Large-Scale Learnable Graph Convolutional Networks

School of Industrial and Management Engineering, Korea University

Sae Rin Lim

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ Large-Scale Learnable Graph Convolutional Networks(2018, KDD)

- CNN을 그래프 데이터에 직접적으로(spatial domain) 적용한 연구는 GCN(2017, ICML)이 최초
- 이웃노드와 자신의 특징 벡터를 통합하여(aggregate) layer마다 업데이트 할 수 있음
- CNN 연산과정을 그래프 데이터에 맞추기 때문에 아래와 같은 단점이 있음
  1. 이웃노드의 수가 노드마다 다르기 때문에 같은 필터 가중치를 사용하지 못함
  2. Aggregation 과정에서 학습이 불가능한 함수(Sum)를 사용 $(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}})$
  3. 전체 그래프 데이터를 입력으로 사용하기 때문에 연산량과 메모리가 많이 필요$(\hat{A}, X_l)$

$$GCN\ layer\text{-}wise\ forward\text{-}propagation$$

$$H_{l+1} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H_l W_l\right)$$

Research Purpose

❖ Large-Scale Learnable Graph Convolutional Networks(2018, KDD)

- Aggregation과정에서 이웃노드의 가중치를 학습가능하도록 어탠션 매커니즘을 적용한 GAT(2018, ICML)이 등장
- 하지만 여전히 같은 필터 파라미터를 사용할 수 없음
- 어텐션 매커니즘을 도입하기 때문에 연산량이 더욱 증가

$$GAT\ layer\text{-}wise\ forward\text{-}propagation$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in Neig(i)} a_{ij}^{(l)} W h_j^{(l)}\right)$$

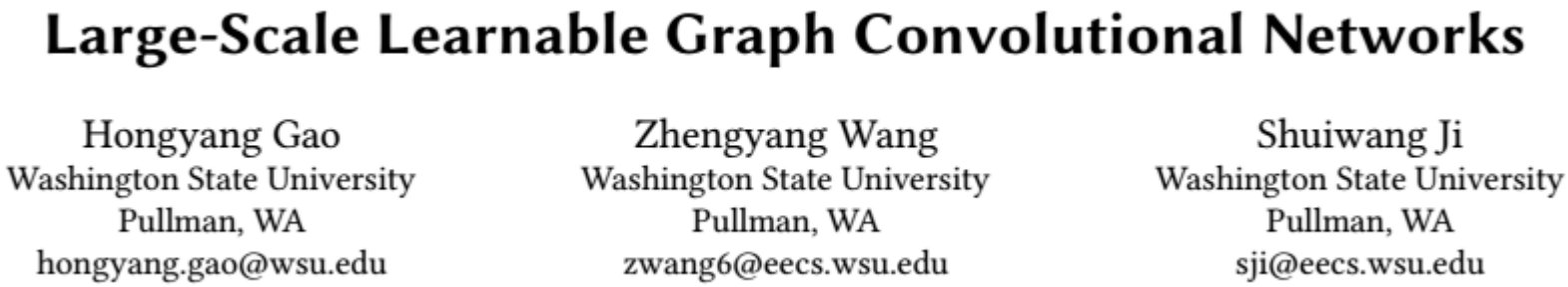
$$a_{i,j}^{(l)} = softmax\left(e_{i,j}^{(l)}\right),$$

$$e_{i,j}^{(l)} = a^{(l)}(Wh_i^{(l)}, Wh_i^{(j)})$$

Research Purpose

❖ Large-Scale Learnable Graph Convolutional Networks(2018, KDD)

- Washington State Univ.에서 연구하였으며 2022년 3월 4일 기준으로 341회 인용
- 본 연구에서는 CNN을 그래프 데이터에 맞게 변경하는 것이 아닌 그래프 데이터를 CNN에 맞게 grid 형식의 데이터로 변경하는 방법론을 제안
- 또한, sub-graph selection algorithm을 고안하여 large-scale의 그래프를 효율적으로 학습



Large-Scale Learnable Graph Convolutional Networks

Hongyang Gao
Washington State University
Pullman, WA
hongyang.gao@wsu.edu

Zhengyang Wang
Washington State University
Pullman, WA
zwang6@eecs.wsu.edu

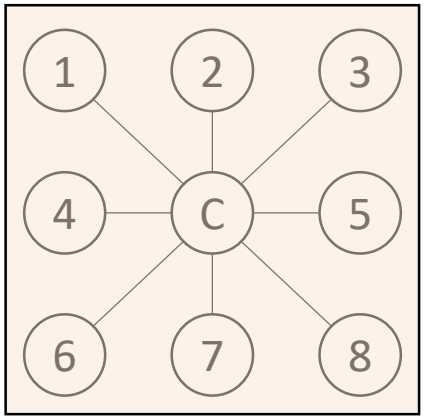
Shuiwang Ji
Washington State University
Pullman, WA
sji@eecs.wsu.edu

Large-Scale Learnable Graph Convolutional Networks(LGCN)

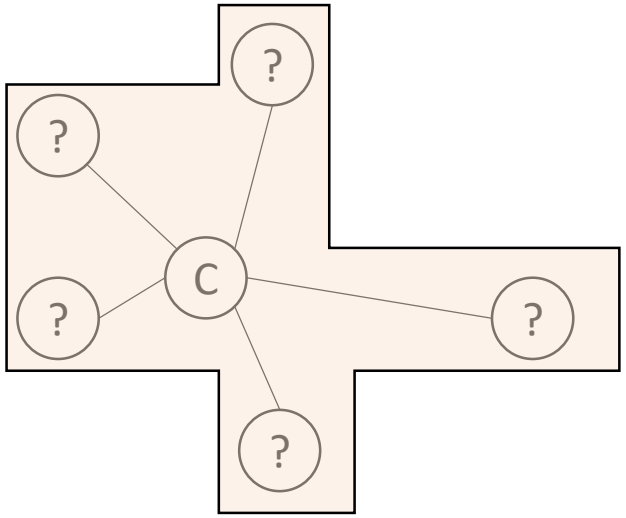
❖ Challenges of Applying Convolutional Operations on Graph Data

- 그래프 데이터를 grid 형식의 데이터로 변경하기 위해서는 두 가지 문제를 해결해야 함
  1. 이웃노드의 수가 같아야 한다
  2. 이웃노드들 간의 순서(order)가 있어야 한다

*이미지나 텍스트와 같은 grid data는 중앙노드를 기준으로 상대적 위치를 통해 ordering 가능



Grid data(2D) with regular filter



Graph data with irregular filter

DMQA

Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ Learnable Graph Convolutional Layers(LGCL)

- 저자들은 앞의 두 문제를 해결하기 위해서 LGCL을 제안(node selection + 1-D CNN)
- LGCL은 그래프 데이터를 grid 데이터로 바꾸는 함수 g와 일반적인 CNN 함수 c로 이루어짐
- k-largest node selection을 통해 고정된 개수의 이웃노드를 선택하고 순서를 정의할 수 있음
- 선택된 이웃노드의 정보를 통합하여 노드마다 aggregated vector를 생성
- Aggregated vector를 일반적인 1-D CNN에 입력하여 forward-propagation 진행

$$\text{L}GCL\ forward\text{-}propagation$$

$$\tilde{H}_l = g(H_l, A, k), \qquad H_{l+1} = c(\tilde{H}_l)$$

$$H_l\text{: hidden node feature vector of } l \text{ layer}$$
$$\tilde{H}_l\text{: grid transformed hidden node feature vector of } l \text{ layer}$$
$$g(\cdot)\text{: operation that performs the } k\text{-largest node selection}$$
$$c(\cdot)\text{: 1-}D\ CNN$$
$$A : Adjecency\ matrix$$
$$k\text{: hyper parameter of } k\text{-largest node selection}$$

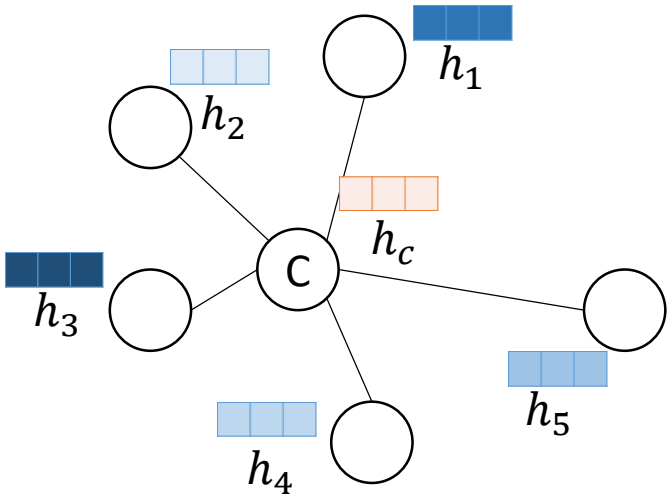
Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ LGCL : k-largest Node Selection $g(H_l, A, k)$

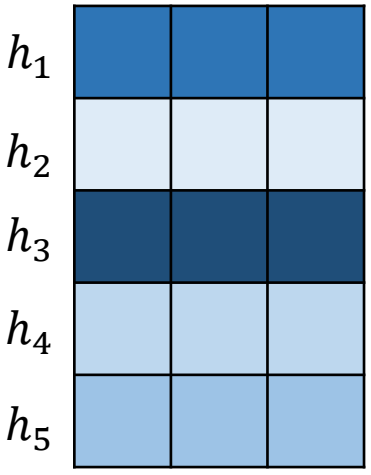
1. 중앙노드 C를 기준으로 이웃노드 n개의 특징 벡터를 모은 행렬 $M_C \in R^{n \times C(featrue\ dim)}$ 생성
2. 행렬 M_C에서 큰 값을 가지는 k개의 row를 선택하고 자신의 특징벡터를 첫 번째 row에 추가하여 새로운 행렬 $\tilde{M}_C \in R^{n \times (k+1)}$ 생성

   *이 때, 이웃노드의 수가 k개 보다 작을 경우 zero vector로 채움

3. $\tilde{M}_C \in R^{n \times (k+1)}$에 1-D CNN을 aggregation function으로 사용하여 노드 정보를 통합



노드 c에 대한
이웃노드 특징 행렬



$M_C \in n \times C$

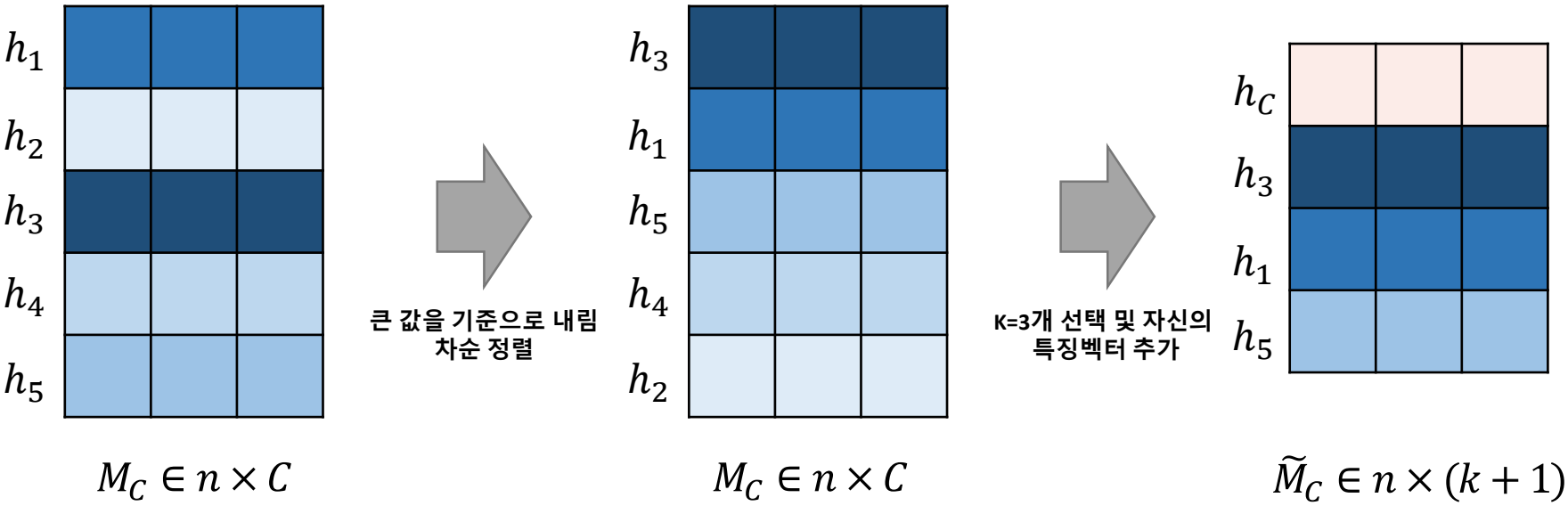
Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ LGCL : k-largest Node Selection $g(H_l, A, k)$

1. 중앙노드 C를 기준으로 이웃노드 n개의 특징 벡터를 모은 행렬 $M_C \in R^{n \times C(featrue\ dim)}$ 생성
2. 행렬 M_C에서 큰 값을 가지는 k개의 row를 선택하고 자신의 특징벡터를 첫 번째 row에 추가하여 새로운 행렬 $\tilde{M}_C \in R^{n \times (k+1)}$ 생성

   *이 때, 이웃노드의 수가 k개 보다 작을 경우 zero vector로 채움

3. $\tilde{M}_C \in R^{n \times (k+1)}$에 1-D CNN을 aggregation function으로 사용하여 노드 정보를 통합



DMQA

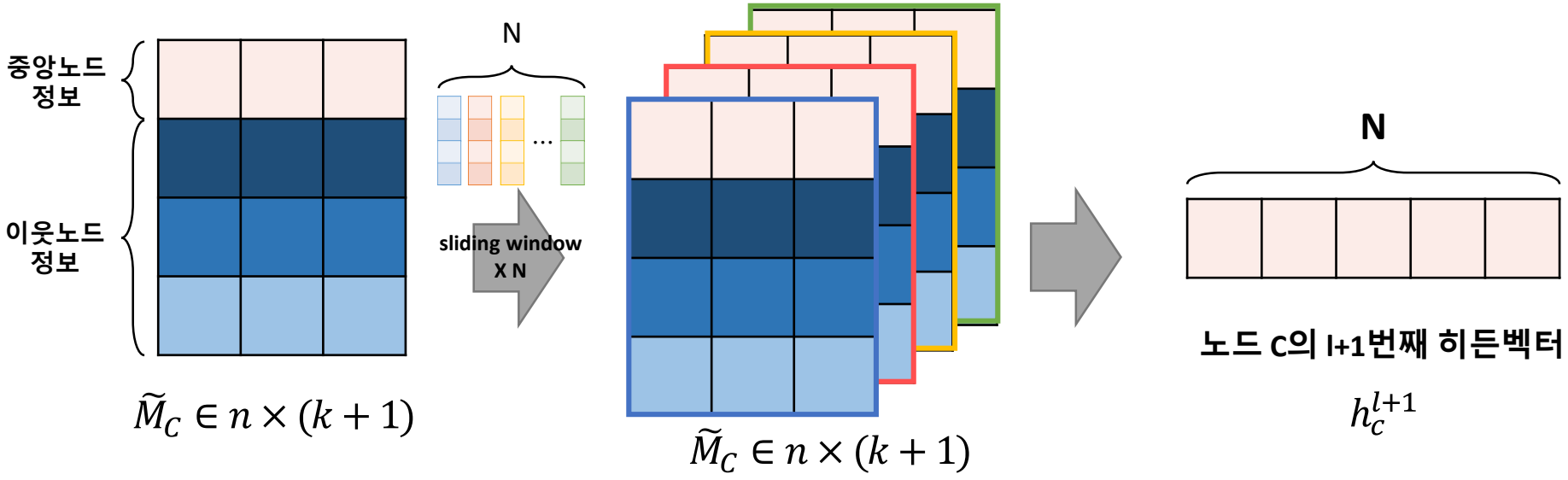
Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ LGCL : 1-D CNN as a aggregation function

1. 중앙노드 C를 기준으로 이웃노드 n개의 특징 벡터를 모은 행렬 $M_C \in R^{n \times C(featrue\ dim)}$ 생성
2. 행렬 M_C에서 큰 값을 가지는 k개의 row를 선택하고 자신의 특징벡터를 첫 번째 row에 추가하여 새로운 행렬 $\tilde{M}_C \in R^{n \times (k+1)}$ 생성

   *이 때, 이웃노드의 수가 k개 보다 작을 경우 zero vector로 채움

3. $\tilde{M}_C \in R^{n \times (k+1)}$에 1-D CNN을 aggregation function으로 사용하여 노드 정보를 통합

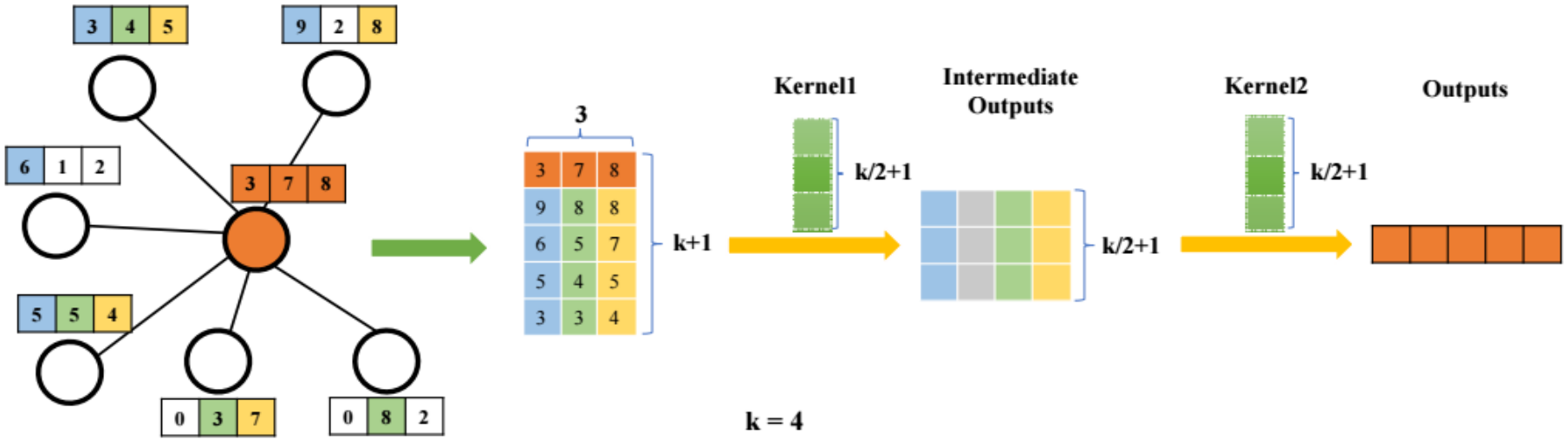


DMQA

Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ LGCL

- K-largest Node Selection 과정을 모든 노드에 대해 진행하여 $H_l \rightarrow \tilde{H} \in R^{N\times(k+1)\times C}$ 로 변형
- N, k+1, C를 batch size, input length, number of channels 로 생각하면 k-largest node selection operator $g(H_l, A, k)$가 그래프 데이터를 1-D grid 데이터로 변형했다고 볼 수 있음
- CNN을 사용하여 이웃노드와 자신의 특징벡터 가중치를 학습 가능한 aggregation function 구현

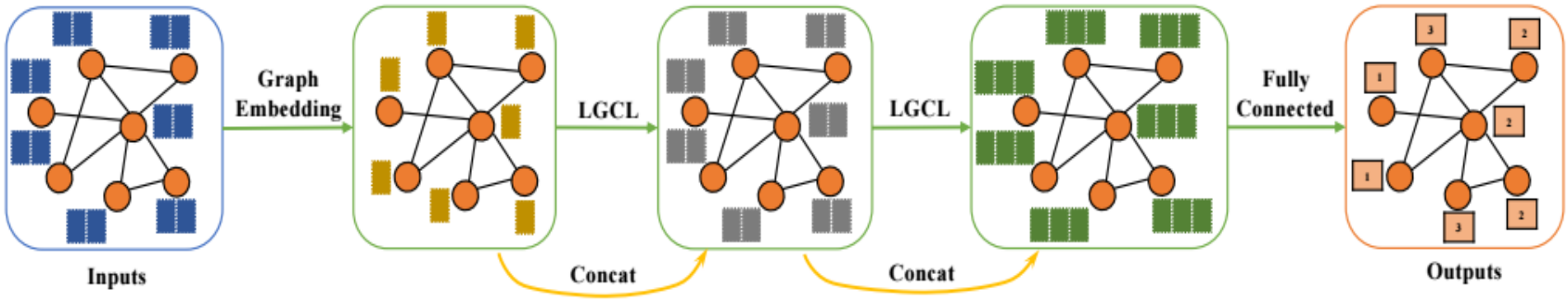


DMQA

Large-Scale Learnable Graph Convolutional Networks(LGCN)

❖ Learnable Graph Convolutional Network(LGCN)

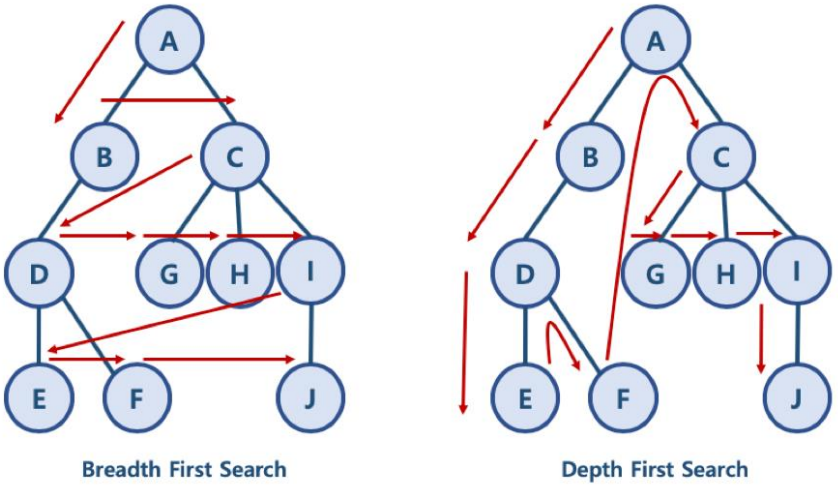
- 본 연구에서는 LGCL을 기반으로 아래와 같은 구조를 가지는 모델을 제안
- Graph Embedding으로는 간단한 Linear Layer를 사용
- GCN(2017, ICML)과는 다르게 모델을 깊게 쌓아도 성능의 하락이 없어 깊은 모델 구축 가능
- 저자들을 모델 구축에 있어서 k와 LGCL의 개수가 중요한 하이퍼 파라미터라고 서술하며 k는 평균 이웃노드의 개수가 좋은 성능을 낸다고 서술



Large-Scale Learnable Graph Convolutional Networks(LGCN)

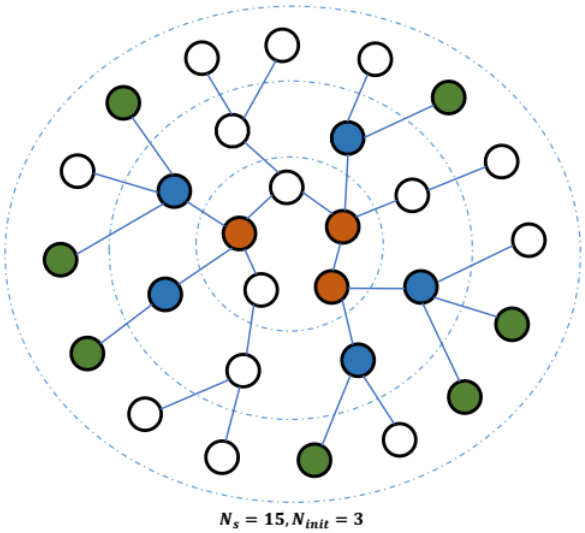
❖ Sub-Graph Training on Large-Scale Data

- Image Segmentation에서 큰 데이터셋을 학습할 때, 이미지를 random crop한 patch단위로 학습한 것에서 영감을 받아 sub-graph selection algorithm을 고안
- 초기 노드를 N_{init}개 만큼 선택한 뒤, 원하는 sub-graph의 크기인 N_s가 될 때까지 너비 우선 탐색(BFS, Breadth-First Search) 진행하여 sub-graph 생성
- 이 방법을 통해서 큰 데이터셋에서도 효율적으로 학습이 가능



https://www.fun-coding.org/00_Images/BFSDFS.png

그래프 탐색 기법 : 넓이 우선 탐색, 깊이 우선 탐색



Example of Sub-graph selection

DMQA

Experiments

❖ Dataset

- Citation Network (Citeseer, Core, Pubmed) : 각 노드는 문서, Edge는 Citation Link
- PPI : protein-protein interaction dataset

Dataset	#Nodes	#Features	#Classes	#Training Nodes	#Validation Nodes	#Test Nodes	Degree	Setting
Cora	2708	1433	7	140	500	1000	4	Transductive
Citeseer	3327	3703	6	120	500	1000	5	Transductive
Pubmed	19717	500	3	60	500	1000	6	Transductive
PPI	56944	50	121	44906 (20 graphs)	6514 (2 graphs)	5524 (2 graphs)	31	Inductive

Experiments

❖ Evaluation

- 4가지 graph-based benchmark task에 대해 성능 평가 수행
  - ➢ Transductive learning 측면에서는 Cora, Citeseer, Pubmed dataset에 대하여 성능 평가
  - ➢ Inductive learning 측면에서는 protein-protein interaction (PPI) dataset에 대하여 성능 평가

Models	Cora	Citeseer	Pubmed
DeepWalk [21]	67.2%	43.2%	65.3%
Planetoid [30]	75.7%	64.7%	77.2%
Chebyshev [4]	81.2%	69.8%	74.4%
GCN [15]	81.5%	70.3%	79.0%
$LGCN_{sub}$(Ours)	**83.3 ± 0.5%**	**73.0 ± 0.6%**	**79.5 ± 0.2%**

Transductive learning

Models	PPI
GraphSAGE-GCN [9]	0.500
GraphSAGE-mean [9]	0.598
GraphSAGE-pool [9]	0.600
GraphSAGE-LSTM [9]	0.612
$LGCN_{sub}$(Ours)	**0.772 ± 0.002**

Inductive learning

Experiments

❖ Ablation Study

- LGCN의 성능향상이 모델을 깊게 쌓은 것이 아닌 LGCL의 효과라는 것을 증명하기 위해서 모델 구조는 동일하게 가져가고 LGCN을 GCN layer로 변경한 ablation study 진행
- 해당 실험을 통해서 단순히 모델의 깊이에 의한 성능향상이 아닌 LGCL에 의한 것임을 보임

Models	Cora	Citeseer	Pubmed
$LGCN_{sub}$-GCN	82.2 ± 0.5%	71.1 ± 0.5%	79.0 ± 0.2%
$LGCN_{sub}$(Ours)	**83.3 ± 0.5%**	**73.0 ± 0.6%**	**79.5 ± 0.2%**

Experiments

❖ Ablation Study

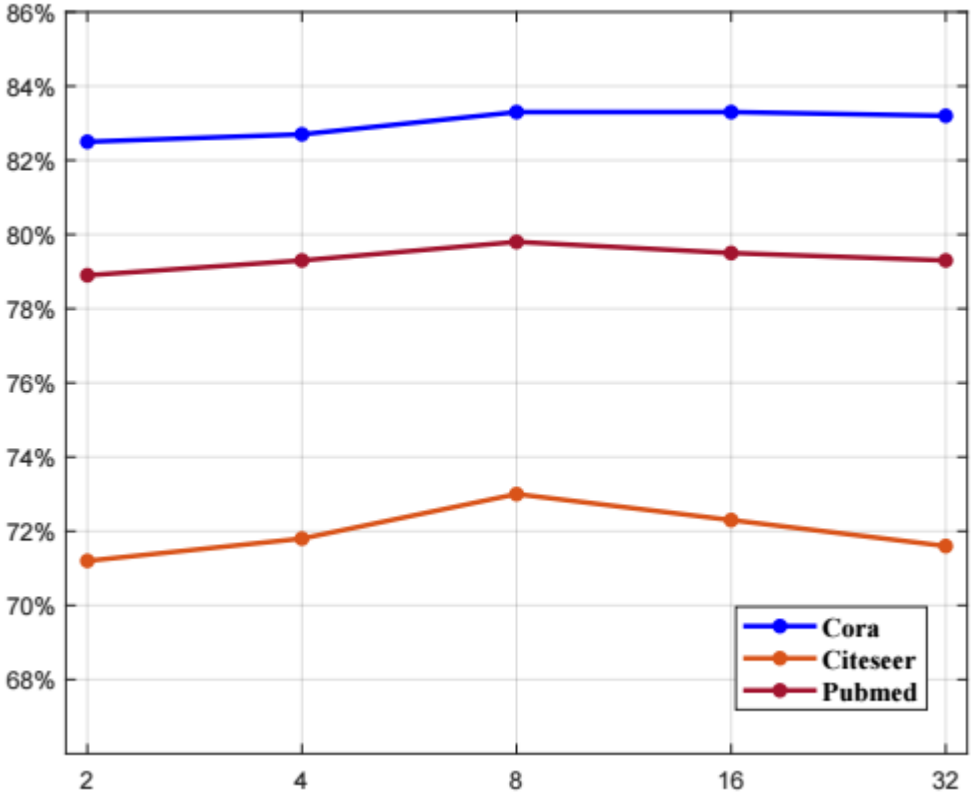
- Sub-graph selection algorithm을 통해 큰 데이터셋에서 효율적으로 학습할 수 있지만 모델이 전체 그래프의 구조를 볼 수 없다는 단점을 가짐
- 이 단점의 악영향을 알아보기 위해서 전체 그래프를 학습한 모델 성능과 Sub-graph selection algorithm을 통해 학습한 모델 성능을 비교
- 큰 성능의 하락이 없이 효율적으로 학습이 가능하다는 것을 보임

		Cora	Citeseer	Pubmed
GCN	# Nodes	2708	3327	19717
	Accuracy	81.5%	70.3%	79.0%
	Time	**7s**	4s	38s
LGCN$_{whole}$	# Nodes	2708	3327	19717
	Accuracy	83.8 ± 0.5%	73.0 ± 0.6%	79.5 ± 0.2%
	Time	58s	30s	1080s
LGCN$_{sub}$	# Nodes	644	442	354
	Accuracy	83.3 ± 0.5%	73.0 ± 0.6%	79.5 ± 0.2%
	Time	14s	**3.6s**	**2.6s**

Experiments

❖ Ablation Study

- 본 연구에서 중요한 하이퍼 파라미터인 k에 따라 성능의 변화를 관찰
- 이웃노드 개수의 평균이 일반적으로 가장 좋은 성능을 나타냄을 확인



Conclusion

❖ conclusion

- GCN 이후 많은 연구들이 그래프 데이터에 CNN 오퍼레이션을 맞추는 연구로 수행됨
- 반면에 본 연구에서는 그래프 데이터를 적절한 방법으로 grid 데이터로 변환하여 기존의 CNN을 그대로 적용할 수 있게끔 함으로써 CNN의 장점을 모두 가질 수 있음
- 방법론 자체가 간단하면서 문제를 해결하기 위해 관점을 바꾸는 것이 큰 도움이 된다는 것을 보여줌
- 하지만 왜 k-largest selection을 한 것인지에 대한 명확한 이유와 큰 값으로 이웃 노드의 순서를 정하는 것이 reasonable하다는 근거가 없어 아쉬웠음

Reference

1. Gao, H., Wang, Z., & Ji, S. (2018, July). Large-scale learnable graph convolutional networks. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1416-1424).


Thank You