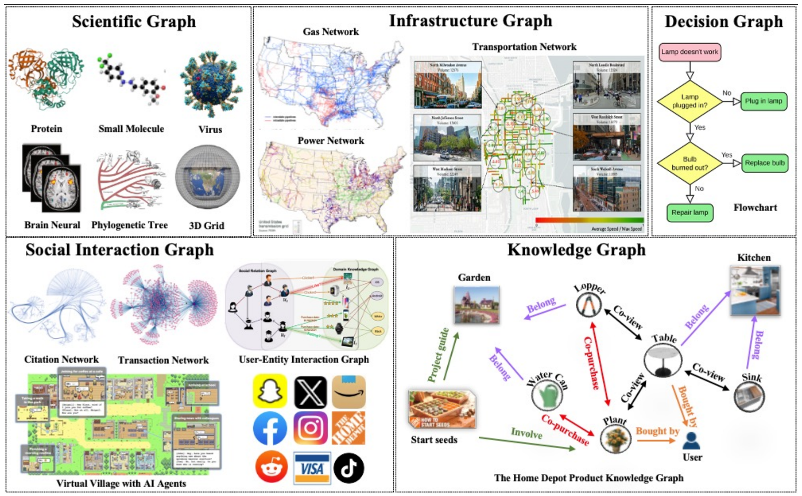
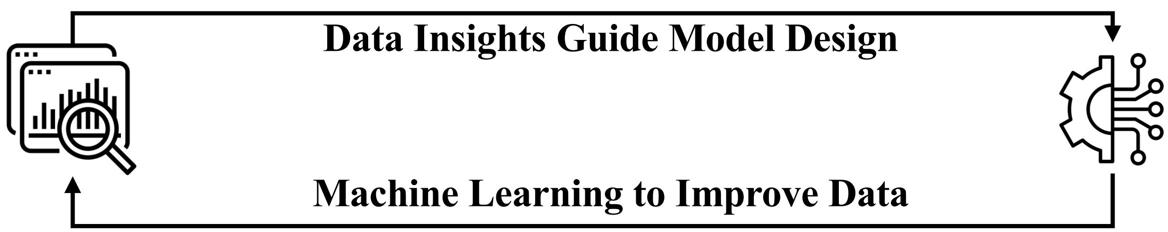
Mining and Learning on Graphs

Linear Algebra and Graph Theory

Yu Wang, Ph.D.
Assistant Professor
Computer and Information Science
University of Oregon
CS 410/510 - Fall 2024

Data Mining & Machine Learning on Graphs





Data mining
Analyze data
Derive patterns and relationships
Solve real-world problems

Machine Learning
Design Model
Allow Computer to Learn and Improve
Without being explicit programmed

Data Mining & Machine Learning on Graphs

Network Analysis

- **Background**
  - Linear Algebra
  - Graph Theory
- **Statistical Graph Model**
  - Erdos-Renyi, Barabasi-Albert
  - Small-World, Chung-Lu
- **Network Analysis**
  - Degree, Closeness
  - Betweeness, Katz
  - Eigenvector, PageRank

Computation Methods

- **Link Prediction**
- **Node Classification**
- **Graph Classification**
- **Network Diffusion**
- **Graph Clustering**

Machine Learning on Graph

- **Network Embedding**
- **Graph Neural Networks**
- **Self-supervised Learning**
- **Trustworthy Issue**
- **Data-quality Issue**

10/01 **10/22** **11/07** **12/03**

Linear Algebra – Basic Notations

Vector

$$\mathbf{v} = \begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$$

$$\boxed{\boldsymbol{u} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}}$$

Please note that we will use this one by default

Matrix

$$\mathbf{A} = \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix}}_{\text{3 columns}} \left.\vphantom{\begin{bmatrix} 1 \\ 0 \\ 2 \\ 3 \end{bmatrix}}\right\} \text{4 rows}$$

$$\mathbf{v} \in \mathbb{R}^{\mathbf{1} \times \mathbf{3}}$$

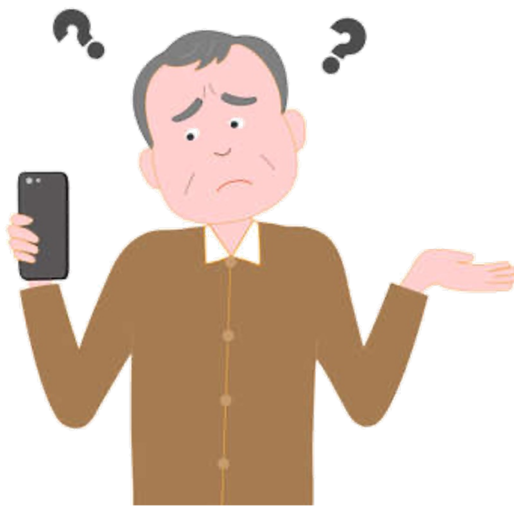
$$\boldsymbol{u} \in \mathbb{R}^{\mathbf{3} \times \mathbf{1}}$$

$$\mathbf{A} \in \mathbb{R}^{\mathbf{4} \times \mathbf{3}}$$

Linear Algebra – Basic Operations

1 Basics

$$
\begin{aligned}
(\mathbf{AB})^{-1} &= \mathbf{B}^{-1}\mathbf{A}^{-1} && (1)\\
(\mathbf{ABC}...)^{-1} &= ...\mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1} && (2)\\
(\mathbf{A}^{T})^{-1} &= (\mathbf{A}^{-1})^{T} && (3)\\
(\mathbf{A}+\mathbf{B})^{T} &= \mathbf{A}^{T}+\mathbf{B}^{T} && (4)\\
(\mathbf{AB})^{T} &= \mathbf{B}^{T}\mathbf{A}^{T} && (5)\\
(\mathbf{ABC}...)^{T} &= ...\mathbf{C}^{T}\mathbf{B}^{T}\mathbf{A}^{T} && (6)\\
(\mathbf{A}^{H})^{-1} &= (\mathbf{A}^{-1})^{H} && (7)\\
(\mathbf{A}+\mathbf{B})^{H} &= \mathbf{A}^{H}+\mathbf{B}^{H} && (8)\\
(\mathbf{AB})^{H} &= \mathbf{B}^{H}\mathbf{A}^{H} && (9)\\
(\mathbf{ABC}...)^{H} &= ...\mathbf{C}^{H}\mathbf{B}^{H}\mathbf{A}^{H} && (10)
\end{aligned}
$$



Matrix Codebook

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

The Matrix Cookbook

[http://matrixcookbook.com]

Kaare Brandt Petersen
Michael Syskind Pedersen

Version: November 15, 2012

Linear Algebra – Matrix Multiplication

Matrix Multiplication

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \quad \Rightarrow \quad \mathbf{C} = [\quad] \quad ?$$

4×3 $\qquad$ 3×2

Dimensions much match!

What is the dimension of C? $(4\times 3)(3\times 2) \rightarrow 4\times 2$

Matrix Multiplication

$1\times1 + 2\times2 + 3\times5$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \quad \Rightarrow \quad \mathbf{C} = \begin{bmatrix} \mathbf{20} & \\ & \\ & \end{bmatrix}$$

4×3 $\qquad$ 3×2

$1\times2 + 2\times3 + 3\times7$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \quad \Rightarrow \quad \mathbf{C} = \begin{bmatrix} 20 & \mathbf{29} \\ & \\ & \end{bmatrix}$$

4×3 $\qquad$ 3×2

Linear Algebra – Matrix Multiplication

Matrix Multiplication

$$0\times1 + 5\times2 + 1\times5$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \quad \Rightarrow \quad \mathbf{C} = \begin{bmatrix} 20 & 29 \\ \mathbf{15} & \\ & \\ & \end{bmatrix}$$

4×3 $\qquad$ 3×2

$$0\times2 + 5\times3 + 1\times7$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \quad \Rightarrow \quad \mathbf{C} = \begin{bmatrix} 20 & 29 \\ 15 & \mathbf{22} \\ & \\ & \end{bmatrix}$$

4×3 $\qquad$ 3×2

Matrix Multiplication

$$2\times1 + 3\times2 + 7\times5$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \Rightarrow \mathbf{C} = \begin{bmatrix} 20 & 29 \\ 15 & 22 \\ \mathbf{43} & \\ & \end{bmatrix}$$

4×3 3×2

$$2\times2 + 3\times3 + 7\times7$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \Rightarrow \mathbf{C} = \begin{bmatrix} 20 & 29 \\ 15 & 22 \\ 43 & \mathbf{62} \\ & \end{bmatrix}$$

4×3 3×2

Linear Algebra – Matrix Multiplication

Matrix Multiplication

$$3\times 1 + 2\times 9 + 5\times 8$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \Rightarrow \quad \mathbf{C} = \begin{bmatrix} 20 & 29 \\ 15 & 22 \\ 43 & 62 \\ \mathbf{61} & \end{bmatrix}$$

4×3 $\qquad$ 3×2

$$3\times 2 + 3\times 9 + 7\times 8$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix} \times \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 7 \end{bmatrix} \Rightarrow \quad \mathbf{C} = \begin{bmatrix} 20 & 29 \\ 15 & 22 \\ 43 & 62 \\ 61 & \mathbf{89} \end{bmatrix}$$

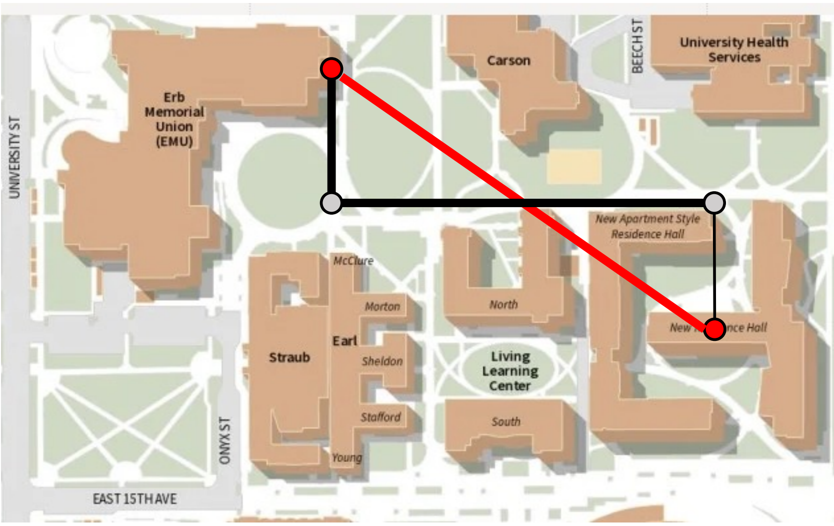
4×3 $\qquad$ 3×2

Linear Algebra – Vector Norms

- $\|\mathbf{v}\|^p$
- Function from a vector space to a single positive real value: $f: \mathbb{R}^d \rightarrow \mathbb{R}$
- Length of **v**

$$\|\mathbf{v}\|^p = \left(\sum_{i=1}^{d} \mathbf{v}_i^p\right)^{\frac{1}{p}}$$

- Examples:
  - Manhattan distance (L_1): $\|\mathbf{v}\|^1 = \left(\sum_{i=1}^{d} |\boldsymbol{v}_i|\right)$
  - Euclidean distance (L_2): $\|\mathbf{v}\|^2 = \left(\sum_{i=1}^{d} \boldsymbol{v}_{\boldsymbol{i}}^{\mathbf{2}}\right)^{\frac{1}{2}}$
  - **How about L_0?**



Linear Algebra – Transpose Matrix

- $\mathbf{A}^{\mathrm{T}}$ or $\mathbf{A}'$
- Flip rows and columns

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 1 \\ 2 & 3 & 7 \\ 3 & 9 & 8 \end{bmatrix}$$

4×3

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 2 & 5 & 3 & 9 \\ 3 & 1 & 7 & 8 \end{bmatrix}$$

3×4

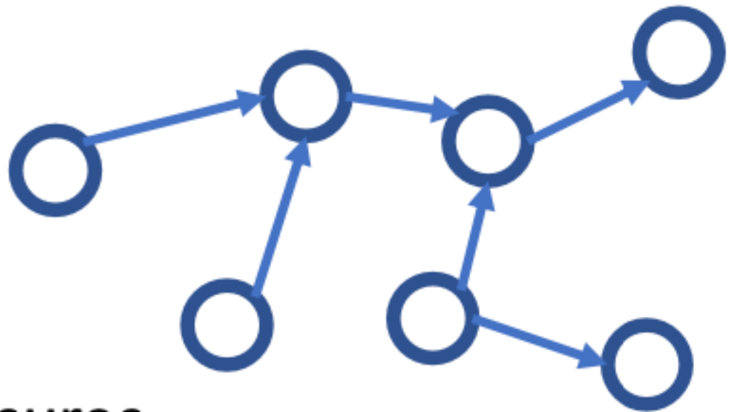
$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

3×1

$$\mathbf{u}^{\mathbf{T}} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

1×3

- Motivation
  - Given a graph (e.g., web pages that have a keyword that was searched for)
  - Which node is the most important?



  - Later we will discuss centrality measures
    - E.g., HITS and PageRank
  - That are based on SVD and Eigendecomposition

Linear Algebra – Singular Value Decomposition (SVD)

- $\mathbf{A}_{[\mathbf{n\ x\ m}]} = \mathbf{U}_{[\mathbf{n\ x\ r}]}\, \boldsymbol{\Lambda}_{[\boldsymbol{r\ x\ r}]} \left(\mathbf{V}_{[\mathbf{m\ x\ r}]}\right)^{\mathbf{T}}$

- **A:** n x m matrix (e.g., n documents, m terms)
- **U:** n x r matrix (n documents, r concepts)
- **Λ**: r x r diagonal matrix (r is rank of the matrix)
  - Can be seen as strength of the topic
- **V:** m x r matrix (m terms, r concepts)

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\text{x}
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\text{x}
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$

Linear Algebra – Singular Value Decomposition (SVD)

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \\ \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cc} \text{CS-concept} & \text{MD-concept} \end{array} \\
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\end{array}
\text{ x }
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\text{ x }
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

Linear Algebra – Singular Value Decomposition (SVD)

term-to-concept
similarity matrix

CS-concept

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\text{x}
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\text{x}
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

Linear Algebra – Singular Value Decomposition (SVD)

'strength' of CS-concept

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung}\end{array} \\
\begin{array}{c} \\ \\ \text{CS} \\ \\ \\ \text{MD} \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\times
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\times
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$

Linear Algebra – Singular Value Decomposition (SVD)

```
>>> import numpy as np
>>> a = np.array([[0.18, 0], [0.36, 0], [0.18, 0], [0.90, 0], [0, 0.53], [0, 0.80], [0, 0.27]])
>>> b = np.array([[0.58, 0.58, 0.58, 0, 0], [0, 0, 0, 0.71, 0.71]])
>>> a
array([[0.18, 0.  ],
       [0.36, 0.  ],
       [0.18, 0.  ],
       [0.9 , 0.  ],
       [0.  , 0.53],
       [0.  , 0.8 ],
       [0.  , 0.27]])
>>> b
array([[0.58, 0.58, 0.58, 0.  , 0.  ],
       [0.  , 0.  , 0.  , 0.71, 0.71]])
>>> np.matmul(a, b)
array([[0.1044, 0.1044, 0.1044, 0.    , 0.    ],
       [0.2088, 0.2088, 0.2088, 0.    , 0.    ],
       [0.1044, 0.1044, 0.1044, 0.    , 0.    ],
       [0.522 , 0.522 , 0.522 , 0.    , 0.    ],
       [0.    , 0.    , 0.    , 0.3763, 0.3763],
       [0.    , 0.    , 0.    , 0.568 , 0.568 ],
       [0.    , 0.    , 0.    , 0.1917, 0.1917]])
```

$$
\begin{array}{c}
\\
\text{CS} \\
\\
\text{MD}
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
$$

Linear Algebra – Singular Value Decomposition (SVD)

- **'documents', 'terms', 'concepts'**
- **U:** document-to-concept similarity matrix
- **V:** term-to-concept similarity matrix
- **Λ**: the diagonal elements are the 'strength' of each concept

- **A:** n x m matrix (e.g., n documents, m terms)
- What is $\mathbf{A}^{\mathrm{T}}\mathbf{A}$?
  - Term-to-term (m x m) similarity matrix
- What is $\mathbf{A}\mathbf{A}^{\mathrm{T}}$?
  - ...

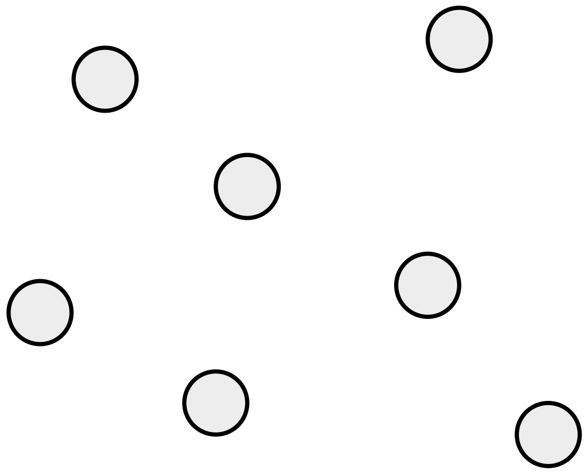
Linear Algebra – Singular Value Decomposition (SVD)

- Text domain:
  - Latent Semantic Indexing (LSI)
    - Analyze the relationship between a set of documents and their terms contained by generating a set of concepts
- Dimensionality reduction
  - In general, for a data matrix **X**
  - Or can be used for an adjacency matrix **A**

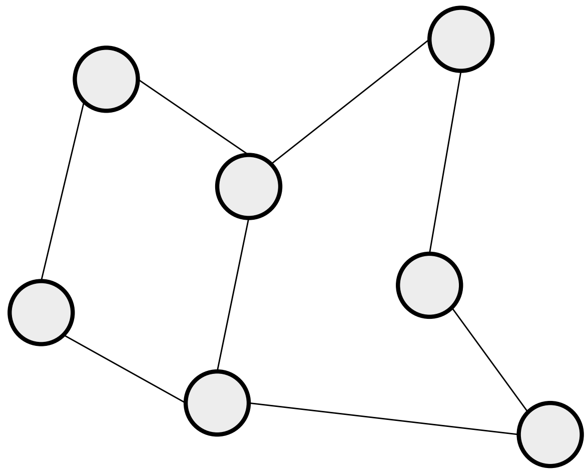
Graph and Network Theory- Why?

They provide a general language for describing highly complex systems in a unified way

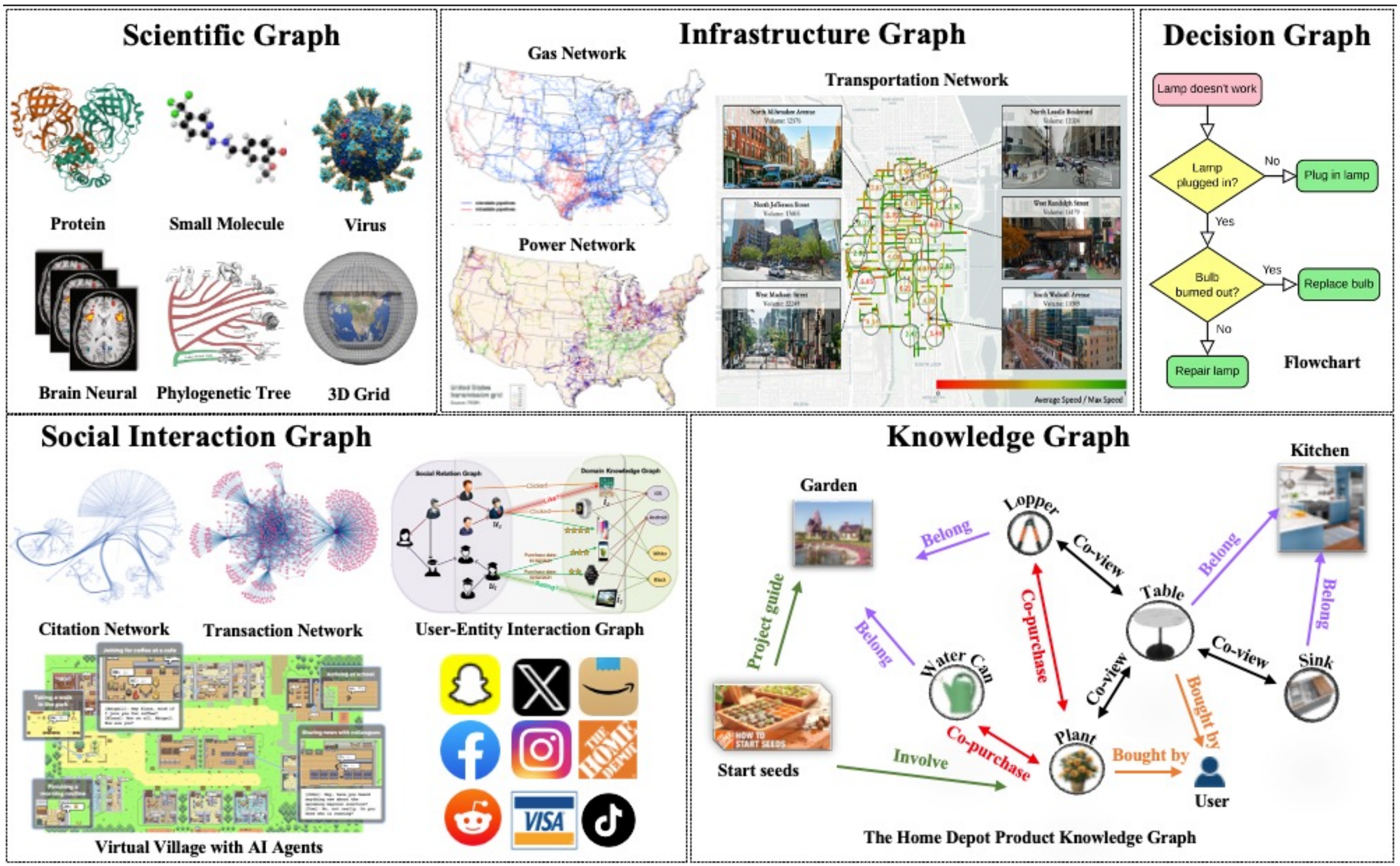
Traditional Data View



Network Data View

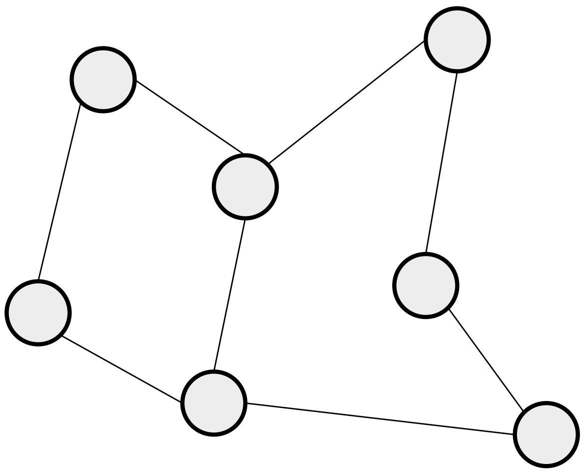


Data Mining & Machine Learning on Graphs



Graph and Network Theory - Basic

Network/Graph-Structured Data



Vertices/Nodes: $\mathcal{V}$ with feature $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times \boldsymbol{d}}$

Edges/Links: $\mathcal{E}$ with feature $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times \boldsymbol{d'}}$

Overall system: $G = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E})$

Network	⟷	**Graph**
Real-world system		Model representation of a network in mathematics

Graph and Network Theory - Examples

- **Connect Conference/Journal Papers with each other where nodes are papers, and the links represent a citation from one to another……**
  - Citation Network (e.g., DBLP)

- **If we connect people based on their dating relations where the nodes are people, and the links are their relations…**
  - Dating Network (e.g., Tinder)

- **If we connect all the words in the dictionary where the nodes are the words and the links connect words having semantic relations between them …**
  - Word Network (e.g., WordNet)

Can you name some examples?

Graph and Network Theory - Network Construction

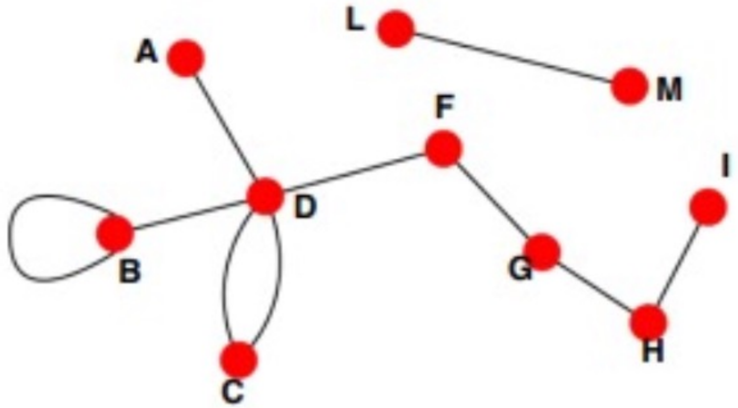
Given a specific scenario, how should we construct the network?

- **How to decide what are the nodes and edges?**
  - Sometimes it is unambiguous and unique
    - User-user interaction, customer-item interaction
  - Other times it is left up to the application needs
    - Route-Net Example
- **In either way, constructing the network/graph is very importance for downstream tasks**
  - If we connect two users based on whether they have the same first name instead of based on whether they have following relations on a Twitter dataset...

Graph and Network Theory – Types of Graphs

Undirected

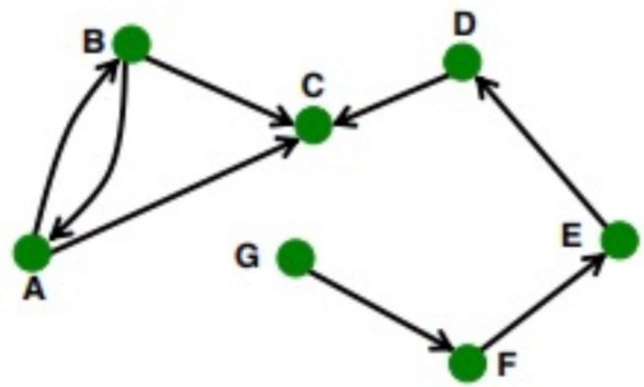
- **Links:** undirected (symmetrical, reciprocal)



- **Examples:**
  - Collaborations
  - Friendship on Facebook

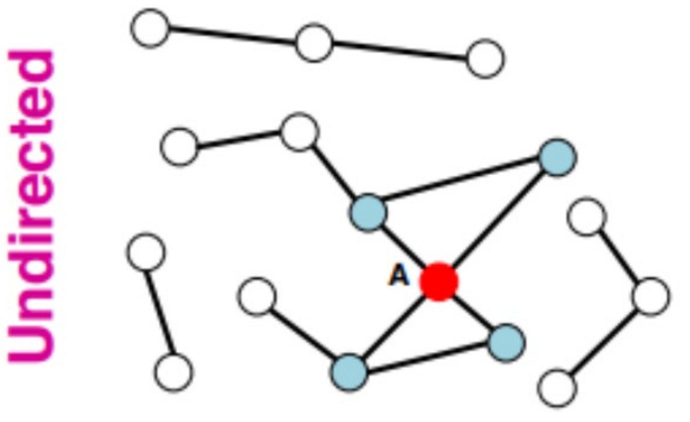
Directed

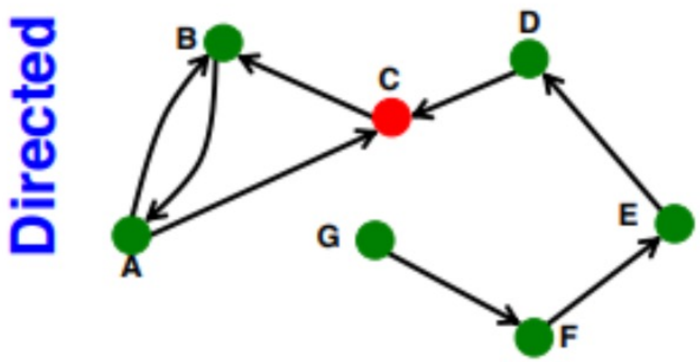
- **Links:** directed (arcs)



- **Examples:**
  - Phone calls
  - Following on Twitter

Graph and Network Theory – Node Degree





Source: Node with $k^{in} = 0$

Sink: Node with $k^{out} = 0$

Node degree, k_i: the number of edges adjacent to node $\boldsymbol{i}$

$$k_A = 4$$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}$

In directed networks we define an **in-degree** and **out-degree.**

The (total) degree of a node is the sum of in- and out-degrees.

$$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$$

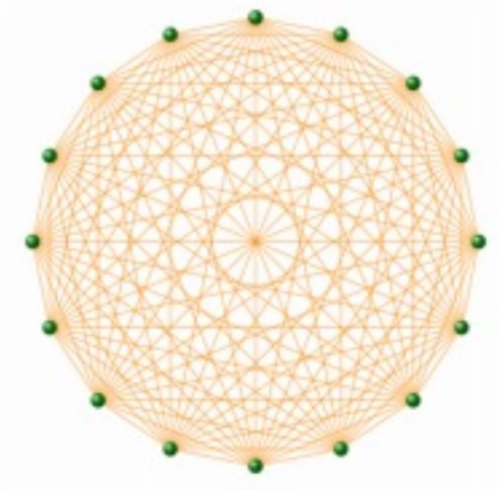
$$\bar{k} = \frac{E}{N} \qquad \overline{k^{in}} = \overline{k^{out}}$$

Graph and Network Theory – Examples

NETWORK	NODES	LINKS	DIRECTED UNDIRECTED	N	L	⟨k⟩
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.33
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Mobile Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Paper	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

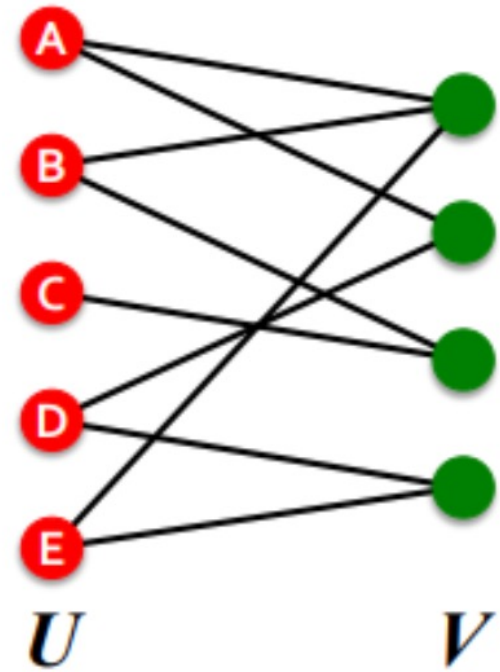
The **maximum number of edges** in an undirected graph on N nodes is

$$E_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$

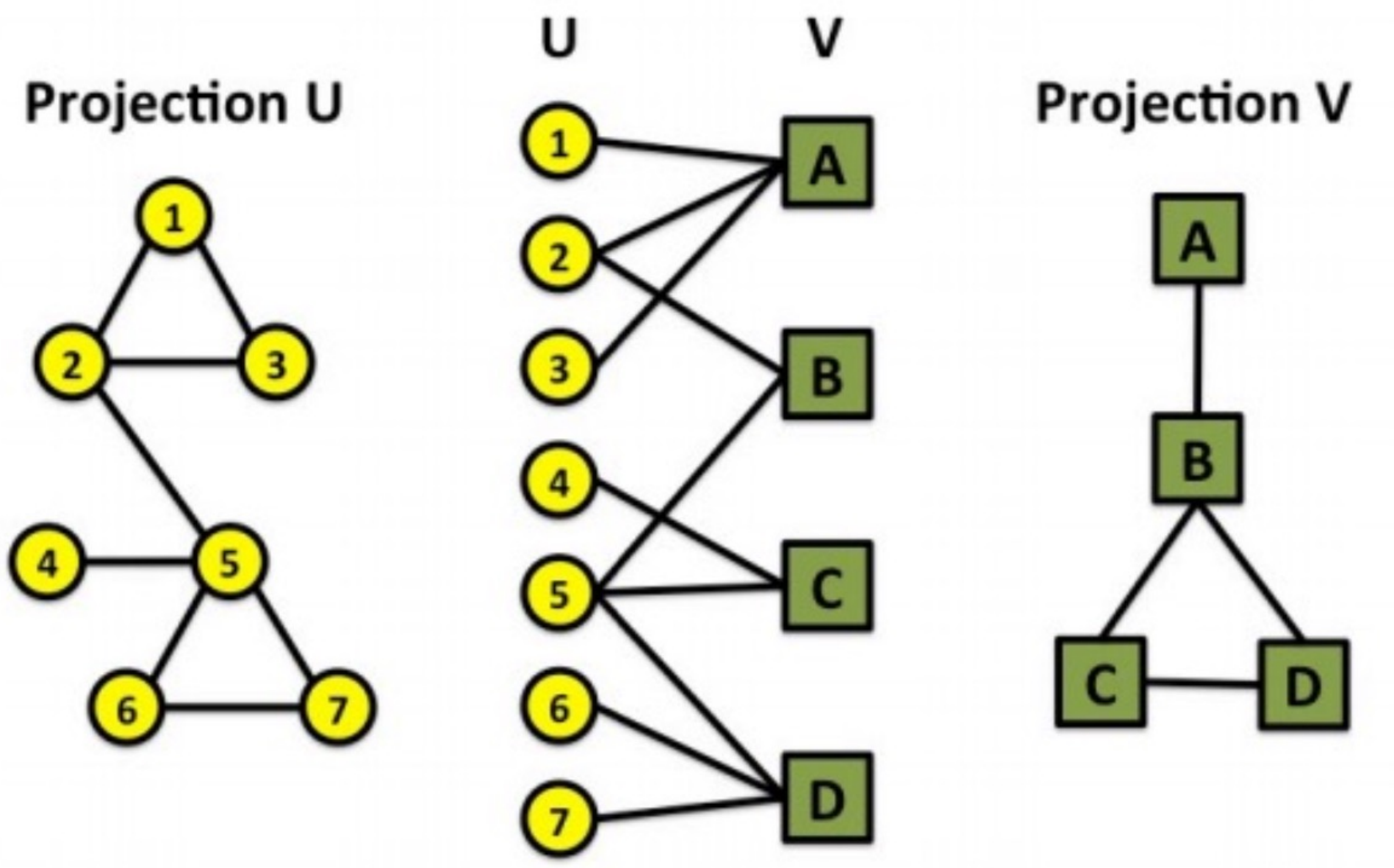


An undirected graph with the number of edges $E = E_{max}$ is called a **complete graph**, and its average degree is $N\text{-}1$

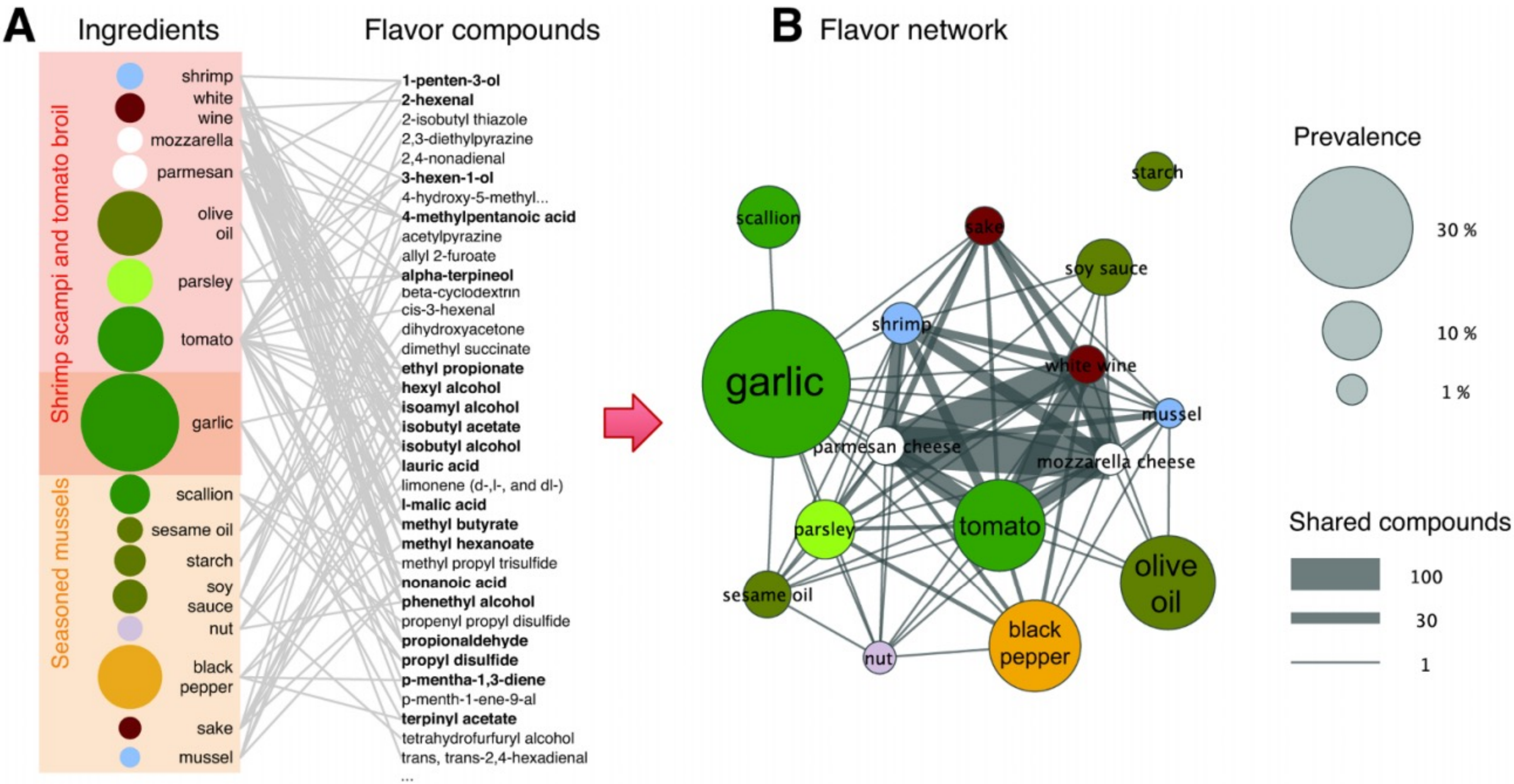
- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**
- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks



Graph and Network Theory – Bipartite Projections

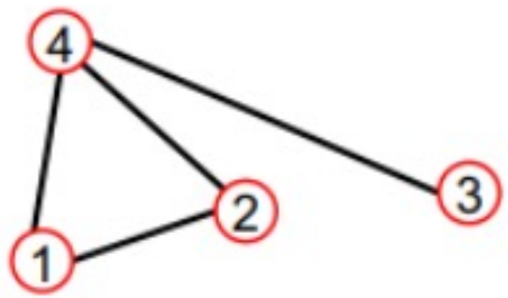


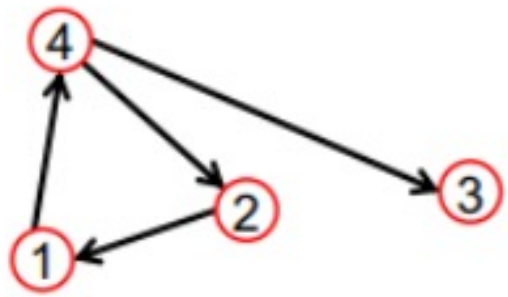
Graph and Network Theory – Examples



Flavor network and the principles of food pairing

Graph and Network Theory – Graph Adjacency Matrix





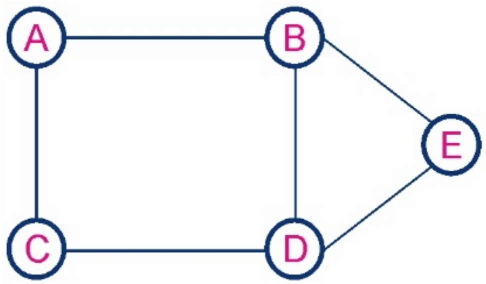
$A_{ij} = 1$ if there is a link from node ***i*** to node ***j***

$A_{ij} = 0$ otherwise

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

Note that for a directed graph (right) the matrix is not symmetric.

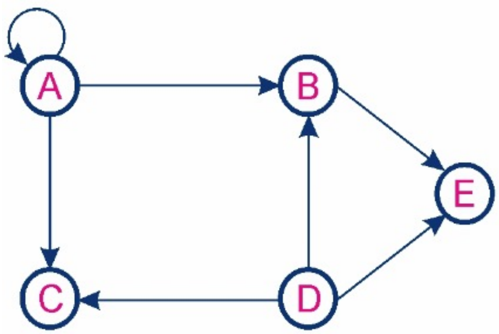
Graph and Network Theory – Graph Adjacency Matrix



Undirected Graph

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	1
C	1	0	0	1	0
D	0	1	1	0	1
E	0	1	0	1	0

$\mathbf{A}_{ij}$

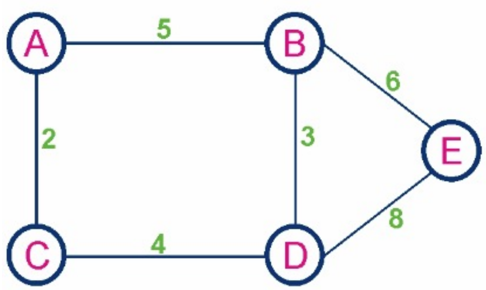


Directed Graph

	A	B	C	D	E
A	1	1	1	0	0
B	0	0	0	0	1
C	0	0	0	0	0
D	0	1	1	0	1
E	0	0	0	0	0

$\mathbf{D}_i$

$\mathbf{D}_i^+$

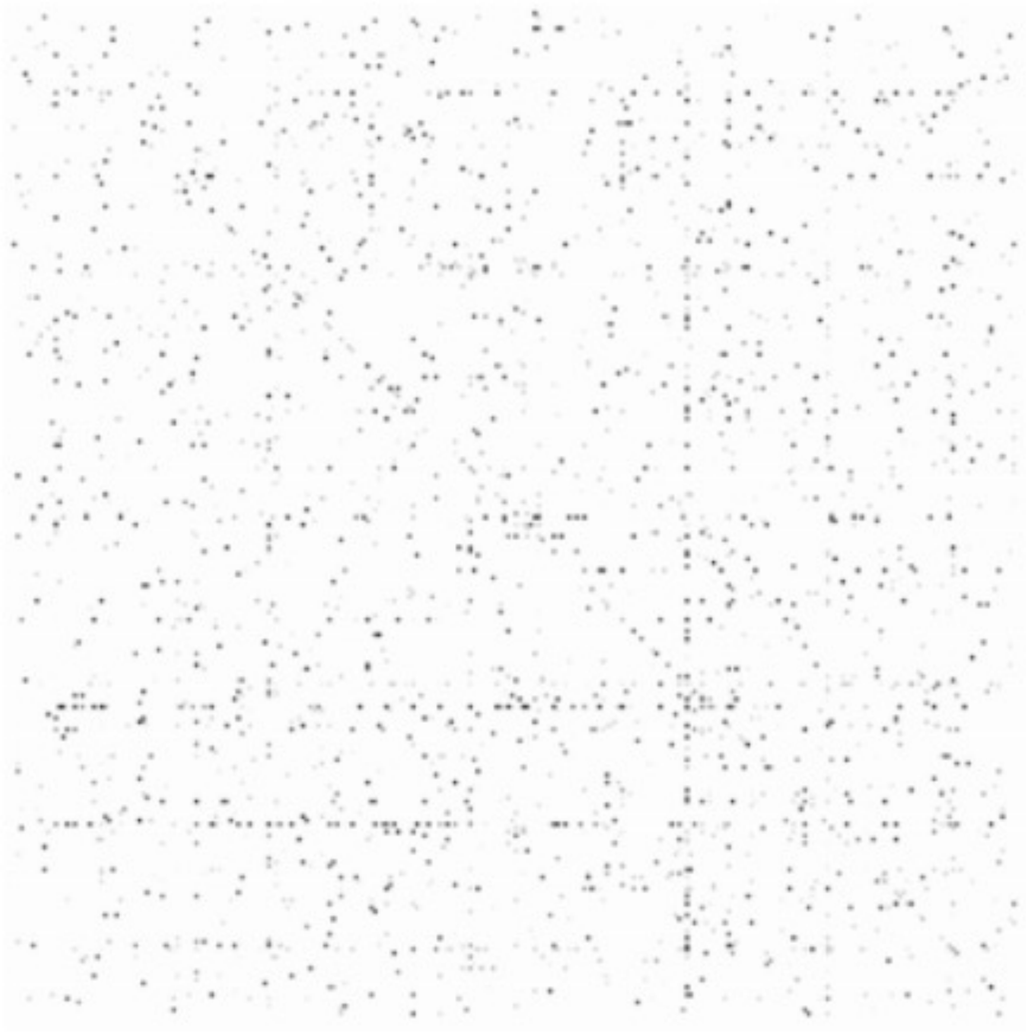


Weighted Graph

	A	B	C	D	E
A	0	5	2	0	0
B	5	0	0	3	6
C	2	0	0	4	0
D	0	3	4	0	8
E	0	6	0	8	0

$\mathbf{D}_i^-$

Graph and Network Theory – Adjacency Matrix



Most adjacency matrices are sparse

Most real-world networks are sparse

$$E \ll E_{max} \quad (\text{or } \bar{k} \ll N\text{-}1)$$

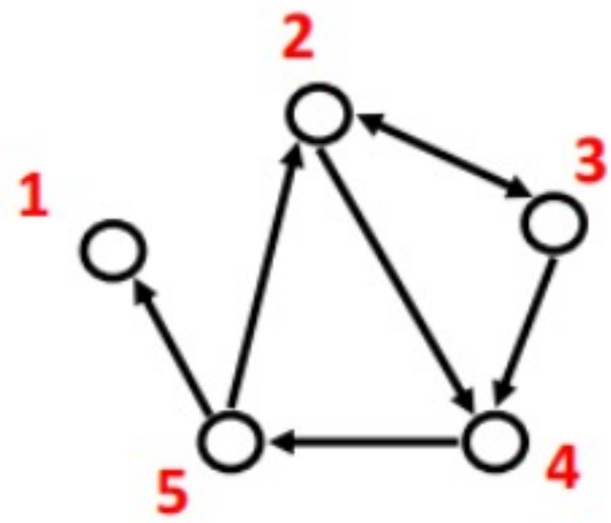
WWW (Stanford-Berkeley):	N=319,717	$\langle k \rangle$=9.65
Social networks (LinkedIn):	N=6,946,668	$\langle k \rangle$=8.87
Communication (MSN IM):	N=242,720,596	$\langle k \rangle$=11.1
Coauthorships (DBLP):	N=317,080	$\langle k \rangle$=6.62
Internet (AS-Skitter):	N=1,719,037	$\langle k \rangle$=14.91
Roads (California):	N=1,957,027	$\langle k \rangle$=2.82
Proteins (S. Cerevisiae):	N=1,870	$\langle k \rangle$=2.39

(Source: *Leskovec et al., Internet Mathematics, 2009*)

Consequence: Adjacency matrix is filled with zeros!

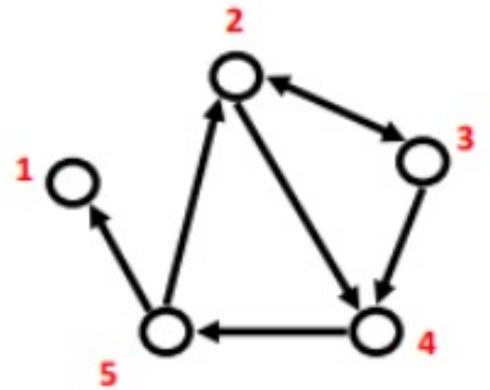
(Density of the matrix (E/N^2): WWW=1.51×10^{-5}, MSN IM = 2.27×10^{-8})

- Represent graph as a set of edges:
  - (2, 3)
  - (2, 4)
  - (3, 2)
  - (3, 4)
  - (4, 5)
  - (5, 2)
  - (5, 1)



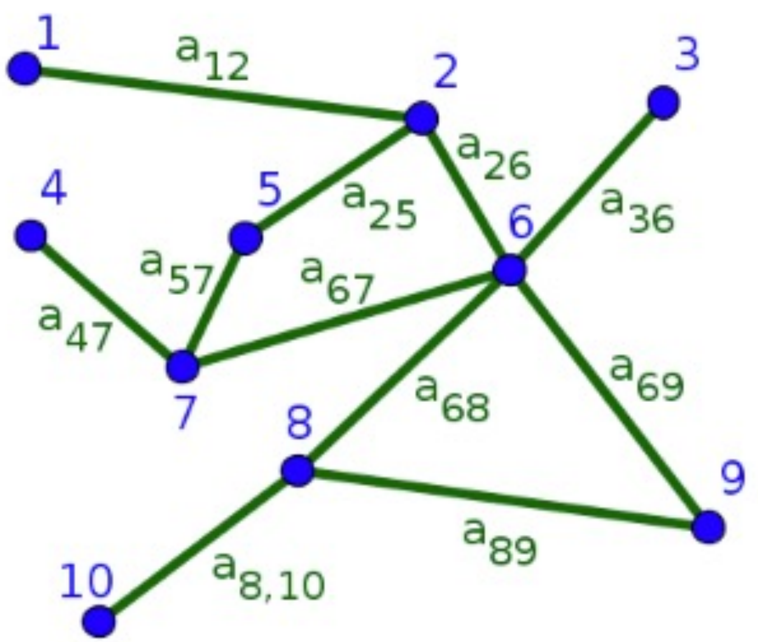
Graph and Network Theory – Adjacency List

- **Adjacency list:**
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node
    - 1:
    - 2: 3, 4
    - 3: 2, 4
    - 4: 5
    - 5: 1, 2



Graph and Network Theory – Degree Distribution

A node's degree is the number of edges or connections it has to other nodes in a network.



$$k_i = \sum_j a_{ij}$$

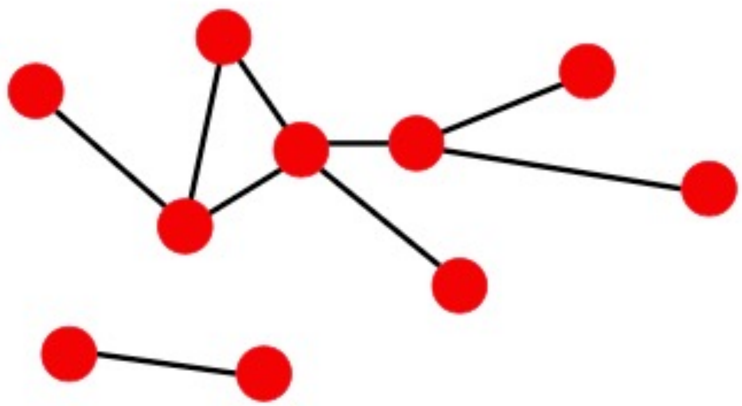
$$\sum_{v_i \in \mathcal{V}} k_i = 2\ \#edges$$

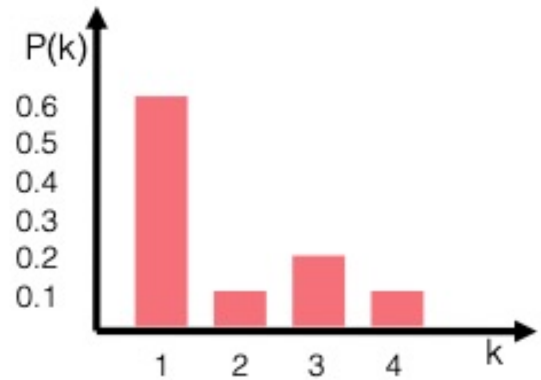
Graph and Network Theory – Degree Distribution

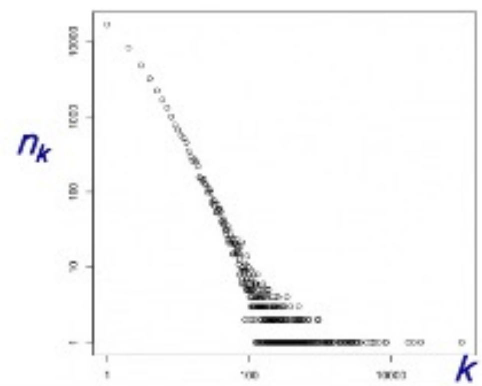
- Degree distribution *P(k)*: Probability that a randomly chosen node has degree *k*

n_k = # nodes with degree *k*

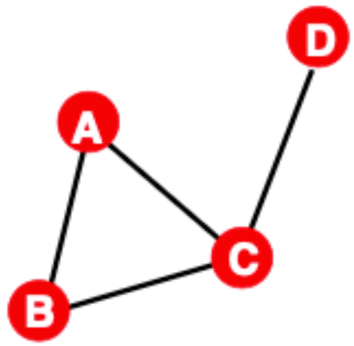
- Normalized histogram:
  $P(k) = n_k / n$ ➔ plot







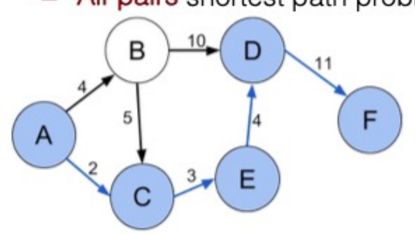
Graph and Network Theory – Distance and Shortest Path



$h_{B,D} = 2$

- Distance (shortest path, geodesic) between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes
  – If the two nodes are disconnected, the distance is usually defined as infinite

- Definition: find a path between two nodes in a graph, in such a way that the sum of the weights of its constituent edges is minimized
  – Many applications (e.g., road networks, community detection, communications)
- Variants
  – Single-source shortest path problem
  – Single-destination shortest path problem
  – All-pairs shortest path problem



Shortest path (A, C, E, D, F) between vertices A and F in the weighted directed graph

Various algorithms:
- Dijkstra
- Bellman-Ford
  (works with negative edge weights)

- Diameter: the maximum (shortest path) distance between any pair of nodes in a graph

- Average path length for a connected graph (component) or a strongly connected (component of a) directed graph
  - Many times we compute the average only over the connected pairs of nodes (that is, we ignore "infinite" length paths)

$$\bar{h} = \frac{1}{n(n-1)} \sum_{i,j \neq i} h_{ij}$$

where h_{ij} is the distance from node i to node j

Graph and Network Theory – Node/Edge Attributes

- **Edge Attributes**
  - Weight (e.g., frequency of communication)
  - Ranking (best friend, second best friend)
  - Type (friend, relative, co-worker)
  - Sign: Friend vs Foe, Trust vs Distrust
  - Properties depending on the structure of the rest of the graph: number of common friends

- **Node Attributes**
  - Bag-of-words feature for documents
  - Customer profile
  - Product meta data

Any Question?