

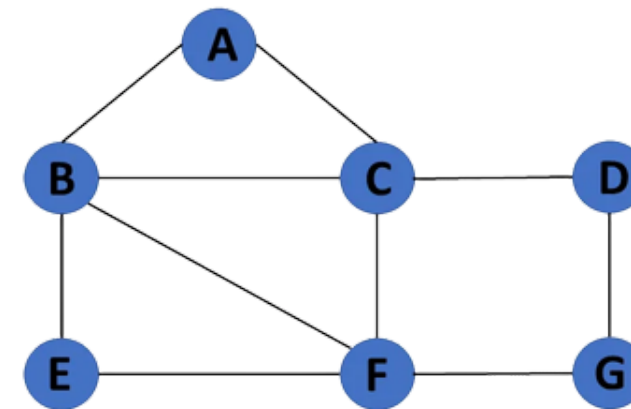
지식 그래프를 활용한 추천 서비스

방 학 2 주 차



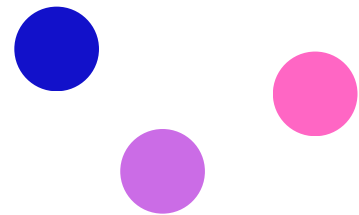
(X)

컴퓨터 과학에서 말하는 그래프는?



(O)

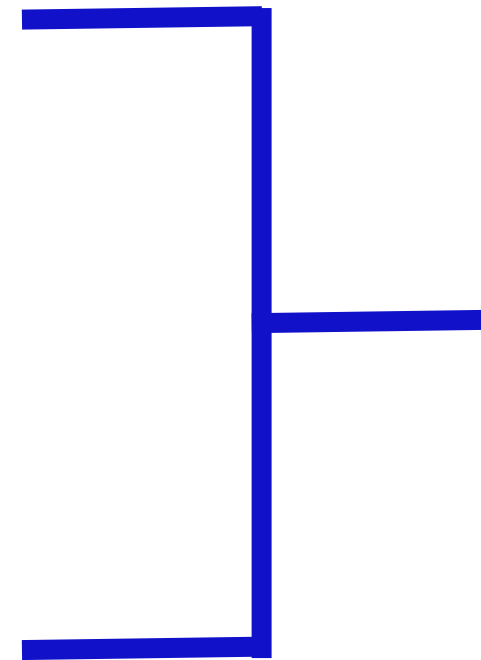
그래프란?



정점 V
(node, vertex)



간선 E
(edge)

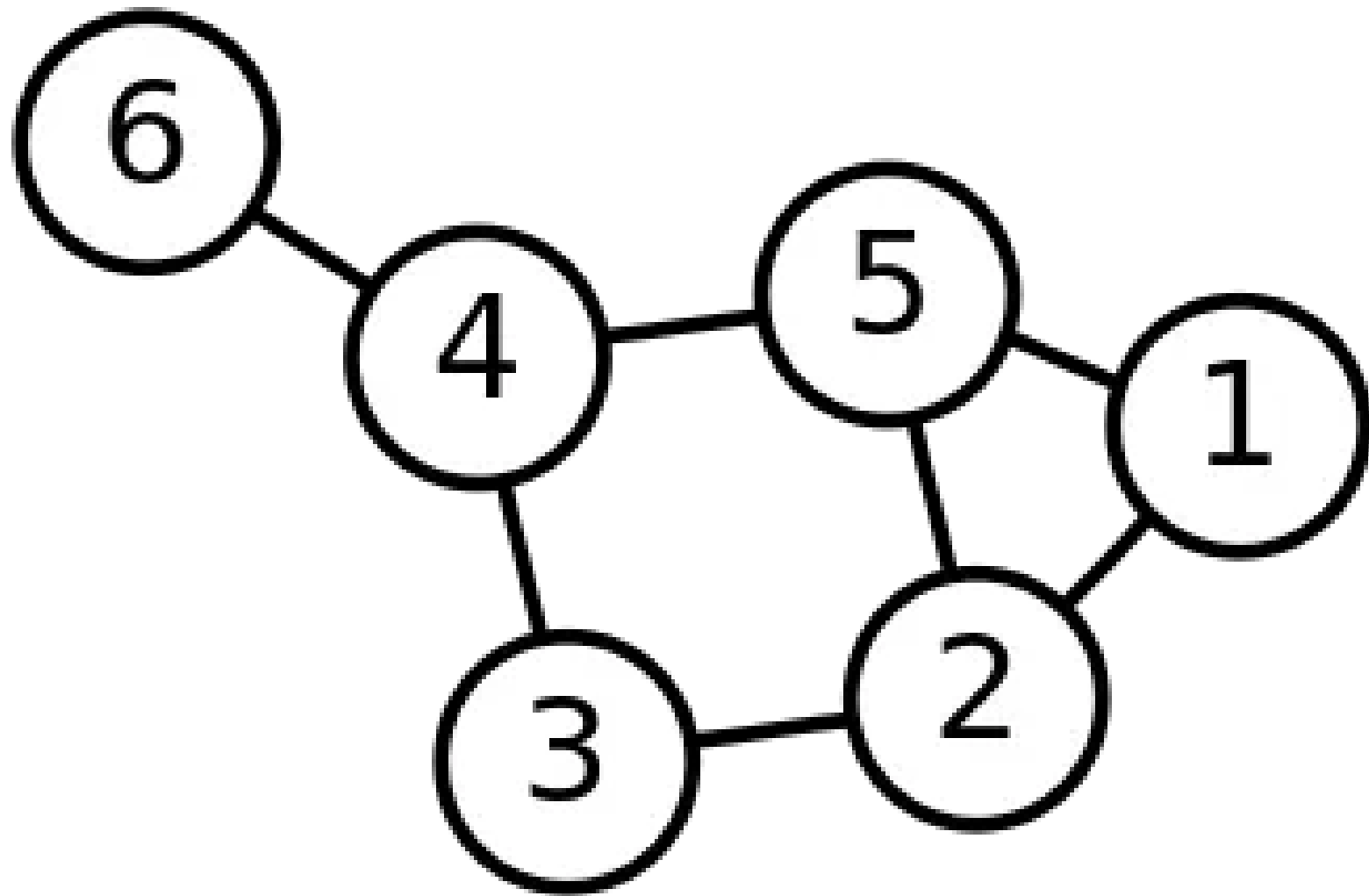


두 집합으로 구성된
구조

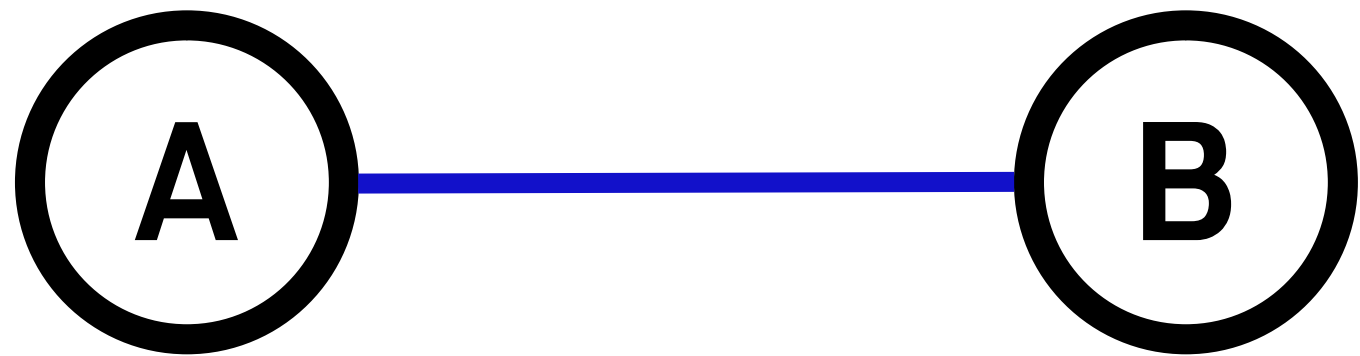
그래프란?

$$V = \{1, 2, 3, 4, 5, 6\}$$

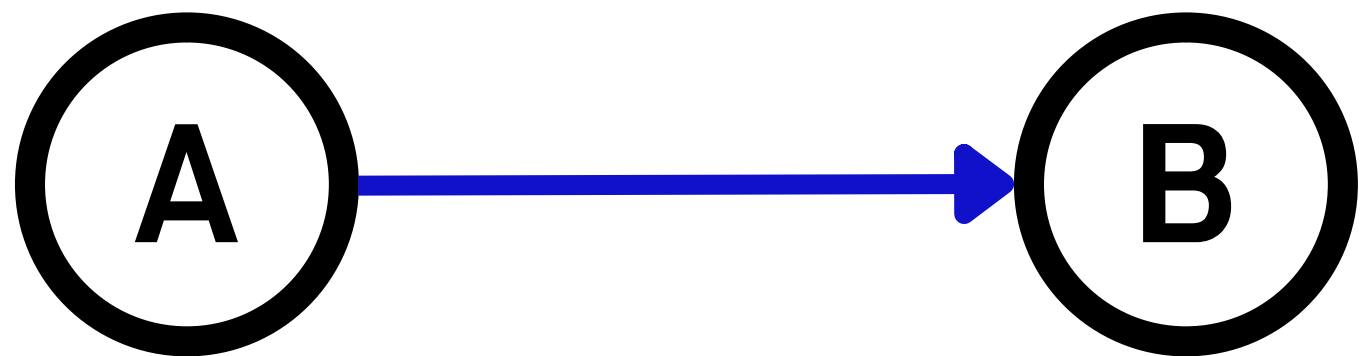
$$E = \{ \\ (4, 6), (3, 4), \\ (4, 5), (2, 3), \\ (2, 5), (1, 5), \\ (1, 2) \\ \}$$



그래프의 종류 1

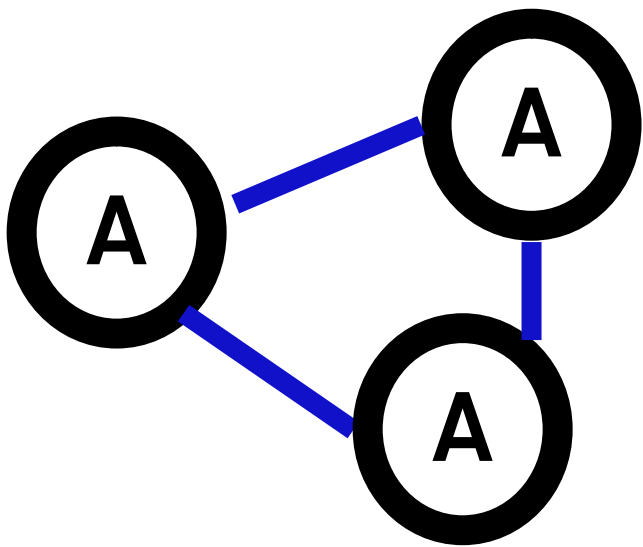


무향 그래프 (Undirected graph)



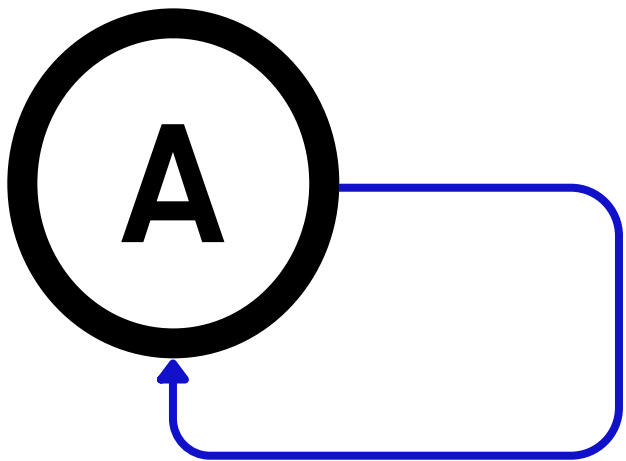
유향 그래프 (Directed graph)

그래프의 종류 2

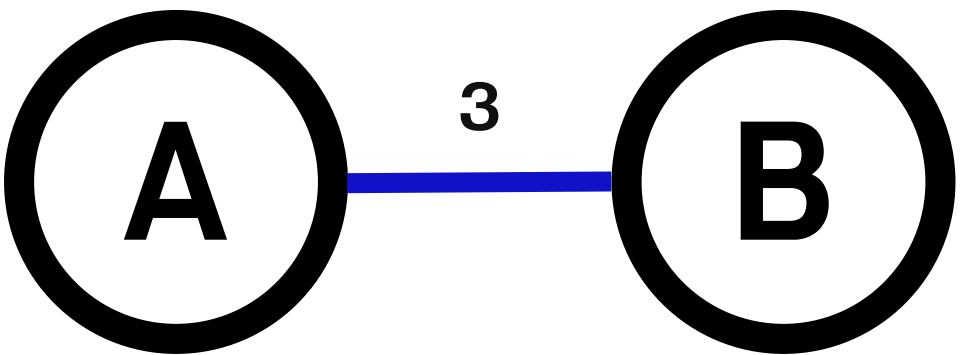


2개가 합쳐지면
pseudograph

다중 그래프
(Multi-graph)

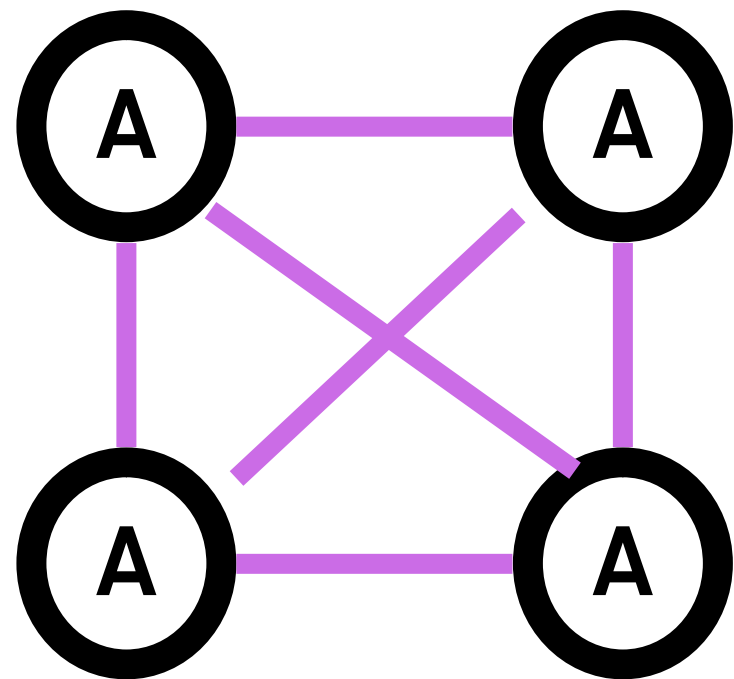


루프
(Loop)

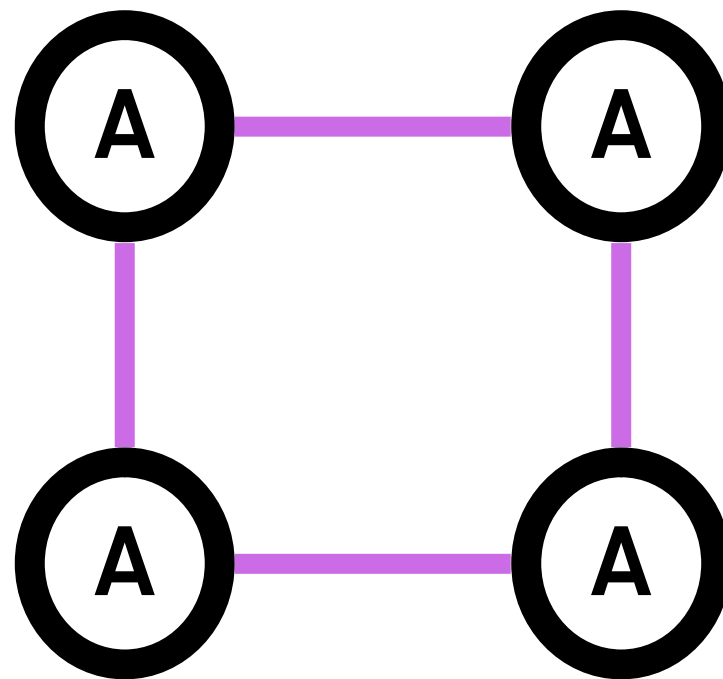


가중치 그래프
(Weighted Graph)

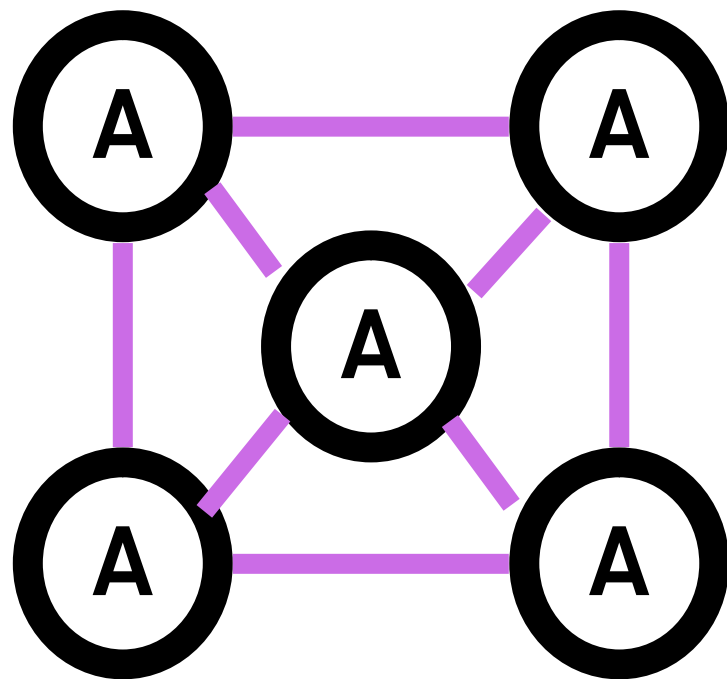
그래프의 종류 3



완전그래프
(Complete Graph)

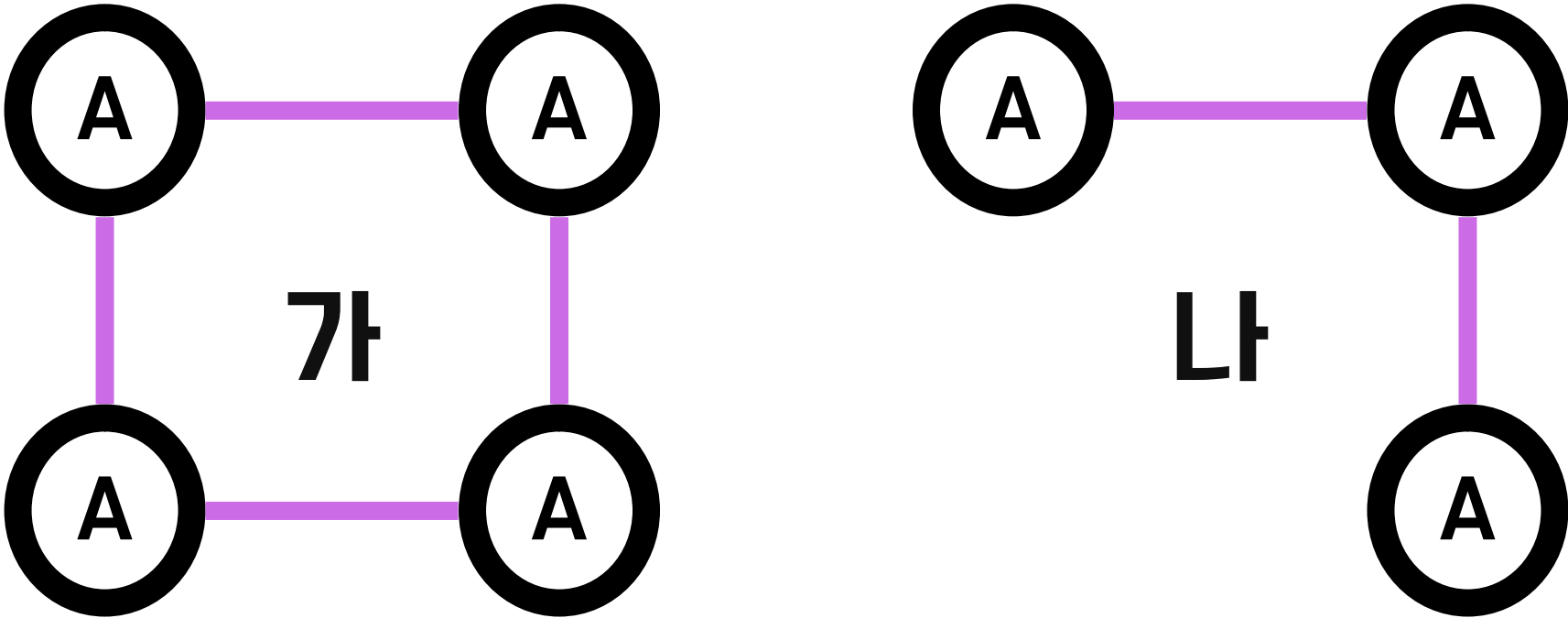


사이클
(Cycle)



휠
(Wheel)

그래프의 종류 4



부분그래프
(Subgraph)

그래프 관련 개념

차수 (Degree)

node에 연결된 edge 수
 $\Rightarrow d(v)$ 혹은 $\deg(v)$

무향그래프에서 $\deg(v)$:
인접한 edge 개수

유향그래프에서 $\deg(v)$:
입력 차수와 출력차수로 나뉨

무향 그래프에서
모든 \deg 의 합은
edge 개수의 2배

입력차수 (in-degree) : $\deg^- (V)$
정점으로 들어오는 edge 개수

출력차수 (out-degree) : $\deg^+ (V)$
정점에서 나가는 edge 개수

인접 (adjacent)

정점 u 와 v 가 하나의 edge로
연결된 경우의 u 와 v



부속 (adjacent)

정점 u 와 v 가 하나의 edge로
연결된 경우의 edge

그래프 관련 개념

Adjacency Matrix

n개의 node를 가진 그래프를
 $n * n$ 의 행렬로 표현한 것

Isomorphic

똑같은 node와 edge로 구성된 그래프는
모양이 달라도 isomorphic(동형)하다고 함.

Incident Matrix

n개의 node와 m개의 edge를
 $n * m$ 의 행렬로 표현한 것

Path

서로 연결되어 있는 간선 혹은
정점들을 순서대로 나열한 것

Connected

모든 정점들의 쌍에 대해서
어떻게든 경로가 존재하면
connected 됐다고 함.

그래프를 사용하는 것들

사회적 관계망
(Social Network)

소프트웨어 디자인
(Software Design)

교통 네트워크
(Transportation Network)

스포츠 토너먼트
(Sport Tournaments)

과제

1. Adjacency Matrix와 Incident Matrix를 통해
앞서 배운 그래프들을 python으로 만들어보기

다중그래프, 가중치 그래프, 완전 그래프, 사이클, 휠, 부분그래프

과제

1. **Adjacency Matrix와 Incident Matrix를 통해
앞서 배운 그래프들을 python으로 만들어보기**

다중그래프, 가중치 그래프, 완전 그래프, 사이클, 휠, 부분그래프

GNN이란?

그래프에 적용하는 인공지능망

CNN

그림에서 특징을 추출

LSTM

시계열 데이터에서
특징을 추출

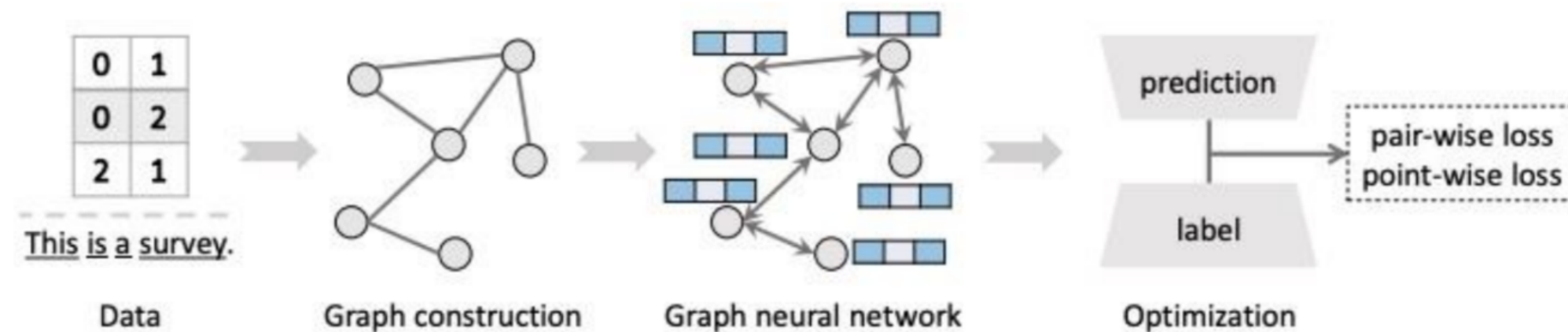
GNN

그럼 GNN은?



그래프에서
특징을 추출함

GNN의 절차



데이터를 그래프로 바꾸기 > GNN 진행 > 최적화하기

GNN의 종류

(1) Recurrent GNN

- Banach Fixed-Point Theorem을 기반으로 임베딩을 진행함

(2) Spatial Convolutional Network

- 중심 노드와 주변 노드들에 대한 정보를 통해 convolution을 진행함

(3) Spectral Convolutional Network

- RGNN의 방식을 CNN으로 가져옴

GNN의 종류

모든 GNN은 Node Embedding을 목표로 함.

Node Embedding

주로 연결관계와 이웃들의 상태를 이용하여 각 점의 상태를 업데이트(학습)하고 마지막 상태를 통해 예측 업무를 수행했을 때, 그 마지막 상태

GNN의 종류

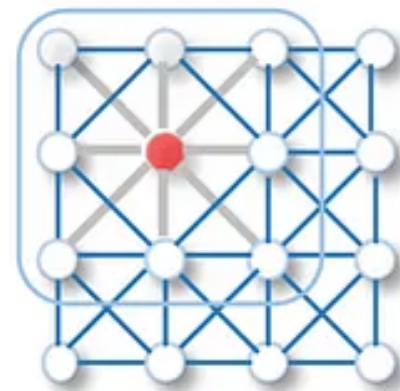
Recurrent GNN

$$\mathbf{x}_n = f_{\mathbf{w}}(\mathbf{l}_n, \mathbf{l}_{\text{co}[n]}, \mathbf{x}_{\text{ne}[n]}, \mathbf{l}_{\text{ne}[n]})$$

$$\mathbf{o}_n = g_{\mathbf{w}}(\mathbf{x}_n, \mathbf{l}_n)$$

GNN의 종류

Spatial Convolutional Network



(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size.



(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbors. Different from image data, the neighbors of a node are unordered and variable in size.

Fig. 1: 2D Convolution vs. Graph Convolution.

GNN의 종류

Spectral Convolutional Network

$$Z = f(X, A) = \text{softmax}(\hat{A} \text{Relu}(\hat{A} X W^{(0)}) W^{(1)})$$

$$A \times X = H$$

1	1	1	0	×	1	0	0	=	1	1	1
1	1	1	0		0	1	0		1	1	1
1	1	1	1		0	0	1		1	2	2
0	0	1	1		0	1	1		0	1	2

$$h_{1,1} = A_{1,1}X_{1,1} + A_{1,2}X_{2,1} + A_{1,3}X_{3,1} + A_{1,4}X_{4,1}$$

$$h_{1,2} = A_{1,1}X_{1,2} + A_{1,2}X_{2,2} + A_{1,3}X_{3,2} + A_{1,4}X_{4,2}$$

$$h_{1,3} = A_{1,1}X_{1,3} + A_{1,2}X_{2,3} + A_{1,3}X_{3,3} + A_{1,4}X_{4,3}$$

GNN를 사용하는 Task

(1) Node Classification

- Node Embedding을 통해 node를 분류

(2) Link Prediction

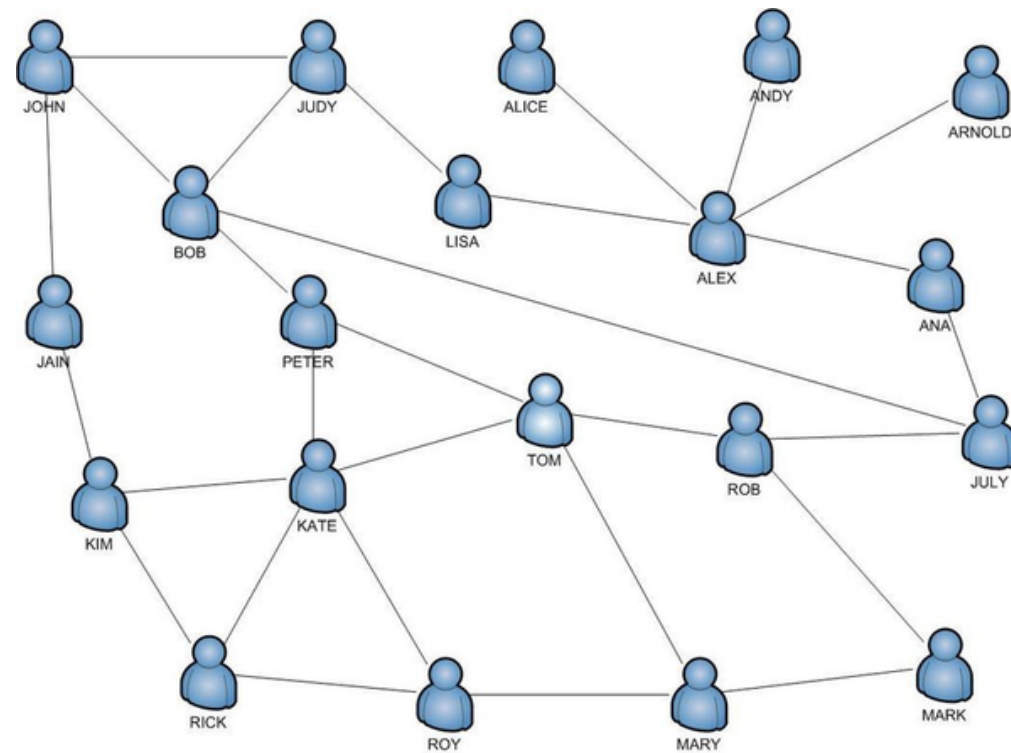
- node들 사이에 연관성을 예측

(3) Graph Classification

- 그래프 자체를 카테고리에 따라 분류

GNN를 사용하는 Task

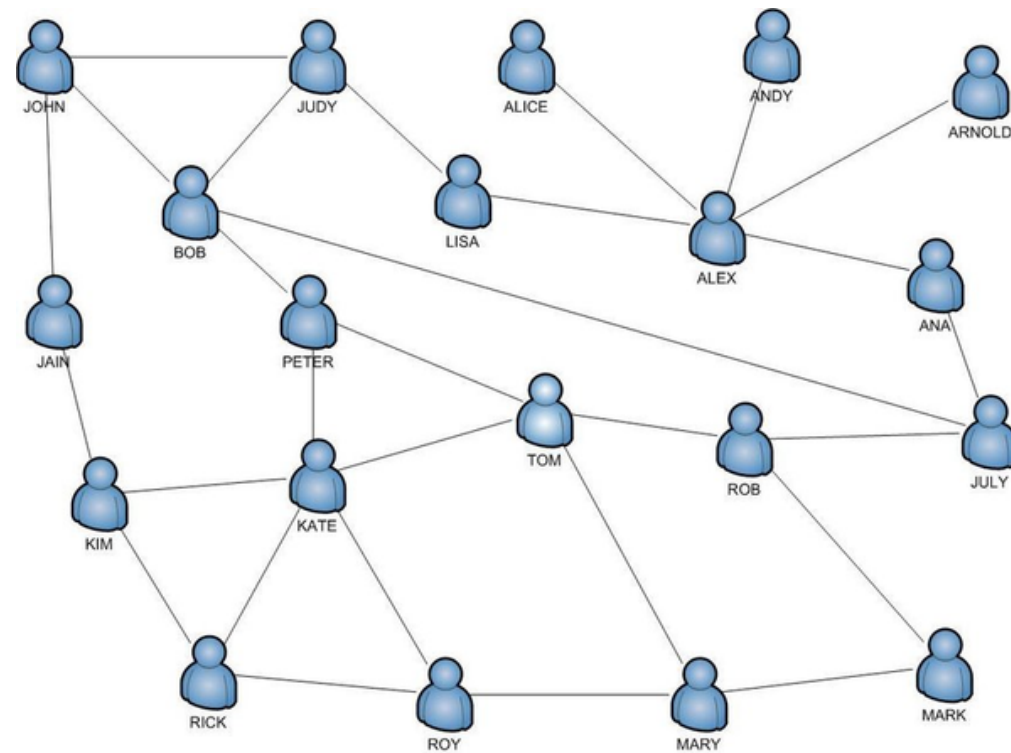
Node Classification



이중에서 가짜 계정은 무엇일까?

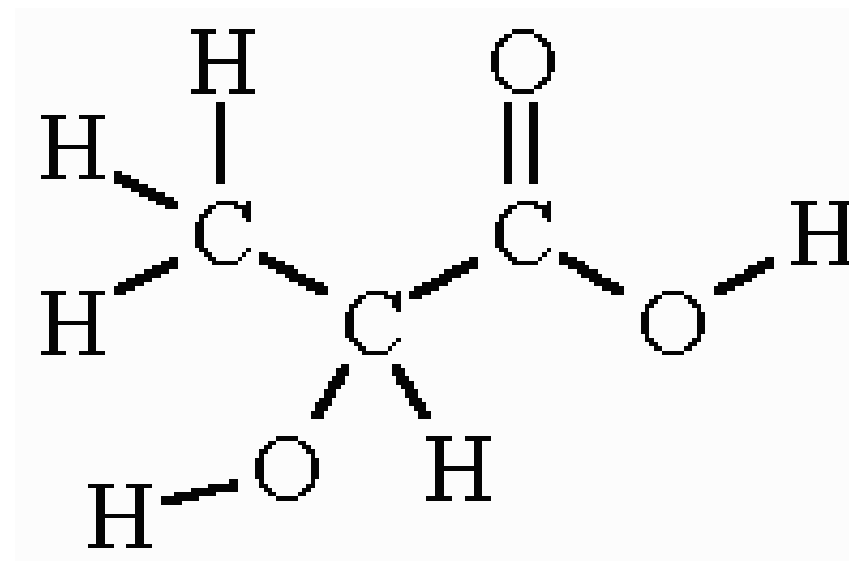
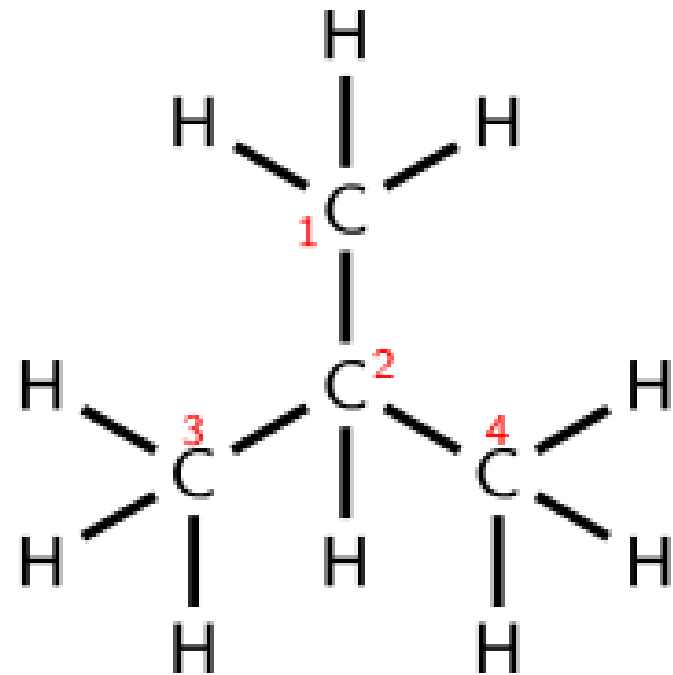
GNN를 사용하는 Task

Link Prediction



서로 얼마나 친할까?

GNN를 사용하는 Task Graph Classification



**이건 아미노산? 핵산?
아니면 뭘까?**

과제

2. 아래 링크들을 읽어보고 정리해보기

'2-2 GNN 방법론' 파트

https://velog.io/@whattsup_kim/Graph-Neural-Networks-%EA%B8%B0%EB%B3%B8-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0

<https://medium.com/watcha/gnn-%EC%86%8C%EA%B0%9C-%EA%B8%B0%EC%B4%88%EB%B6%80%ED%84%B0-%EB%85%BC%EB%AC%B8%EA%B9%8C%EC%A7%80-96567b783479>

지식 그래프란?

단어와 단어를 관계로 연결하여
단어의 관계적 의미를 보여주는 그래프

정점 => 개별 객체에 대한 정보

간선 => 각 객체간의 연관성

지식 그래프 vs 테이블

테이블

명시적으로 정의되지 않은
관계에 대한 추론이 불가능

'인용의 인용'과 같은 많은 참조가
필요한 내용에 대해서는
문서 테이블-인용 테이블을
계속 확인하며 찾아야 함

CRUD 과정에서 비용이 많이 들고,
정합성이나 연결관계를
고려해야 함

지식 그래프

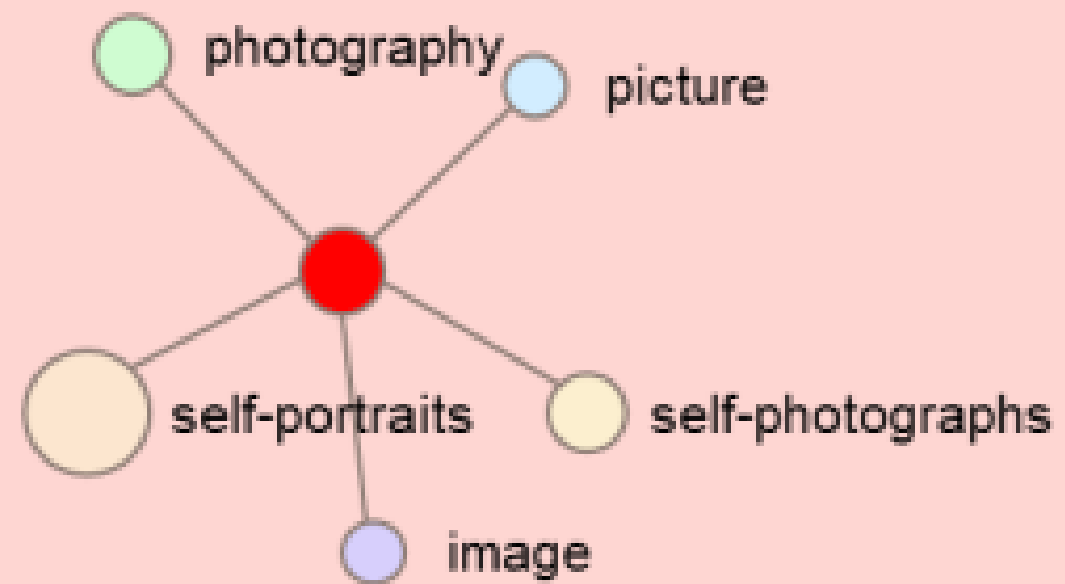
기존에 정의된 것을 토대로
새로운 관계를 탐색, 추론할 수 있음

문서가 인용 관계로 연결되어서
몇번의 접근만으로도
답을 찾을 수 있음

아주 간단하게 요청을 처리하고,
관리가 매우 쉬움

지식 그래프

[[selfie]] is a [[self-portraits]]



지식 그래프의 종류

WordNet, YAGO, Probase, ConceptNet 등

과제

3. 아래 링크들을 읽어보고 정리해보기 (Abstract, Result/Conclusion 위주로)

YAH0 :

- https://asiabiega.github.io/papers/yago3_cidr2015.pdf

Probase :

- https://www.researchgate.net/publication/241623566_Probase_A_probabilistic_taxonomy_for_text_understanding

ConceptNet :

- <https://arxiv.org/pdf/1612.03975.pdf>

가천대학교 이다

THANK YOU

GNN & KNOWLEDGE GRAPH

가천대학교 이다