
Inductive Representation Learning on Large Graphs (GraphSage)

School of Industrial and Management Engineering, Korea University

Jae Hoon Kim

Contents

- ❖ Research Purpose
- ❖ GraphSage
- ❖ Experiments
- ❖ Conclusion

Research Purpose

- ❖ Inductive Representation Learning on Large Graphs (NeurIPS 2017)
 - Stanford Univ.에서 연구하였으며 2022년 1월 23일 기준으로 5311회 인용됨

Inductive Representation Learning on Large Graphs

William L. Hamilton* Rex Ying* Jure Leskovec
wleif@stanford.edu rexying@stanford.edu jure@cs.stanford.edu

Department of Computer Science
Stanford University
Stanford, CA, 94305

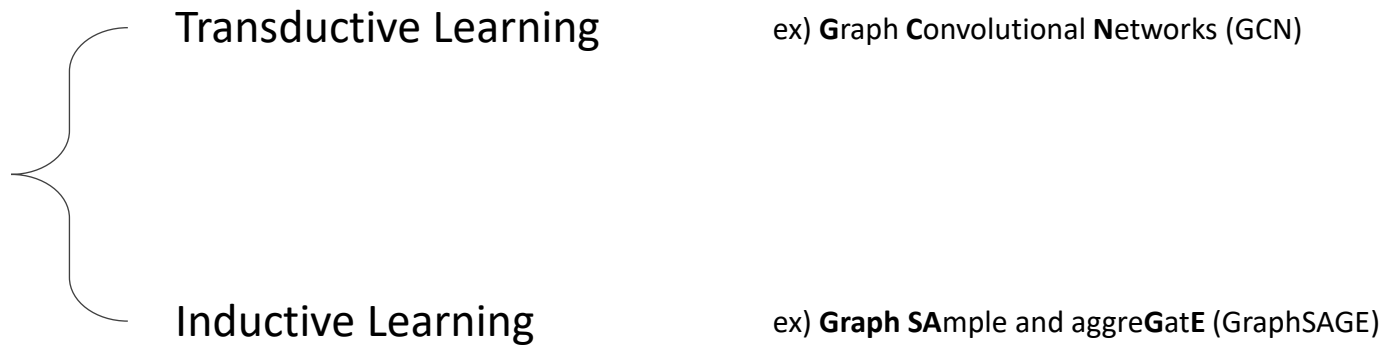
Abstract

Low-dimensional embeddings of nodes in large graphs have proved extremely useful in a variety of prediction tasks, from content recommendation to identifying protein functions. However, most existing approaches require that all nodes in the graph are present during training of the embeddings; these previous approaches are inherently *transductive* and do not naturally generalize to unseen nodes. Here we present GraphSAGE, a general *inductive* framework that leverages node feature information (e.g., text attributes) to efficiently generate node embeddings for previously unseen data. Instead of training individual embeddings for each node, we learn a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. Our algorithm outperforms strong baselines on three inductive node-classification benchmarks: we classify the category of unseen nodes in evolving information graphs based on citation and Reddit post data, and we show that our algorithm generalizes to completely unseen graphs using a multi-graph dataset of protein-protein interactions.

Research Purpose

❖ Inductive Representation Learning on Large Graphs (NeurIPS 2017)

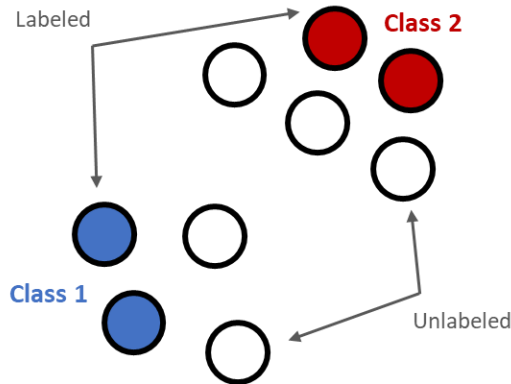
- 기존 연구는 고정된 그래프 데이터를 잘 표현하는 방법을 연구함
- 해당 연구는 학습한 모델로 새롭게 생성된 노드에 대해서도 잘 표현하는 방법을 연구함
- 전자는 transductive learning, 후자는 inductive learning에 속함 (좀 더 명확한 비교는 뒷장에서...)



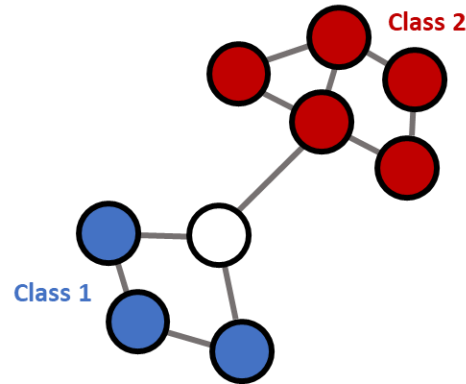
Research Purpose

❖ Transductive Learning vs. Inductive Learning

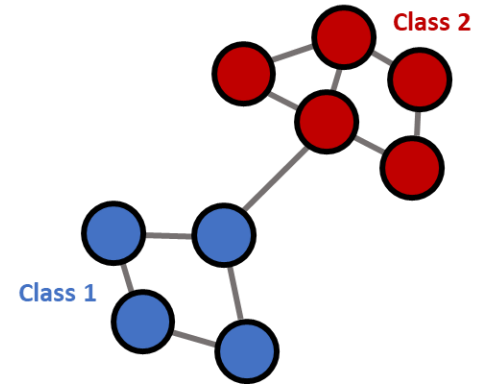
- Transductive learning의 경우 학습 시 훈련 및 검증 데이터의 구분이 없으며 둘을 합쳐서 사용함
 - ✓ GCN의 논문 제목이 **Semi-Supervised** Classification with Graph Convolutional Networks인 이유
- 별도의 모델을 생성하지 않음 (관찰한 데이터에 맞추어 분류 혹은 예측)
 - ✓ 따라서 새로운 데이터가 추가되었을 때 이에 맞춰 전체 데이터에 대한 재연산이 필요함



< Given Data >



Step 1



Step 2

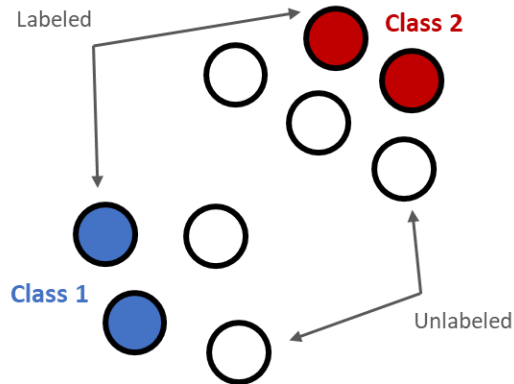
< Label Propagation >

** unlabeled data가 또 다른 unlabeled data의 레이블링에 관여함

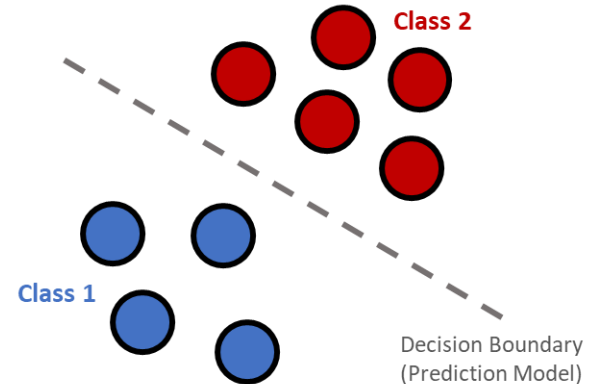
Research Purpose

❖ Transductive Learning vs. Inductive Learning

- Inductive learning의 경우 학습 시 훈련 및 검증 데이터를 구분하여 사용함
 - ✓ 일반적인 Supervised Learning 방식과 동일함
- 훈련 데이터를 통해 새로운 데이터를 분류 혹은 예측하기 위한 모델을 생성함
 - ✓ 새로운 데이터가 추가되었을 때 생성된 모델을 사용하여 추론이 가능함



< Given Data >



< Logistic Regression >

**** 오직 labeled data만이 unlabeled data의 레이블링에 관여함**

Research Purpose

❖ Inductive Representation Learning on Large Graphs (NeurIPS 2017)

- Transductive 모델은 학습에 필요한 연결 정보를 **자신과 연결된 모든 노드**로 정의함 (full batch)
- GraphSAGE는 특정 노드의 주변에서 **샘플링한 이웃 노드**를 연결정보로 사용함 (mini batch)
- **규모가 큰 그래프 데이터를 처리할 때에는** mini batch 방식이 더 효율적인 학습이 가능함
- Transductive 모델은 특정 노드를 표현할 때 **그래프 전체에 대한 연결 정보**를 사용하여 표현함
- GraphSAGE는 특정 노드에 **이웃한 노드의 정보만을 통합**하여 표현함
- **새로운 노드가 생성될 경우** Transductive 모델은 기존 모델 사용이 불가하나 GraphSAGE는 가능함

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

< Node feature matrix (H) of Transductive GCN >

→ **그래프 전체의 연결정보를 나타냄 (Adjacency Matrix)**

- 새로운 노드가 추가될 경우 adjacency matrix의 크기가 바뀜
- 따라서 전체 데이터에 대한 재연산이 필요함 (기존 모델 사용 불가)

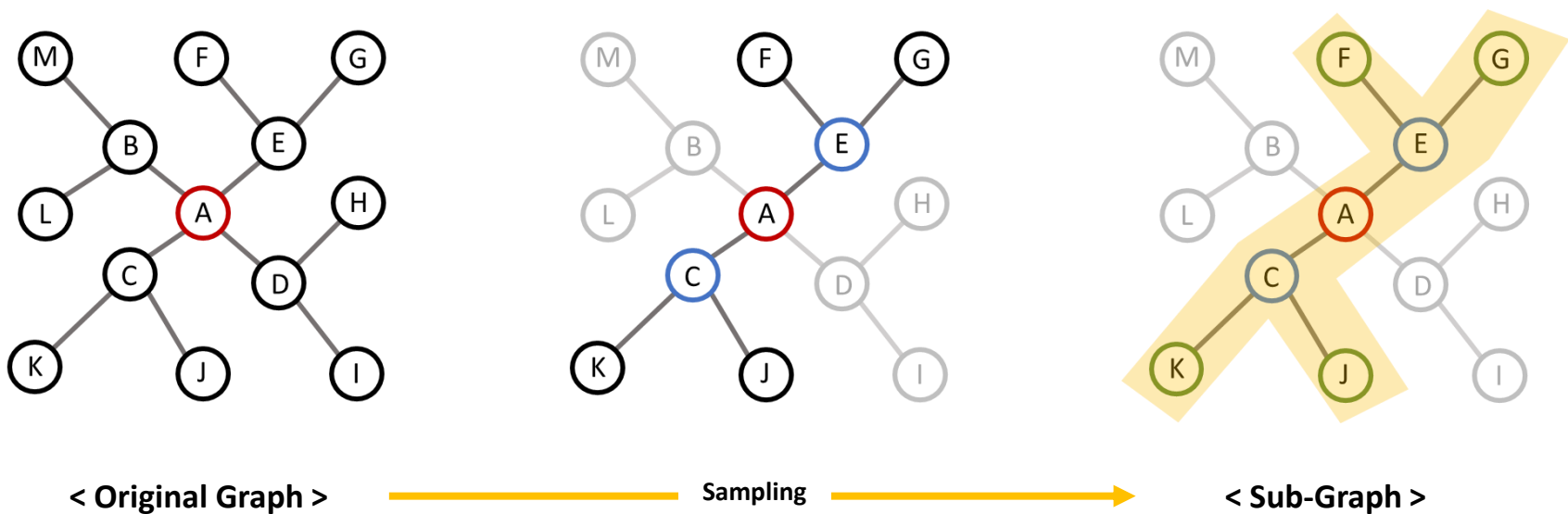
0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

→ 새로운 노드를 추가

Graph Sample and aggregate (GraphSAGE)

❖ Sampling Neighbors

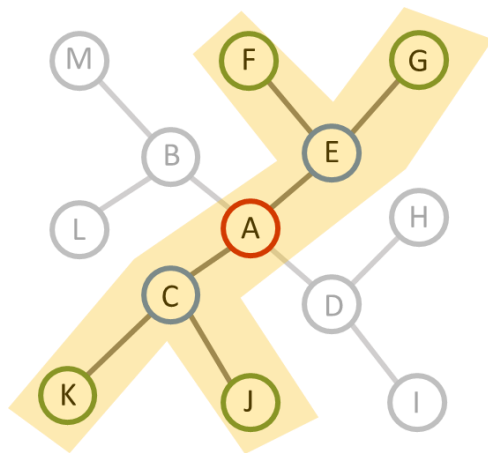
- 특정 노드(A)의 주변에서 hop k의 깊이로 특정 개수의 이웃 노드를 샘플링함
- 각 깊이에서 2개의 노드를 샘플링한다고 할 때 hop 1에서는 C와 E 노드를 샘플링함
- 이후 hop 2에서는 C와 E 각각에서 2개의 노드를 샘플링하여 sub-graph를 구축함



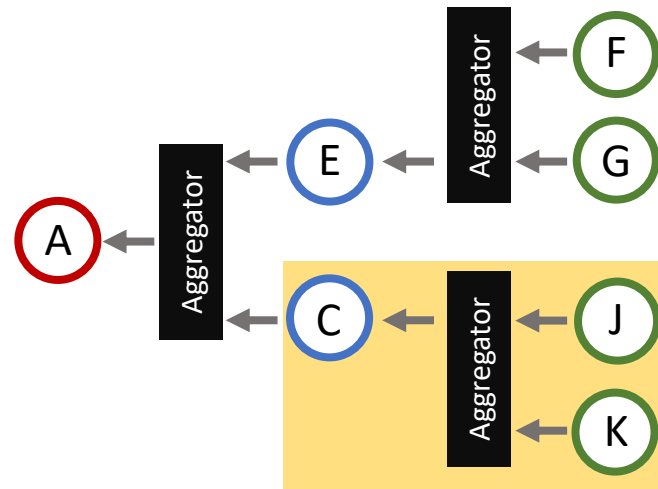
Graph Sample and aggreGatE (GraphSAGE)

❖ Aggregate Neighbors

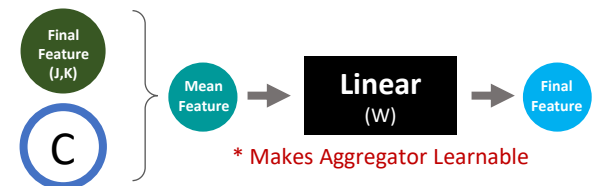
- Aggregation시 노드 순서와 무관한 연산이 이루어져야 함
- Aggregation function은 종류가 Mean, LSTM, Pooling(max)이 있으며 학습 가능한 네트워크로 구성 됨
 - ✓ LSTM aggregation의 경우 입력 노드의 순서를 임의로 섞음



< Sub-Graph >



ex) Mean Aggregation



Graph Sample and aggreGatE (GraphSAGE)

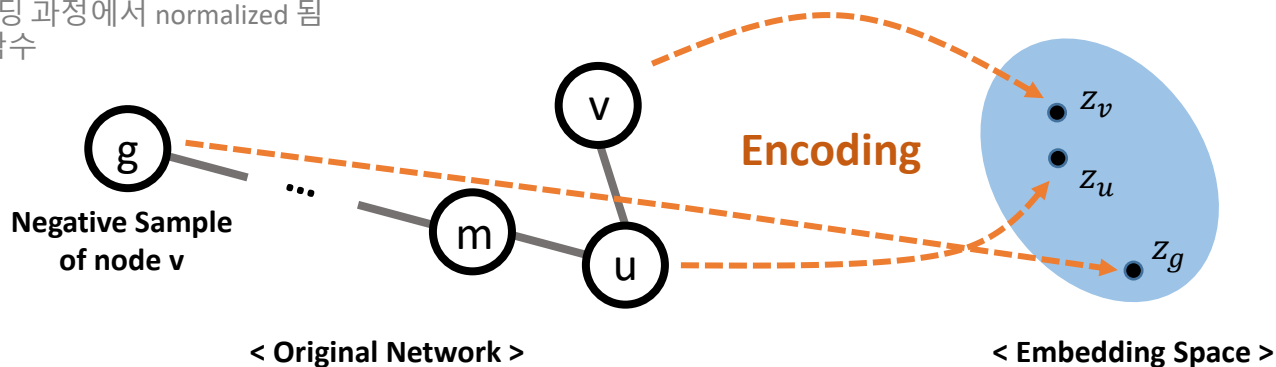
❖ Objective function

- GraphSAGE의 목적함수는 비지도학습 혹은 지도학습(ex. Cross Entropy) 방식으로 구성될 수 있음
- **비지도학습의 경우** 특정 노드와 이에 이웃한 노드의 각 특징 벡터가 유사해지도록 학습함
- 또한 특정 노드와 이웃하지 않은 노드를 Q개 샘플링하여 각 특징 벡터와 안 유사해지도록 학습함

Ex) Unsupervised Learning

** 모든 z 는 인코딩 과정에서 normalized 됨

** σ 는 Sigmoid 함수



$$Loss(z_v) = -\log(\sigma(z_u^T z_v) + \epsilon) - \underbrace{Q}_{\text{Negative Sample 개수}} * \underbrace{\mathbb{E}_{v_n \sim P_n(v)}}_{\text{Negative Sample Distribution}} \log(\sigma(-z_u^T z_{v_n}) + \epsilon)$$

이웃한 노드 간의 내적 (코사인유사도)

내적 값이 클수록 0에 가까워짐

Negative Sample 개수

Negative Sample Distribution

내적 값이 작을수록 (-1에 가까울수록) 0에 가까워짐

<https://medium.com/analytics-vidhya/ohmygraphs-graphsage-and-inductive-representation-learning-ea26d2835331>

Experiments

❖ Benchmark & Various Aggregators

- Web of Science 인용 정보, Reddit 게시물, Protein-protein interaction 구조 정보 데이터로 성능 비교
- GraphSAGE에 GCN, Mean aggregator, LSTM aggregator, Pooling aggregator를 사용하여 성능 비교

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K ; weight matrices $\mathbf{W}^k, \forall k \in \{1, \dots, K\}$; non-linearity σ ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, \dots, K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations \mathbf{z}_v for all $v \in \mathcal{V}$

```
1  $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$  ;
2 for  $k = 1 \dots K$  do
3   for  $v \in \mathcal{V}$  do
4      $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$ ;
5      $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k))$ 
6   end
7    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$ 
8 end
9  $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$ 
```

$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}))$

**GCN-GraphSAGE의 경우 다음의 식으로 대체

Default training settings (GraphSAGE)

• Optimizer:	Adam
• Batch size:	512
• Hop size:	2
• Neighbor sample size(1 st hop):	25
• Neighbor sample size(2 nd hop):	10
• Loss function(Supervised):	Cross Entropy

Experiments

❖ Benchmark & Various Aggregators

- Raw features는 그래프 구조를 사용하지 않고 로지스틱 회귀 모델에 데이터를 입력한 것
- LSTM, Pooling을 이용한 버전이 좋은 성능을 보여줌 (일반적으로 pooling 버전을 많이 사용함)
- Hop size는 GCN과 마찬가지로 2로 설정하는 것이 가장 좋은 성능을 보인다고 함

Table 1: Prediction results for the three datasets (micro-averaged F1 scores). Results for unsupervised and fully supervised GraphSAGE are shown. Analogous trends hold for macro-averaged scores.

Name	Citation		Reddit		PPI	
	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1	Unsup. F1	Sup. F1
Random	0.206	0.206	0.043	0.042	0.396	0.396
Raw features	0.575	0.575	0.585	0.585	0.422	0.422
DeepWalk	0.565	0.565	0.324	0.324	—	—
DeepWalk + features	0.701	0.701	0.691	0.691	—	—
GraphSAGE-GCN	0.742	0.772	0.908	0.930	0.465	0.500
GraphSAGE-mean	0.778	0.820	0.897	0.950	0.486	0.598
GraphSAGE-LSTM	0.788	0.832	0.907	0.954	0.482	0.612
GraphSAGE-pool	0.798	0.839	0.892	0.948	0.502	0.600
% gain over feat.	39%	46%	55%	63%	19%	45%

Conclusion

❖ Conclusion

- 기존 GCN의 경우 transductive한 학습 특성상 대규모 데이터 학습과 새로 발생한 데이터에 대한 추론이 어려웠음
- GraphSAGE 모델은 위의 한계점을 다음과 같이 해결함
 - ✓ 샘플링 방법론을 통해 대규모 데이터도 **mini batch** 방식으로 효율적인 학습이 가능하도록 함
 - ✓ 이웃 노드만으로 특정 노드를 표현함으로써 **inductive** 학습으로 새로 발생한 데이터도 기존 모델로 추론이 가능하도록 함

❖ Reference

- Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.
- <https://medium.com/analytics-vidhya/ohmygraphs-graphsage-and-inductive-representation-learning-ea26d2835331>
- <https://towardsdatascience.com/inductive-vs-transductive-learning-e608e786f7d>

Thank You