Large Graph Mining: Power Tools and a Practitioner's Guide

Christos Faloutsos
Gary Miller
Charalampos (Babis) Tsourakakis
CMU

Outline

- **Reminders**
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Matrix Representations of G(V,E)

Associate a matrix to a graph:

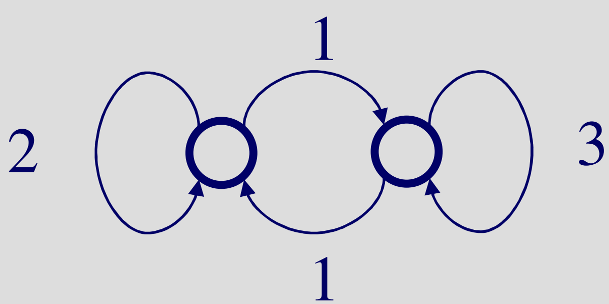
- Adjacency matrix
- Laplacian
- Normalized Laplacian

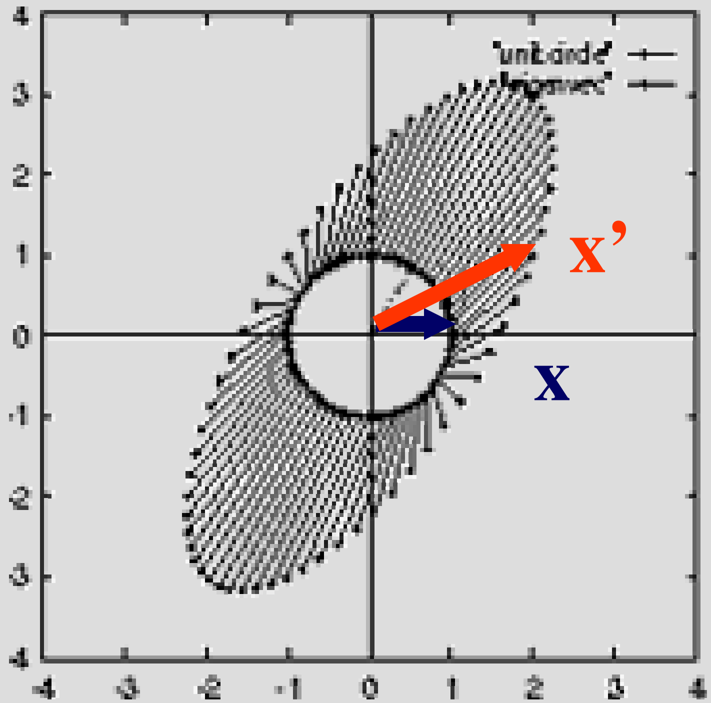
Main focus (Adjacency matrix, Laplacian)

Recall: Intuition

- **A** as vector transformation

$$\underset{\mathbf{x'}}{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \underset{\mathbf{A}}{\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}} \underset{\mathbf{x}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}$$

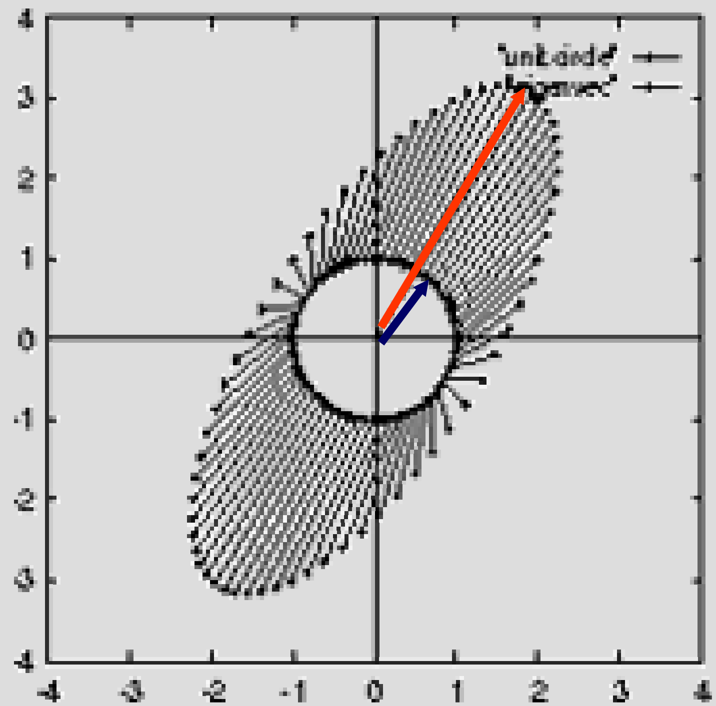




Intuition

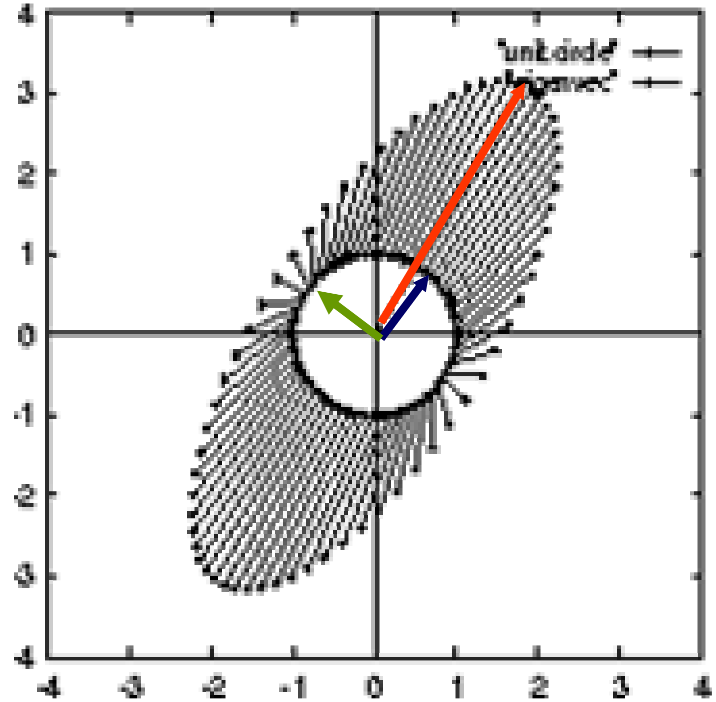
- By defn., eigenvectors remain parallel to themselves (‘**fixed points**’)

$$\underbrace{3.62}_{\lambda_1} * \underbrace{\begin{bmatrix}0.52\\0.85\end{bmatrix}}_{\mathbf{v_1}} = \underbrace{\begin{bmatrix}2 & 1\\1 & 3\end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix}0.52\\0.85\end{bmatrix}}_{\mathbf{v_1}}$$



Intuition

- By defn., eigenvectors remain parallel to themselves (‘**fixed points**’)
- And orthogonal to each other



Keep in mind!

- For the rest of slides we will be talking for square nxn matrices

$$M = \begin{bmatrix} m_{11} & & m_{1n} \\ & \ldots & \\ m_{n1} & & m_{nn} \end{bmatrix}$$

and symmetric ones, i.e,

$$M = M^T$$

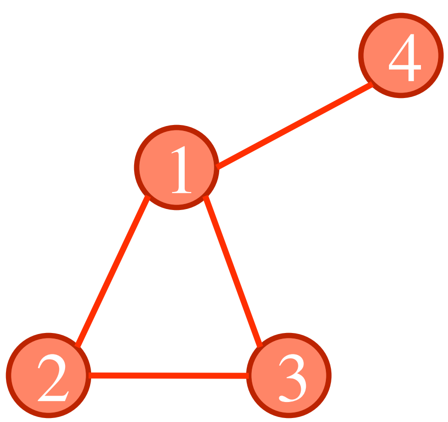
Outline

- Reminders
- **Adjacency matrix**
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Adjacency matrix

Undirected

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

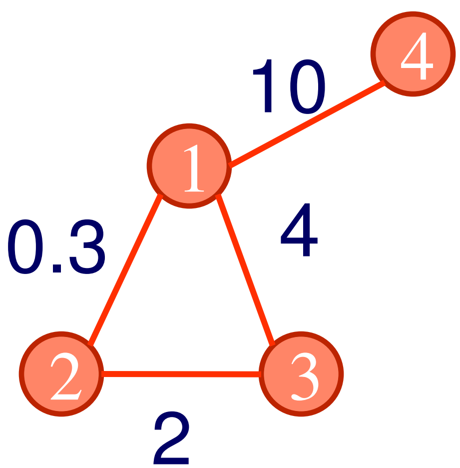


$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Adjacency matrix

Undirected Weighted

$$A_{uv} = \begin{cases} w_{uv} & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

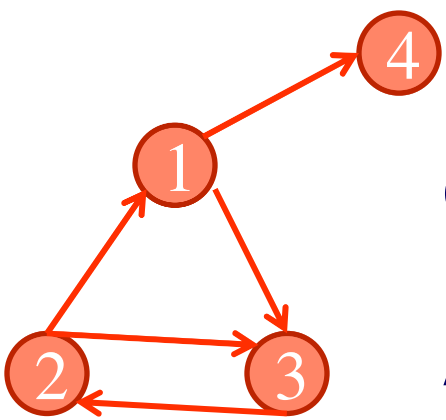


$$A = \begin{pmatrix} 0 & 0.3 & 4 & 10 \\ 0.3 & 0 & 2 & 0 \\ 4 & 2 & 0 & 0 \\ 10 & 0 & 0 & 0 \end{pmatrix}$$

Adjacency matrix

Directed

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$



Observation
If G is undirected,
$A = A^T$

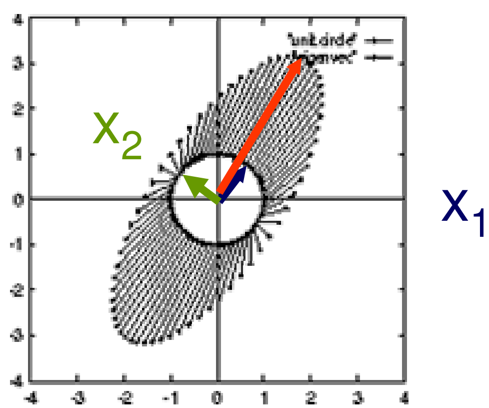
$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Spectral Theorem

Theorem [Spectral Theorem]

- If $M=M^T$, then

$$M = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} — & x_1^T & — \\ & \cdots & \\ — & x_n^T & — \end{bmatrix} = \lambda_1 x_1 x_1^T + \cdots + \lambda_n x_n x_n^T$$



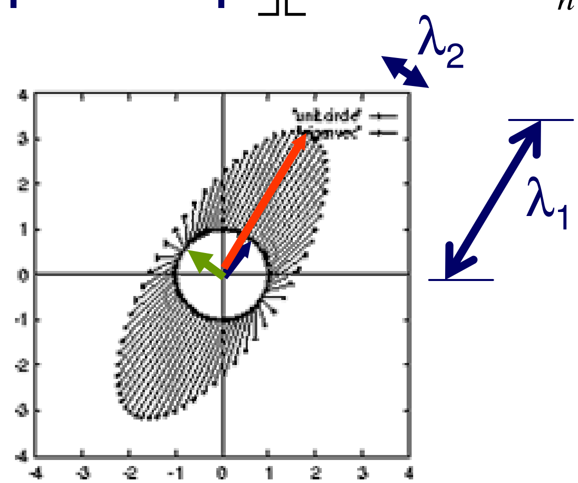
Reminder 1:
x_i,x_j orthogonal

Spectral Theorem

Theorem [Spectral Theorem]

- If $M=M^T$, then

$$M = \begin{bmatrix} | & & | \\ x_1 & \dots & x_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} — & x_1^T & — \\ & \dots & \\ — & x_n^T & — \end{bmatrix} = \lambda_1 x_1 x_1^T + \dots + \lambda_n x_n x_n^T$$



Reminder 2:

x_i i-th principal axis

λ_i length of i-th principal axis

Outline

- Reminders
- Adjacency matrix
  - **Intuition behind eigenvectors**: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

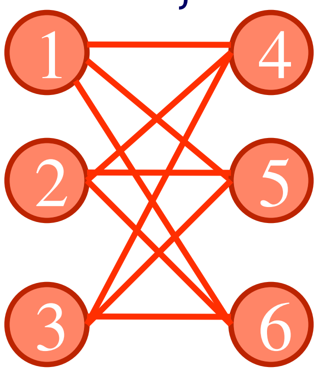
Eigenvectors:

- Give groups
- Specifically for bi-partite graphs, we get each of the two sets of nodes
- Details:

Bipartite Graphs

Any graph with no cycles of odd length is bipartite

$K_{3,3}$



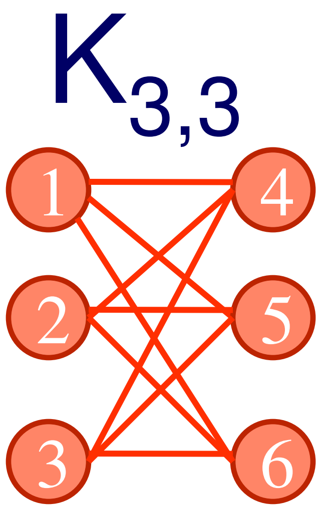
$$A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$$

Q1: Can we check if a graph is bipartite via its spectrum?

Q2: Can we get the partition of the vertices in the two sets of nodes?

Bipartite Graphs

Adjacency matrix $A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$



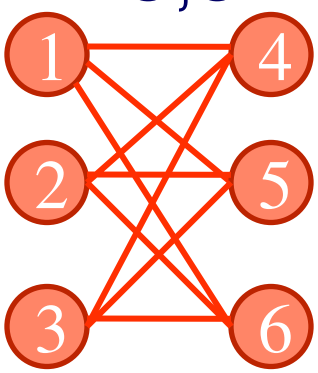
where $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Eigenvalues: **Λ**=[3,-3,0,0,0,0]

Bipartite Graphs

Adjacency matrix $A = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$

$K_{3,3}$

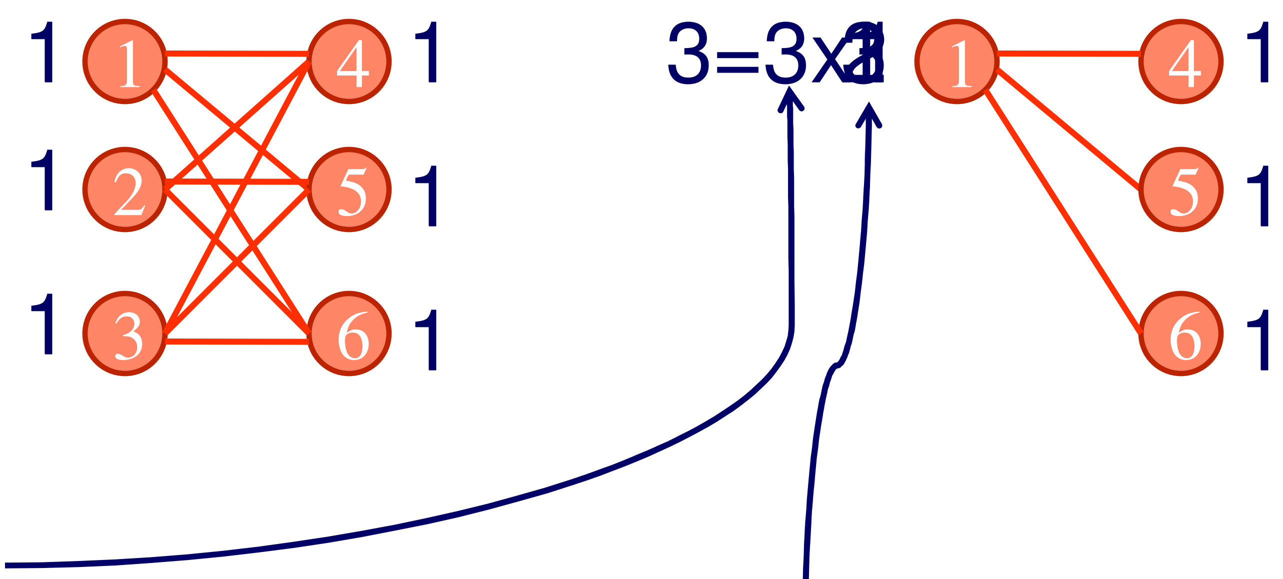


where $B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

Why $\lambda_1 = -\lambda_2 = 3$?

Recall: $A\mathbf{x} = \lambda\mathbf{x}$, $(\lambda, \mathbf{x})$ eigenvalue-eigenvector

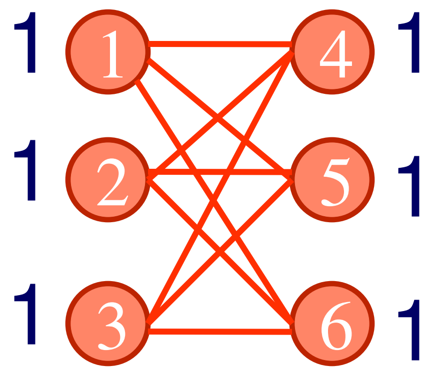
Bipartite Graphs

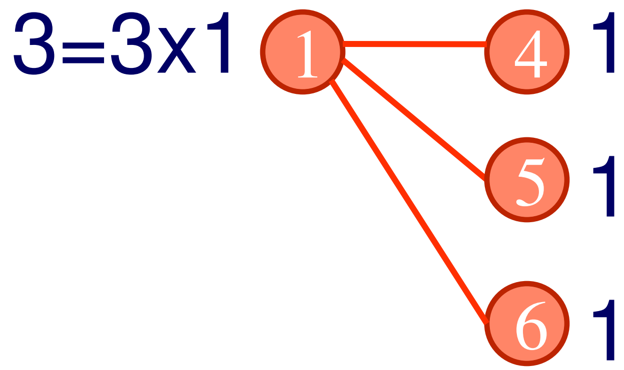


$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

Value @ each node: eg., enthusiasm about a product

Bipartite Graphs

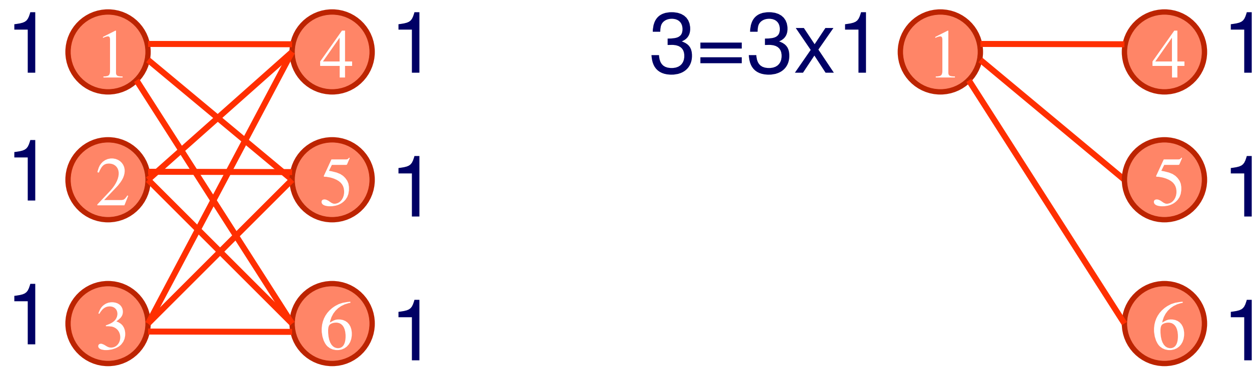




$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

1-vector remains unchanged (just grows by '3' = λ_1)

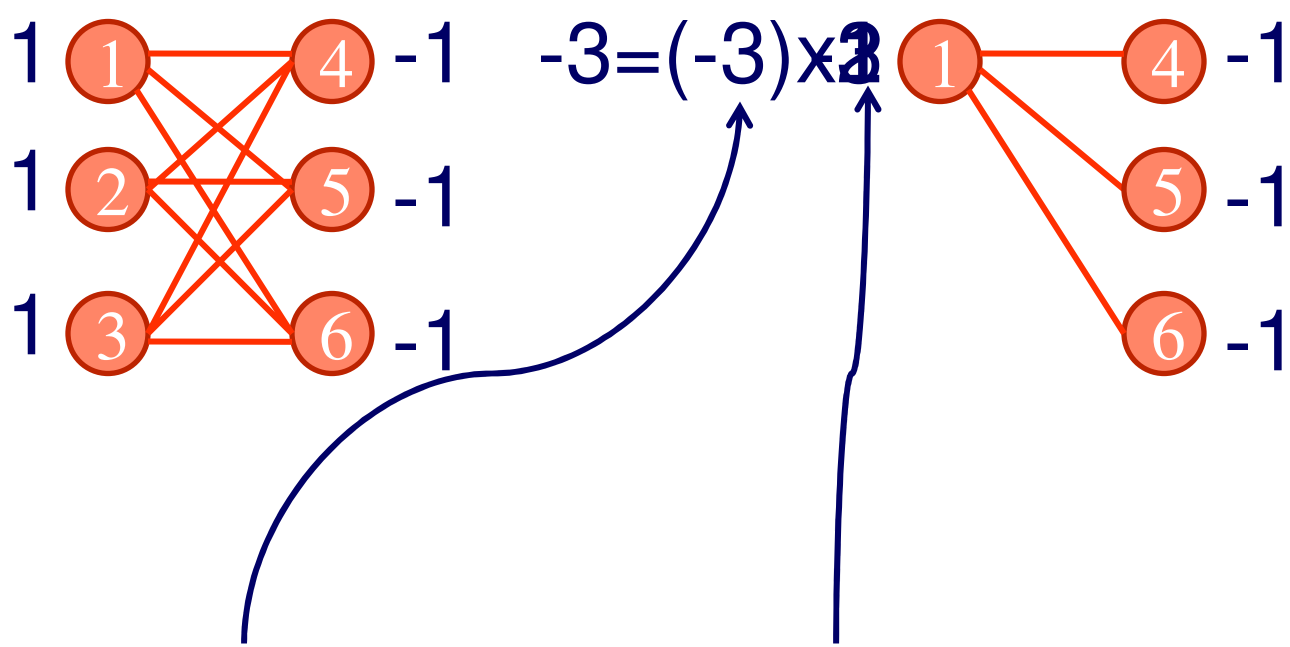
Bipartite Graphs



$$\lambda_1 = 3, u_1 = \mathbf{1} = [1,\ 1,\ 1,\ 1,\ 1,\ 1]^T$$

Which other vector remains unchanged?

Bipartite Graphs



$$\lambda_2 = -3, u_2 = \mathbf{1} = [1,\ 1,\ 1,\ \text{-}1,\ \text{-}1,\ \text{-}1]^T$$

Bipartite Graphs

- Observation
  u_2 gives the partition of the nodes in the two sets S, V-S!

1 2 3 4 5 6

$$\lambda_2 = -3, u_2 = \mathbf{1} = [1, 1, 1, -1, -1, -1]^T$$

S V-S

Question: Were we just “lucky”? Answer: No

Theorem: $\lambda_2 = -\lambda_1$ iff G bipartite. u_2 gives the partition.

Outline

- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - **Walks of length *k***
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

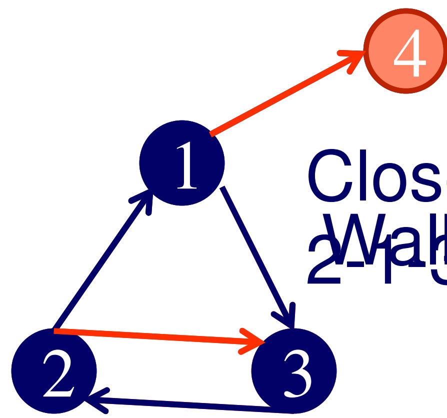
Walks

- A walk of length r in a directed graph:

$$u_0 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_r$$

where a node can be used more than once.

- Closed walk when: $u_0 = u_r$



Closed walk of length 3
2-1-3-2

Walk of length 2
2-1-4

Walks

Theorem: G(V,E) directed graph, adjacency matrix A. The number of walks from node u to node v in G with length r is $(A^r)_{uv}$

Proof: Induction on k. See Doyle-Snell, p.165

Walks

Theorem: G(V,E) directed graph, adjacency matrix A. The number of walks from node u to node v in G with length r is $(A^r)_{uv}$

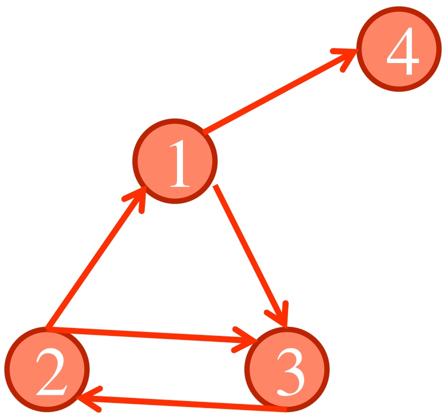
$$A = \begin{bmatrix} \; a_{ij}^1 \; \end{bmatrix}, \quad A^2 = \begin{bmatrix} \; a_{ij}^2 \; \end{bmatrix}, \ldots, A^r = \begin{bmatrix} \; a_{ij}^r \; \end{bmatrix}$$

a_{ij}^1 → (i,j)

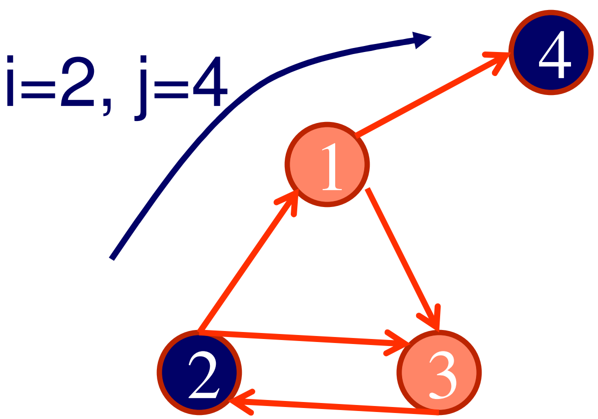
a_{ij}^2 → $(i, i_1),(i_1,j)$

a_{ij}^r → $(i,i_1),\ldots,(i_{r-1},j)$

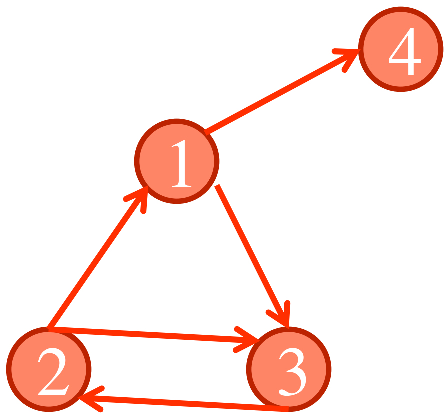
Walks



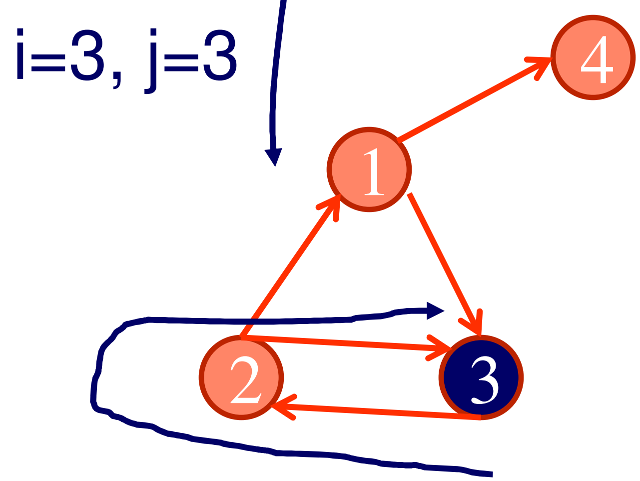
$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$



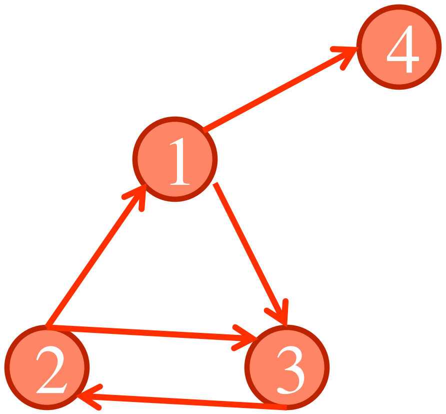
Walks



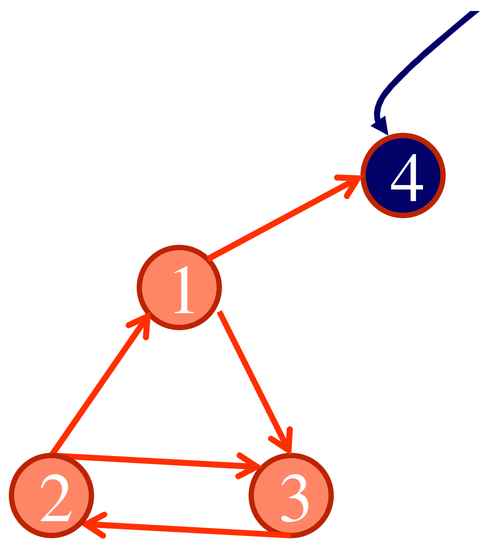
$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$



Walks



$$A^6 = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 3 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$



Always 0,
node 4 is a sink

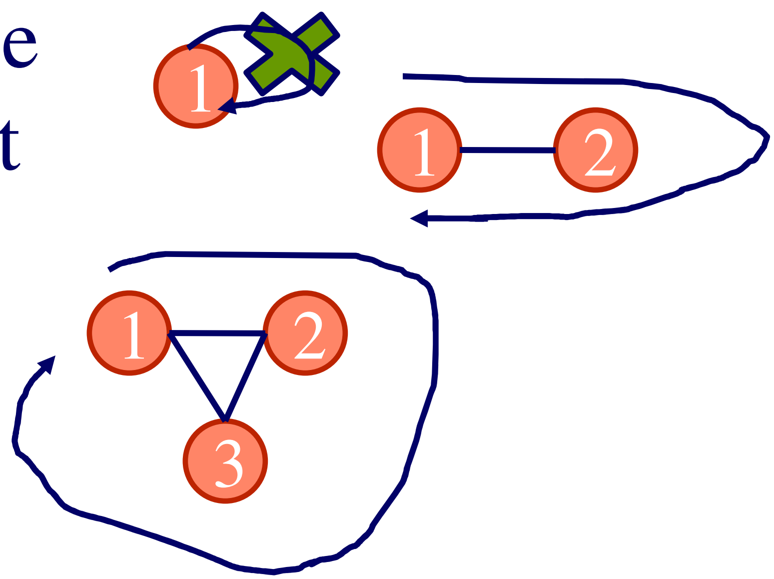
Walks

Corollary: If A is the adjacency matrix of undirected G(V,E) (no self loops), e edges and t triangles. Then the following hold:

a) $trace(A) = 0$

b) $trace(A^2) = 2e$

c) $trace(A^3) = 6t$



Walks

Corollary: If A is the adjacency matrix of undirected G(V,E) (no self loops), e edges and t triangles. Then the following hold:

a) trace(A) = 0

b) trace(A^2) = 2e

c) trace(A^3) = 6t

Computing A^r may be expensive!

Remark: virus propagation

The earlier result makes sense now:

- The higher the first eigenvalue, the more paths available ->
- Easier for a virus to survive

Outline

- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length *k*
- **Laplacian**
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

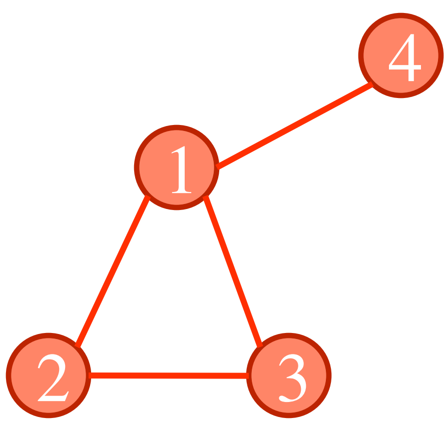
Main upcoming result

the second eigenvector of the Laplacian (u_2) gives a good cut:

- Nodes with positive scores should go to one group
- And the rest to the other

Laplacian

$$L_{uv} = \begin{cases} \mathrm{d}_u & \text{if } u = v \\ -1 & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

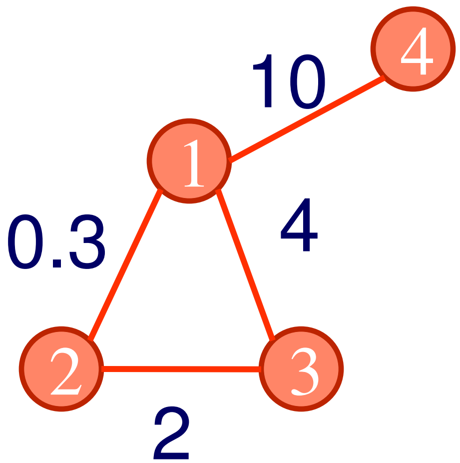


$$L = D - A = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Diagonal matrix, $d_{ii}=d_i$

Weighted Laplacian

$$L_{uv} = \begin{cases} d_u = \sum_v w_{uv} & \text{if } u = v \\ -w_{uv} & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$



$$L = \begin{pmatrix} 14.3 & -0.3 & -4 & -10 \\ -0.3 & 2.3 & -2 & 0 \\ -4 & -2 & 6 & 0 \\ -10 & 0 & 0 & 10 \end{pmatrix}$$

Outline

- **Reminders**
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length k
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

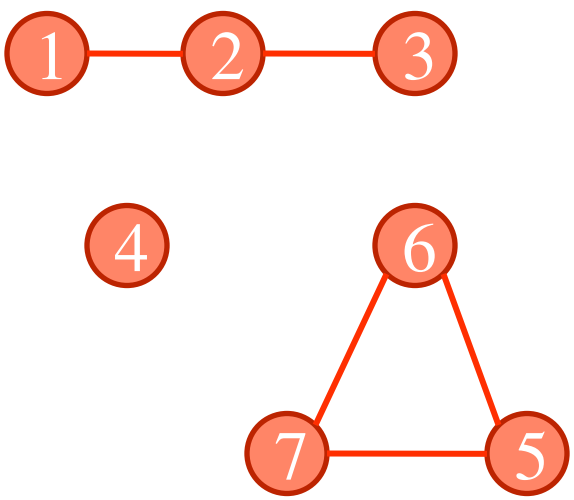
Connected Components

- **Lemma**: Let G be a graph with n vertices and c connected components. If L is the Laplacian of G, then rank(L)= n-c.

- **Proof**: see p.279, Godsil-Royle

Connected Components

G(V,E)



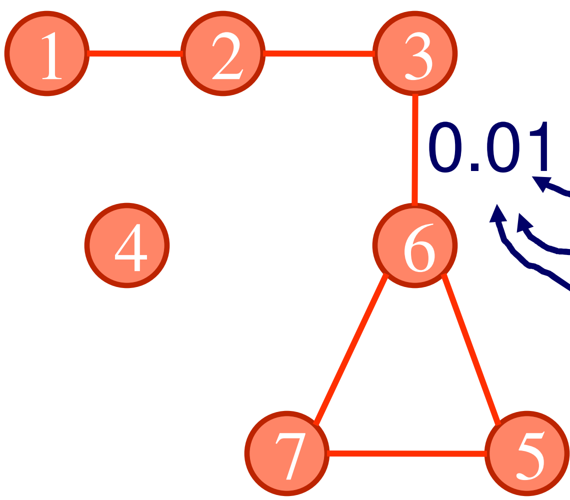
$$L=\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

#zeros = #components

$$\text{eig}(L)=\begin{pmatrix} 0 & 0 & 0 & 1 & 3 & 3 & 3 \end{pmatrix}$$

Connected Components

G(V,E)



$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1.01 & 0 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -0.1 & 0 & -1 & 2.01 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

#zeros = #components

Indicates a “good cut”

$$eig(L) = \begin{pmatrix} 0 & 0 & 0.0066 & 1.005 & 3 & 3 & 3.0084 \end{pmatrix}$$

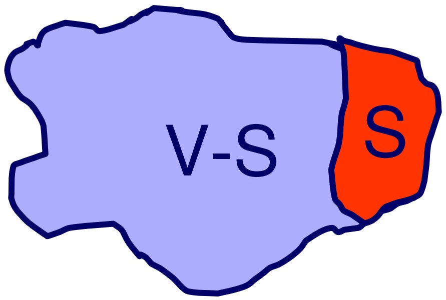
Outline

- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length *k*
- Laplacian
  - Connected Components
  - **Intuition: Adjacency vs. Laplacian**
  - Cheeger Inequality and Sparsest Cut:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Adjacency vs. Laplacian Intuition

details

Let **x** be an indicator vector:



$$x_i = 1, \text{if } i \in S$$

$$x_i = 0, \text{if } i \notin S$$

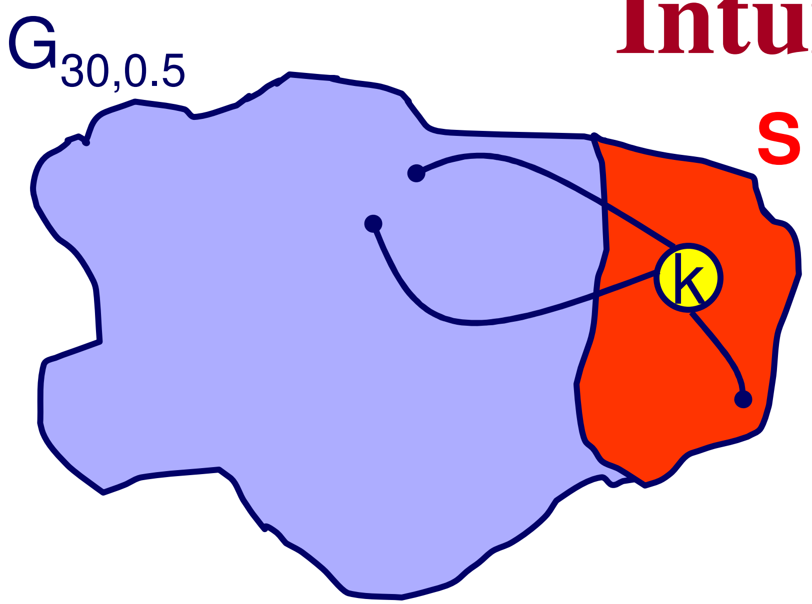
Consider now y=Lx

k-th coordinate

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

details

Adjacency vs. Laplacian Intuition



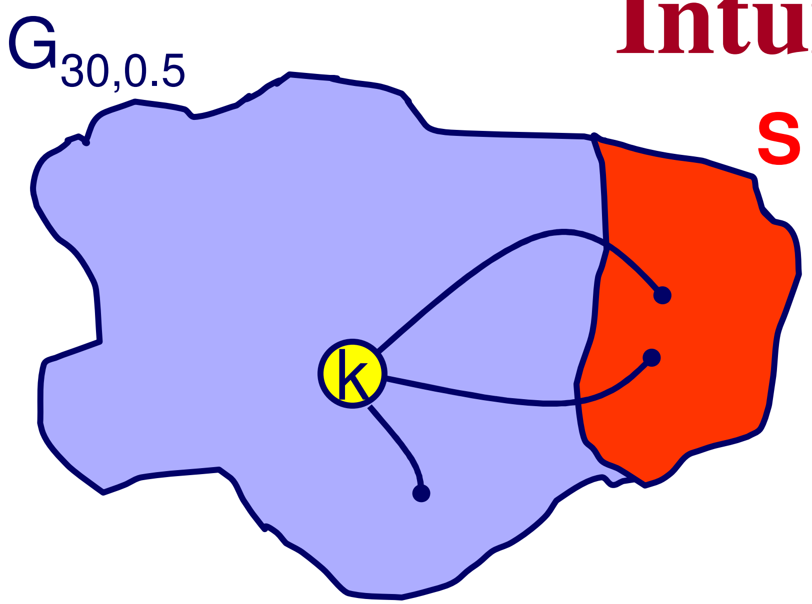
Consider now y=Lx

$$y_k > 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

details

Adjacency vs. Laplacian Intuition



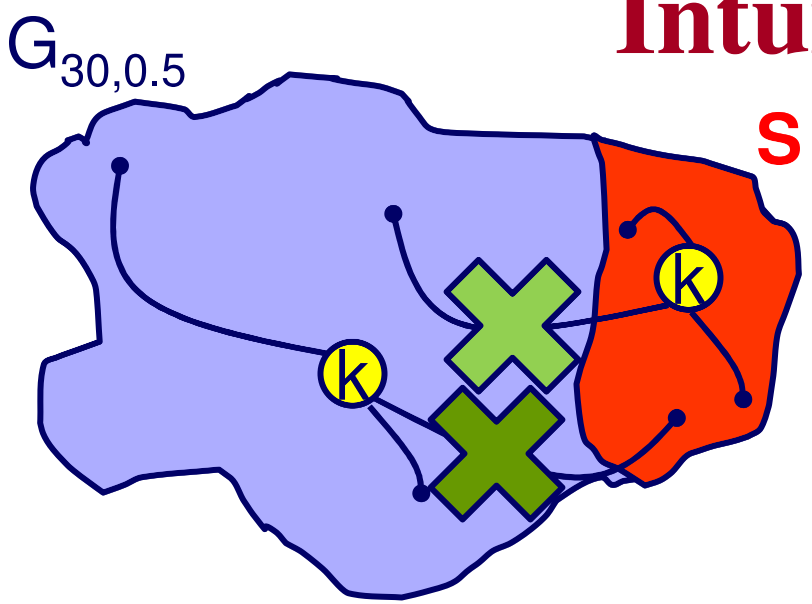
Consider now y=Lx

$$y_k < 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

details

Adjacency vs. Laplacian Intuition



Consider now y=Lx

$$y_k = 0$$

$$y_k = (Lx)_k = d_k x_k - \sum_{j:(j,k)\in E(G)} x_j$$

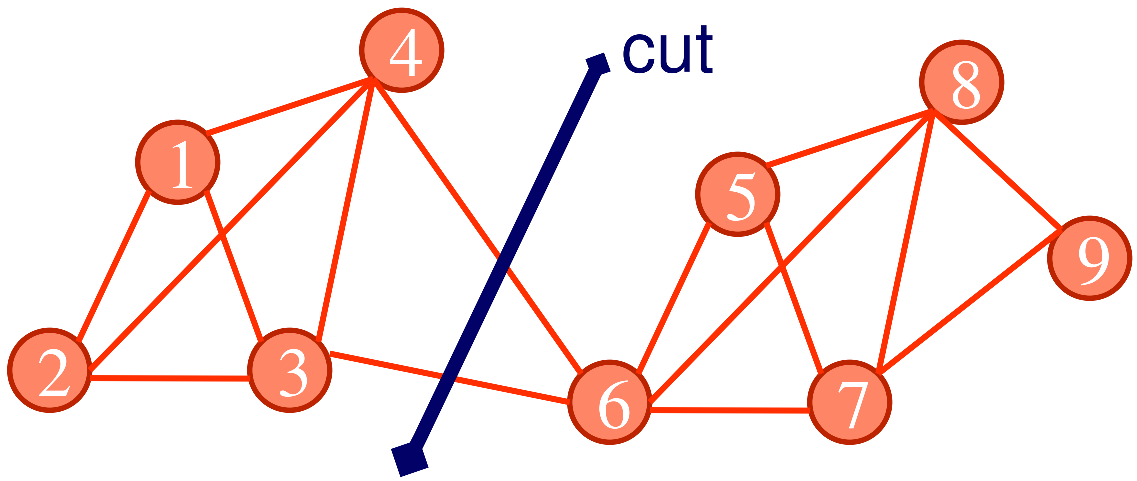
Laplacian: connectivity,
Adjacency: #paths

Outline

- Reminders
- Adjacency matrix
  - Intuition behind eigenvectors: Eg., Bipartite Graphs
  - Walks of length *k*
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - **Sparsest Cut and Cheeger inequality**:
    - Derivation, intuition
    - Example
- Normalized Laplacian

Why Sparse Cuts?

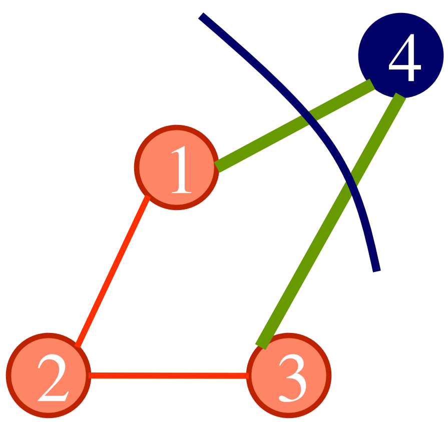
- Clustering, Community Detection



- And more: Telephone Network Design, VLSI layout, Sparse Gaussian Elimination, Parallel Computation

Quality of a Cut

- Isoperimetric number φ of a cut *S*:

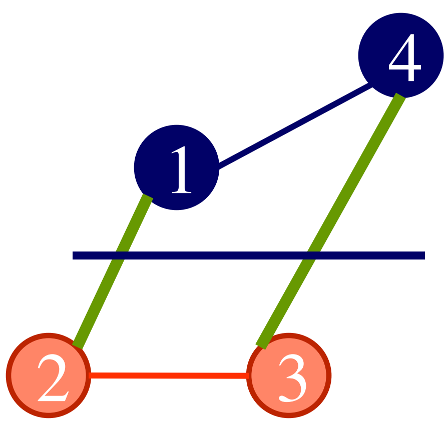


$$\phi(S) = \frac{e(S, V-S)}{\min(|S|, |V-S|)}$$

$$\phi(\{4\}) = \frac{2}{min(1,3)} = 2$$

Quality of a Cut

- Isoperimetric number φ of a **graph** = score of best cut:



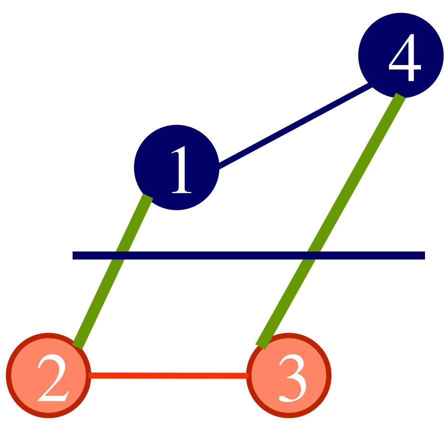
$$\phi(G) = \min_{S \subseteq V} \phi(S)$$

$$\phi(\{1,4\}) = \frac{2}{min(2,2)} = 1$$

and thus $\phi(G) = 1$

Quality of a Cut

- Isoperimetric number φ of a **graph** = score of best cut:



Best cut: hard to find

BUT: Cheeger's inequality gives bounds

λ_2: Plays major role

Let's see the intuition behind λ_2

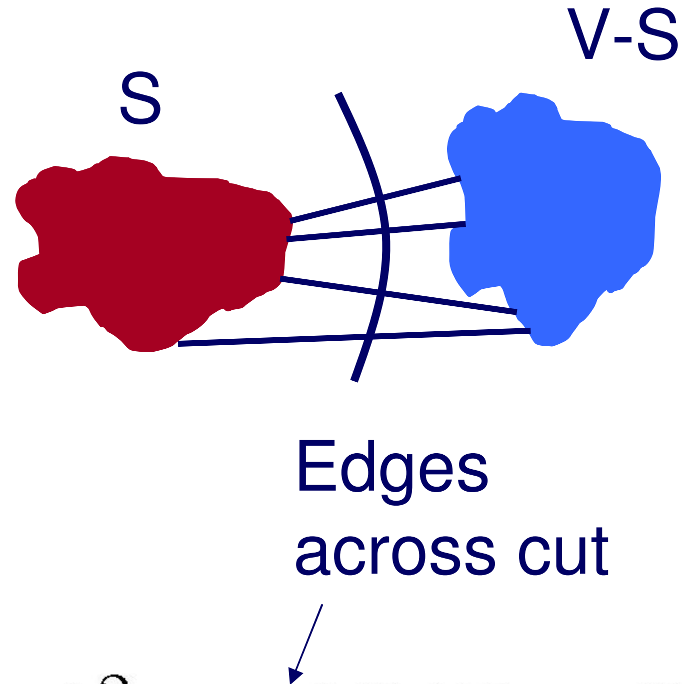
Laplacian and cuts - overview

- A cut corresponds to an indicator vector (ie., 0/1 scores to each node)
- Relaxing the 0/1 scores to real numbers, gives eventually an alternative definition of the eigenvalues and eigenvectors

Why λ_2?

Characteristic Vector **x**

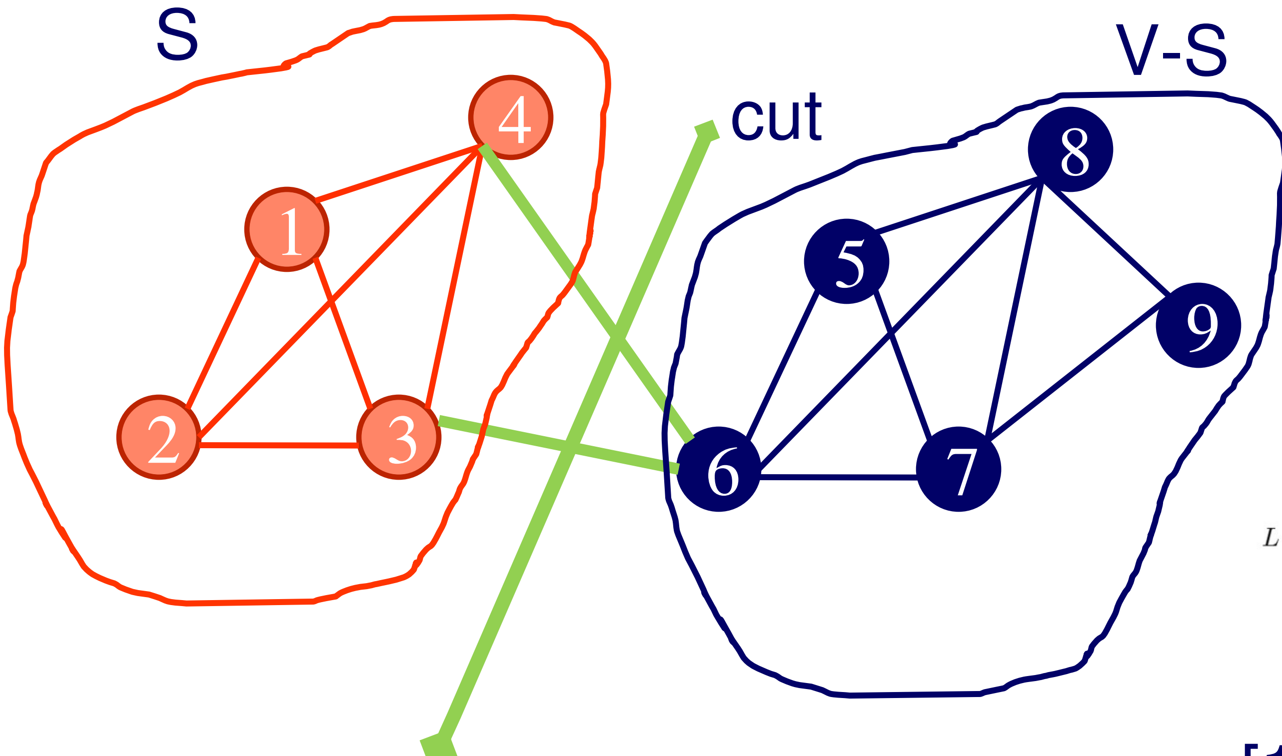
- $x_i = 1, \text{if } i \in S$
- $x_i = 0, \text{if } i \notin S$



Then:

$$x^T L x = \sum_{(i,j) \in E(G)} (x_i - x_j)^2 = e(S, V - S)$$

Why λ_2?



$$L = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 4 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & -1 & 5 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$x^TLx=2$

$x=[1,1,1,1,0,0,0,0,0]^T$

details

Why λ_2?

$$r(S) = \frac{e(S,V-S)}{|S||V-S|} \Rightarrow \frac{\phi(S)}{n} \le r(S) \le \frac{\phi(S)}{\frac{n}{2}}$$

Ratio cut

Sparsest ratio cut $$r(G) = \min_{S \subset V} r(S) = \min_{x \in \{0,1\}^n} \frac{1}{n} \frac{x^T L x}{x^T x}$$

NP-hard

Relax the constraint: $x \in \{0,1\}^n \rightarrow x \in \mathbb{R}^n$

Normalize: $\sum_i x_i = 0$

→ ?

details

Why λ_2?

Sparsest ratio cut $$r(G) = \min_{S \subset V} r(S) = \min_{x \in \{0,1\}^n} \frac{1}{n} \frac{x^T L x}{x^T x}$$

NP-hard

Relax the constraint: $x \in \{0,1\}^n \rightarrow x \in \mathbb{R}^n$

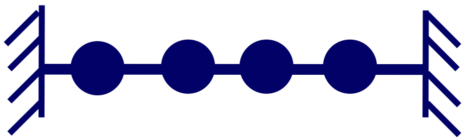
Normalize: $\sum_i x_i = 0$

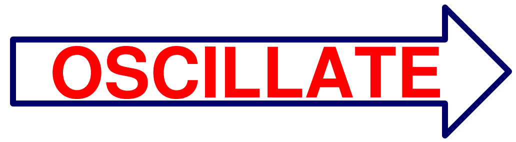
→ λ_2

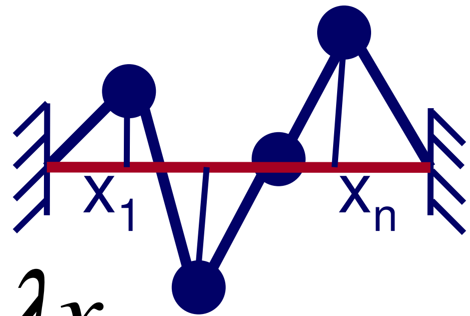
because of the Courant-Fisher theorem (applied to *L*)

$$\lambda_2 = \min_{\sum_i u_i = 0, u \neq 0} \frac{u^T L u}{u^T u} = \min_{\sum_i u_i = 0, u \neq 0} \frac{\sum_{(i,j) \in E(G)} (u_i - u_j)^2}{\sum_i u_i^2}$$

Why λ_2?







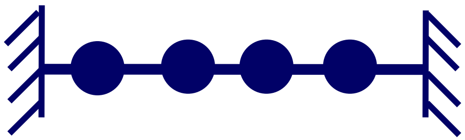
Each ball 1 unit of mass

$$Lx = \lambda x$$

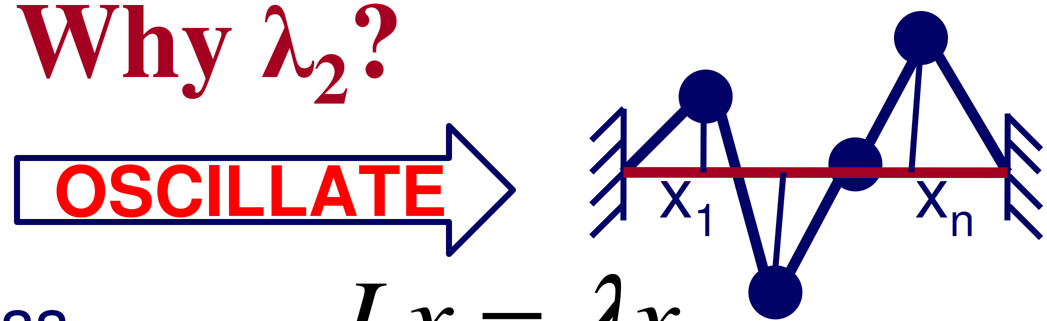
Dfn of eigenvector

Matrix viewpoint:

Why λ_2?



Each ball 1 unit of mass



$$Lx = \lambda x$$

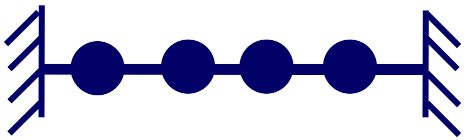
Force due to neighbors (Lx)

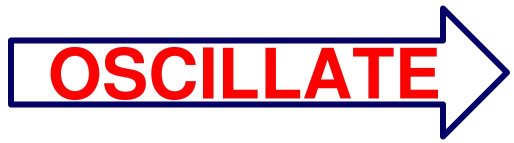
Hooke's constant (λ)

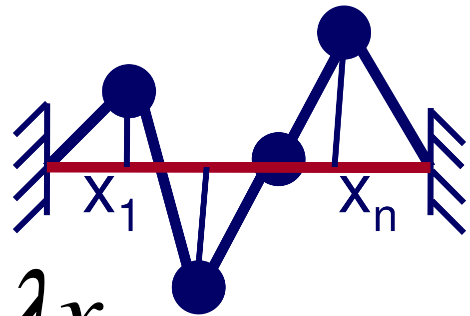
displacement (x)

Physics viewpoint:

Why λ_2?

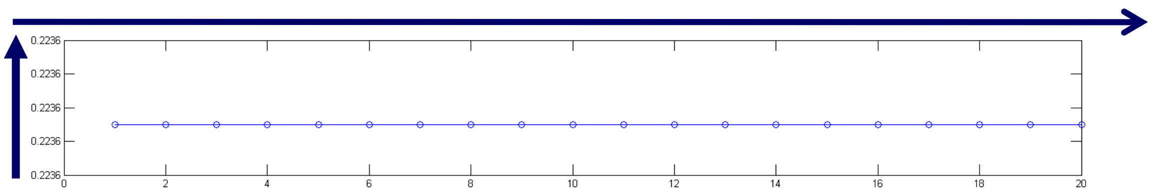






Each ball 1 unit of mass

$$Lx = \lambda x$$

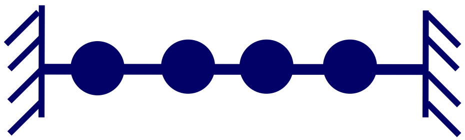


Node id

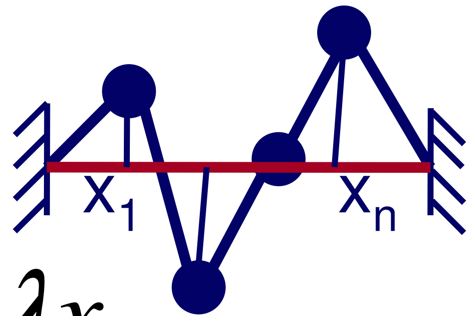
Eigenvector value

For the first eigenvector:
All nodes: same displacement (= value)

Why λ_2?



OSCILLATE

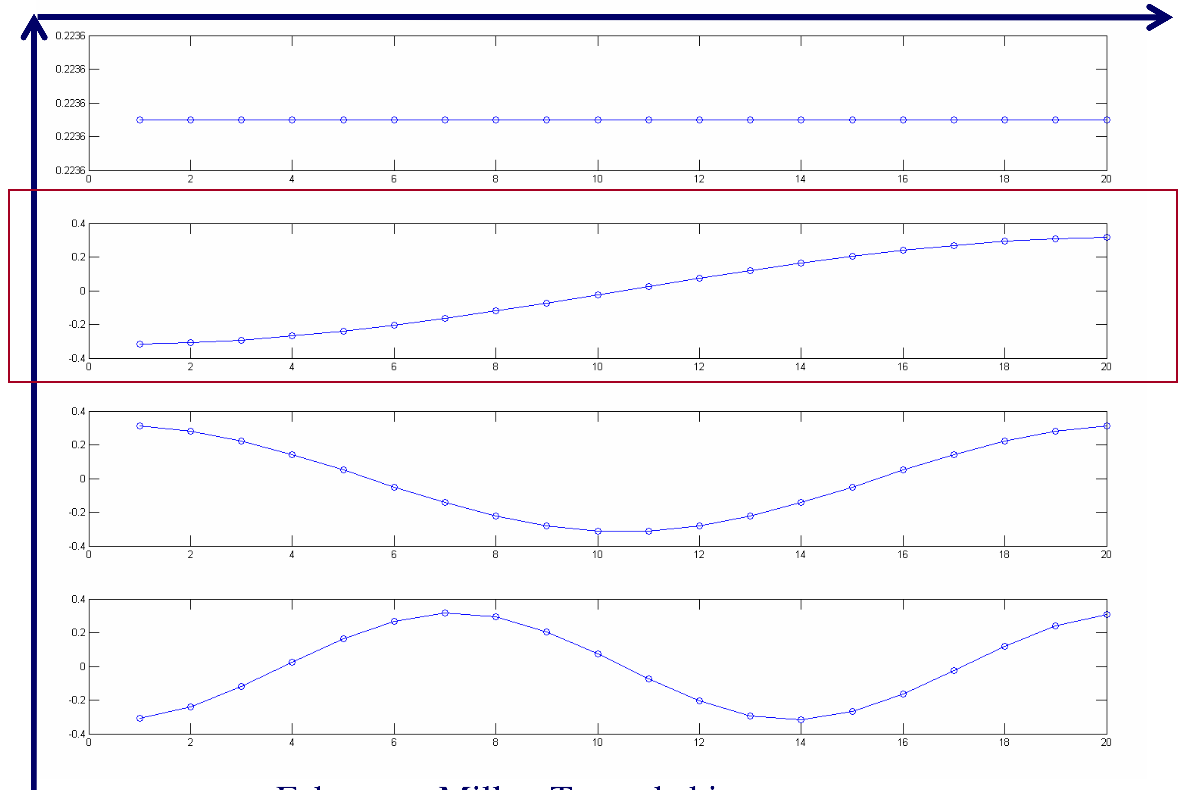


Each ball 1 unit of mass

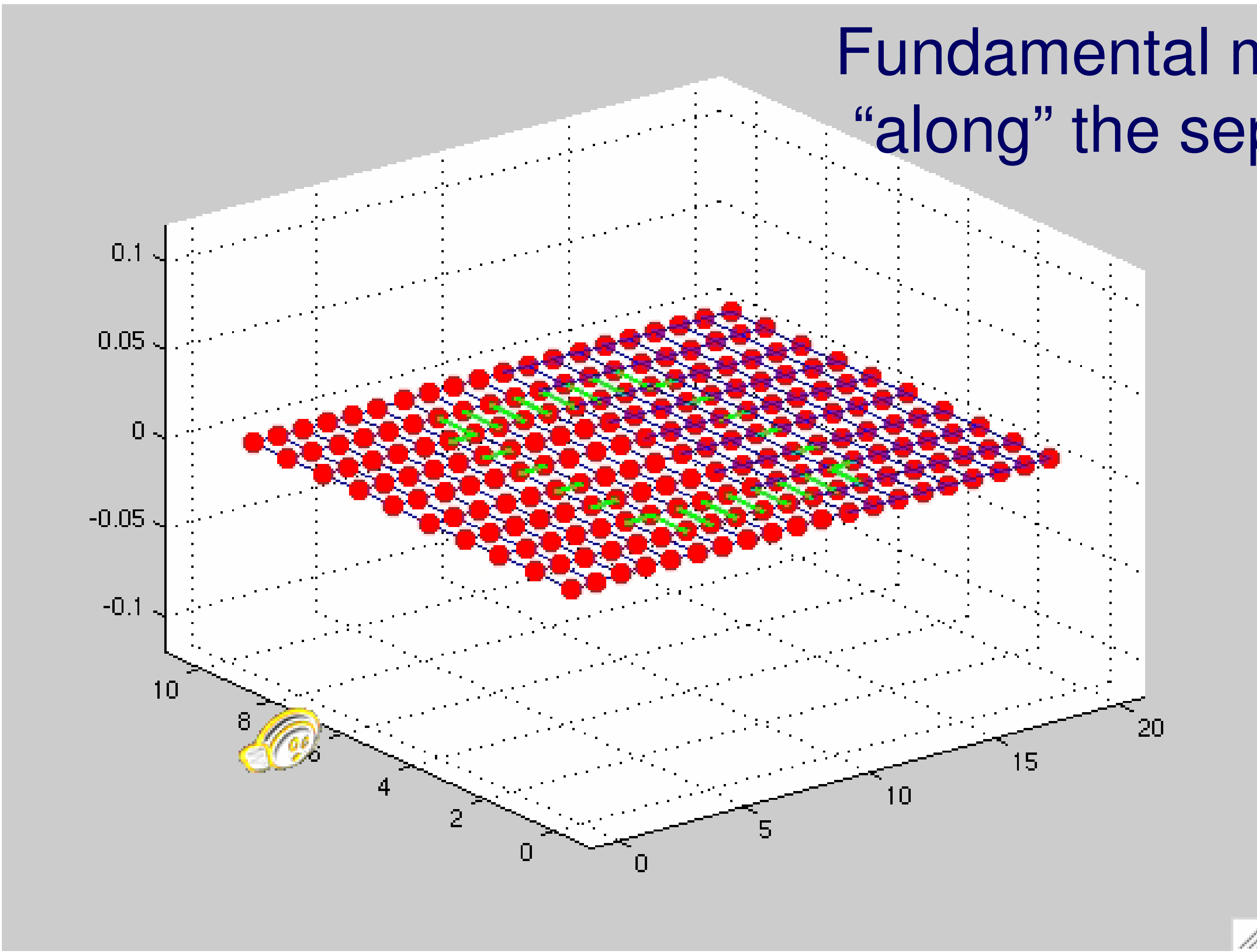
$$Lx = \lambda x$$

Node id

Eigenvector value



Why λ_2?



Fundamental mode of vibration:
"along" the separator

Cheeger Inequality

Score of best cut
(**hard** to compute)

$$\frac{\phi^2}{2d_{max}} \leq \lambda_2 \leq 2\phi(G)$$

Max degree

2nd smallest eigenvalue
(**easy** to compute)

Cheeger Inequality and graph partitioning heuristic:

$$\frac{\phi^2}{2d_{max}} \leq \lambda_2 \leq 2\phi(G)$$

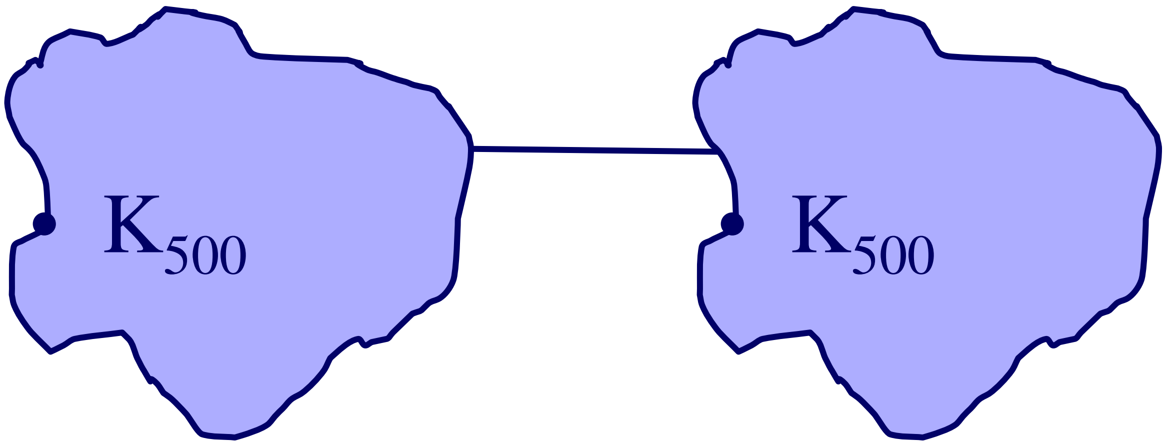
- Step 1: Sort vertices in non-decreasing order according to their score of the second eigenvector
- Step 2: Decide where to cut.
  - Bisection
  - **Best ratio cut**

Two common heuristics

Outline

- Reminders
- Adjacency matrix
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Sparsest Cut and Cheeger inequality:
    - Derivation, intuition
    - **Example**
- Normalized Laplacian

Example: Spectral Partitioning



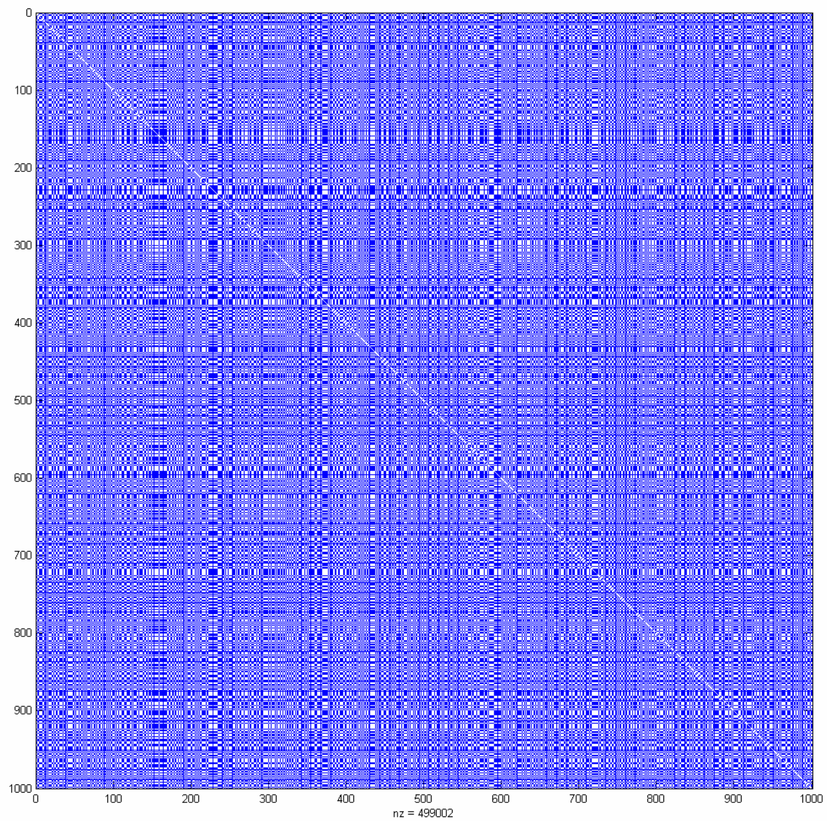
dumbbell graph

A = zeros(1000);
A(1:500,1:500)=ones(500)-eye(500);
A(501:1000,501:1000)= ones(500)-eye(500);
myrandperm = randperm(1000);
B = A(myrandperm,myrandperm);

In social network analysis,
such clusters are called communities

Example: Spectral Partitioning

- This is how adjacency matrix of B looks

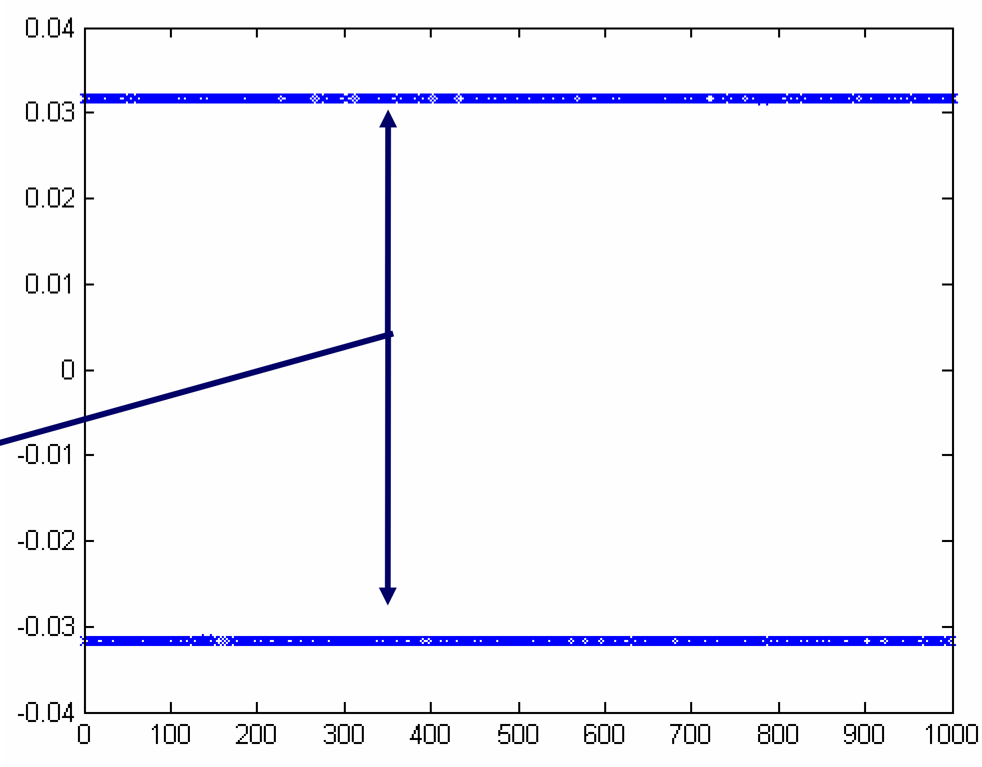


spy(B)

Example: Spectral Partitioning

- This is how the 2nd eigenvector of B looks like.

```
L = diag(sum(B))-B;
[u v] = eigs(L,2,'SM');
plot(u(:,1),'x')
```

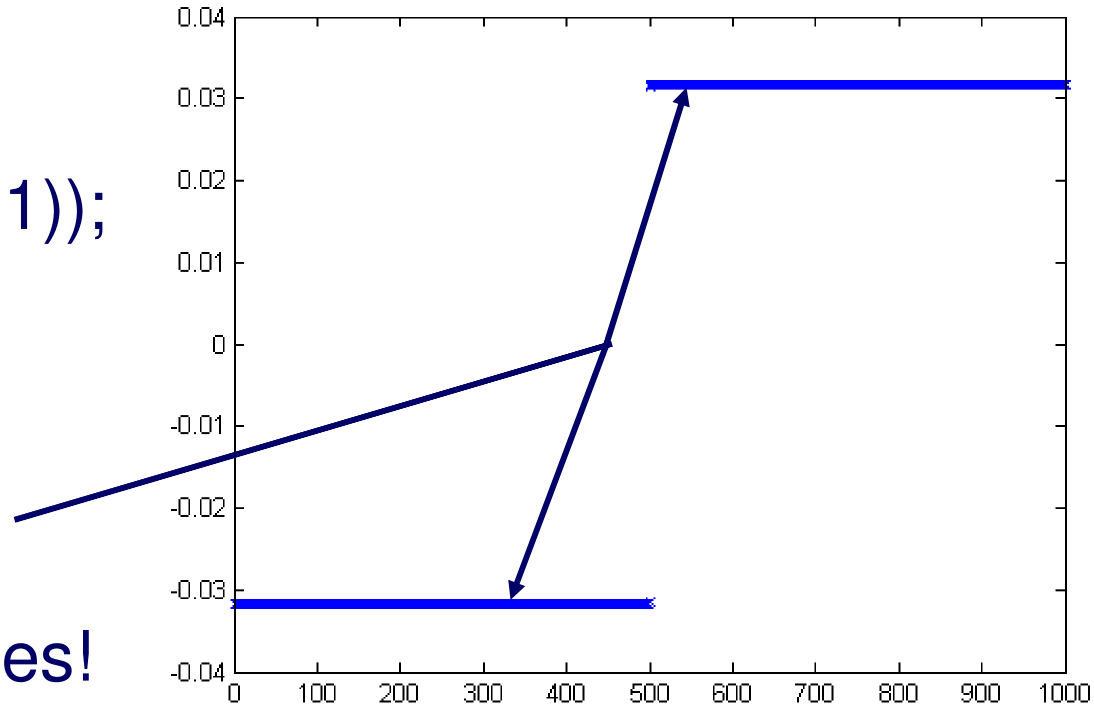


Not so much information yet…

Example: Spectral Partitioning

- This is how the 2nd eigenvector looks if we sort it.

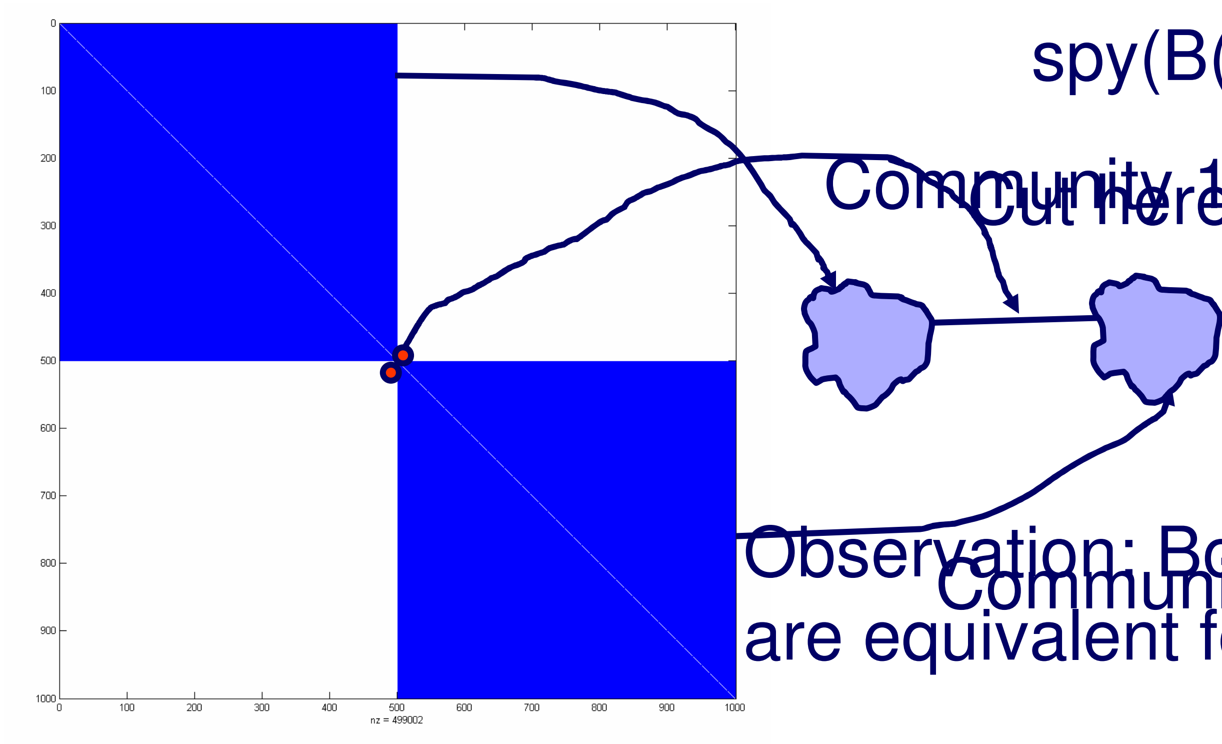
```
[ign ind] = sort(u(:,1));
plot(u(ind),'x')
```



But now we see the two communities!

Example: Spectral Partitioning

- This is how adjacency matrix of B looks now

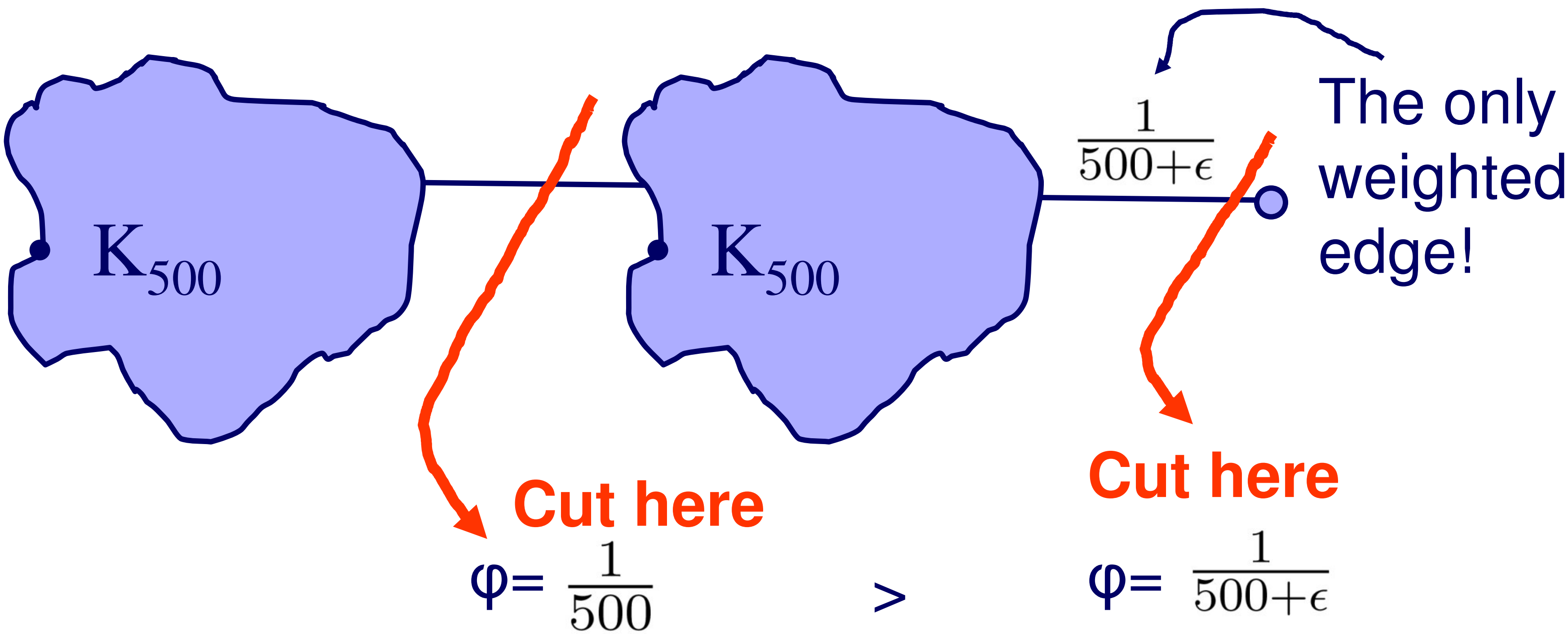


Observation: Both heuristics are equivalent for the dumbbell

Outline

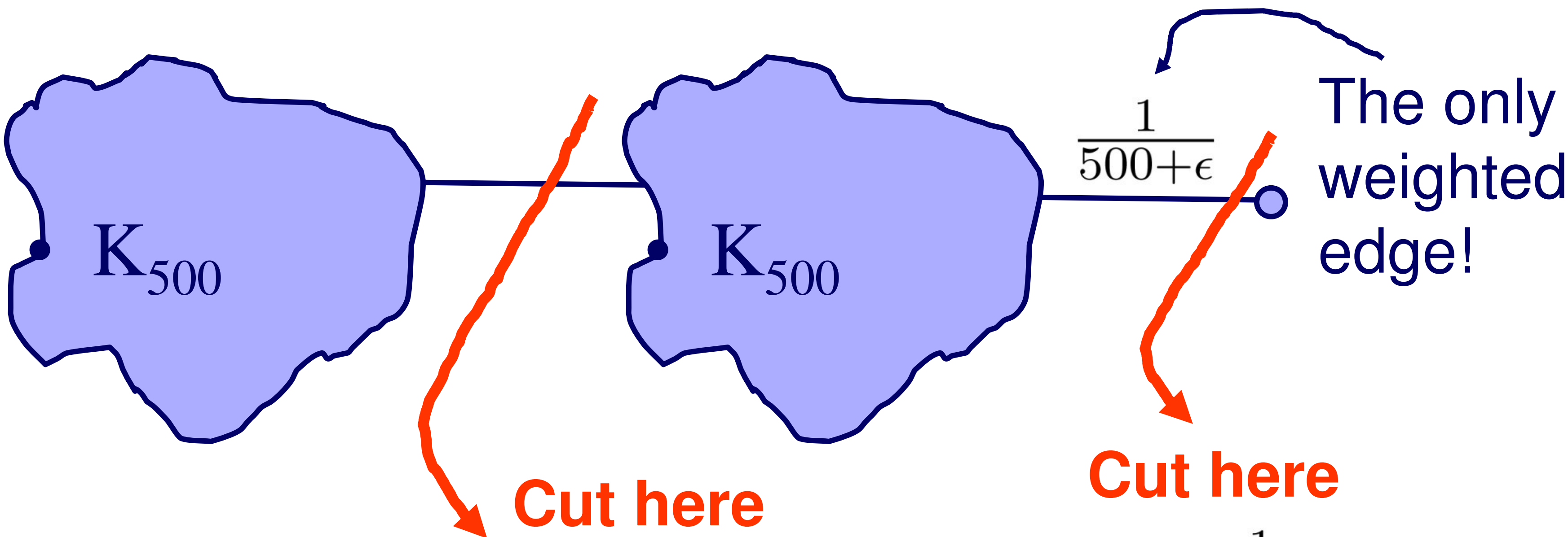
- Reminders
- Adjacency matrix
- Laplacian
  - Connected Components
  - Intuition: Adjacency vs. Laplacian
  - Sparsest Cut and Cheeger inequality:
- Normalized Laplacian

Why Normalized Laplacian



So, φ is not good here…

Why Normalized Laplacian



Optimize Cheeger constant h(G), balanced cuts

$$h_G = \min_S h_G(S)$$

$\varphi = \frac{1}{500}$ > $\varphi = \frac{1}{500+\epsilon}$

where

$$h(S) = \frac{e(S, V-S)}{\min(vol(S), vol(V-S))}$$

$$vol(S) = \sum_{v \in S} d_v$$

Extensions

- Normalized Laplacian
  - Ng, Jordan, Weiss Spectral Clustering
  - Laplacian Eigenmaps for Manifold Learning
  - Computer Vision and many more applications…

Standard reference: Spectral Graph Theory Monograph by Fan Chung Graham

Conclusions

Spectrum tells us a lot about the graph:

- Adjacency: #Paths
- Laplacian: Sparse Cut
- Normalized Laplacian: Normalized cuts, tend to avoid unbalanced cuts

References

- Fan R. K. Chung: *Spectral Graph Theory* (AMS)
- Chris Godsil and Gordon Royle: *Algebraic Graph Theory* (Springer)
- Bojan Mohar and Svatopluk Poljak: *Eigenvalues in Combinatorial Optimization*, IMA Preprint Series #939
- Gilbert Strang: *Introduction to Applied Mathematics* (Wellesley-Cambridge Press)