Semi-Supervised Classification with Graph Convolutional Networks

School of Industrial and Management Engineering, Korea University

Young Jae Lee

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ Semi-Supervised Classification with Graph Convolutional Networks (2017, ICLR)

- 암스테르담 대학교에서 연구하였고 2022년 01월 14일 기준으로 12124회 인용

Published as a conference paper at ICLR 2017

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Thomas N. Kipf
University of Amsterdam
T.N.Kipf@uva.nl

Max Welling
University of Amsterdam
Canadian Institute for Advanced Research (CIFAR)
M.Welling@uva.nl

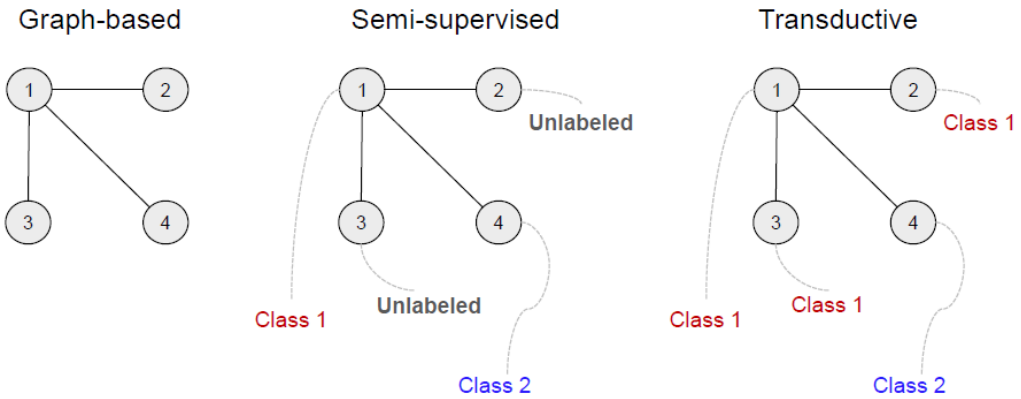
ABSTRACT

We present a scalable approach for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on graphs. We motivate the choice of our convolutional architecture via a localized first-order approximation of spectral graph convolutions. Our model scales linearly in the number of graph edges and learns hidden layer representations that encode both local graph structure and features of nodes. In a number of experiments on citation networks and on a knowledge graph dataset we demonstrate that our approach outperforms related methods by a significant margin.

Research Purpose

❖ Semi-Supervised Classification with Graph Convolutional Networks (2017, ICLR)

- 그래프에서 노드를 분류하는 문제를 해결하고자 함
  - ✓ 하나의 그래프 안 노드 분류 문제에서 레이블은 오직 작은 집합의 노드에만 가능
  - ✓ 하나의 그래프 안 다수 노드에 레이블이 없을 때 **Transductive 관점**으로 볼 수 있음
  - ✓ 레이블 정보가 **Smoothing**되는 Semi-Supervised Learning으로 볼 수 있음



- 기존 방법론의 Computational Cost를 해결하기 위한 해결책 제시
  - ✓ Spectral Graph Convolution의 빠르고 효율적인 1차 근사 증명
  - ✓ 기존 Semi-Supervised 조건 연구에서 사용하는 Loss Term 대신 GCN의 인접 행렬을 사용하여 해결

DMQA

Graph Convolutional Networks (GCNs)

❖ 기존 Semi-Supervised Loss Term

- $L = L_0 + \lambda L_{reg}$
- L_0: 레이블이 있는 노드에 대한 지도 학습 분류 손실 함수
- L_{reg}: Graph Laplacian Regularization Term
  - ✓ 연결된 노드가 비슷한 Representation을 갖도록 하는 손실 함수
  - ✓ **제한:** 연결된 노드는 유사하다는 가정이 필요하며 유사도 이외의 추가적 정보를 담지 못함
- 본 논문에서는 GCN 모델 F(X, A)를 정의하여 해결

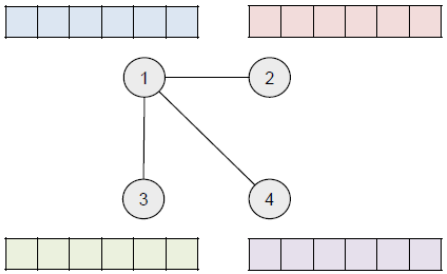
Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks (Fast Approximation Convolutions)

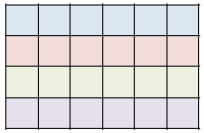
- 그래프와 인접 행렬을 모델의 입력으로 이용하는 GCN, $f(X, A)$ 제시
- Multi-Layer GCN with the Following Layer-wise Propagation Rule

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

X, A 정의



$Node - feature\ matrix$
$X \in R^{n \times F}$



$Adjacency\ matrix$
$A \in R^{n \times n}$

0	1	1	1
1	0	0	0
1	0	0	0
1	0	0	0

- $l+1$번째 레이어의 Propagation 결과
- $H^{(l)}$: l번째 레이어의 Hidden State
- $H^{(0)}$: X
- $\tilde{A}$: $A + I_N$ (Self-Connection / Renormalization Trick for Gradient Vanishing or Exploding)
- $\tilde{D}$: $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ 각 노드의 degree를 나타내는 대각 행렬
- $W^{(l)}$: l번째 레이어의 학습 가능한 파라미터
- σ: 비선형 함수

DMQA

Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks (Spectral Graph Convolutions)

- Spectral Convolutions 정의
  - ✓ Signal 데이터 x에 대한 g_θ에 대한 곱: $g_\theta * x = U g_\theta U^T x$

$$g_\theta * x = U g_\theta U^T x$$

- U: 그래프의 Normalized Laplacian 행렬 L의 고유벡터 행렬
- $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$
- Λ: 고유값을 나타내는 대각 행렬
- $U^T x$: x에 대한 그래프 Fourier 변환
- 위 식은 행렬이 커질수록 Computational Cost가 커진다는 단점이 있음(고유값 분해 수행 시)

$$\boldsymbol{g_{\theta'} * x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) x}$$

- Chebyshev 다항식으로 필터를 근사하는 방법으로 해결
  - ✓ $g_{\theta'} \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$
- $\tilde{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_N$: 은 Λ의 크기를 재조정한 행렬 / λ_{max}는 L에서 가장 큰 고유값을 의미
- θ'는 Chebyshev 계수를 나타내는 벡터
- **재귀적으로 정의: $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 1, T_1(x) = x$**
- $\tilde{L} = \frac{2L}{\lambda_{max}} - I_N$

Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks (Layer-wise Linear Model)

- 논문의 핵심은 Spectral Graph Convolution을 딥러닝과 접목
- Chebyshev 다항식을 선형 모델로 변환
  - ✓ K=1을 적용하여 선형 Convolution 필터를 만들고 여러 개의 레이어를 쌓는 구조

Page 6

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$$Z = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$$

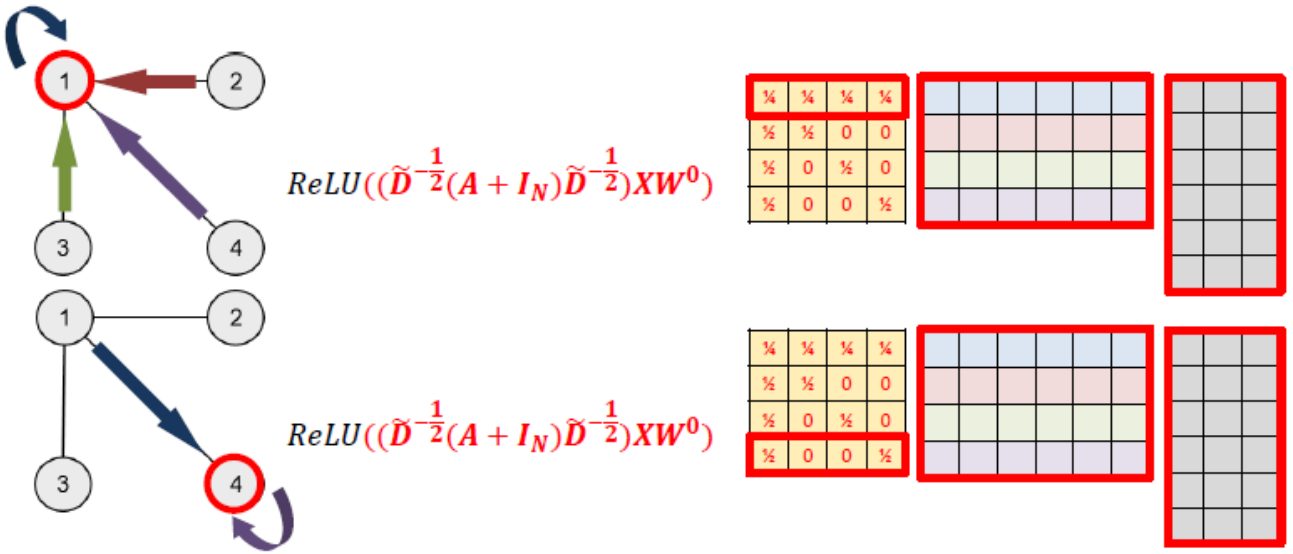
- $g_{\theta'} * x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x$
- $\lambda_{max} \approx 2$: 이 역할은 네트워크의 파라미터들로 대체 가능
- $g_{\theta'} * x \approx \theta'_0 x + \theta'_1(L - I_N)x = \theta'_0 x + \theta'_1 D^{-\frac{1}{2}}AD^{-\frac{1}{2}}x$
- $\theta = \theta'_0 = -\theta'_1$: 전체 그래프에 걸쳐 공유하는 파라미터
- $g_{\theta'} * x = \theta\left(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right)x$
- $\tilde{A}: A + I_N, \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$: Renormalization Trick

Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks

- $F(X, A)$가 핵심
- 그래프가 주어질 때 $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$값을 계산(with Renormalization Trick)
- 2-Layer GCN

$$Z = f(X, A) = softmax\left(\tilde{A}ReLU\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$



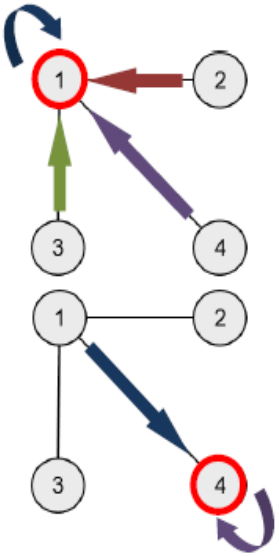
DMQA

Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks

- $F(X, A)$가 핵심
- 그래프가 주어질 때 $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$값을 계산(with Renormalization Trick)
- 2-Layer GCN

$$Z = f(X, A) = softmax\left(\tilde{A}ReLU\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$



$$ReLU\left(\left(\tilde{D}^{-\frac{1}{2}}(A + I_N)\tilde{D}^{-\frac{1}{2}}\right)XW^0\right) = ReLU\left(\hat{A}XW^0\right) = H_1$$

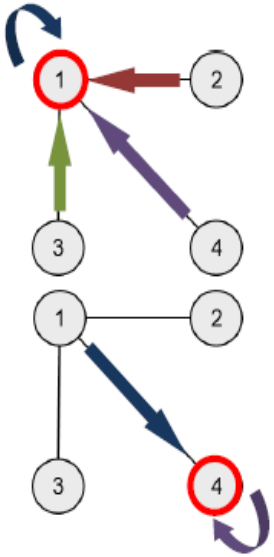
$$= ReLU\left[\begin{matrix} ¼ & ¼ & ¼ & ¼ \\ ½ & ½ & 0 & 0 \\ ½ & 0 & ½ & 0 \\ ½ & 0 & 0 & ½ \end{matrix}\right] =$$

Graph Convolutional Networks (GCNs)

❖ Graph Convolutional Networks

- $F(X, A)$가 핵심
- 그래프가 주어질 때 $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$값을 계산(with Renormalization Trick)
- 2-Layer GCN

$$Z = f(X, A) = softmax\left(\tilde{A}ReLU\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$



$$Z = f(X, A) = softmax(\hat{A}H_1W^1)$$

$$= \boldsymbol{softmax}\left[\begin{matrix} ¼ & ¼ & ¼ & ¼ \\ ½ & ½ & 0 & 0 \\ ½ & 0 & ½ & 0 \\ ½ & 0 & 0 & ½ \end{matrix}\right] = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Class 1

Class 2

Graph Convolutional Networks (GCNs)

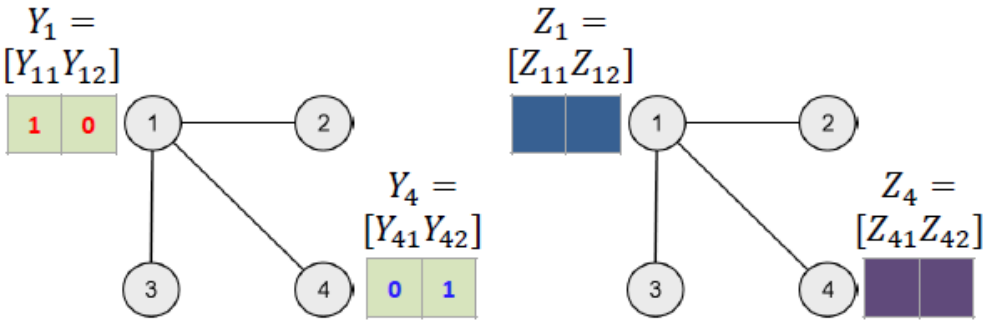
❖ Graph Convolutional Networks

- $F(X, A)$가 핵심
- 그래프가 주어질 때 $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$값을 계산(with Renormalization Trick)
- 레이블이 있는 학습 데이터(Y_L)에 대해 다음과 같은 손실 함수 계산

$$Cross\ Entropy\ Loss = -\sum_{l \in Y_L}\sum_{f=1}^{F} Y_{lf} ln Z_{lf}$$

F : *number of class*
Y_L : *all labeled examples*

$$= -(\boldsymbol{lnZ_{11} + lnZ_{42}})$$



DMQA

Experiments

❖ Dataset

- Citation Network (Citeseer, Core, Pubmed): 각 노드는 문서, Edge는 Citation Link
- NELL: Knowledge Graph로 부터 추출된 Bipartite Graph

Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Experiments

Classification Accuracy

- ❖ 동등 비교를 위해 이전 SOTA 모델과 동일한 데이터 Split 이용
- ❖ 이전 모델보다 우수한 결과를 보임

Table 2: Summary of results in terms of classification accuracy (in percent).

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Conclusion

- 그래프에서 노드를 분류하는 문제를 해결하기 위해 기존 방법론보다 효율적인 방법을 제안
- Computational Cost를 고려하며 Convolution 연산으로 대체함
- Spectral Graph Convolution의 빠르고 효율적인 1차 근사 증명 및 딥러닝과 결합하기 위한 수학적 증명 제시
- 기존 방법론보다 우수한 성능을 보여주며 효율적인 연산이 가능함을 증명
  - 후기: 논문에서 한계점을 메모리 문제, 방향성 그래프, Self-Connection 적용을 얘기했지만 최근 여러 연구에서 해결해 나가고 있기 때문에 언급은 안했음. 최근 GCN 연구에서 가장 기초가 되는 논문으로 꼭 한 번 정독할 필요가 있는 논문임.

- Reference

  - Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
  - http://dmqa.korea.ac.kr/activity/seminar/267


Thank You