Simplifying Graph Convolutional Networks

School of Industrial and Management Engineering, Korea University

Sangmin Kim

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ Simplifying Graph Convolutional Networks (ICLR 2019)

- Cornell University에서 연구하였으며 2022년 2월 11일 기준으로 898회 인용

Simplifying Graph Convolutional Networks

Felix Wu [*1] **Tianyi Zhang** [*1] **Amauri Holanda de Souza Jr.** [*1 2] **Christopher Fifty** [1] **Tao Yu** [1]
Kilian Q. Weinberger [1]

Abstract

Graph Convolutional Networks (GCNs) and their variants have experienced significant attention and have become the de facto methods for learning graph representations. GCNs derive inspiration primarily from recent deep learning approaches, and as a result, may inherit unnecessary complexity and redundant computation. In this paper, we reduce this excess complexity through successively removing nonlinearities and collapsing weight matrices between consecutive layers. We theoretically analyze the resulting linear model and show that it corresponds to a fixed low-pass filter followed by a linear classifier. Notably, our experimental evaluation demonstrates that these simplifications do not negatively impact accuracy in many downstream applications. Moreover, the resulting model scales to larger datasets, is naturally interpretable, and yields up to two orders of magnitude speedup over FastGCN.

rithms has followed a clear trend from initial simplicity to need-driven complexity. For instance, limitations of the linear Perceptron (Rosenblatt, 1958) motivated the development of the more complex but also more expressive neural network (or multi-layer Perceptrons, MLPs) (Rosenblatt, 1961). Similarly, simple pre-defined linear image filters (Sobel & Feldman, 1968; Harris & Stephens, 1988) eventually gave rise to nonlinear CNNs with learned convolutional kernels (Waibel et al., 1989; LeCun et al., 1989). As additional algorithmic complexity tends to complicate theoretical analysis and obfuscates understanding, it is typically only introduced for applications where simpler methods are insufficient. Arguably, most classifiers in real world applications are still linear (typically logistic regression), which are straight-forward to optimize and easy to interpret.

However, possibly because GCNs were proposed after the recent "renaissance" of neural networks, they tend to be a rare exception to this trend. GCNs are built upon multi-layer neural networks, and were never an extension of a simpler (insufficient) linear counterpart.

Research Purpose

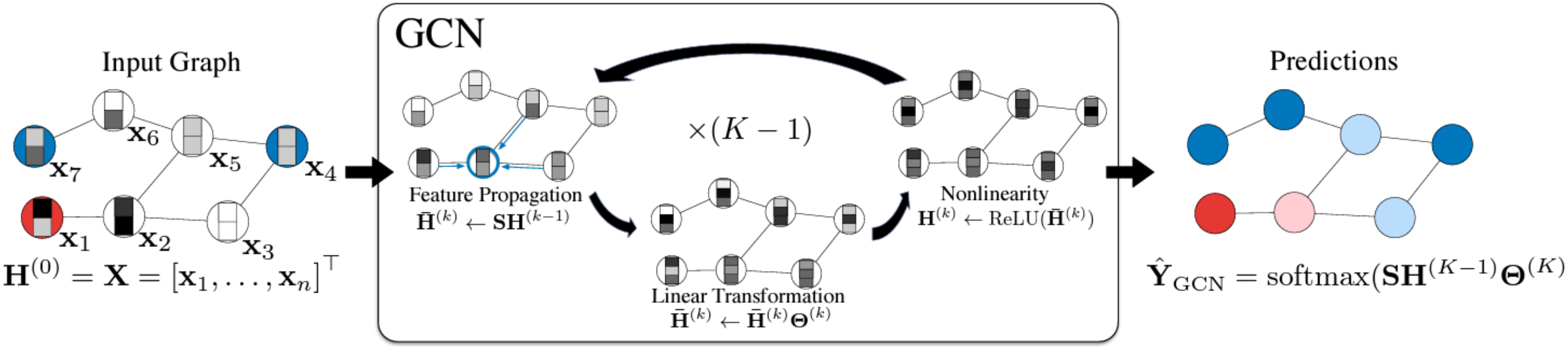
❖ Simplifying Graph Convolutional Networks (ICLR 2019)

- 대부분의 머신러닝 및 딥러닝 알고리즘은 초기 Simplicity 에서 필요에 의해(need-driven) Complexity 로 발전하는 경향을 보임
- 그러나, 최근 Neural network의 부흥기 이후, 등장한 GCN은 간단하지 않은 모델 형태로 시작하여 발전함
- 따라서, 저자들은 GCN이 필요 이상으로 복잡한 형태일 수 있음을 가정하여 모델 구조를 간소화함
- GCN 모델 내 nonlinearity 대신 linear model 형태로 변경하여 모델 파라미터를 줄여 training time 을 대폭 개선함

Graph Convolutional Networks(GCN)

❖ Graph Convolutional Network (GCN)

- GCN은 3가지 단계로 구분
  - ✓ Feature propagation
  - ✓ Linear transformation
  - ✓ Nonlinear transition

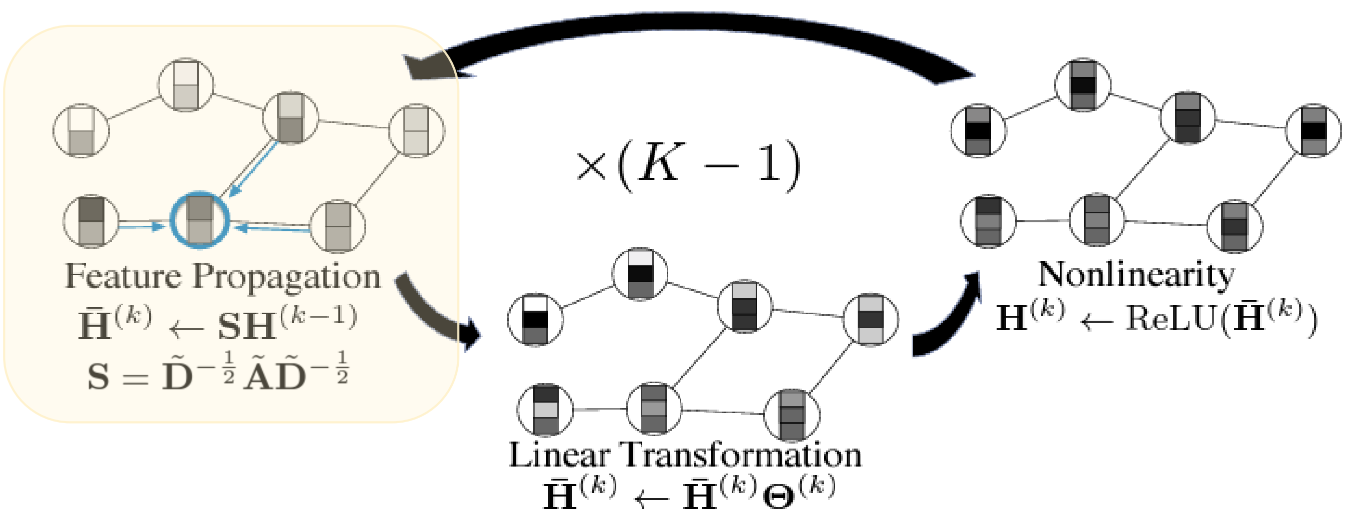


Graph Convolutional Networks(GCN)

❖ Graph Convolutional Network (GCN)

- Feature propagation
  - ✓ GCN layer 에서의 각 노드 v_i의 feature h_i는 v_i의 이웃 노드에 대한 average aggregation 에 해당
  - ✓ 이웃 노드들에 의한 smoothing 된 hidden representation 을 얻게 됨

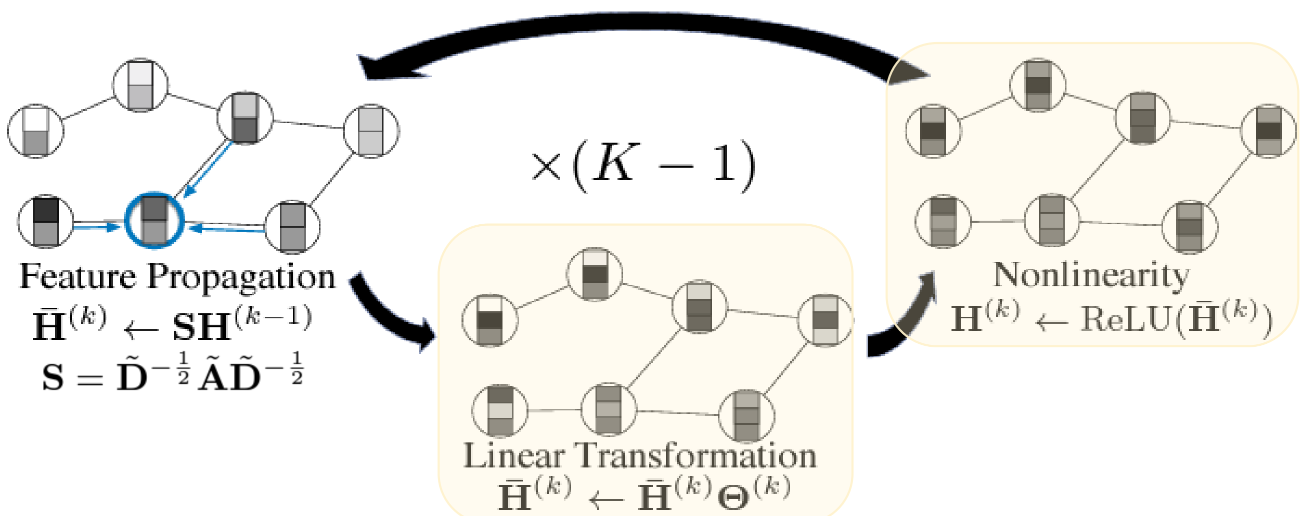
$$\bar{\mathbf{h}}_i^{(k)} \leftarrow \frac{1}{d_i+1}\mathbf{h}_i^{(k-1)} + \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{(d_i+1)(d_j+1)}}\mathbf{h}_j^{(k-1)}$$



Graph Convolutional Networks(GCN)

❖ Graph Convolutional Network (GCN)

- Linear transformation
  - ✓ 일반적인 MLP와 동일하게 가중치 파라미터 Θ를 곱해 줌
- Nonlinear transition
  - ✓ ReLU와 같은 활성화 함수를 적용하여 *k*-th, GCN layer의 feature representation $H^{(k)}$를 구함



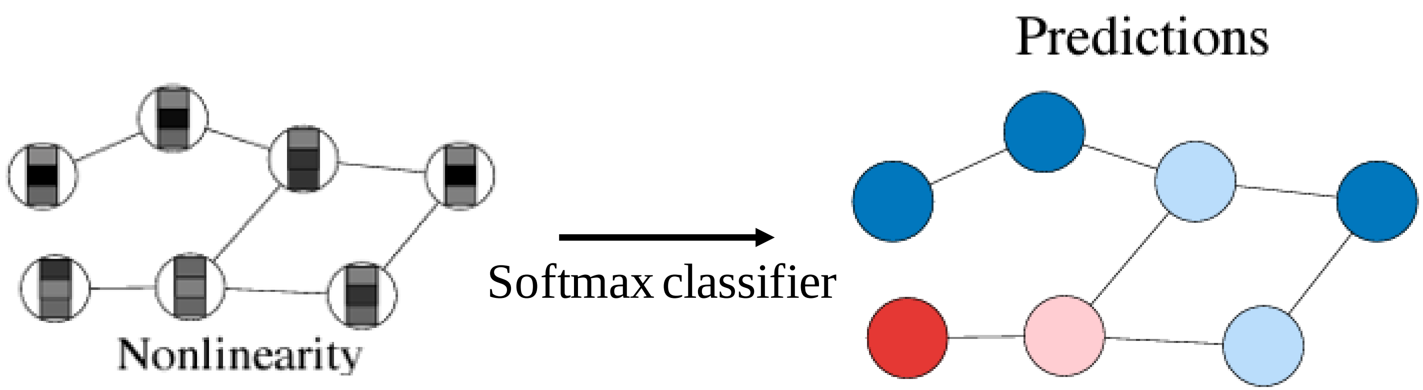
DMQA

Graph Convolutional Networks(GCN)

❖ Graph Convolutional Network (GCN)

- Classifier
  - ✓ 마지막 GCN layer에 softmax 함수를 적용

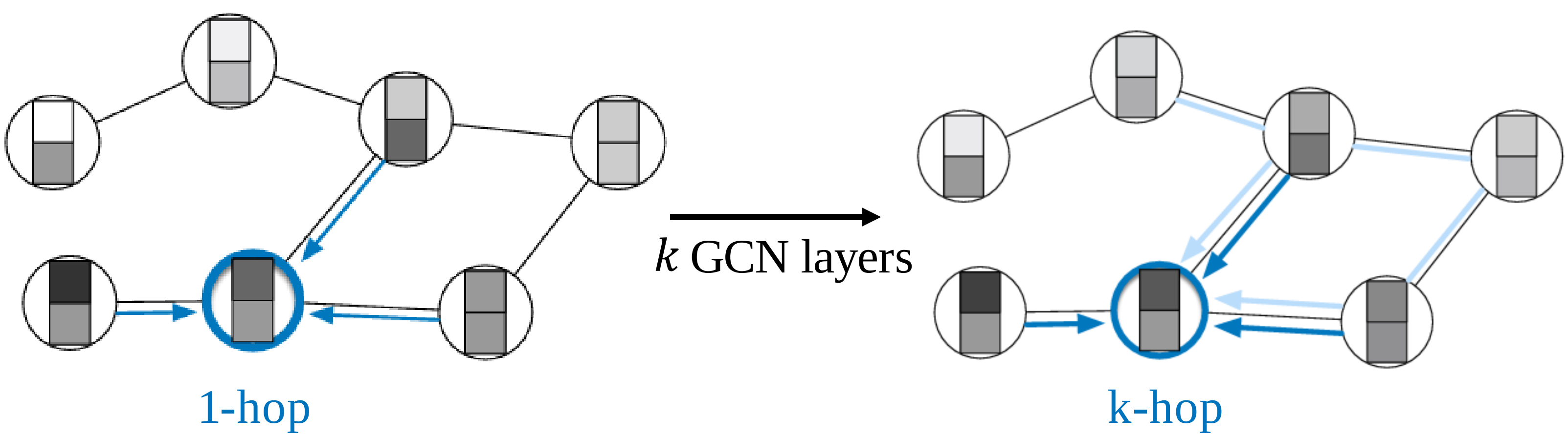
$$\hat{\mathbf{Y}}_{\mathrm{GCN}} = \mathrm{softmax}\left(\mathbf{S}\mathbf{H}^{(K-1)}\mathbf{\Theta}^{(K)}\right)$$



Simplifying Graph Convolutional Networks(SGC)

❖ Simple Graph Convolution

- 각 GCN layer는 1-hop 또는 1-order의 이웃 노드 정보를 averaging
- 따라서, *k*-th layer를 거칠 경우, *k*-hop 의 이웃 노드 정보가 반영됨



DMQA

Simplifying Graph Convolutional Networks(SGC)

❖ Simple Graph Convolution

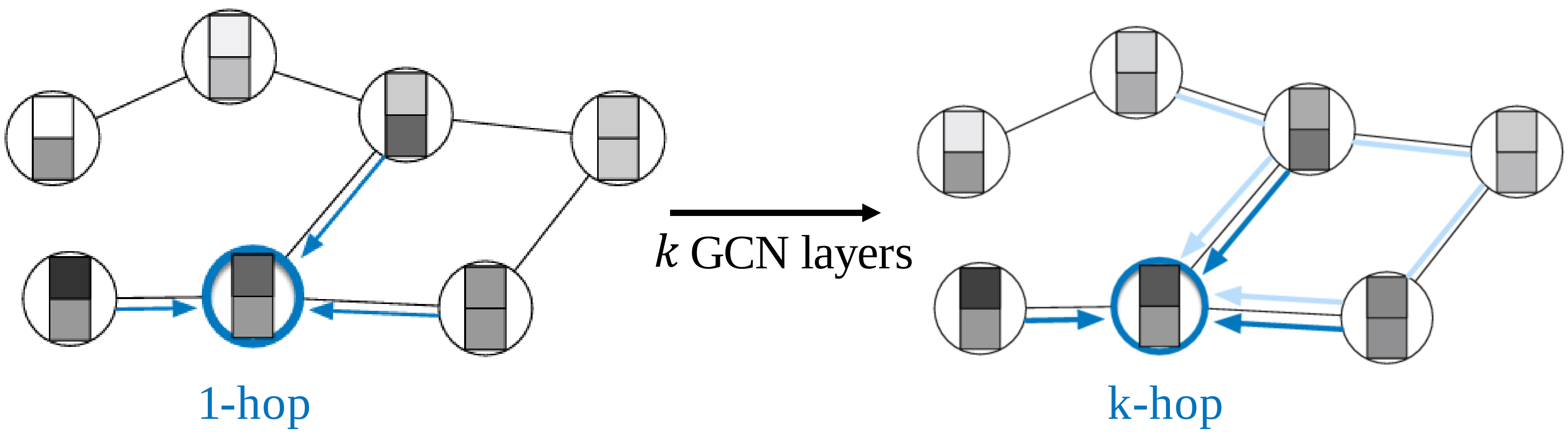
- Linearization
  - ✓ SGC 에서는 layer 간 nonlinearity 보다는 local averaging 이 더 중요할 것이라 가정
  - ✓ 활성화 함수에 해당하는 nonlinear transition 을 제거함

$$\boxed{\mathbf{\Theta} = \mathbf{\Theta}^{(1)}\mathbf{\Theta}^{(2)}\ldots\mathbf{\Theta}^{(K)}}$$

$$\mathbf{H}^{(0)} = \mathbf{X}.$$

$$\hat{\mathbf{Y}} = \mathrm{softmax}\left(\mathbf{S}\ldots\mathbf{S}\mathbf{S}\mathbf{X}\mathbf{\Theta}^{(1)}\mathbf{\Theta}^{(2)}\ldots\mathbf{\Theta}^{(K)}\right)$$

$$\boxed{\hat{\mathbf{Y}}_{\mathrm{SGC}} = \mathrm{softmax}\left(\mathbf{S}^K\mathbf{X}\mathbf{\Theta}\right)}$$



DMQA

Simplifying Graph Convolutional Networks(SGC)

❖ Simple Graph Convolution

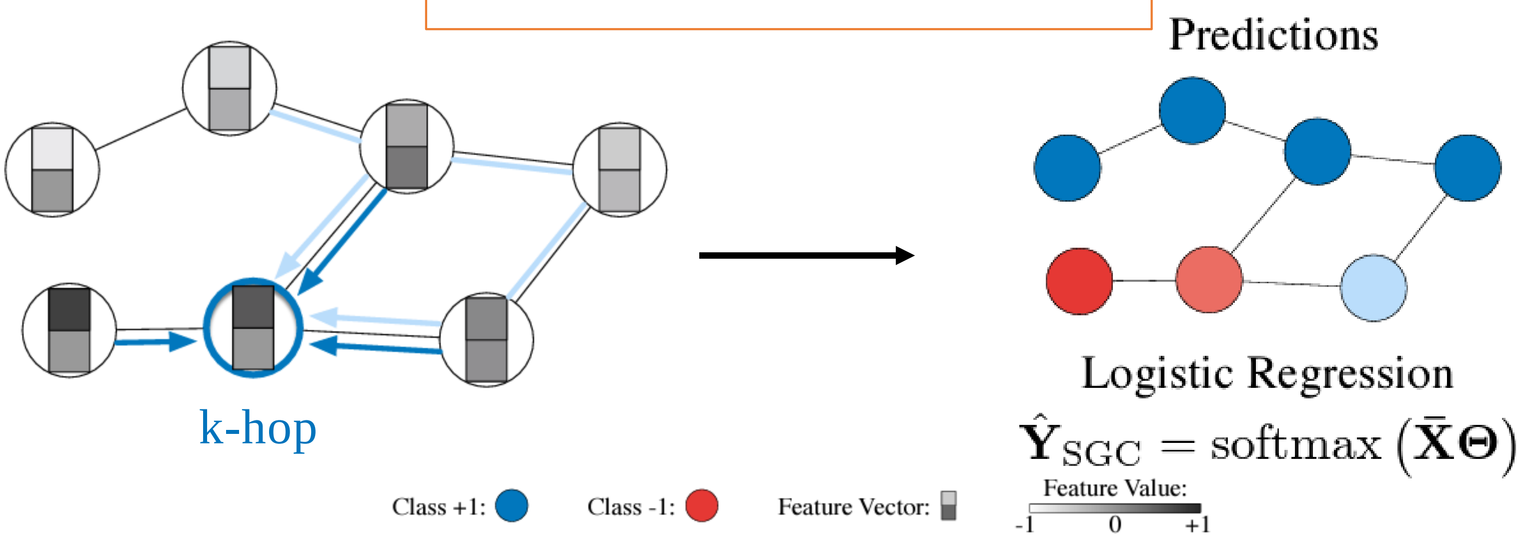
- Logistic regression
  - ✓ SGC 는 parameter free 의 local averaging 이 이루어짐
  - ✓ 이후, Multi class Logistic Regression 과 동일한 구조를 가짐
  - ✓ $\bar{X} = s^K X$ 는 weight 가 없기 때문에 feature pre-processing 단계와 동일하게 볼 수 있음

$$\boxed{\boldsymbol{\Theta} = \boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(2)} \dots \boldsymbol{\Theta}^{(K)}}$$

$$\mathbf{H}^{(0)} = \mathbf{X}$$

$$\hat{\mathbf{Y}} = \mathrm{softmax}\left(\mathbf{S} \dots \mathbf{S}\mathbf{S}\mathbf{X}\boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(2)} \dots \boldsymbol{\Theta}^{(K)}\right)$$

$$\boxed{\hat{\mathbf{Y}}_{\mathrm{SGC}} = \mathrm{softmax}\left(\mathbf{S}^K\mathbf{X}\boldsymbol{\Theta}\right)}$$



DMQA

Experiments

❖ Citation networks & Social networks

- Dataset
  - ✓ Citation network datasets: Cora, Citeseer, Pubmed
  - ✓ Community structure datasets: Reddit

Dataset	# Nodes	# Edges	Train/Dev/Test Nodes
Cora	2,708	5,429	140/500/1,000
Citeseer	3,327	4,732	120/500/1,000
Pubmed	19,717	44,338	60/500/1,000
Reddit	233K	11.6M	152K/24K/55K

Dataset

Experiments

❖ Citation networks & Social networks

- Results - Performance
  - ✓ Citation network 에서 GCN(Kipf, T. N. and Welling, M, 2017) 보다 1% 정도 우수, 더 적은 파라미터로 인한 overfitting 영향이 줄어든 것을 원인으로 판단
  - ✓ Community structure datasets(Reddit)에서 SAGE-GCN & FastGCN(sampling-based GCN variants), DGI 보다 1% 가량 우수

	Cora	Citeseer	Pubmed
Our experiments:			
GCN	81.4 ± 0.4	70.9 ± 0.5	79.0 ± 0.4
GAT	83.3 ± 0.7	72.6 ± 0.6	78.5 ± 0.3
FastGCN	79.8 ± 0.3	68.8 ± 0.6	77.4 ± 0.3
GIN	77.6 ± 1.1	66.1 ± 0.9	77.0 ± 1.2
LNet	$80.2 \pm 3.0^{\dagger}$	67.3 ± 0.5	$78.3 \pm 0.6^{\dagger}$
AdaLNet	$81.9 \pm 1.9^{\dagger}$	$70.6 \pm 0.8^{\dagger}$	$77.8 \pm 0.7^{\dagger}$
DGI	82.5 ± 0.7	71.6 ± 0.7	78.4 ± 0.7
SGC	81.0 ± 0.0	71.9 ± 0.1	78.9 ± 0.0

Test accuracy(%) averaged over 10 runs on citation networks

Setting	Model	Test F1
Supervised	GaAN	96.4
	SAGE-mean	95.0
	SAGE-LSTM	95.4
	SAGE-GCN	93.0
	FastGCN	93.7
	GCN	**OOM**
Unsupervised	SAGE-mean	89.7
	SAGE-LSTM	90.7
	SAGE-GCN	90.8
	DGI	94.0
No Learning	Random-Init DGI	93.3
	SGC	94.9

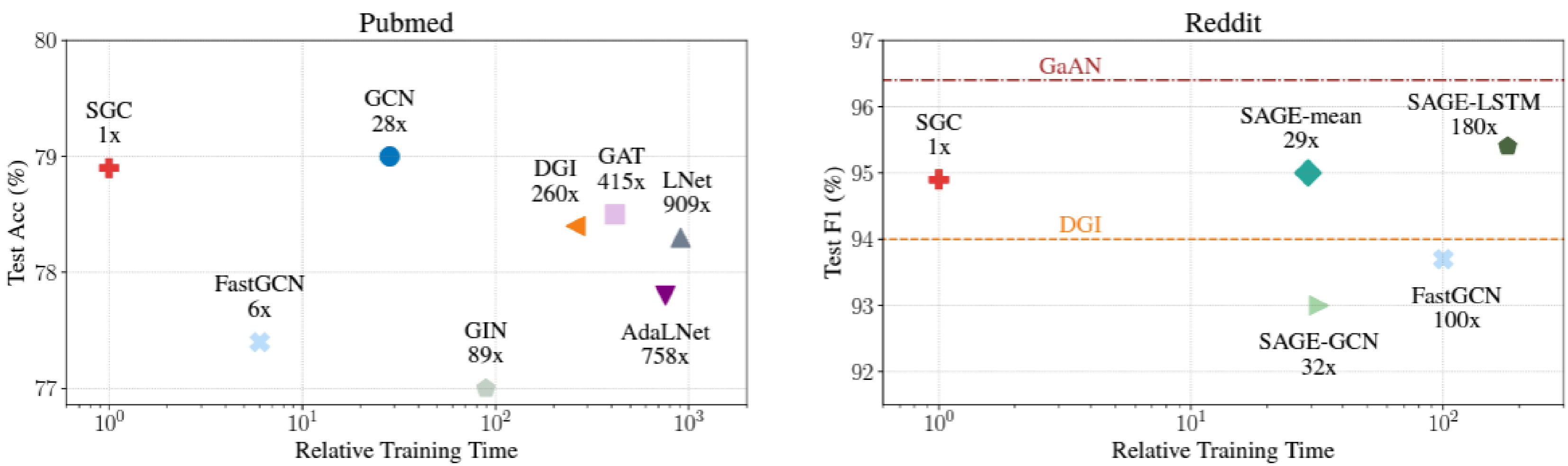
Test Micro F1 Score(%) averaged over 10 runs on Reddit
OOM: Out of memory

DMQA

Experiments

❖ Citation networks & Social networks

- Results - Efficiency
  - ✓ 비교 방법론 통틀어 압도적으로 빠른 훈련 시간을 보여줌
  - ✓ Reddit dataset 에서 GCN은 메모리 초과로 학습에 실패, 다른 방법론들 조차 neighborhood size 및 model size를 조절하여 학습해야하는 한계점 존재
  - ✓ 반면, SGC는 단일 weight matrix(Θ)만 학습하며 메모리 사용량 최소화



Performance over training time on Pubmed and Reddit

Experiments

❖ Downstream tasks

- 추가적으로 총 5가지의 downstream applications를 통해 모델 성능 검증
  - ✓ Text classification
  - ✓ Semi-supervised user geolocation
  - ✓ Relation extraction
  - ✓ Zero-shot image classification
  - ✓ Graph classification

Dataset	Model	Test Acc. ↑	Time (seconds) ↓
20NG	GCN	87.9 ± 0.2	1205.1 ± 144.5
	SGC	88.5 ± 0.1	19.06 ± 0.15
R8	GCN	97.0 ± 0.2	129.6 ± 9.9
	SGC	97.2 ± 0.1	1.90 ± 0.03
R52	GCN	93.8 ± 0.2	245.0 ± 13.0
	SGC	94.0 ± 0.2	3.01 ± 0.01
Ohsumed	GCN	68.2 ± 0.4	252.4 ± 14.7
	SGC	68.5 ± 0.3	3.02 ± 0.02
MR	GCN	76.3 ± 0.3	16.1 ± 0.4
	SGC	75.9 ± 0.3	4.00 ± 0.04

Test Accuracy (%) on text classification datasets

Dataset	Model	Acc.@161↑	Time ↓
GEOTEXT	GCN+H	60.6 ± 0.2	153.0s
	SGC	61.1 ± 0.1	5.6s
TWITTER-US	GCN+H	61.9 ± 0.2	9h 54m
	SGC	62.5 ± 0.1	4h 33m
TWITTER-WORLD	GCN+H	53.6 ± 0.2	2d 05h 17m
	SGC	54.1 ± 0.2	22h 53m

Test Accuracy (%) within 161 miles on semi-supervised user geolocation

DMQA

Experiments

❖ Downstream tasks

- 총 5가지의 downstream applications를 통해 추가 모델 평가 진행
  - ✓ Text classification
  - ✓ Semi-supervised user geolocation
  - ✓ Relation extraction
  - ✓ Zero-shot image classification
  - ✓ Graph classification

TACRED	Test Accuracy ↑
C-GCN (Zhang et al., 2018c)	66.4
C-GCN	66.4 ± 0.4
C-SGC	67.0 ± 0.4

Test Accuracy (%) on Relation Extraction

Model	# Param. ↓	2-hop Acc. ↑	3-hop Acc. ↑
Unseen categories only:			
EXEM 1-nns	-	27.0	7.1
ADGPM	-	26.6	6.3
GCNZ	-	19.8	4.1
GCNZ (ours)	$9.5M$	20.9 ± 0.2	4.3 ± 0.0
MLP-SGCZ (ours)	$4.3M$	21.2 ± 0.2	4.4 ± 0.1
Unseen categories & seen categories:			
ADGPM	-	10.3	2.9
GCNZ	-	9.7	2.2
GCNZ (ours)	$9.5M$	10.0 ± 0.2	2.4 ± 0.0
MLP-SGCZ (ours)	$4.3M$	10.5 ± 0.1	2.5 ± 0.0

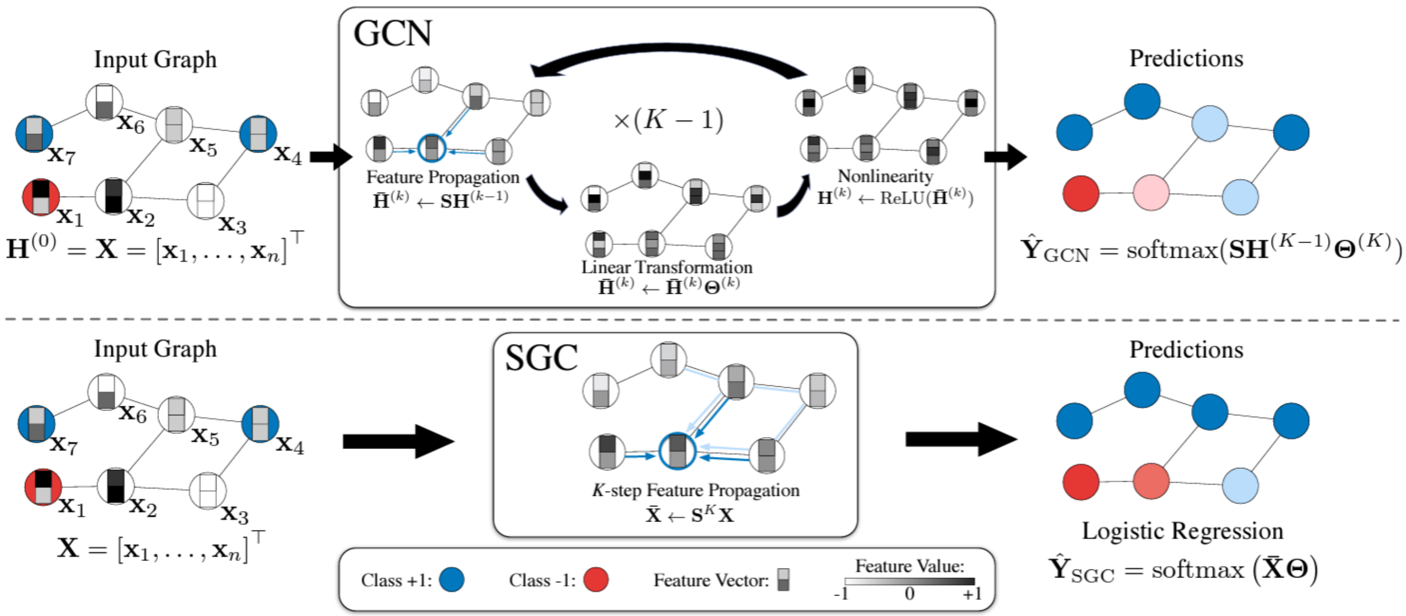
Top-1 accuracy (%) averaged over 10 runs in the 2hop and 3-hop setting of the zero-shot image task on ImageNet

DMQA

Conclusion

❖ Conclusion

- 기존 GCN(Kipf, T. N. and Welling, M, 2017)의 구조를 pre-processing step과 multi-class logistic regression과정으로 간소화함(Nonlinearity → linearity)
- 이를 통해, GCN의 expressive power는 nonlinear feature extraction이 아닌 반복적인 graph propagation(Message Passing Neural Network)으로부터 기인함



Schematic layout of a GCN v.s. a SGC

Conclusion

❖ Conclusion

- Node classification task 이외 다양한 그래프 기반 task 에서 SOTA 방법론 대비 훨씬 적은 훈련 시간으로 동등하거나 준하는 성능을 보여줌
- 또한, 본 리뷰에서 다루지 않은 Spectral Analysis 부분을 통해 spectral domain에서 low-pass-type filter(저주파 신호를 포착하여, 그래프 smoothing feature를 나타냄)를 다루며, SGC 를 수학적으로도 뒷받침함

❖ Reference

- Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019, May). Simplifying graph convolutional networks. In International conference on machine learning (pp. 6861-6871). PMLR.
- http://dsba.korea.ac.kr/seminar/?mod=document&uid=1329

Thank You