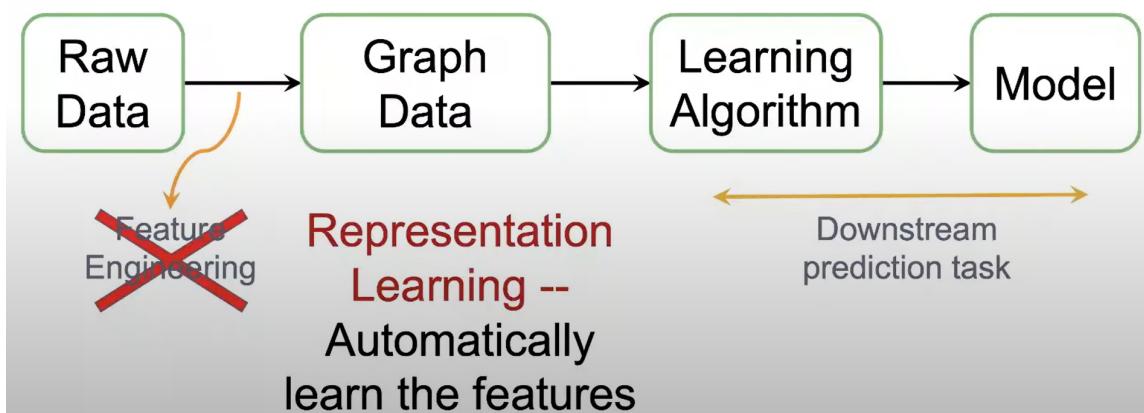


1

Lecture 1

Motivation for Graph ML

- feature engineering 없이도 모델 구축하기 위해



- 이 수업에서 다룰 것

We are going to cover various topics in Machine Learning and Representation Learning for graph structured data:

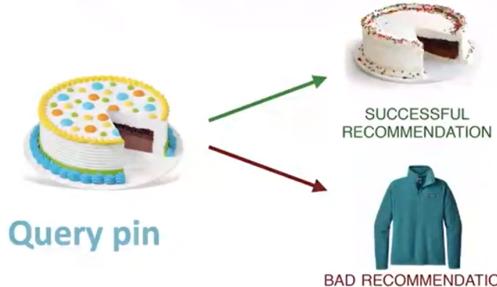
- Traditional methods: Graphlets, Graph Kernels
- Methods for node embeddings: DeepWalk, Node2Vec
- Graph Neural Networks: GCN, GraphSAGE, GAT, Theory of GNNs
- Knowledge graphs and reasoning: TransE, BetaE
- Deep generative models for graphs
- Applications to Biomedicine, Science, Industry

Applications of Graph ML

- Classic Graph ML task
 - **Node classification:** Predict a property of a node
 - Example: Categorize online users / items
 - **Link prediction:** Predict whether there are missing links between two nodes
 - Example: Knowledge graph completion
 - **Graph classification:** Categorize different graphs
 - Example: Molecule property prediction
 - **Clustering:** Detect if nodes form a community
 - Example: Social circle detection
 - **Other tasks:**
 - **Graph generation:** Drug discovery
 - **Graph evolution:** Physical simulation
- Example of Node Level

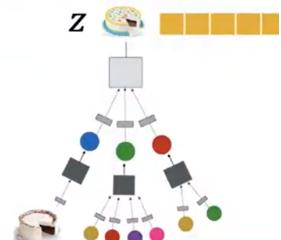
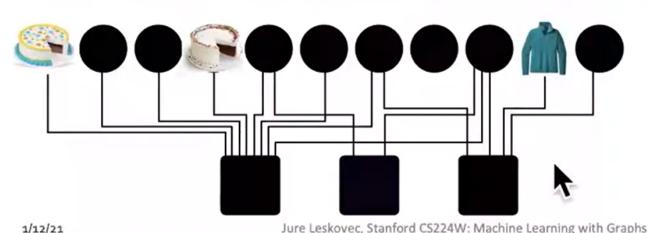
- Protein folding → alphafold로 해결
 - node : protein sequence에서의 amino acid
 - edge : residue들 간 proximity
- Example of Edge Level
 - Recommender systems
 - node : users + items
 - edge : user-item interaction
 - PinSage : Graph-based Recommender

Task: Recommend related pins to users

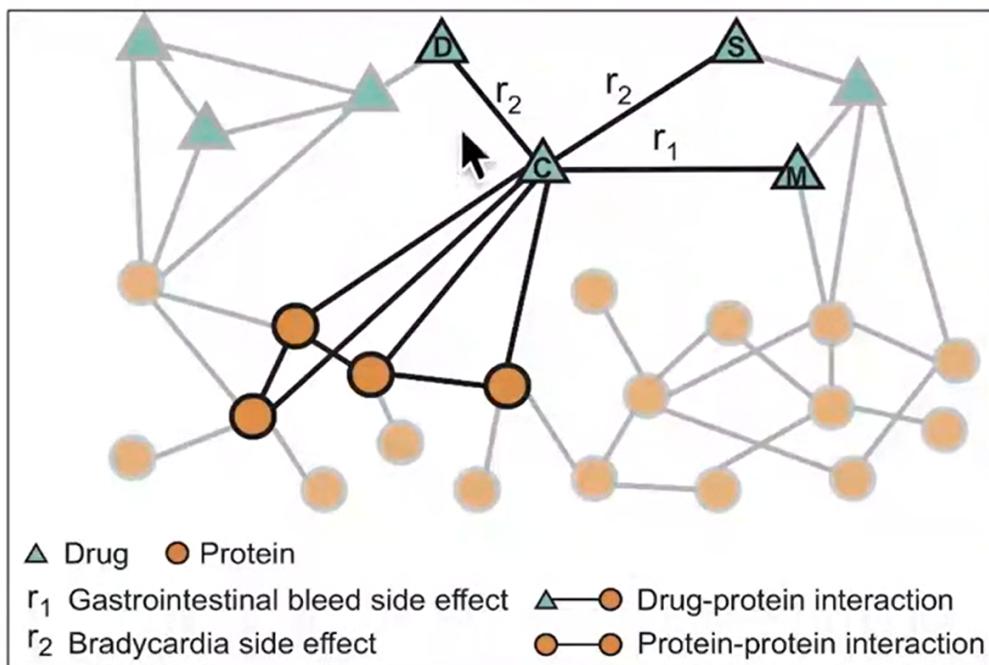


Task: Learn node embeddings z_i such that
 $d(z_{\text{cake}1}, z_{\text{cake}2}) < d(z_{\text{cake}1}, z_{\text{sweater}})$

Predict whether two nodes in a graph are related



- Drug side effect
 - 합병증 환자들은 여러 약물들을 한번에 복용하는데 이때 같이 복용할 때의 부작용 예측
 - node는 drug와 protein / edge는 interaction



- missing connection을 찾는 것이 문제
- Example of Subgraph Level
 - Traffic prediction
 - node : road segment
 - edge : connectivity between road segment
- Example of Graph Level
 - antibiotics
 - node : atom
 - edge : chemical bond
 - molecule generation / optimization

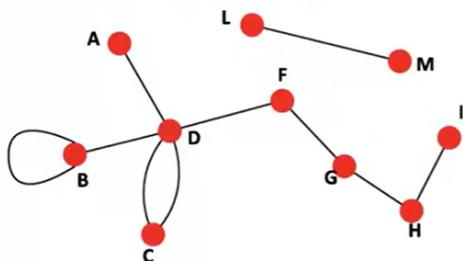
Choice of graph representation

- Component of network
 - objects (N) : nodes, vertices

- interaction (E) : links, edges
- system (G(N,E)) : network, graph
- Choosing a proper representation 또한 매우 중요함
 - node와 edge를 어떻게 설정할 것인가
- Directed vs Undirected Graphs

Undirected

- Links: undirected
(symmetrical, reciprocal)



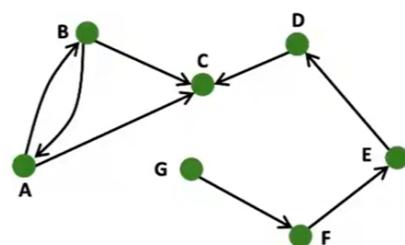
- Examples:

- Collaborations
- Friendship on Facebook

- Node degree

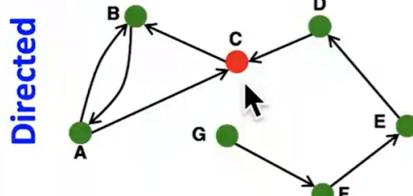
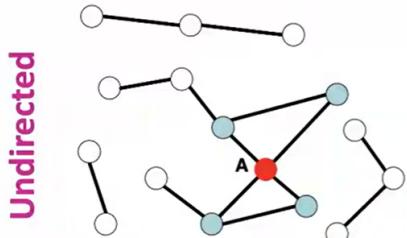
Directed

- Links: directed
(arcs)



- Examples:

- Phone calls
- Following on Twitter



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

Node degree, k_i : the number of edges adjacent to node i

$$k_A = 4$$

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

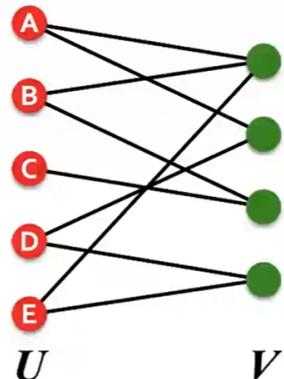
In directed networks we define an **in-degree** and **out-degree**. The (total) degree of a node is the sum of in- and out-degrees.

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

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

- Bipartite Graph

- **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**



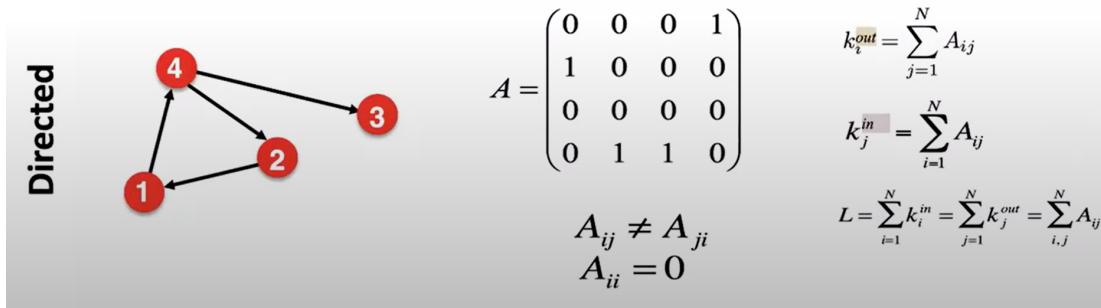
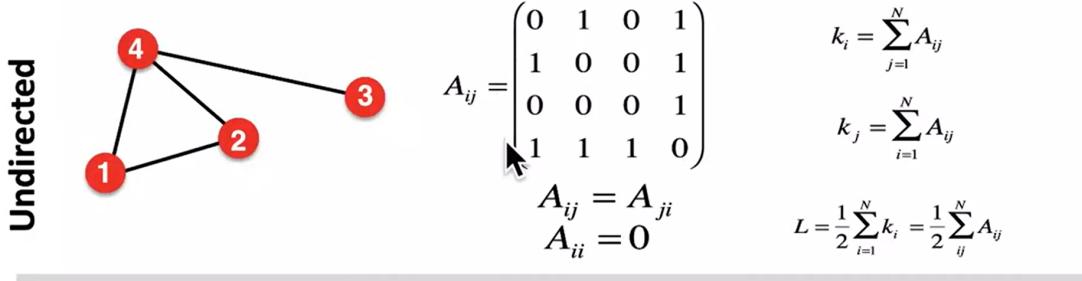
- **Examples:**

- 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

- Representing Graph : Adjacency Matrix

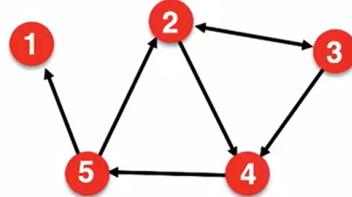


- 대부분의 adjacency matrix는 sparse matrix임

- Representing Graph : 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

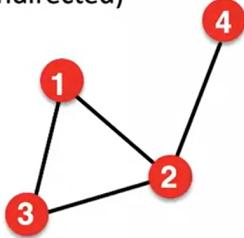


- Node and Edge attribute
 - weight

- ranking
 - type
 - sign
 - properties depending on the structure of the rest of the graph
- More types of graphs

■ Unweighted

(undirected)



$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

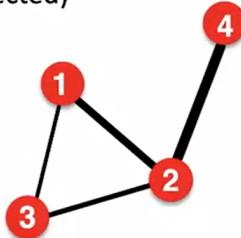
$$A_{ii} = 0 \quad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^N A_{ij} \quad \bar{k} = \frac{2E}{N}$$

Examples: Friendship, Hyperlink

■ Weighted

(undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

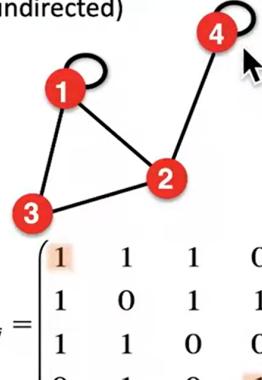
$$A_{ii} = 0 \quad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^N \text{nonzero}(A_{ij}) \quad \bar{k} = \frac{2E}{N}$$

Examples: Collaboration, Internet, Roads

■ Self-edges (self-loops)

(undirected)



$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

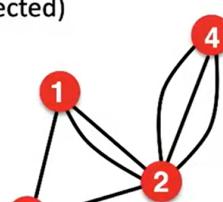
$$E = \frac{1}{2} \sum_{i,j=1, i \neq j}^N A_{ij} + \sum_{i=1}^N A_{ii}$$

$$A_{ii} \neq 0 \quad A_{ij} = A_{ji}$$

Examples: Proteins, Hyperlinks

■ Multigraph

(undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$$E = \frac{1}{2} \sum_{i,j=1}^N \text{nonzero}(A_{ij}) \quad \bar{k} = \frac{2E}{N}$$

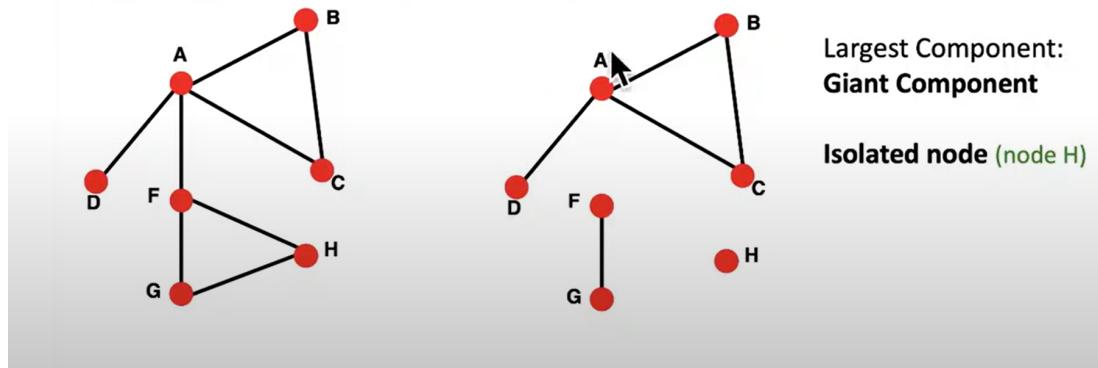
$$A_{ii} = 0 \quad A_{ij} = A_{ji}$$

Examples: Communication, Collaboration

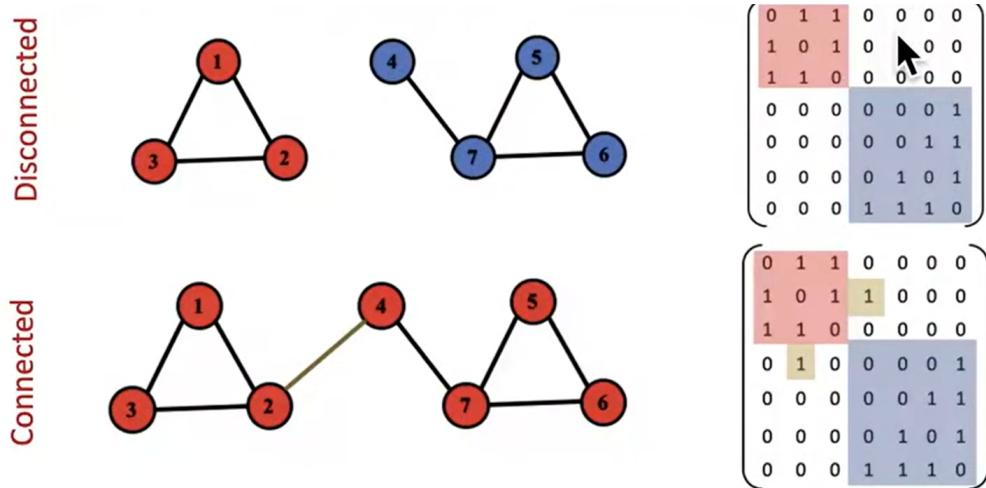
- Connectivity of Undirected graphs

■ Connected (undirected) graph:

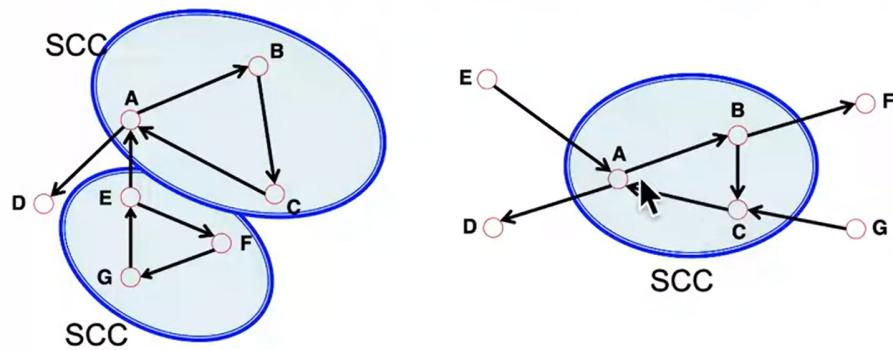
- Any two vertices can be joined by a path
- A disconnected graph is made up by two or more connected components



- adjacency matrix가 block diagonal form으로 만들어질 수 있음



- 종류
 - strongly connected directed graph
노드들끼리 link 있음 (A-B path와 B-A path 둘 다 존재)
SCC (strongly connected components)



In-component: nodes that can reach the SCC,

Out-component: nodes that can be reached from the SCC.

- Weakly connected directed graph