

Algebraic Representation of Networks

$$\begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Hiroki Sayama
sayama@binghamton.edu

Describing networks with matrices (1)

- **Adjacency matrix**

A matrix with rows and columns labeled by nodes, where a_{ij} represents the number of edges between node i and node j
(must be symmetric for undirected graph)

- **Incidence matrix (not discussed much)**

A matrix with rows labeled by nodes and columns labeled by edges, where a_{ij} indicates whether edge j is connected to node i (1) or not (0)

Describing networks with matrices (2)

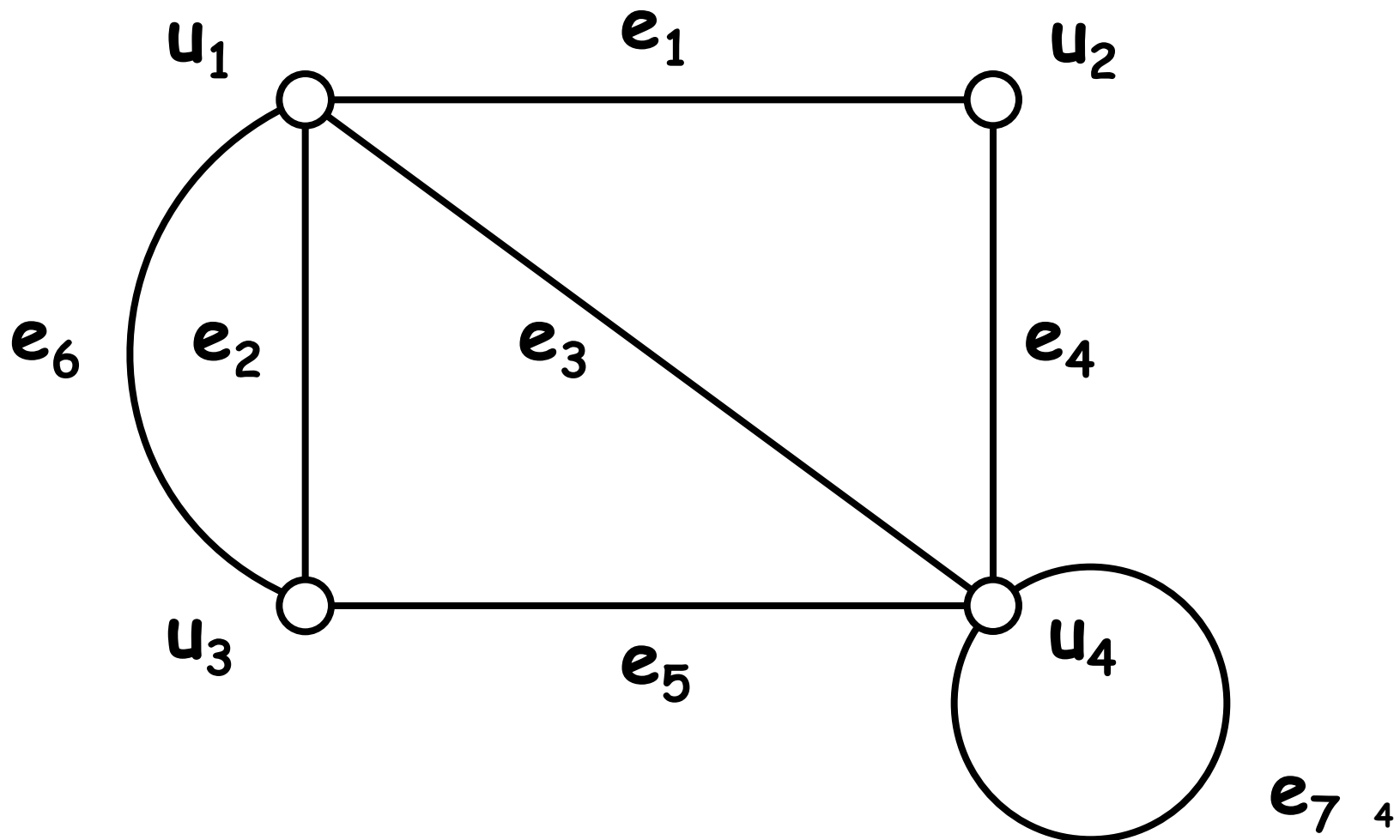
- **Transition probability matrix**

A matrix with rows and columns labeled by states (nodes), where a_{ij} represents the probability of transition from state (node) i to state (node) j

- **Laplacian matrix**

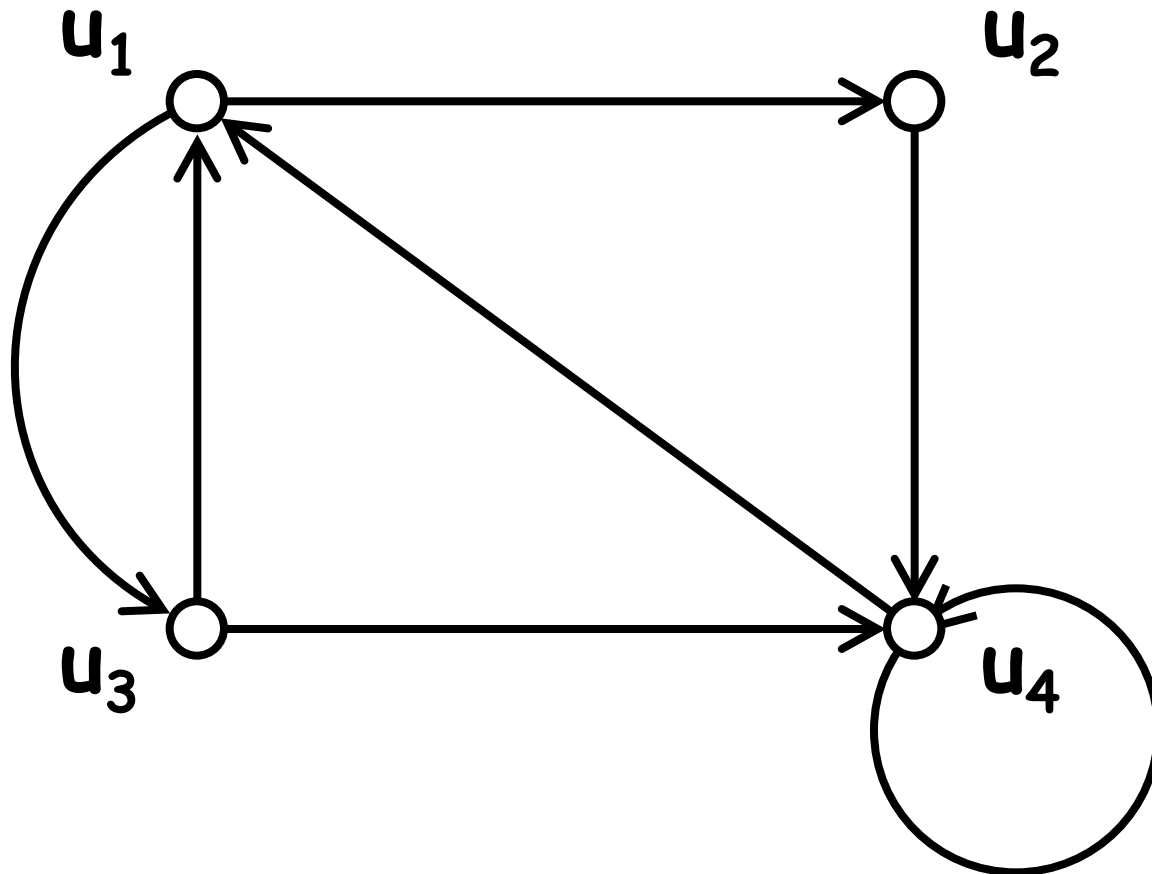
A matrix with rows and columns labeled by nodes, where a_{ij} represents node degree if $i = j$, or is -1 if node i and node j are connected

Exercise Write adjacency and incidence matrices of the (multi-)graph below



Exercise

- Write an adjacency matrix of the (multi-)graph below



Exercise

- Think about which node would be most suitable to be a source or a sink in a network represented by the adjacency matrix on the right
- Find the maximal flow of this network

0	0	3	0	0	0	0	0
0	0	0	0	0	2	4	0
0	0	0	0	0	0	0	0
0	0	0	0	0	3	5	0
0	6	0	4	0	0	0	5
0	0	4	0	0	0	0	0
2	0	7	0	0	0	0	0
2	0	2	2	0	0	0	0

Arithmetic Operations Applied to Adjacency Matrices

Sum and difference of adjacency matrices

- One can calculate a **sum** and a **difference** of adjacency matrices if the two graphs have the same number of nodes.

Adjacency matrix
of graph A

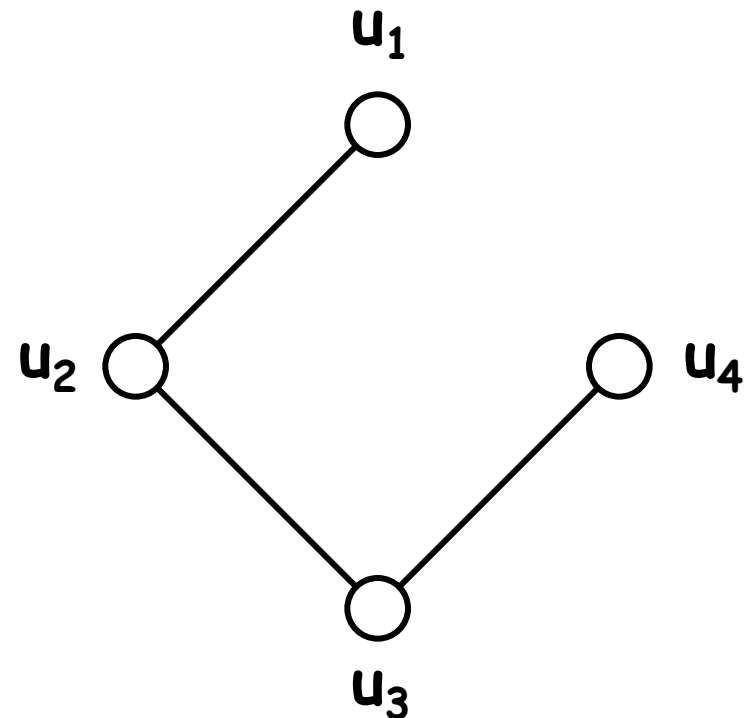
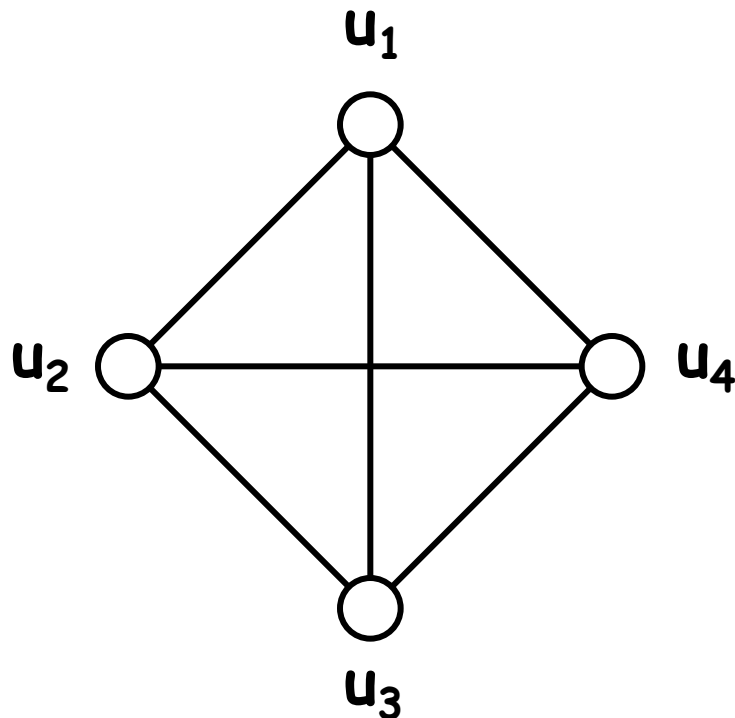
Adjacency matrix
of graph B

$$A + B$$

Sum of the two adjacency matrices

Exercise

- Calculate the sum of and the difference between the adjacency matrices of the following two graphs, and draw the actual shape of the resultant graphs



Product of adjacency matrices

- Similarly, one can calculate a **product** of two adjacency matrices (multiplication is not commutative)

Adjacency matrix
of graph A

Adjacency matrix
of graph B

A B

Product of the two
adjacent matrices (1)

$\neq$

not equal
in general

Adjacency matrix
of graph B

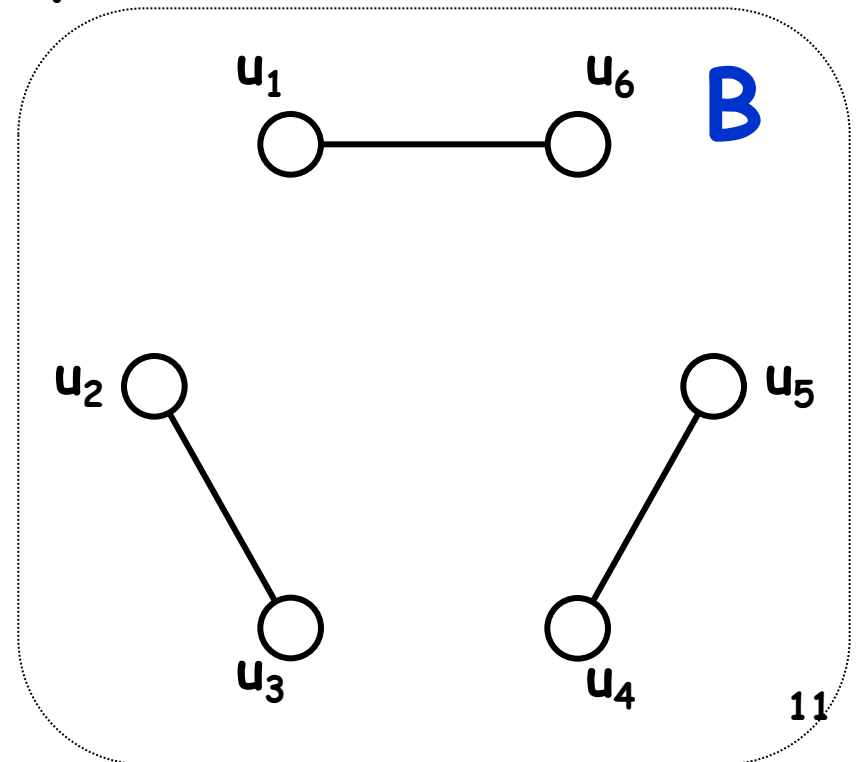
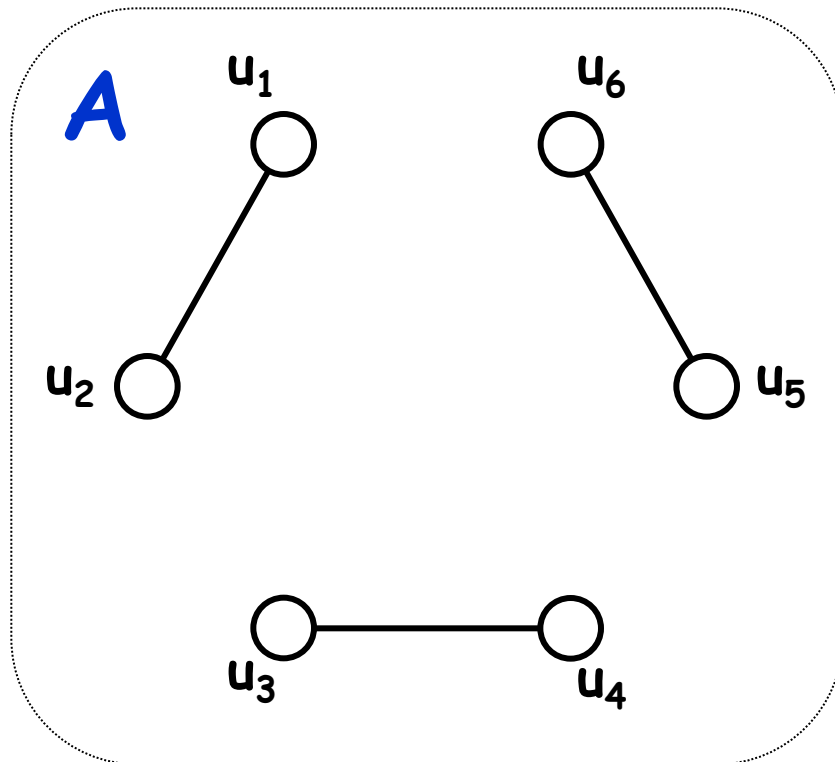
Adjacency matrix
of graph A

B A

Product of the two
adjacent matrices (2)

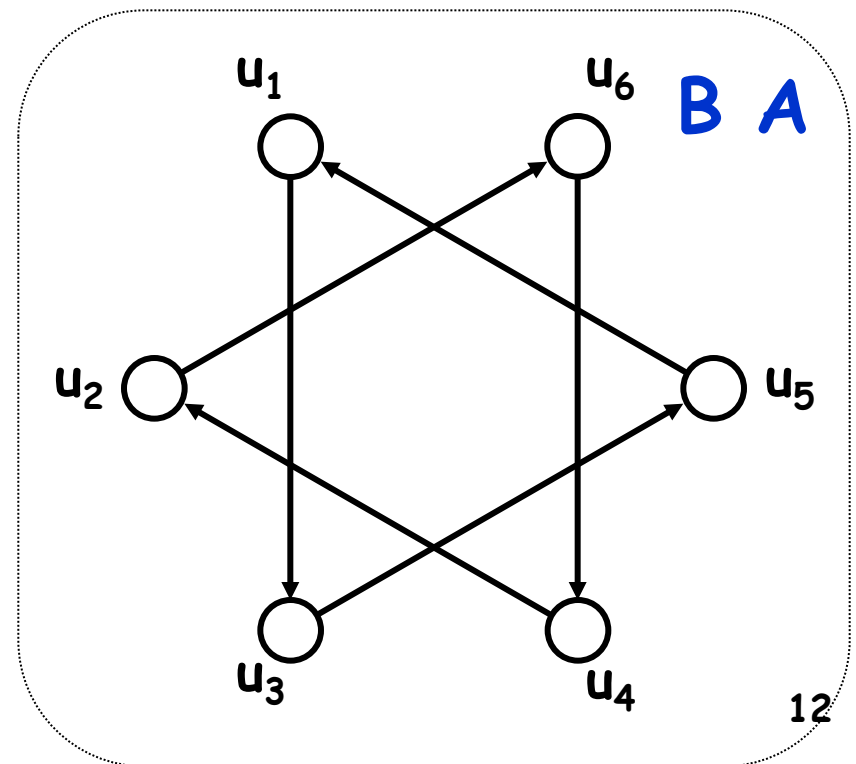
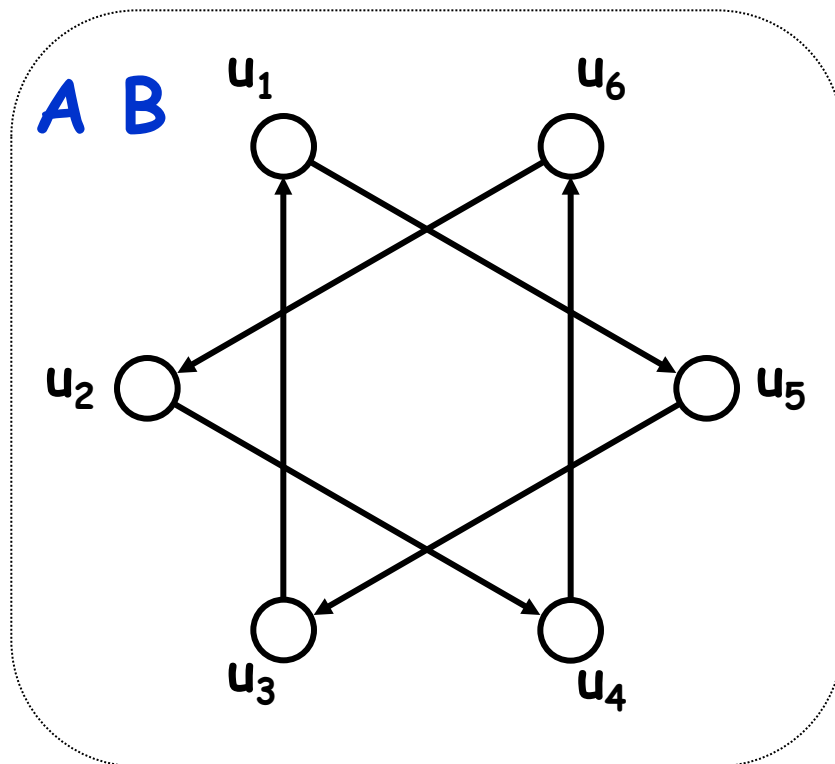
Exercise

- Calculate two different products of the adjacency matrices of the following two graphs, and draw the actual shape of the results (**Note: such multiplication may create directed graphs**)
- Then think about what the product means



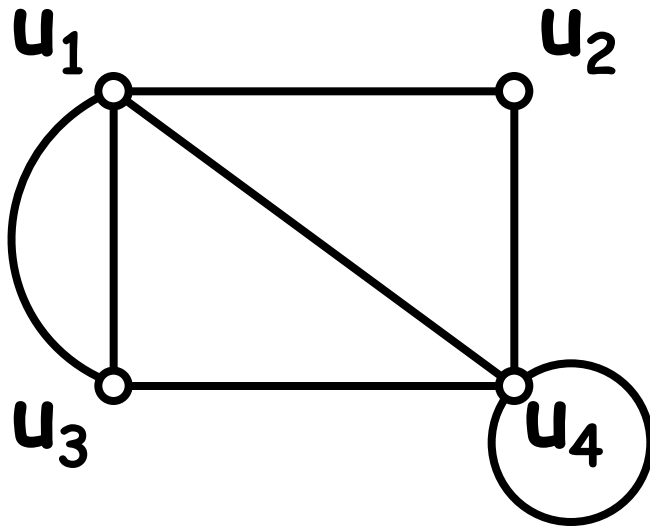
Answer

- Product $X \ Y$ indicates a directed graph that maps each node to a set of possible destinations that may be reached by a two-step move, first following Y and then X



Power of Adjacency Matrices

What does a power of an adjacency matrix mean?



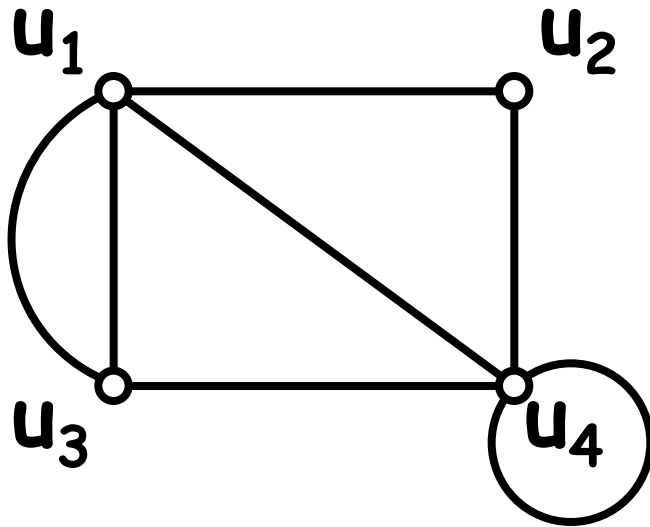
$$A = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$A^n \times ?$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Try to calculate Ax ,
 A^2x , A^3x , etc.

What does a power of an adjacency matrix mean?



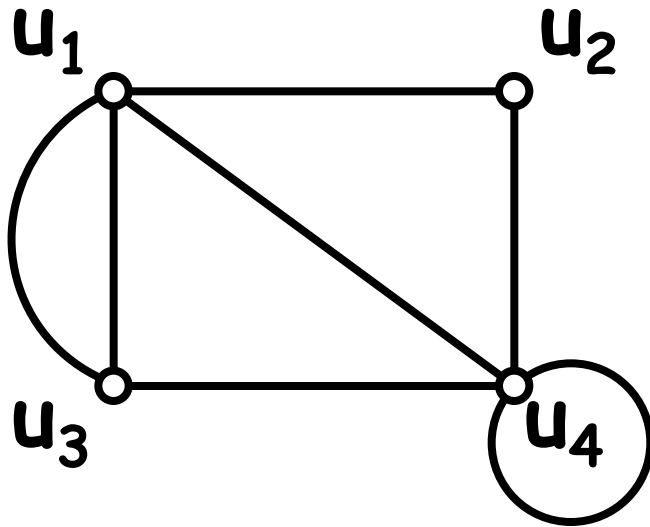
$$A = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$A^n \times ?$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

This formula gives a set of nodes that can be reached in n steps from node u_2 (and the # of such walks)

What does a power of an adjacency matrix mean?



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

$$A^n \mathbf{I} = A^n$$

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

Arranging all the results starting from every node gives a power of adjacency matrix A

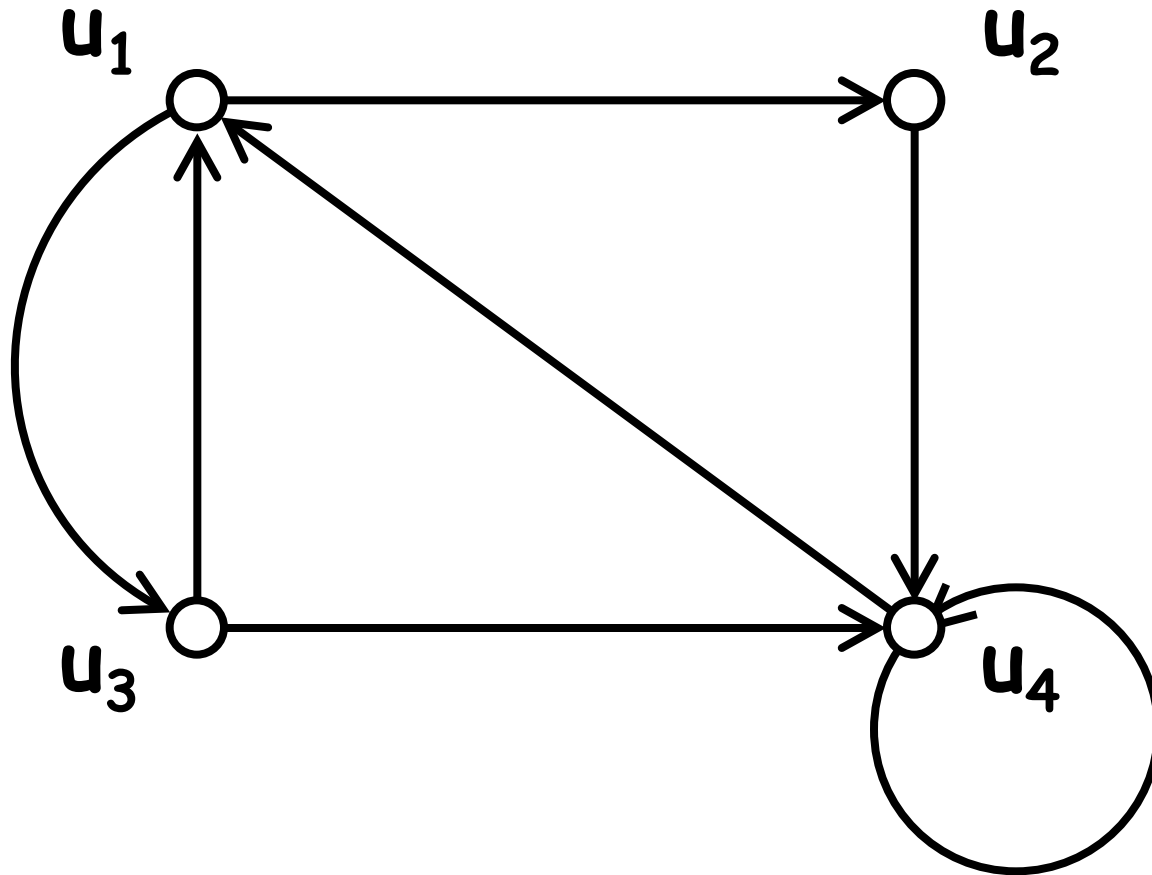
A theorem on the power of adjacency matrix

- In adjacency matrix A raised to the power of n , $(A^n)_{ij}$ gives the number of different walks of length n that starts at node j and ends at node i

(This applies to both undirected and directed graphs; proof can be easily obtained by using mathematical induction with n)

Exercise

- Calculate how many walks of length two exist between u_1 and every other node in the graph below



Exercise

- Using the power of an adjacency matrix, count the number of triangles included in:
 - (a) A complete graph made of 20 nodes
 - (b) An Erdos-Renyi random network made of 1000 nodes with connection probability 0.01

Determining graph connectivity

- A^n gives the number of different walks of length n between every pair of nodes
- $C_n = \sum_{k=1 \sim n} A^k$
gives the number of different walks of length n or shorter between every pair of nodes

Determining graph connectivity

- $C_n = \sum_{k=1 \sim n} A^k$

gives the number of different walks of length n or shorter between every pair of nodes

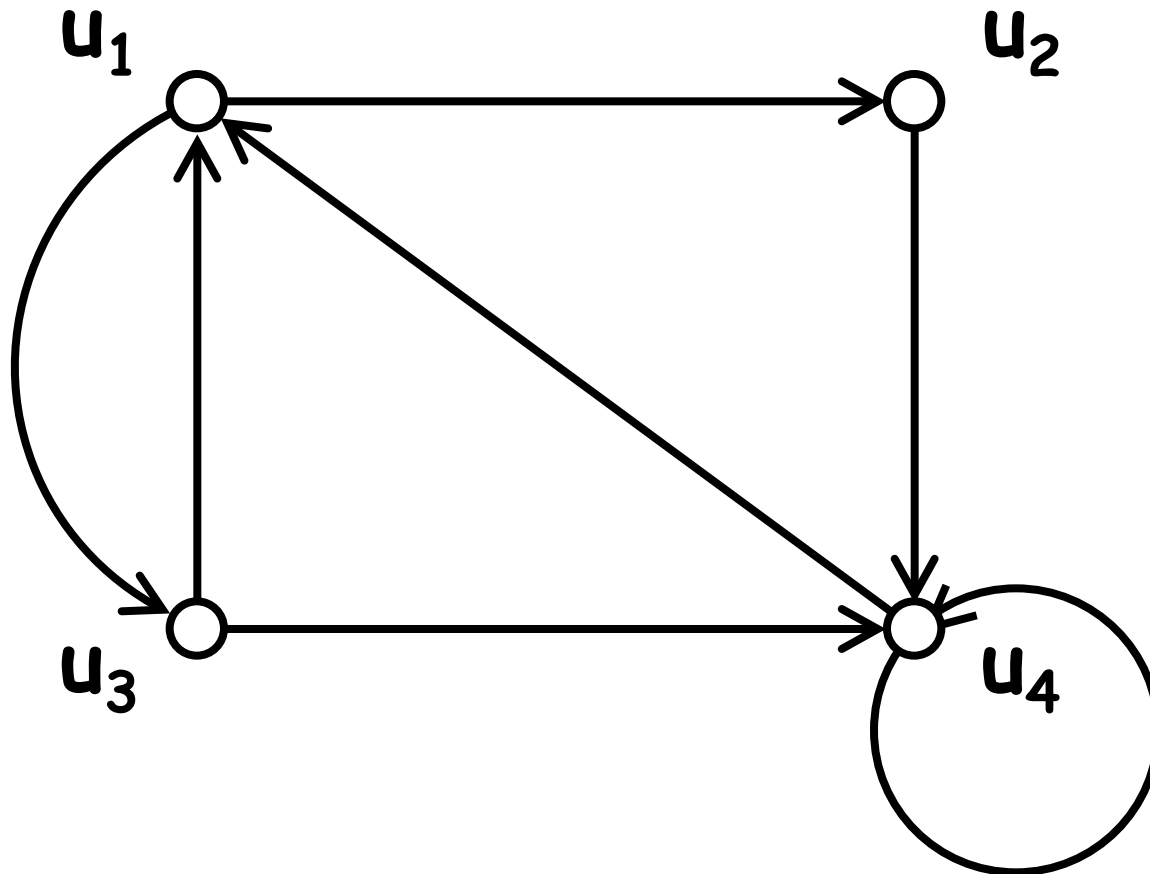
- In $C_{(\# \text{ of nodes} - 1)}$, every possible path in that graph should be counted
 - Because a path must not visit the same node more than once

Determining graph connectivity

- $C_{(\# \text{ of nodes} - 1)} = \sum_{k=1 \sim (\# \text{ of nodes} - 1)} A^k$
 - If $(C_{(\# \text{ of nodes} - 1)})_{ij} > 0$ for all $i \neq j$,
then there is a path between any pair
of nodes (and vice versa)
- $\Rightarrow$ The original graph is *numerically*
determined to be a (strongly)
connected graph

Exercise

- Show the strong connectivity of the graph below by calculating the sum of powers of its adjacency matrix



Exercise

- An alternative method is just to calculate $(A + I)^{(\# \text{ of nodes} - 1)}$ and check if all elements have positive values
 - Those values no longer show # of paths, but still tell us whether there are paths between each pair of nodes
- Why does this work?

Transitive closure

- Transitive closure of a graph is a graph that contains edge $\langle u, v \rangle$ whenever there is a path from node u to node v in the original graph
 - Obtained by making all diagonal components 0 and all non-diagonal **non-zero** components 1 in $C_{(\# \text{ of nodes} - 1)}$
 - Describes accessibility between nodes
 - Is a complete graph if the original graph is (strongly) connected

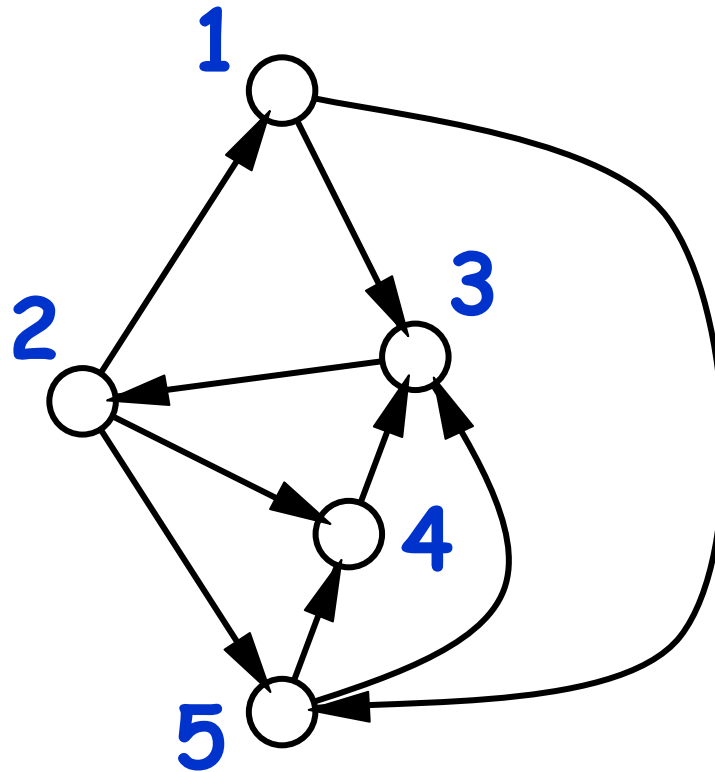
Transition Probability Matrix

Transition probability matrix

- An adjacency matrix of a directed graph with normalized weights (i.e., sum of all weights of outgoing links is always 1 for every node)
 - Considers each node as a “state”, and a directed link as a stochastic “state transition”: Representing a Markov chain
 - Can be constructed from a unweighted directed graph by assigning normalized weights

Exercise

- Create a TPM of the following graph



Properties of TPMs

- A product of two TPMs is also a TPM
- Always has **eigenvalue 1**
- $|\lambda| \leq 1$ for all eigenvalues
- If the original network is **strongly connected** (with some additional conditions), the TPM has **one and only one eigenvalue 1** (no degeneration)

TPM and asymptotic probability distribution

- $|\lambda| \leq 1$ for all eigenvalues
- If the original network is **strongly connected** (with some additional conditions), the TPM has **one and only one eigenvalue 1** (no degeneration)
- This is a **unique dominant eigenvalue**; the probability vector will converge to its corresponding eigenvector

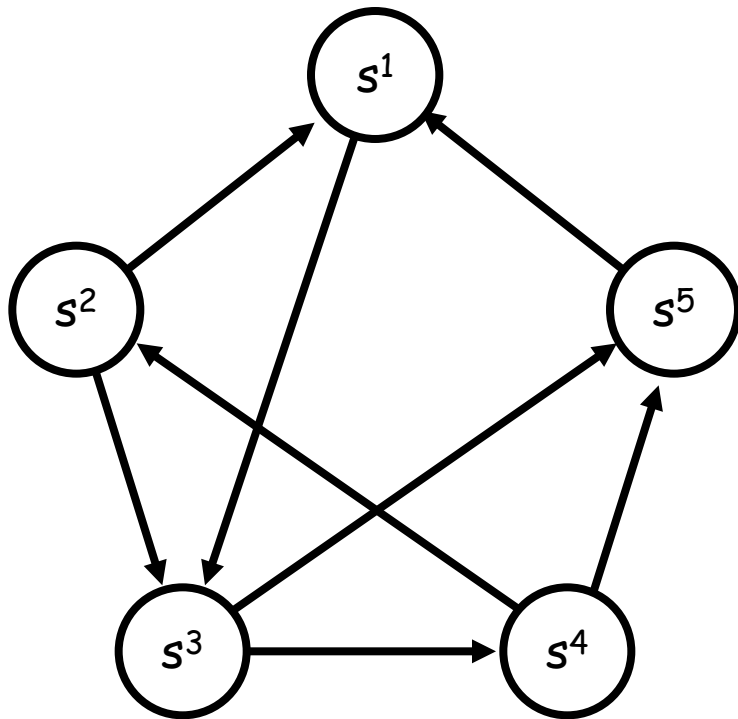
Exercise

- Obtain eigenvalues and eigenvectors of the TPM created in the previous exercise
- Calculate $T^n (1/5, 1/5, 1/5, 1/5, 1/5)^T$ for large n and see what you will get

Application: Google's "PageRank"

- Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, 'The PageRank Citation Ranking: Bringing Order to the Web' (1998): <http://www-db.stanford.edu/~backrub/pageranksub.ps>
- **Node:** Web pages
- **link:** Web links
- **State:** Temporary "importance" of that node
- Its coefficient matrix is a transition probability matrix that can be obtained by dividing each column of the adjacency matrix by the number of 1's in that column

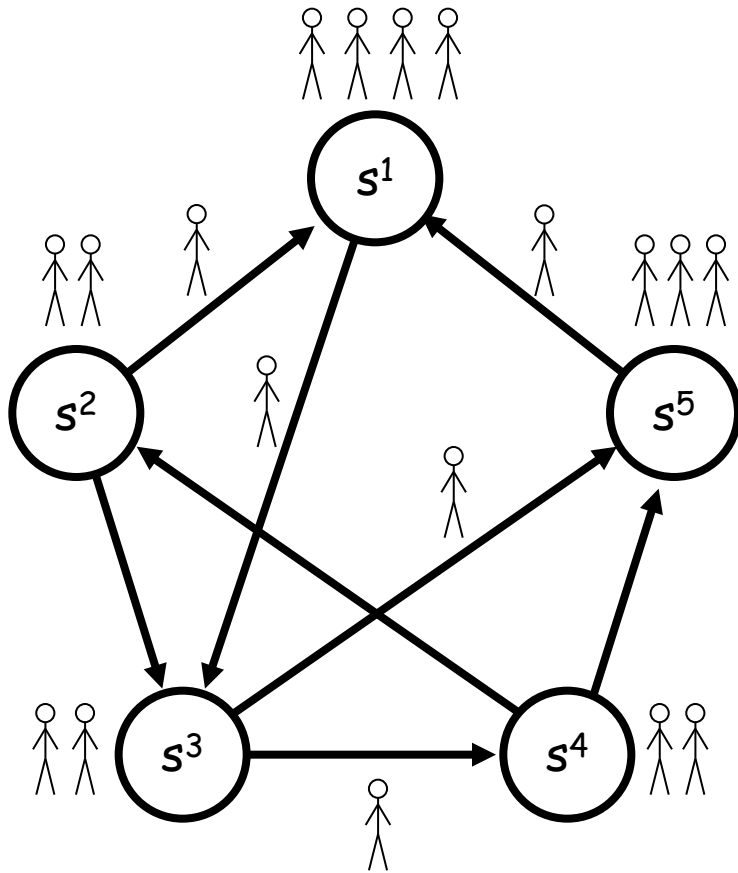
Example



$$\begin{pmatrix} 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.5 & 0 \\ 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}$$

* PageRank is actually calculated by forcedly assigning positive non-zero weights to all pairs of nodes in order to make the entire network strongly connected

Interpreting the PageRank network as a stochastic system

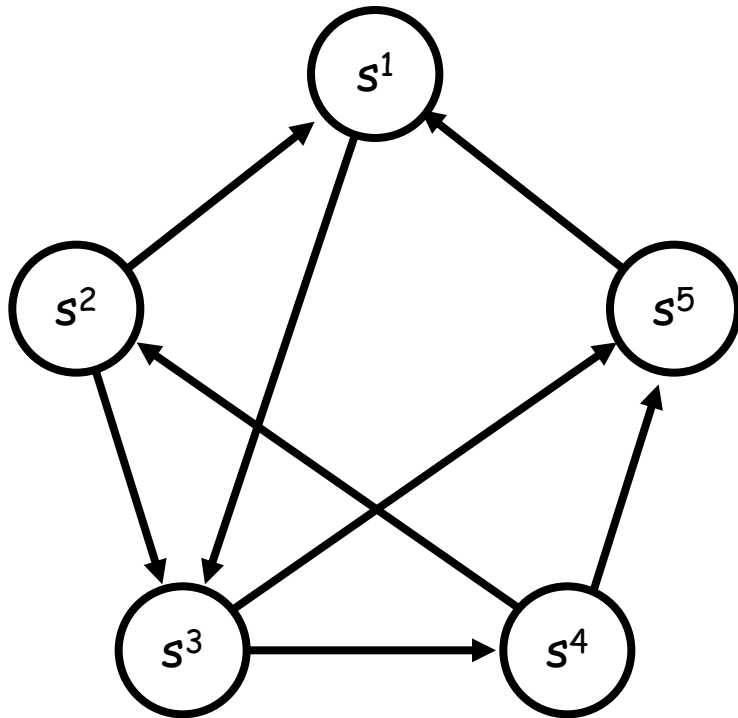


- State of each node can be viewed as a relative population that are visiting the webpage at t
- At next timestep, the population will distribute to other webpages linked from that page evenly

PageRank calculation

- Just one dominant eigenvector of the TPM of a strongly connected network always exists, with $\lambda = 1$
- This shows the equilibrium distribution of the population over WWW
- So, just solve $x = Ax$ and you will get the PageRank for all the web pages on the World Wide Web

Exercise



$$\begin{pmatrix} 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.5 & 0 \\ 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}$$

Calculate the PageRank of each node in the above network (the network is already strongly connected so you can directly calculate its dominant eigenvector; but also try using the NetworkX built-in function for PageRank)

A note on PageRank

- PageRank algorithm gives non-trivial results only for **asymmetric networks**
- If links are symmetric (undirected), the PageRank values will be the same as node degrees
 - Prove this

Laplacian Matrix

Laplacian matrix

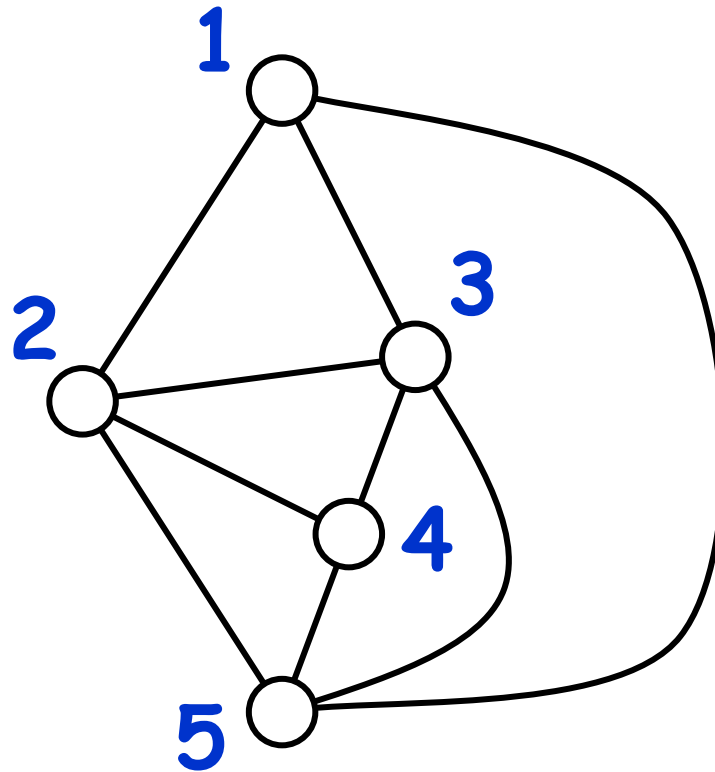
- A matrix with rows and columns labeled by nodes, where a_{ij} represents node degree if $i = j$, or is -1 if node i and node j are connected

$$L = D - A$$

D: degree matrix (diagonal elements are node degrees; all 0 elsewhere)

Exercise

- Write a Laplacian matrix of the graph below



Relationship with Laplacians in vector calculus

- Related to “Laplacian” in vector calculus/PDEs
 - It is a negative, discrete version of it
 - Similar to a “second-order derivative”, defined on a network
 - E.g. diffusion on a network:
$$x(t+1) = x(t) - d L x(t)$$

Relationship with Laplacians in vector calculus

Laplacian discretized over 2-D space:

$$\nabla^2 f = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2$$

$$\sim (f_+(x+\Delta x, y) + f_+(x-\Delta x, y) - 2f_+(x, y)) / \Delta x^2 \\ + (f_+(x, y+\Delta y) + f_+(x, y-\Delta y) - 2f_+(x, y)) / \Delta y^2$$

$$= (f_+(x+\Delta k, y) + f_+(x-\Delta k, y) + f_+(x, y+\Delta k) \\ + f_+(x, y-\Delta k) - 4f_+(x, y)) / \Delta k^2$$

Laplacian (graph) $\sim$ - Laplacian (vector calc.)

Properties of a Laplacian

- Has $(1, 1, 1, \dots, 1)$ as an eigenvector
 - Because each row/column adds up to 0
 - The corresponding eigenvalue is 0
- All eigenvalues ≥ 0
 - # of zero eigenvalues = # of connected components in a graph
 - 2nd smallest ev.: “algebraic connectivity”
 - Smallest non-zero ev.: “spectral gap”
 - Shows how quickly the network can suppress non-homogeneous states and synchronize

Exercise

- Create an Erdos-Renyi random network made of 100 nodes with connection probability 0.03
- Obtain its Laplacian matrix and calculate its eigenvalues
 - See what you find
 - Visualize the network and compare the results

Exercise

- Generate the following network topologies w/ similar size and density:
 - random graph
 - barbell graph
 - ring-shaped graph (i.e., degree-2 regular graph)
- Measure their spectral gaps and see how topologies quantitatively affect their values

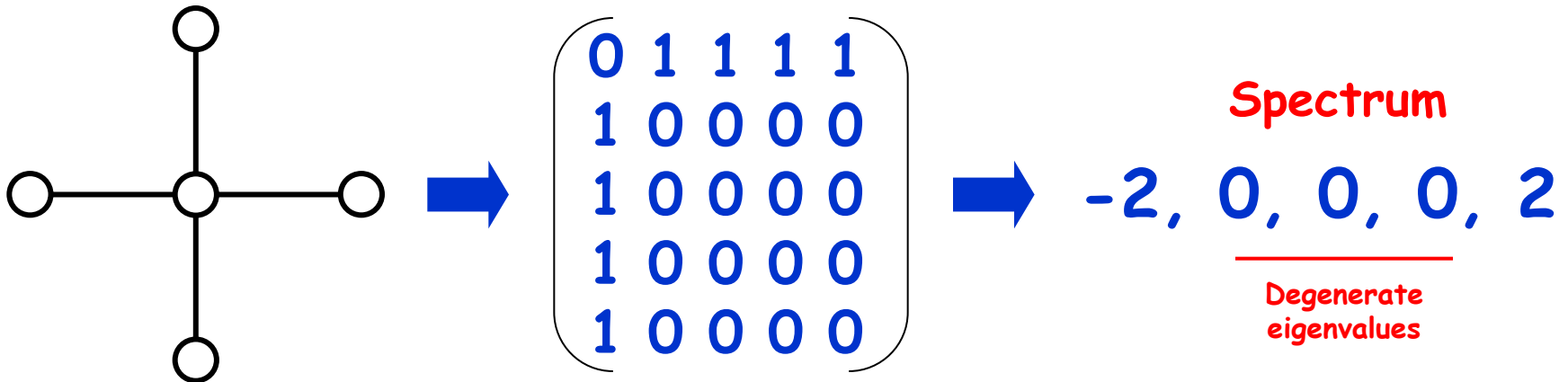
Graph Spectrum

Degree distribution and graph spectrum

- Structural characteristics of a large complex network can be studied by analyzing these distributions
 - Similar networks often have similar degree distributions and graph spectra
 - Degree distribution is structural, intuitive and very easy to obtain
 - Graph spectrum has strong connection to both structure and dynamical behavior

Graph spectrum

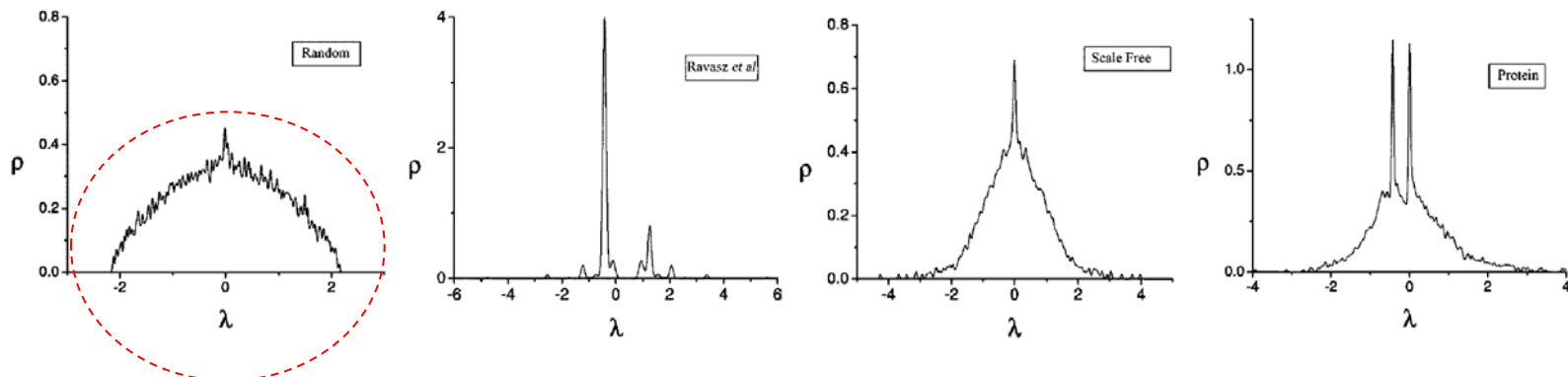
- Distribution of eigenvalues of the adjacency matrix of the network



- Undirected graphs have symmetric adj. matrices $\rightarrow$ all real eigenvalues

Graph spectral analysis

- Plotting an eigenvalue distribution (i.e., histogram)
 - Especially effective for visualizing complex network data obtained experimentally
 - Computing power may be needed to obtain these plots for large networks



Wigner's semi-circle law

de Aguiar & Bar-Yam, Phys. Rev. E 71: 016106 (2005)

Exercise

- Obtain spectra of networks made of 1,000 nodes each
 - Random
 - Scale-free
 - Based on some data
- Plot their density distributions

Exercise

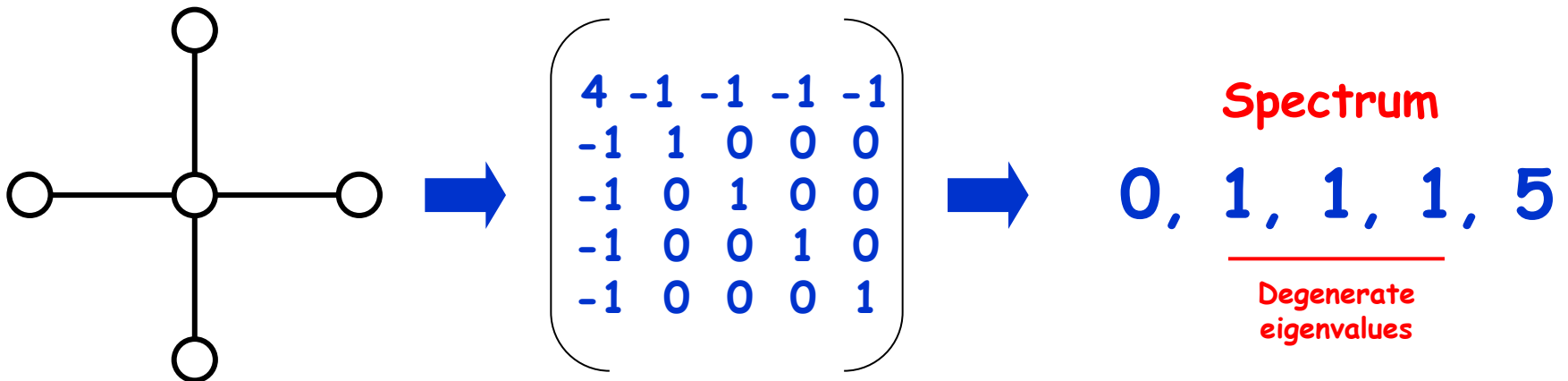
- Obtain the spectrum of the Supreme Court Citation network
 - Can you do this??
- If you can't, make a subgraph induced by randomly selected 1,000 nodes, and conduct the same analysis
 - Crude random sampling technique...

What eigenvalues and eigenvectors can tell us

- An eigenvalue tells whether a particular “state” of the network (specified by its corresponding eigenvectors) grows or shrinks by interactions between nodes over edges
 - $\text{Re}(\lambda) > 0 \Rightarrow$ growing
 - $\text{Re}(\lambda) < 0 \Rightarrow$ shrinking

Laplacian spectrum

- Distribution of eigenvalues of the Laplacian matrix of the network



Review of Laplacian spectrum

- At least one λ is zero
- All the other λ s are zero or positive
- # of zero λ s corresponds to # of connected components in the graph
- 2nd smallest λ : “algebraic connectivity”
- Smallest non-zero λ : “spectral gap”

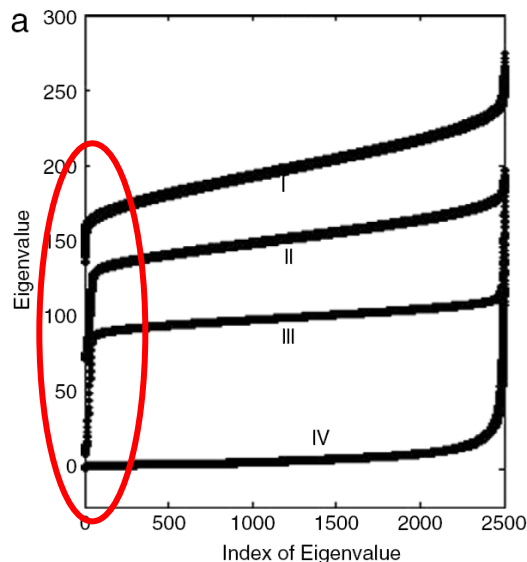
Algebraic connectivity

$$\underbrace{0 = \lambda_1 = \lambda_2 = \dots = \lambda_{k-1}}_{\text{As many 0's as \# of CC's}} < \lambda_k \leq \lambda_{k+1} \leq \dots \leq \lambda_N$$

Spectral gap

Spectral gap

- Determines how easily a dynamical network can get synchronized
 - The larger it is (relatively to the largest λ_N), the easier the synchronization is
(Barahona & Pecora, Phys. Rev. Lett. 89: 054101. 2002)



- I. ER random
- II. NW small-world
- III. WS small-world
- IV. BA scale-free

(Zhan, Chen & Yeung, Physica A 389: 1779-1788, 2010)

Exercise

- Create a small-world network of 1,000 nodes with varying p
- Obtain Laplacian spectra of the network and find its spectral gap λ_2
- Plot λ_2 over p and see how it changes as random rewiring rate increases