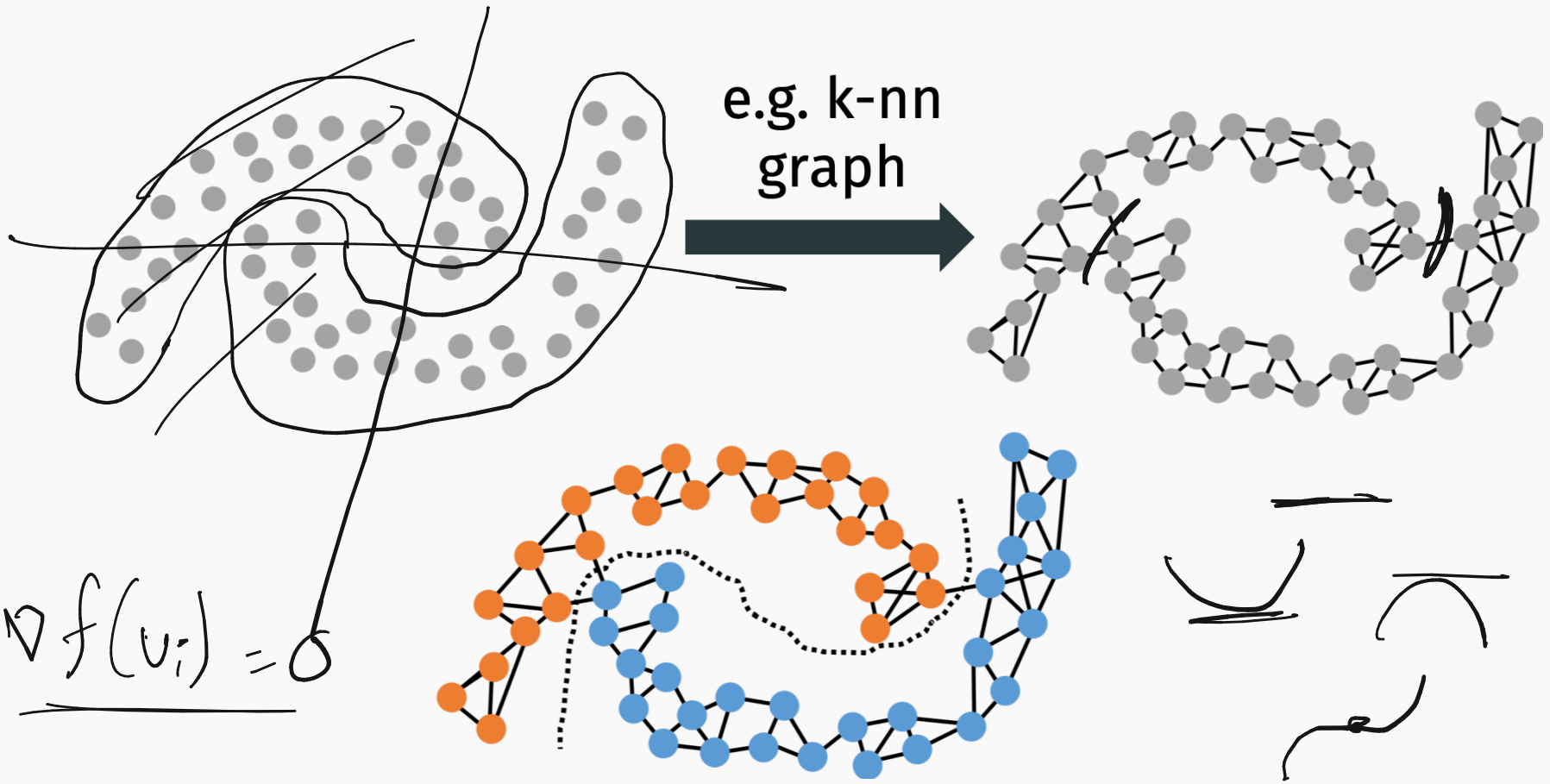



(a) Zachary Karate Club Graph

Important in 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.

There are many way's to formalize Zachary's problem:

**$\beta$-Balanced Cut:**

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

**Sparsest Cut:**

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

Most 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 in at worst $O(n^3)$ time (faster if you use iterative methods).**

S, T → T = V \ S

**Basic spectral clustering method:**

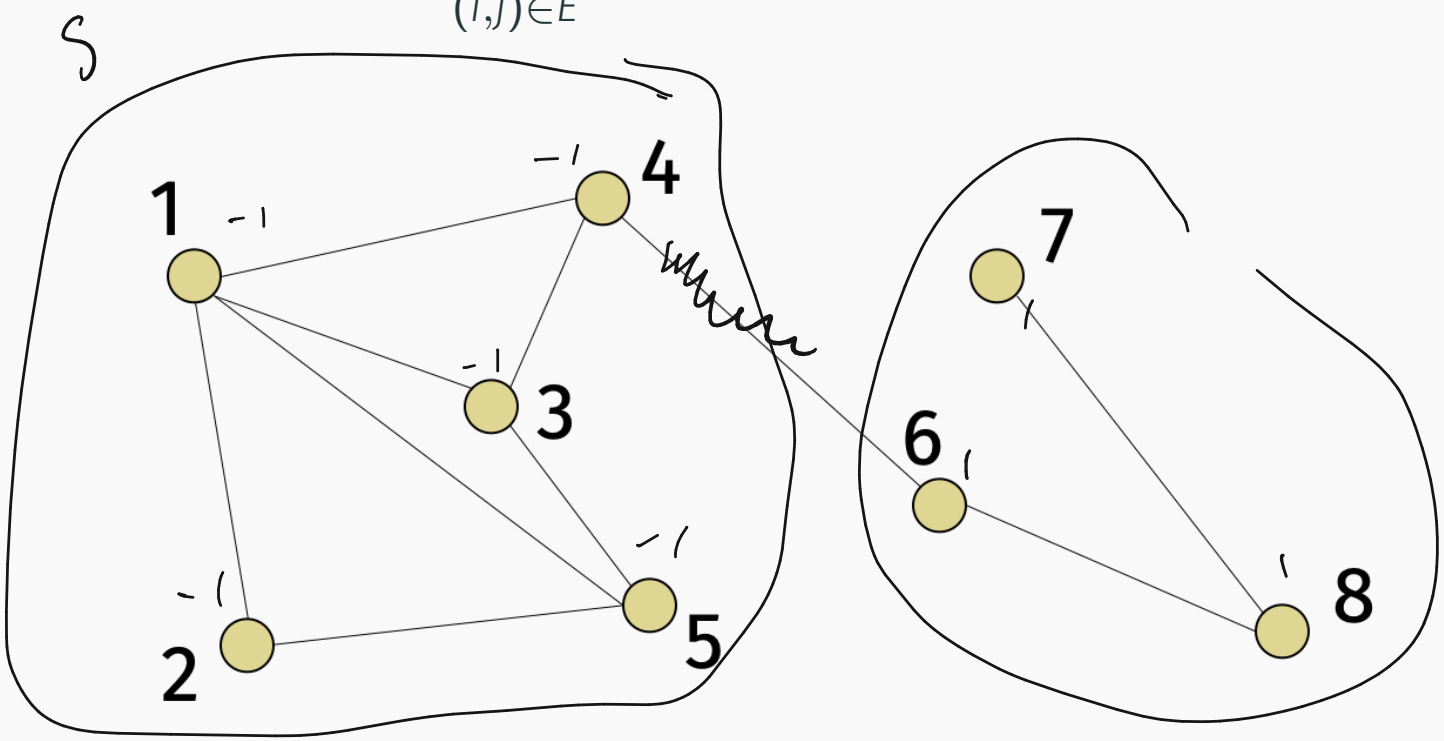
- Compute second smallest eigenvalue (vector) of graph, $\mathbf{v}_{n-1}$.
- $\mathbf{v}_{n-1}$ has an entry for every node $i$ in the graph.
- If the $i^{\text{th}}$ entry is positive, put node $i$ in $T$.
- Otherwise if the $i^{\text{th}}$ entry is negative, put $i$ in $S$.

This shouldn't make much sense yet!

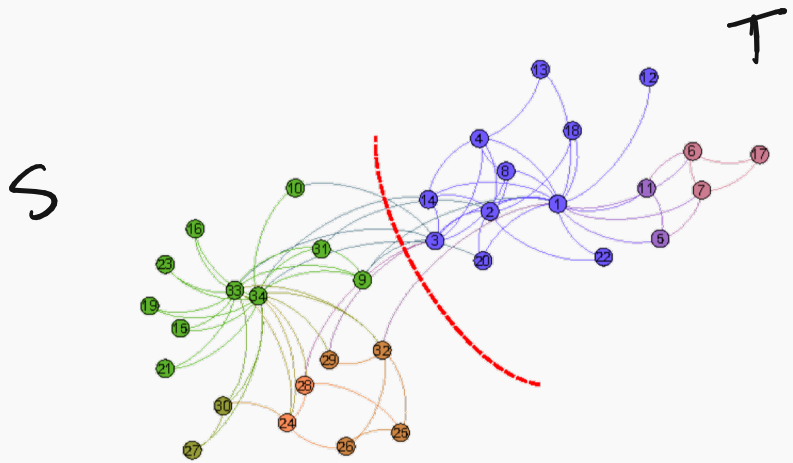
## Another conclusion from $L = B^T B$:

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 $i \in T = V \setminus S$:

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



# THE LAPLACIAN VIEW



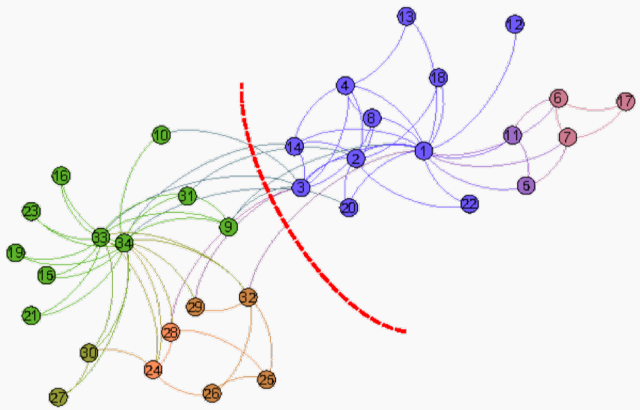
(a) Zachary Karate Club Graph

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

$$\begin{pmatrix} -1/\sqrt{n} \\ -1/\sqrt{n} \\ \vdots \\ 1/\sqrt{n} \end{pmatrix}$$

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:

- $\mathbf{c}^T L \mathbf{c} = \frac{4}{n} \cdot cut(S, T)$.
- $\mathbf{c}^T \mathbf{1} = \frac{1}{\sqrt{n}}(|T| - |S|)$. $= 0$

$-1/\sqrt{n}$

$+1/\sqrt{n}$

$1/\sqrt{n}$

$-1/\sqrt{n}$

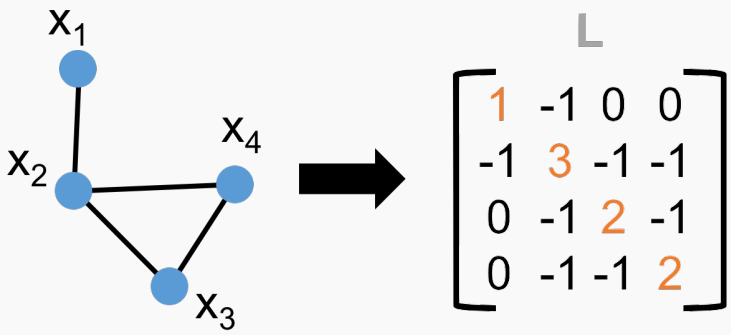
Want to minimize both $\mathbf{c}^T L \mathbf{c}$ (cut size) and $|\mathbf{c}^T \mathbf{1}|$ (imbalance).

# SMALLEST LAPLACIAN EIGENVECTOR

The smallest eigenvector/singular vector $\mathbf{v}_n$ satisfies:

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

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




What is $v_0$?

2 optimization problems

1) Best perfectly balanced cut

2) Second smallest eigenvector prob.

1) = 2) + additional constraint

instead of just requiring $\|v\|_2 = 1$, also require that $v_i = \pm \frac{1}{\sqrt{n}}$ for all $i$.

# SECOND SMALLEST LAPLACIAN EIGENVECTOR

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

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

If $\mathbf{v}_{n-1}$ were binary $\{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n$ it would have:

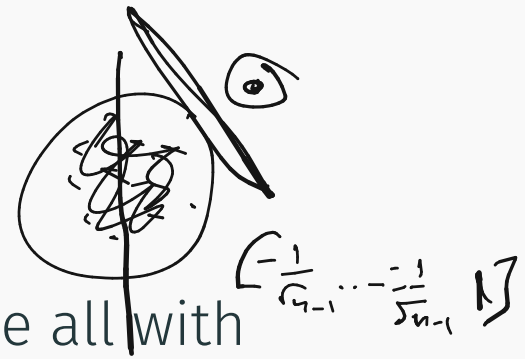
- $\mathbf{v}_{n-1}^T L \mathbf{v}_{n-1} = \frac{1}{n}\,\mathbf{cut}(S,T)$ as small as possible given that $\mathbf{v}_{n-1}^T \mathbf{1} = |T| - |S| = 0$.
- $\mathbf{v}_{n-1}$ would indicate the smallest perfectly balanced cut.

$\mathbf{v}_{n-1} \in \mathbb{R}^n$ is not generally binary, but a natural approach is to 'round' the vector to obtain a cut.

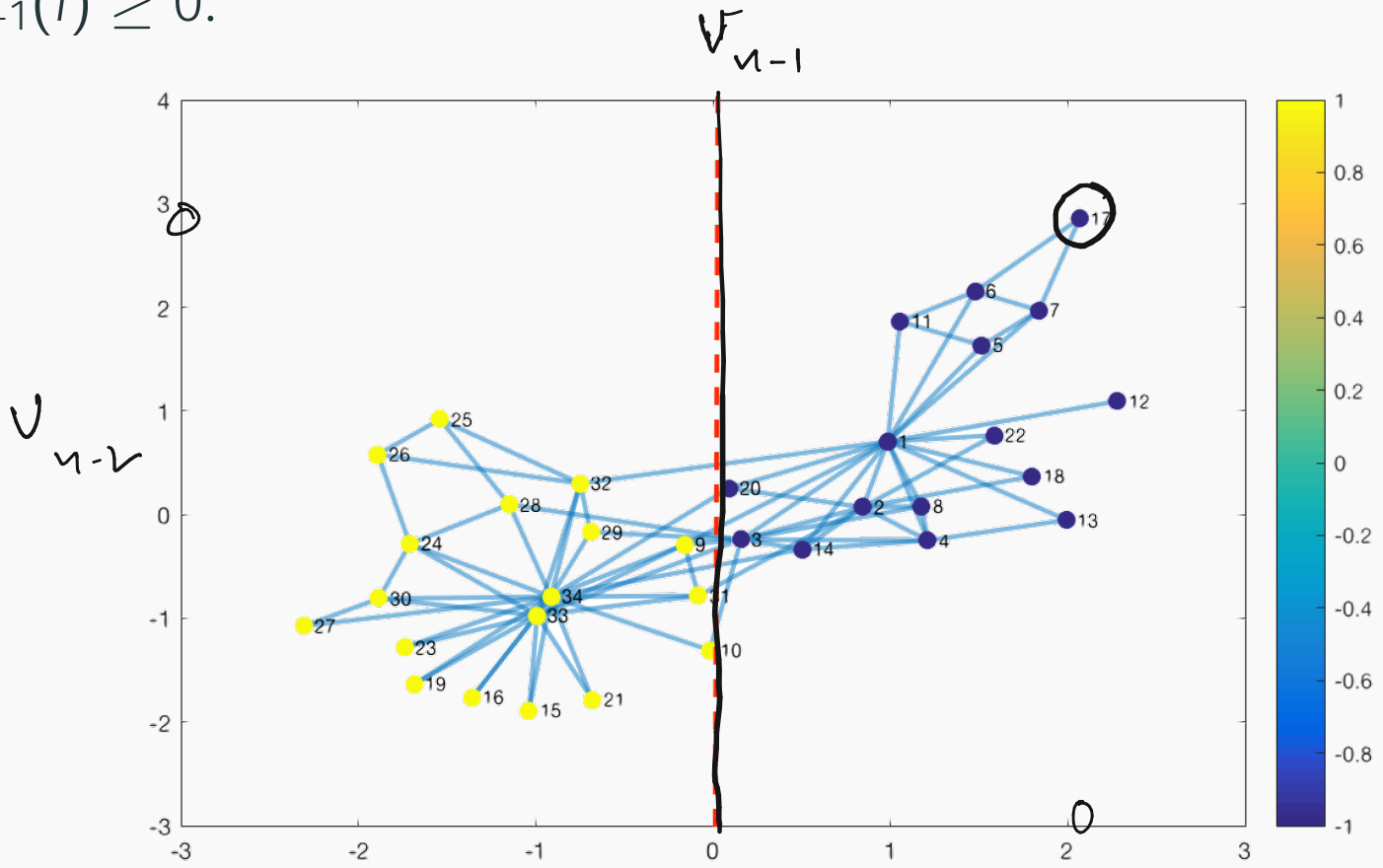
# CUTTING WITH THE SECOND LAPLACIAN EIGENVECTOR

Find a good partition of the graph by computing

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



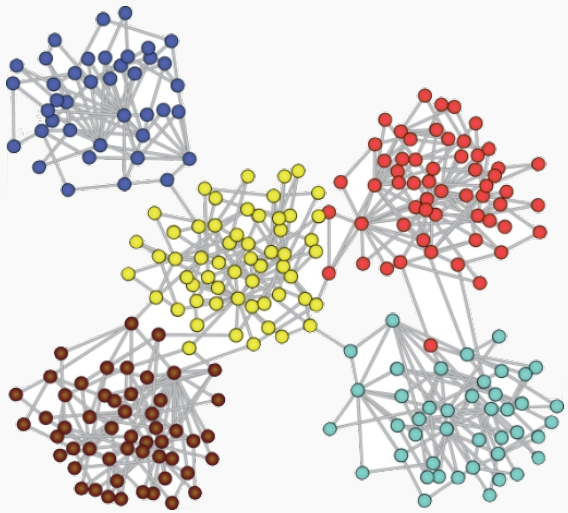
Set $S$ to be all nodes with $\mathbf{v}_{n-1}(i) < 0$, and $T$ to be all with $\mathbf{v}_{n-1}(i) \geq 0$.



# SPECTRAL PARTITIONING IN PRACTICE

The Shi-Malik normalized cuts algorithm is one of the most commonly used variants of this approach, using the normalized Laplacian $\bar{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

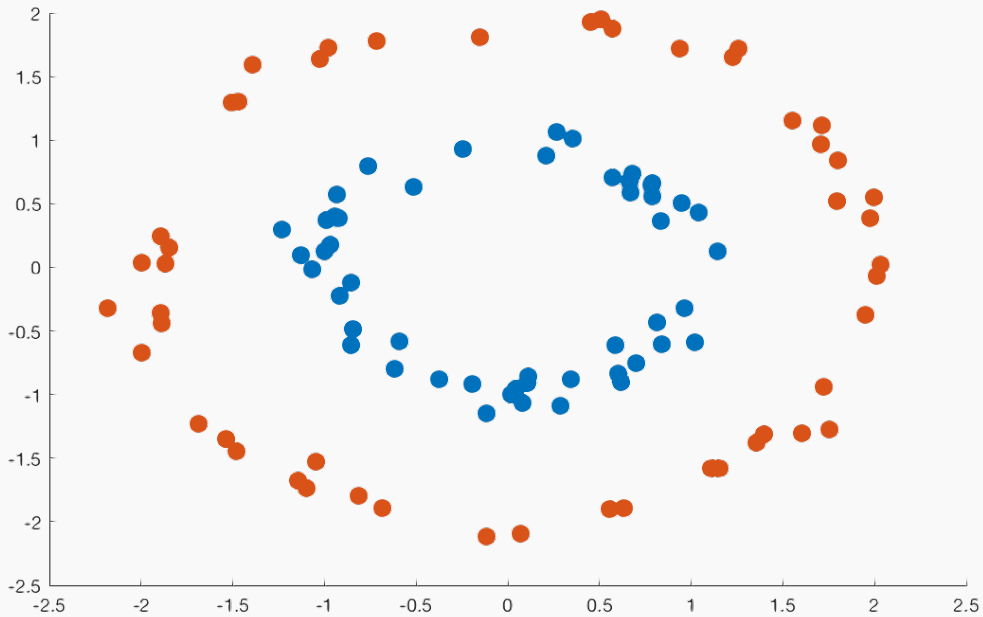
**Important consideration:** What to do when we want to split the graph into more than two parts?



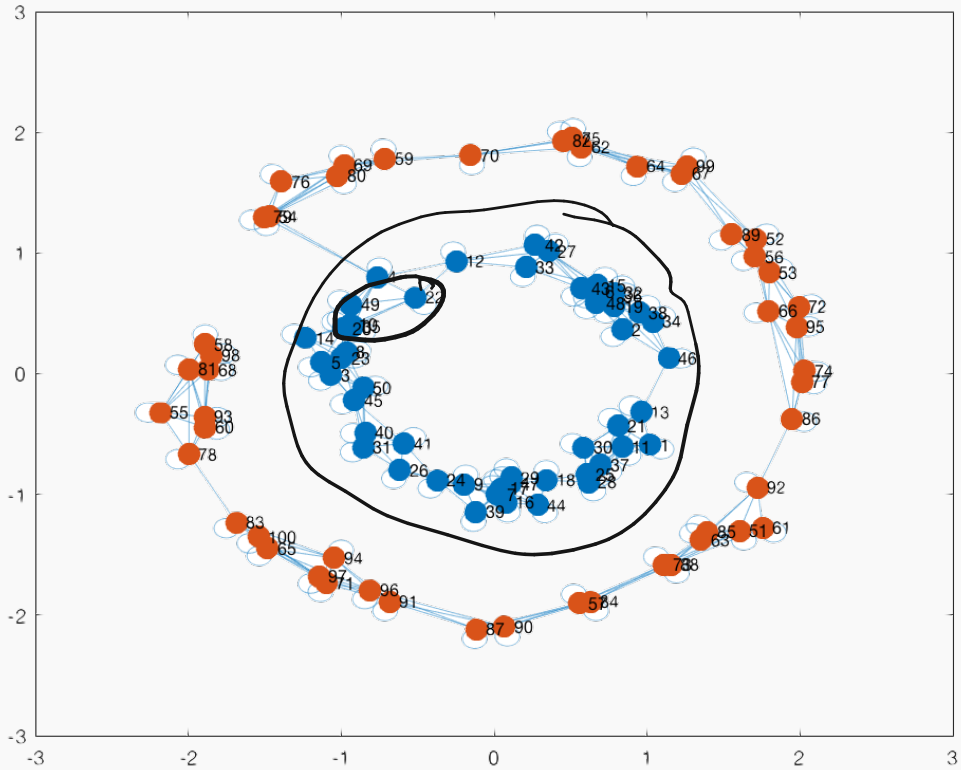
Spectral Clustering:

- Compute smallest $\ell$ 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.

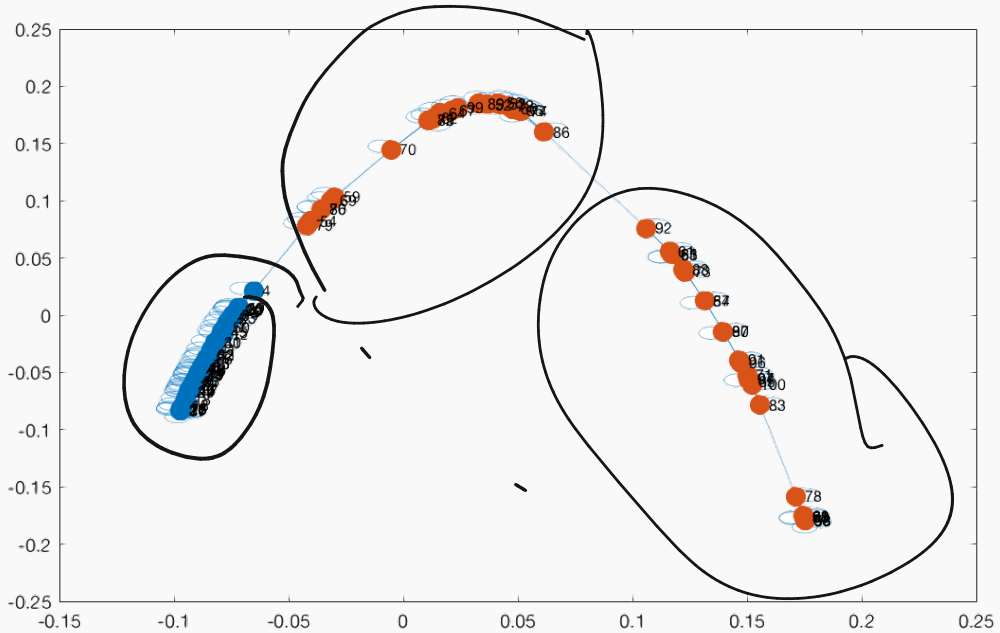
**Original Data:** (not linearly separable)



## $k$-Nearest Neighbors Graph:



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

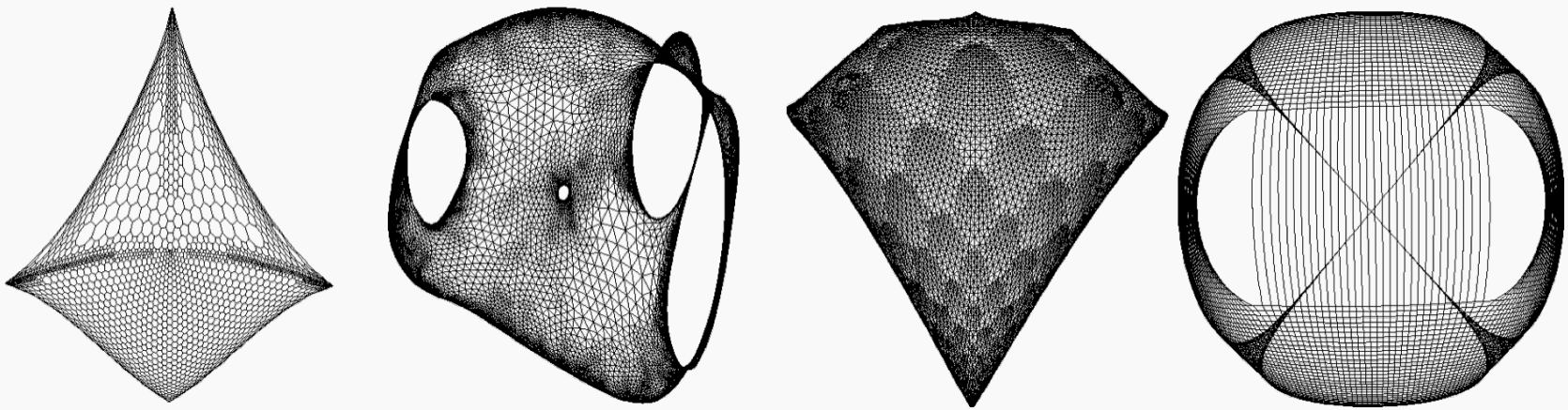


# WHY DOES THIS WORK?

Intuitively, since $\mathbf{v} \in \underbrace{\mathbf{v}_1, \ldots \mathbf{v}_k}$ are smooth over the graph,

$$\sum_{i,j \in E} (\underbrace{\mathbf{v}[i]} - \underbrace{\mathbf{v}[j]})^2$$

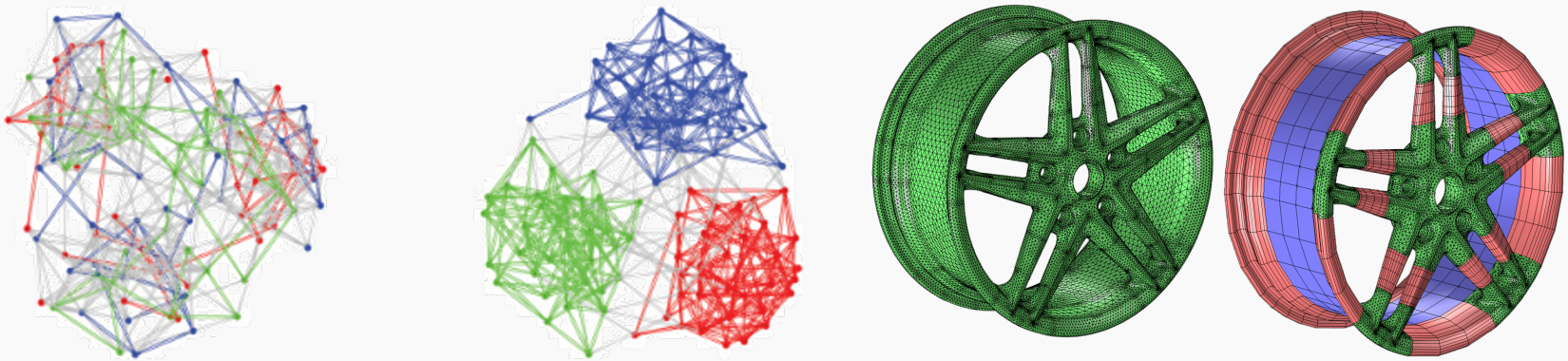
is small for each coordinate. I.e. this embedding explicitly encourages nodes connected by an edge to be placed in nearby locations in the embedding.



Also useful e.g., in graph drawing.

# TONS OF OTHER APPLICATIONS!

Fast balanced partitioning 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).

# 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.
- Difficult to analyze for general 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.
- This is also the whole idea behind Bayesian Machine Learning (can be used to justify $\ell_2$ linear regression, $k$-means clustering, PCA, etc.)

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

$$f(x) = \frac{x^T A^T A x}{x^T x}$$

$$\nabla f(x) = \begin{bmatrix} \partial f / x_1 \\ \vdots \\ \partial f / x_n \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

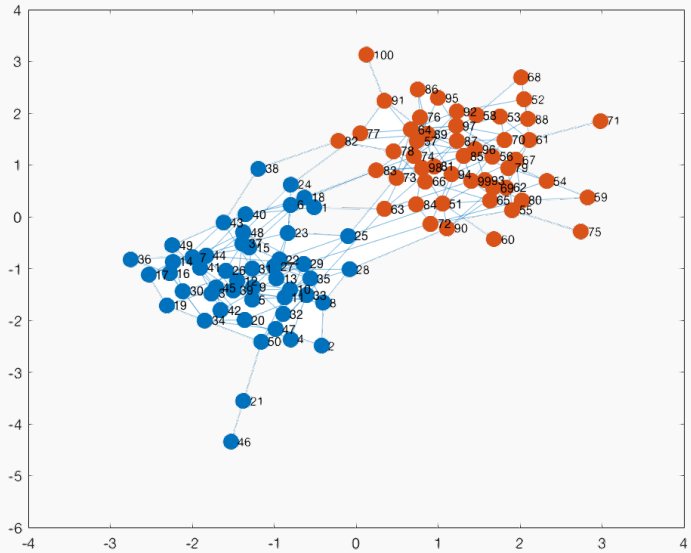
$$f(x) = -\frac{\sum_{ij} x_i x_j A_i^T A_j}{x_1^2 + \ldots + x_n^2}$$

{ cost of cake

**Stochastic Block Model (Planted Partition Model):**

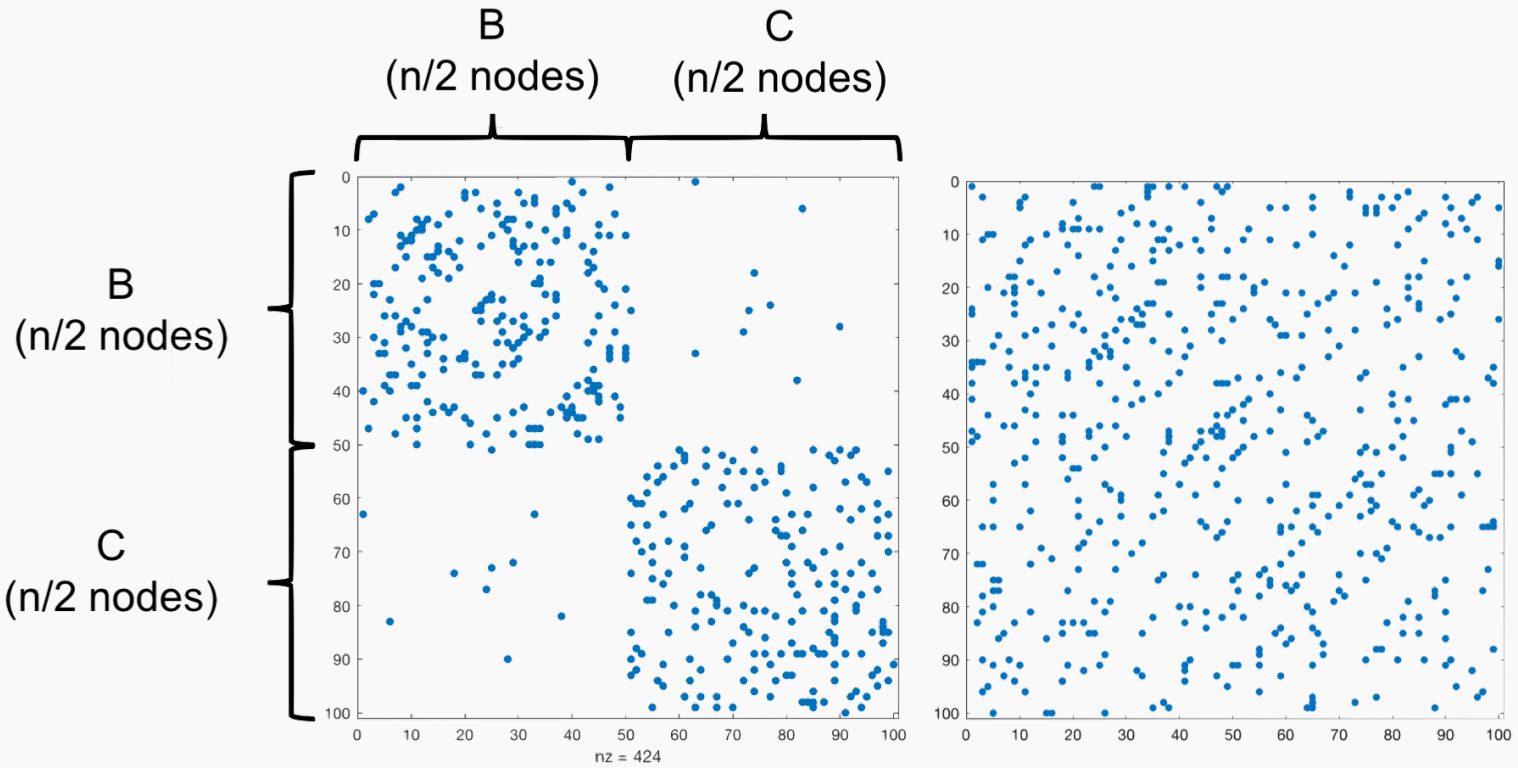
Let $G_n(p, q)$ be a distribution over graphs on $n$ nodes, split equally into two groups $B$ and $C$, each with $n/2$ nodes.

- 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. $q < p$.



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

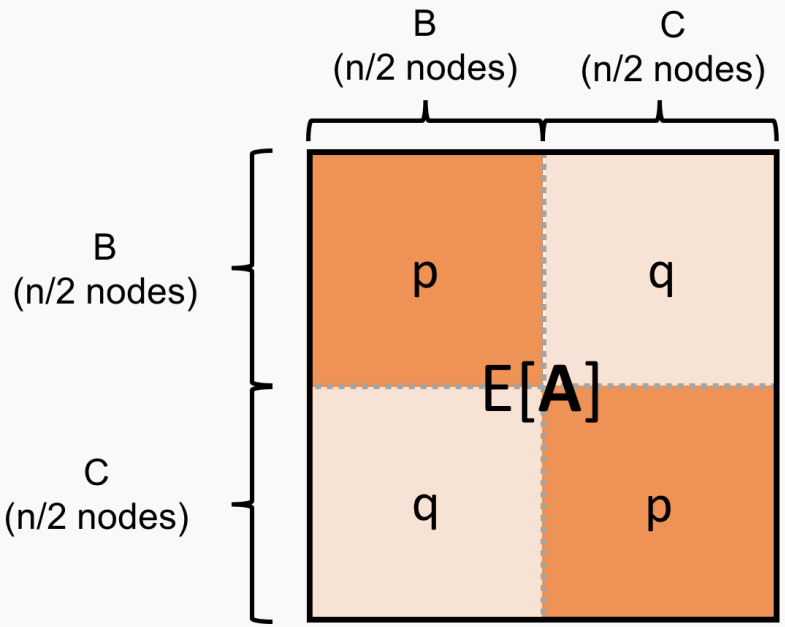
- Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of $G$. What is $\mathbb{E}[\mathbf{A}]$?




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

# EXPECTED ADJACENCY SPECTRUM

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




We are going to determine the eigenvectors and eigenvalues of $\mathbb{E}[\mathsf{A}]$.

# EXPECTED LAPLACIAN

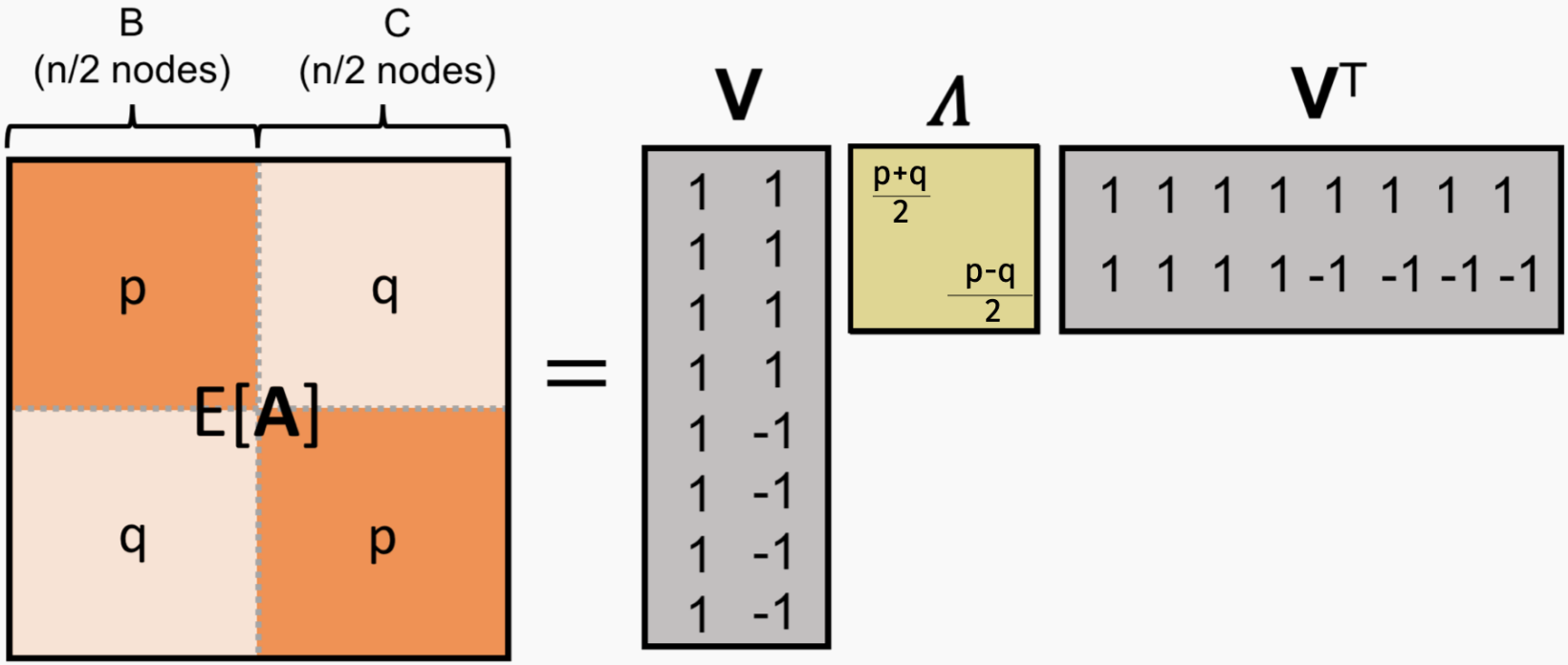
What is the expected Laplacian of $G_n(p, q)$?

$\mathbb{E}[\mathbf{A}]$ 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




- $\mathbf{v}_1 \sim \mathbf{1}$ with eigenvalue $\lambda_1 = \frac{(p+q)n}{2}$.
- $\mathbf{v}_2 \sim \boldsymbol{\chi}_{B,C}$ with eigenvalue $\lambda_2 = \frac{(p-q)n}{2}$.
- $\chi_{B,C}(i) = 1$ if $i \in B$ and $\chi_{B,C}(i) = -1$ for $i \in C$.

If we compute $\mathbf{v}_2$ then we 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 $\chi_{B,C}$ – 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 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).

For $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\|\mathbf{X}\|_2 = \max_{\mathbf{z} \in \mathbb{R}^d : \|\mathbf{z}\|_2 = 1} \|\mathbf{X}\mathbf{z}\|_2$.

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

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

where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of $\bar{\mathbf{A}}$.

The error gets larger if there are eigenvalues with similar magnitudes.

# EIGENVECTOR PERTURBATION

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

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

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

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

$$\sin\theta(v_2, \bar{v}_2) \leq \frac{O(\sqrt{pn})}{\min_{j\neq i}|\lambda_i - \lambda_j|} \leq \frac{O(\sqrt{pn})}{(p-q)n/2} = O\left(\frac{\sqrt{p}}{(p-q)\sqrt{n}}\right)$$

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

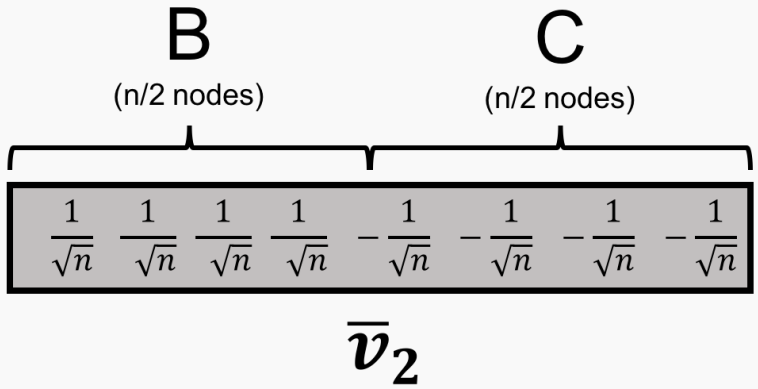
$$\min_{j\neq i}|\lambda_i - \lambda_j| = \min\left(qn, \frac{(p-q)n}{2}\right).$$

Assume $\left|\frac{(p-q)n}{2} - 0\right|$ will be the minimum of the two gaps. I.e. smaller than $\left|\frac{(p+q)n}{2} - \frac{(p-q)n}{2}\right| = qn$.

# 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?

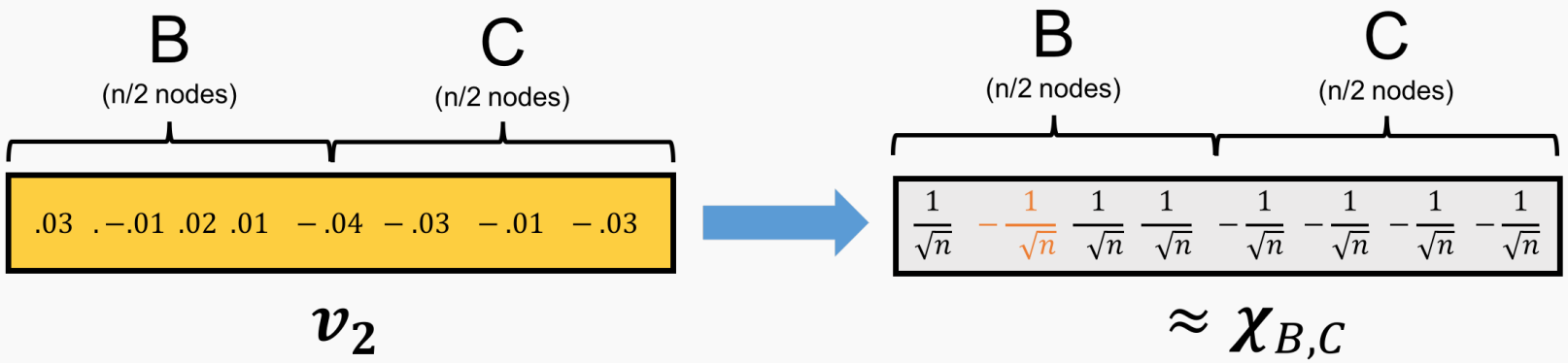
- Can show that this implies $\|\mathbf{v}_2 - \bar{\mathbf{v}}v_2\|_2^2 \leq O\left(\frac{p}{(p-q)^2 n}\right)$ (exercise).
- $\bar{\mathbf{v}}_2$ is $\frac{1}{\sqrt{n}}\chi_{B,C}$: the community indicator vector.




- Every $i$ where $\mathbf{v}_2(i)$ and $\bar{\mathbf{v}}_2(i)$ differ in sign contributes $\geq \frac{1}{n}$ to $\|\mathbf{v}_2 - \bar{\mathbf{v}}_2\|_2^2$.
- So they 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 large eigenvector $v_2$ 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.




- Why does the error increase as $q$ gets close to $p$?
- Even when $p - q = O(1/\sqrt{n})$, assign all but an $O(n)$ fraction of nodes correctly. E.g., assign 99% of nodes correctly.

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?

We will discuss this more next class!