

CUAI 추천시스템 스터디

2024.05.21

발표자 : 이해원

스터디원 소개 및 만남 인증



스터디원 1 : 권하연
스터디원 2 : 정성룡
스터디원 3 : 이해원

목차

진행 상황

머신러닝을 위한 그래프

노드 임베딩

그래프 임베딩

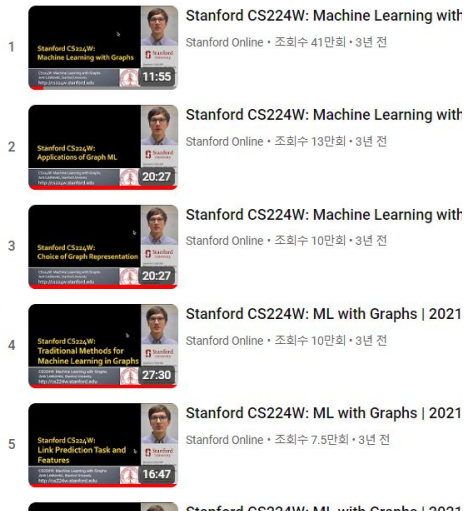
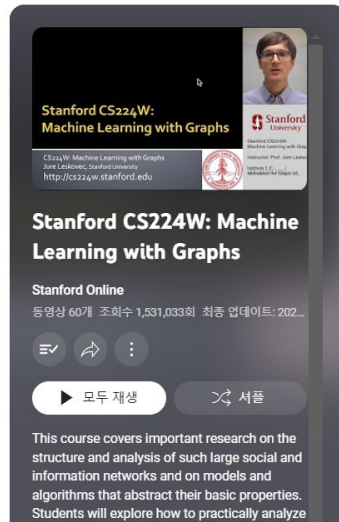
진행 상황

- Recommender Systems 도서를 활용하여 스터디를 진행
 - 이론적인 부분이 너무 많고 이해하기 어려워 주제를 바꾸기로 결정

추천시스템



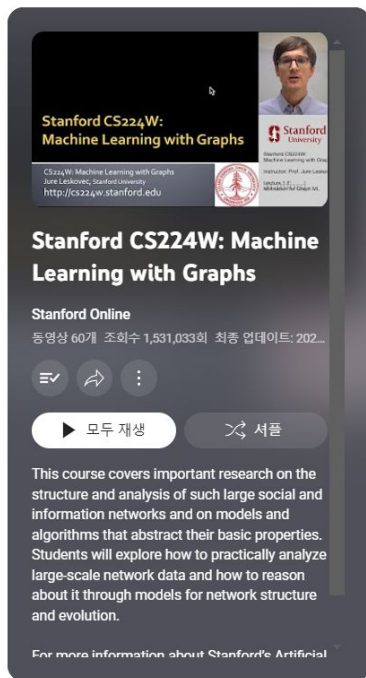
Machine Learning with Graphs



진행 상황

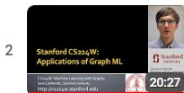
Stanford CS224W

: 매주 Lecture 2강 공부 후 파트 분배하여 스터디



Stanford CS224W: Machine Learning with Graphs | 2021 | Lecture 1.1 - Why Graphs

Stanford Online • 조회수 41만회 • 3년 전



Stanford CS224W: Machine Learning with Graphs | 2021 | Lecture 1.2 - Applications of Graph ML

Stanford Online • 조회수 13만회 • 3년 전



Stanford CS224W: Machine Learning with Graphs | 2021 | Lecture 1.3 - Choice of Graph Representation

Stanford Online • 조회수 10만회 • 3년 전



Stanford CS224W: ML with Graphs | 2021 | Lecture 2.1 - Traditional Feature-based Methods: Node

Stanford Online • 조회수 10만회 • 3년 전



Stanford CS224W: ML with Graphs | 2021 | Lecture 2.2 - Traditional Feature-based Methods: Link

Stanford Online • 조회수 7.5만회 • 3년 전



Stanford CS224W: ML with Graphs | 2021 | Lecture 2.3 - Traditional Feature-based Methods: Graph

Stanford Online • 조회수 6.9만회 • 3년 전

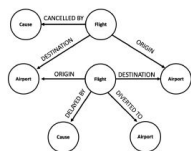


Stanford CS224W: Machine Learning with Graphs | 2021 | Lecture 3.1 - Node Embeddings

머신러닝을 위한 그래프

그래프의 정의와 활용

- 꼭지점(Node)들과 그 노드를 잇는 변(간선, Edge)들을 모아 구성한 자료구조
- 소셜 네트워크, 역학조사, 분자구조, 알고리즘 등 다양한 분야에 활용
- 복잡한 문제를 단순하게 확인할 수 있다는 장점이 있음



Event Graphs

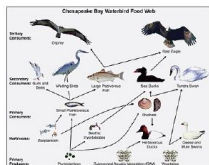


Image credit: Wikipedia

Food Webs



Image credit: Compu



Image credit: Particle

Particle



Image credit: Medium

Social Networks



Citation Networks

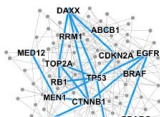


Image credit: Sci

Economic Ne



Image credit: Missoula Current News

Internet

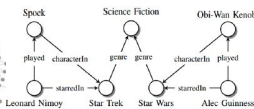


Image credit: Maximilian Nickel et al

Knowledge Graphs

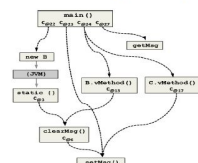


Image credit: ResearchGate

Code Graphs

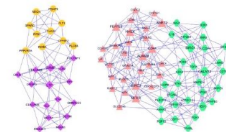


Image credit: ese.wustl.edu

Regulatory Networks

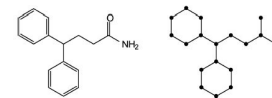


Image credit: MDPI

Molecules

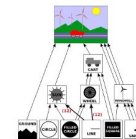


Image credit: math.hws.edu

Scene Graphs

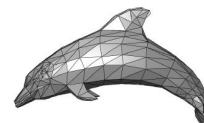


Image credit: Wikipedia

3D Shapes

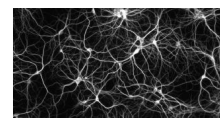


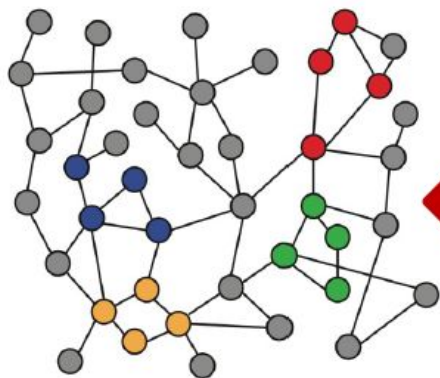
Image credit: The Conversation

Networks of Neurons

머신러닝을 위한 그래프

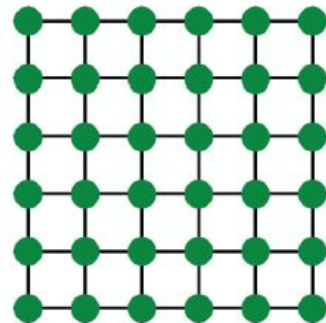
그래프 활용의 어려움

- 격자 형태로 표현 가능한 이미지와 텍스트
- 그러나 그래프는 Non-Euclidean Space에서 표현됨
- 노드와 간선의 데이터를 모두 고려하는 데이터 처리가 필요



Networks

VS.



Images



Text

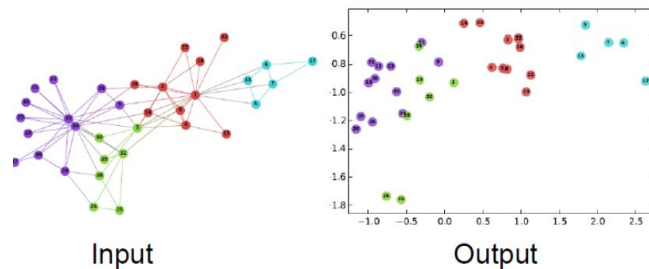
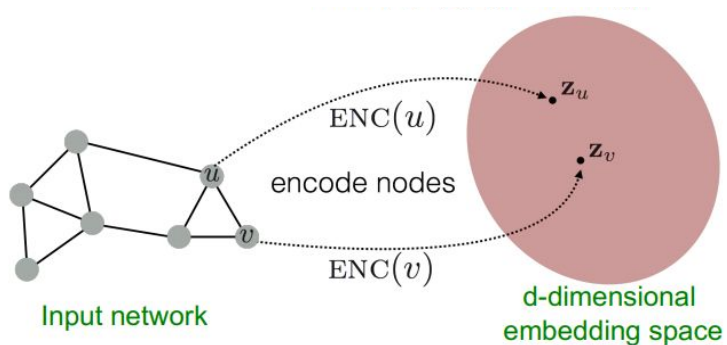
노드 임베딩

노드 임베딩?

그래프를 임베딩 공간의 벡터로 만드는 것

Goal : $\text{similarity}(u, v) \approx \mathbf{z}_v^T \mathbf{z}_u$

즉, 그래프에서 두 노드 u, v 의 유사도 \approx 임베딩 공간에서의 두 벡터 $\mathbf{z}_u, \mathbf{z}_v$ 의 유사도



노드 임베딩

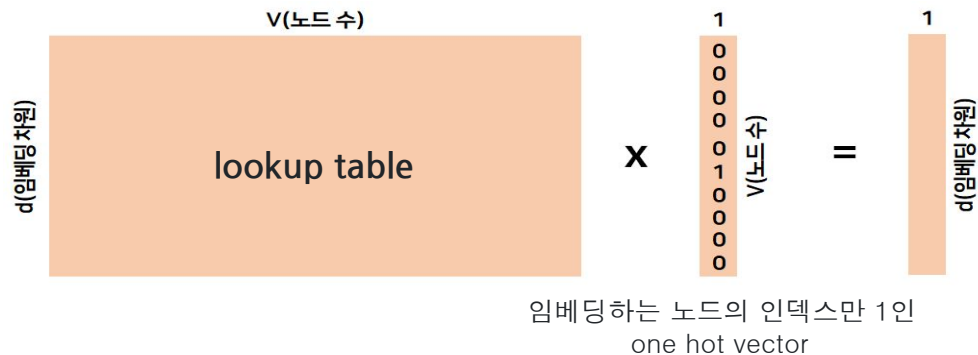
노드 임베딩의 과정

1. 인코더가 노드를 임베딩 공간으로 맵핑하여 벡터를 생성한다.
→ **lookup**
2. 노드 유사도 함수를 정의하고, 이를 통해 그래프에서 노드 간 유사도를 측정한다.
→ **RandomWalk, $P(v|u)$**
3. 디코더가 맵핑된 벡터에 대해 유사도를 측정한다.
→ **Z_u 와 Z_v 벡터의 내적**
4. 2와 3에서 측정된 유사도가 비슷해지도록 인코더의 파라미터를 최적화한다

노드 임베딩

embedding-lookup

룩업 테이블에서 입력으로 주어진 인덱스의 열만 출력

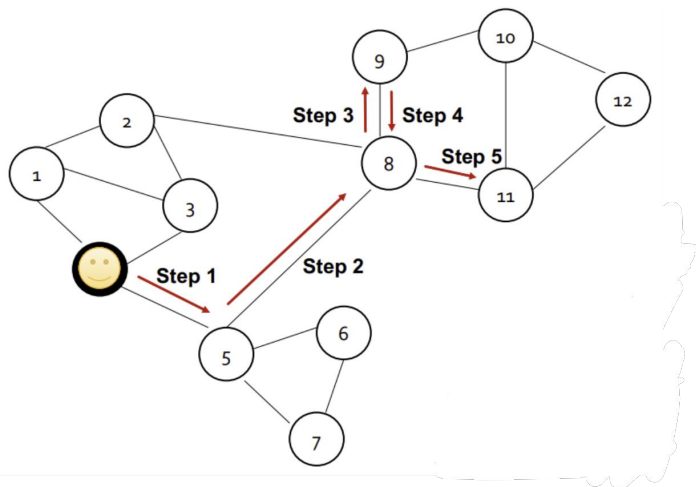


- 각각의 노드가 개별적인 임베딩 벡터를 가지게 됨
- 노드 수가 많을수록 룩업 테이블이 매우 커짐

노드 임베딩

Random Walk

: 특정 노드에서 시작하여 랜덤하게 이웃노드로 이동할때 만들어진 노드 시퀀스



각 노드에 대해 랜덤워크를 기록한다.

두 노드가 전체 랜덤워크에서
동시에 등장할 확률이 높으면 유사하다

$$\mathbf{z}_u^T \mathbf{z}_v \approx \text{랜덤워크에서 노드 } u \text{와 } v \text{가 동시에 등장할 확률}$$

$$\mathbf{z}_u^T \mathbf{z}_v \approx P_R(v|u)$$

노드 임베딩

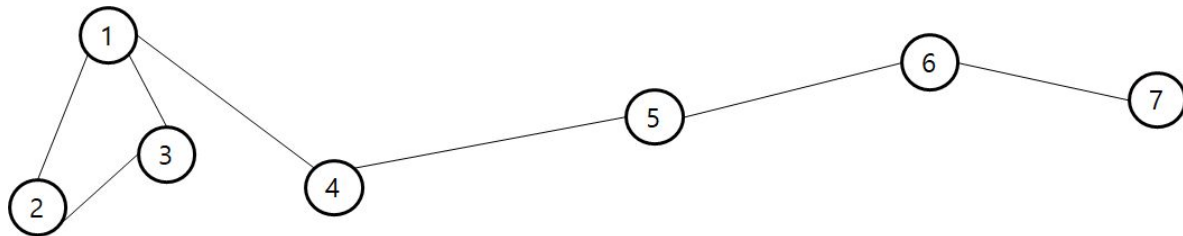
Random Walk의 장점

- **Expressivity :**

확률로 표현되기 때문에 두 노드의 경로가 짧은 경우는 물론이고 경로가 긴 경우에도 이웃 정보를 잡아낼 수 있다.

- **Efficiency :**

훈련 시 모든 노드를 고려할 필요없이
랜덤워크 시에 동시 등장하는 노드 쌍만 고려하면 되어 효율적이다.



노드 임베딩

Random Walk의 최적화 과정

Given $G = (V, E)$

Goal : to learn a mapping $f : u \rightarrow \mathbb{R}^d : f(u) = z_u$

$N_R(u)$: 랜덤워크 전략 R에 의해 구해진 u의 이웃 노드 집합

1. 짧은 거리의 랜덤워크를 고정하여 랜덤하게 각 노드마다 랜덤워크를 구한다
2. 이를 통해 손실함수를 정의한다

손실 함수:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

노드 임베딩

Random Walk의 최적화 과정

$$P(v|\mathbf{z}_u) = \frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^T \mathbf{z}_n)}$$

이때의 확률은 소프트 맥스를 활용

- 이웃노드는 내적값이 커지고 이웃하지 않은 내적값이 작아진다

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^T \mathbf{z}_n)}\right) -$$

손실함수의 시간복잡도가 커지는 문제가 발생한다

Nested sum over nodes gives
 $O(|V|^2)$ complexity!


노드 임베딩

Random Walk의 최적화 과정

Negative Sampling

$$\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^T \mathbf{z}_n)}\right)$$

random distribution
over nodes



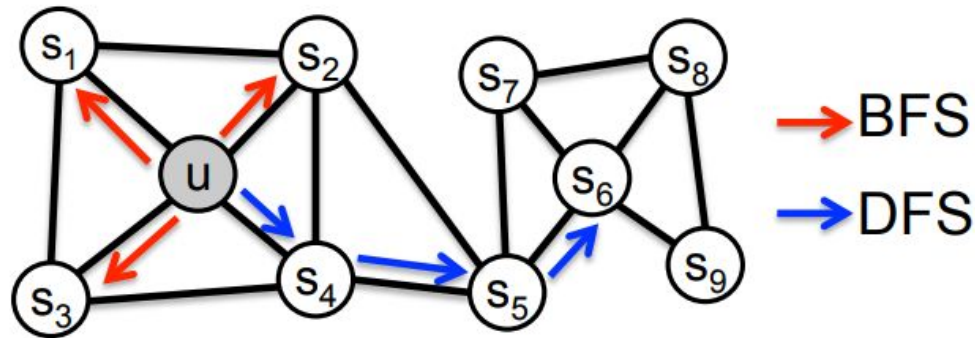
$$\approx \log(\sigma(\mathbf{z}_u^T \mathbf{z}_v)) - \sum_{i=1}^k \log(\sigma(\mathbf{z}_u^T \mathbf{z}_{n_i})), n_i \sim P_V$$

전체 노드에 대한 내적값이 아니라 몇 개의 노드를 골라 손실함수를 최적화

노드 임베딩

Node2Vec

- 랜덤 워크를 수행하여 노드의 이웃 관계를 탐색
- 워크 선택시 넓이와 깊이 중 무엇을 우선 탐색할지 결정
- 파라미터
 - p : 이전 노드로 돌아갈 가능성 > 낮을수록 **BFS**
 - q : 얼마나 새로운 곳을 잘 탐색하는가? > 낮을수록 **DFS**



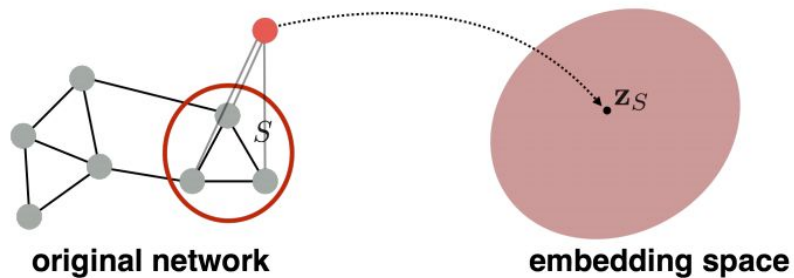
그래프 임베딩

Sum

$$\mathbf{z}_G = \sum_{v \in G} \mathbf{z}_v$$

- 그래프에 포함되는 노드들의 임베딩을 합하여 그래프 임베딩을 표현

Virtual Node



- 임베딩 하고자하는 그래프의 모든 노드와 연결된 가상의 노드를 생성고 가상의 노드를 임베딩