# Deep Graph Infomax

| | |
|---|---|
| Domain | Graph |
| tag | Graph structure, Mutual information, Unsupervised learning |
| Conference / Journal | ICLR |
| Publish year | 2019 |
| 정리 날짜 | @2024년 1월 30일 |
| AI summary | "Deep Graph Infomax (DGI)"라는 새로운 그래프 기반 비지도 학습 방법을 소개합니다. 이 방법은 그래프 구조화된 데이터에서 노드의 표현을 학습하는데, 특히 그래프 컨볼루셔널 네트워크를 사용하여 노드의 로컬한 정보와 그래프의 글로벌한 요약 사이의 상호 정보를 최대화하는 방식으로 작동합니다. 이를 통해 각 노드의 표현이 그래프의 전반적인 구조적 속성을 반영할 수 있도록 합니다. |
| AI key info | Deep Graph Infomax, Graph, ICLR, 2019, unsupervised learning, mutual information, encoder, decoder, Deep Infomax, CNN, graph convolution network, local-global mutual information maximization, discriminator, negative sampling, loss function, patch representation, global information, readout function, corruption function, objective function |

## Summary

Graph structured data를 학습하는 새로운 unsupervised learning 방법 제시

- Mutual information maximize해 global structure 정보를 가진 embedding 생성함

    - Graph convoluation architecture통해 patch representation만들어 mutual info 구함
- Transudctive, inductive classification task에 모두 활용 가능

## Background & Motivation

Graph ML의 challenge: Generalizing NN to graph-structured input

- Supervised-learning에서는 graph convolution network가 성공적이었음
    - large-scale graph에서 구조를 파악하는 task등에서 unsupervised 필요함
- Unsupervised: Random-walk based
    - 노드들이 어떻게 연결되어있는지의 인접성 정보만으로 재구성하는 것으로 너무 단순화 됨
    - 구조 정보를 희생해 노드의 인접성 정보를 과대평가함
    - encoder가 인접 노드를 similar representation으로 만들고자 해 실제 구조 파악 떨어짐
        - 인접 노드가 다른 특성 가질 수 있음
    - 초기 hyperparameter 선택에 따라 성능이 너무 달라짐

    GNN에서 encoder와 decoder의 역할

    - encoder
        - 각 노드의 특성을 representation space에 숫자 벡터인 embedding으로 변환
            - 노드들의 복잡한 관계와 속성을 잘 표현하는 embedding 만들어야
        - 인접한 노드가 유사한 특성을 가진다고 가정
            - 가까울수록 representation space 내에서 비슷한 숫자 값을 갖게 만듬
    - decoder
        - embedding 사용해 원래 그래프 정보 재구성 혹은 예측

### Deep graph infomax: unsupervised learning based on mutual information

- Random walk 방식이 아니라, mutual information 기반

- Deep Infomax: mutual information 사용하는 MINE모델 발전시킴
  - MINE: statistic network를 2개 변수의 joint distribution에 classifier로 사용
- CNN기반으로 이미지 처리에 사용되던 Deep Infomax를 그래프에 적용해보자!
  - 흠 이게 GCN이랑 머가다름 그럼?

## Methodology

### Graph-based Unsupervised learning

- Encoder $\mathcal{E}(\mathbf{X}, \mathbf{A}) = \mathbf{H} = \{\vec{h}_1, ..., \vec{h}_N\}$ 학습이 목표
  - Graph convolution encoder가 local 이웃에 aggregation 반복해 node representation $H$ 만듬
  - $\vec{h}_i \in R^{F'}$는 각 node i의 high-level representation
    - 노드 자체가 아니라 노드 중심 그래프 일부분: patch를 summarize함
- node feature $X = \{\vec{x}_1, ..., \vec{x}_N\}$와 adjacency matrix $A \in R^{N \times N}$ input으로 받음
  - graph는 unweighted라고 가정해 adjacency 1 or 0 구성

### Local-Global mutual information maximization

- Learning enoder : maximizing local-global mutual information alignment
  - node representation이 global information을 갖도록 하는 것
  - Graph-level summary vector $\vec{s} = \mathcal{R}(\mathcal{E}(X, A))$
    - Readout function $\mathcal{R}$으로 patch representation을 summarize해서 global-level로 나타냄
  - Discriminator $\mathcal{D}(\vec{h}_i, \vec{s})$: local, global representation이 얼마나 잘 일치하는가
  - Negative sample for $\mathcal{D}(\vec{\tilde{h}}_i, \vec{s})$: 부정적인 사례 $(\tilde{X}, \tilde{A})$를 학습하도록 함
    - $\vec{s}$와 상관없는 patch를 가져와 비교하도록 함
    - single graph의 경우, 비교할 arbitrary alternative가 없으니 corruption function으로 만듬

        - Corruption function: 원래 그래프로부터 negative example을 만드는 함수
- Loss function

$$\mathcal{L}=\frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\mathcal{D}\left(\vec{h}_i,\vec{s}\right)\right]+\sum_{j=1}^{M}\mathbb{E}_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\tilde{h}}_j,\vec{s}\right)\right)\right]\right)$$

    - Deep InfoMax에서 사용한 방식대로 binary cross-entropy loss 사용
        - noise-contrastive type objectivve를 적용
        - positive example과 negative example의 loss를 구함
    - $\vec{s}$와 $\vec{h}_i$ 간의 mutual information을 최대화
- patch level의 similarity를 discover, preserve할 수 있음
    - 비슷한 patches간의 link를 만드는 것이 목표, not summary가 모든 similarities 저장

**DGI procedure**

1. Corruption function으로 negative sampling
2. Encoder에 그래프 넣어 patch representation $\vec{h}_i$만들기 $H=\mathcal{E}(X,A)$
3. Encoder에 negative examplee 넣어 patch representation $\vec{\tilde{h}}_i$만들기 $\tilde{H}=\mathcal{E}(\tilde{X},\tilde{A})$
4. Readout function으로 $\vec{h}_i$s summarize 함 $\vec{s}=\mathcal{R}(H)$
5. $\mathcal{D}$ 통해 loss 구해서 gradient descent로 parameter $\mathcal{E},\mathcal{R},\mathcal{D}$업데이트

## Questions

- loss function 부분이 이해가 잘 안됨
    - Discrimminator가 probability score을 0부터 1까지의 값으로 가지는거임? 이거를 loss function인지 헷갈렸는데 그럼 이게 objective function인거지? (negative, 0]

의 값밖에 안나와서

- 발표는 loss function이라고 하심
  - Objective가 맞는듯?, -붙여서 loss로 사용