Legacy와 ML-based 방법들의 Community detection/Node classification성능 비교

ICA Node2Vec GCN graphSAGE 비교

2015-11923 최한결

1. Introduction

현실 세계에서는 서로 다른 분야의 복잡한 시스템이 네트워크를 형성한다. 그 중에는 소셜 네트워크, 웹 페이지, 운송 네트워크, 인용 네트워크, 대사 네트워크, 단백질 상호작용 네트워크, 유전자 네트워크 등이 있다. 네트워크는 신경과학에서부터 사회학, 경제, 정보기술에 이르기까지 인간 존재의 거의 모든 영역에 존재한다. 소셜 네트워크의 커뮤니 티는 유사한 선호나 관심사를 가진 사람들 을 대표할 수 있다. 또한 질병의 확산을 막 기 위해 지역사회와 사람들 사이의 연관성 을 인식하는데 사용할 수 있다. 따라서 복잡한 네트워크에서 커뮤니티의 탐지는 다양한 과학 분야 및 실생활에 관련된 서비스에서의 광범위한 응용을 가지고 있는 것이 분명하다. 최근 몇 년 간 네트워크의 크기가 급속도로 커지고 있고 데이터가 큰 만큼 효과적인 커뮤니티 검출 알고리즘을 찾는게 중요하다. 여러가지 방식들이 소개되었고 최근에 머신러닝 관련 연구가 활발히 진행되면서 네트워크 분석 분야에서도 머신러닝을 활용하기 시작했고 매우 높은 성능을 보여주는 방식들이 소개되었다. 이 연구에서는 Community detection/Node classification methods 중 소개된 몇몇 legacy 방식들과 ML-based 방식들의 성능을 비교해 볼 것이다. 대상 methods는 ICA, node2vec, GCN, graphSAGE 4가지이다.

2. Description of methods, measures (metric)

2.1. Method

2.1.1. Iterative classification algorithm(ICA)[1]

ICA는 네트워크 구조와 각 노드의 속성을 동시에 이용해서 node labeling하는 방식이

다. 절차는 다음과 같다.
각 노드 속성을 기준으로 SVM, kNN등의 알고리즘을 사용해 노드를 labeling한다.
그 후 각 주변 노드의 label을 나타내는 network feature vector를 추가해 노드들의 label을 업데이트한다. 수렴할 때까지 반복시킨 후 결과를 반환한다.

2.1.2. Node2Vec[5]

2015년에 발표된 방식으로 2014년에 발표된 Deepwalk[4]와 유사하다. 기존 NLP technique인 word2vec[2,3]에서 word를 node, sentence를 random walk에 대응한 방식이 Deepwalk 방식이고 조금 더 발전시킨 방식이다. 기존 Deepwalk방식에서는 무작위로 이웃 노드들을 순회하여 word2vec의 하나의 sentence로 대응시켰지만 Node2vec 방식에서는 random walk가 아닌 BFS와 DFS를 활용하여 가까운 노드에 높은 weight, 먼 노드에 낮은 weight를 준다. BFS는 가까운 노드를 우선 탐색하고 DFS는 먼 노드를 우선 탐색하는 경향이 있으므로 DFS와 BFS를 이용해 weight를 적절히 줄 수 있다. 이 후에는 word2vec방식과 동일하게 주변 노드를 학습시키고 같은 공간에 존재하는 노드들을 같이 분류한다.

2.1.3. GNN [6,7]

2.1.3.1. 기본 개념

GNN은 그래프 구조에서 사용하는 인공 신경망이다. 주변 이웃 노드들의 정보와 자기 자신 노드의 정보를 이용하여 node embedding을 수행한다. 이 때 주변 노드들의 정보를 모으는 과정을 aggregate, 그 정보와 자기 자신의 값을 이용해 새로운 embedding상태를 구하는 과정을 concat이라고 한다. GNN의 학습도 일반적인 인공 신경망의 학습과 비슷하게 이뤄진다. 따라서 aggregate함수와 concat함수를 정의, loss function, optimizer를 사용해 loss가 0에 가까워질 때까지 학습시킨다. 이 때 aggregate, concat함수에 따라 다음 두 방식으로 나뉜다.

2.1.3.2. GCN [8]

GCN에서의 aggregation fucntion은 다음과 같다.

$$h_v^k = \sigma\left(W_k \sum_{u \in N(v) \cup v} \frac{h_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

(이 때 σ는 activation function으로 여기에서는 ReLU function을 사용했다.)
이를 행렬을 이용해 표현하면 다음과 같다.

$$\tilde{A} = A + I$$

$$D_{i,i} = \sum_j A_{i,j}$$

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

2.1.3.3. GraphSAGE [9]

GraphSAGE에서 사용하는 Aggregation function 중 average aggregation을 사용한다. Average aggregation strategy는 다음과 같다.

$$h_v^0 = x_v$$

$$h_v^k = \sigma\left(W_k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B_k h_v^{k-1}\right)$$

따라서 GCN과 다르게 B,W가 학습되고 자신의 이전 embedding을 이웃 노드들과 따로 더해준다.

3. **Details of experiment**

3.1. 실험 데이터

3.1.1. Zachary's karate club [10]

Zachary의 논문에 나온 karate club의 dataset을 사용했다. Karate club이 Mr. Hi의 클럽과 Officer의 두 클럽으로 나뉘었고 이를 통해 각 노드를 labeling 하였다. 이 labeled data를 이용해 지도 학습을 시켰다. 34개의 노드가 존재, 78개의 엣지가 있다. 노드 수가 매우 적어 무작위 표본을 반복 추출해서 학습시켰다.

Figure 1 karate club graph를 시각화한 모습

3.1.2. Cora [11]

다음 데이터는 7가지로 분류된 머신러닝 논문의 데이터이다. 2708개의 노드가 존재하고 5429개의 엣지가 존재한다. 또한 사용빈도가 높은 단어들의 존재 유무로 각 논문별 크기 1433의 feature vector를 갖고 있다.

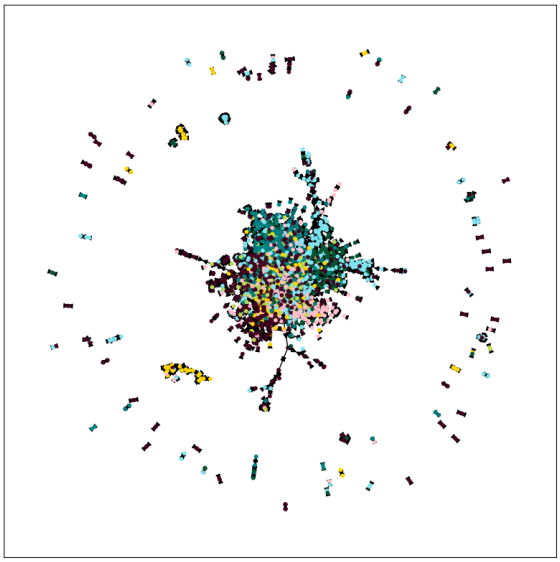


Figure 2 Cora dataset graph를 시각화한 모습

3.1.3. Dataset 비교

	Karate club	Cora
Nodes	34	2708
Edges	78	5429
Classes	2	7
features	X	1433

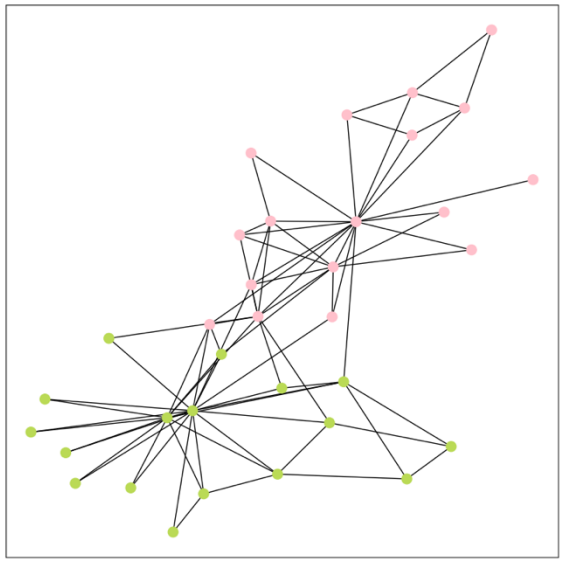
Table 1 dataset 그래프 비교

4. Performance analysis

4.1. Zachary's karate club

4.1.1. ICA

Zachary's karate club 데이터에 네트워크 구조를 제외한 feature가 없어 초기 분류 모델을 설정하지 못했다. 따라서 Mr. Hi와 Officer 노드를 우선 0과 1로 labeling하고 나머지 unlabeled node들을 update하는 방식을 사용했다. Node label Vector가 수렴 후 accuracy는 0.94가 되었다. ICA를 이용해 node labelling을 한 것을 나타내면 다음과 같다.



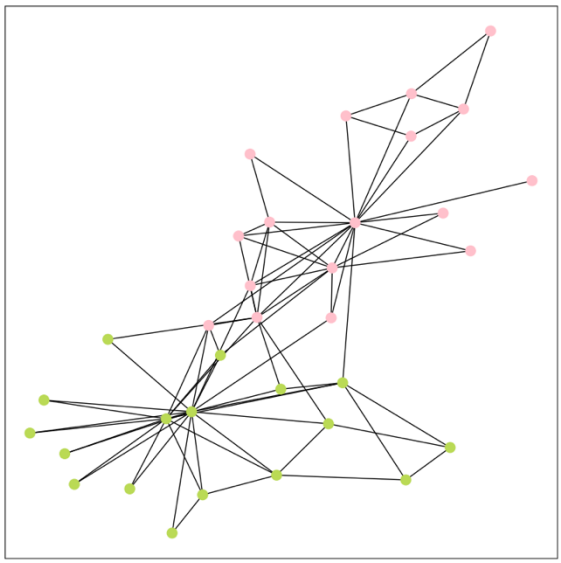


Figure 3 실제 분류 데이터로 나타낸 그래프

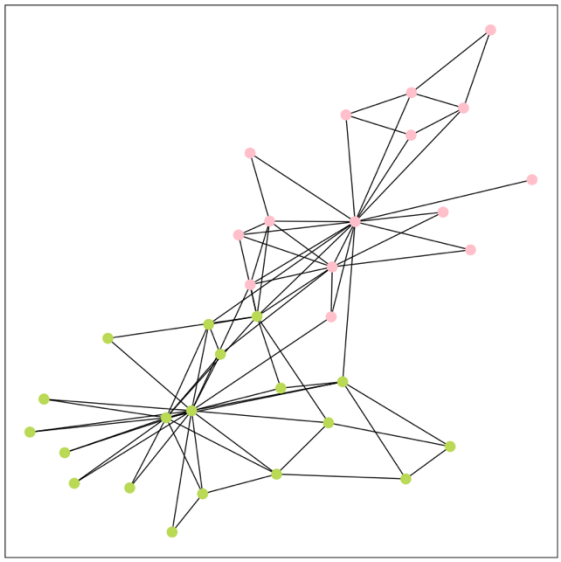


Figure 4 ICA를 이용해 node labeling을 한 그래프

4.1.2. Node2vec

그래프를 무작위 추출을 반복하여 node2vec 방식을 이용해 학습시켰다. 전체 그래프에 대해 학습된 모델을 적용시킨 후 결과를 이차원에 매핑한 결과는 다음과 같다.

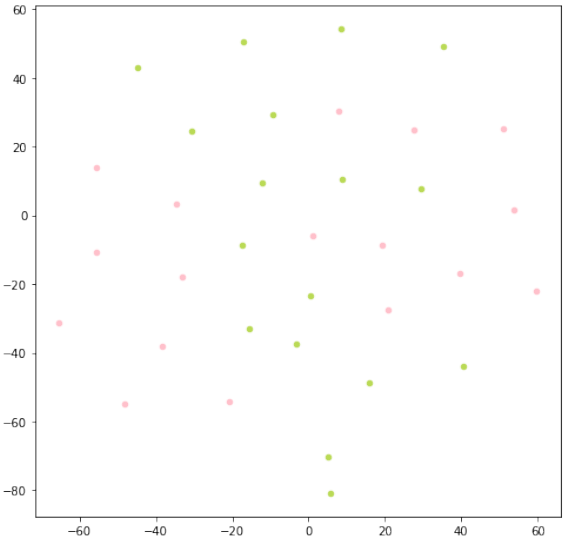


Figure 5 전체 그래프를 node2vec방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 0.529이다.

4.1.3. GCN

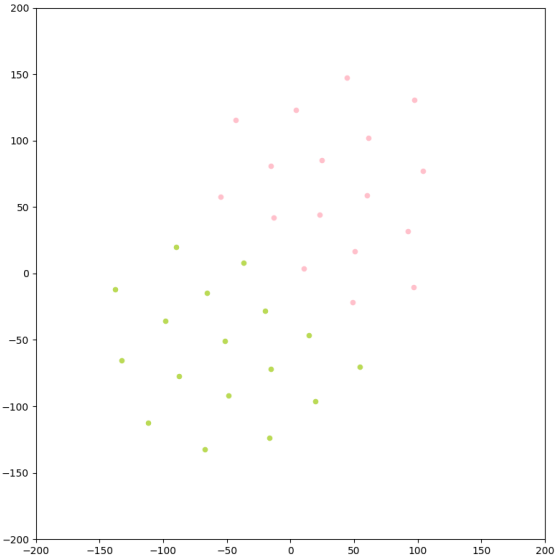


Figure 6 전체 그래프를 GCN방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 0.970이다.

4.1.4. graphSAGE

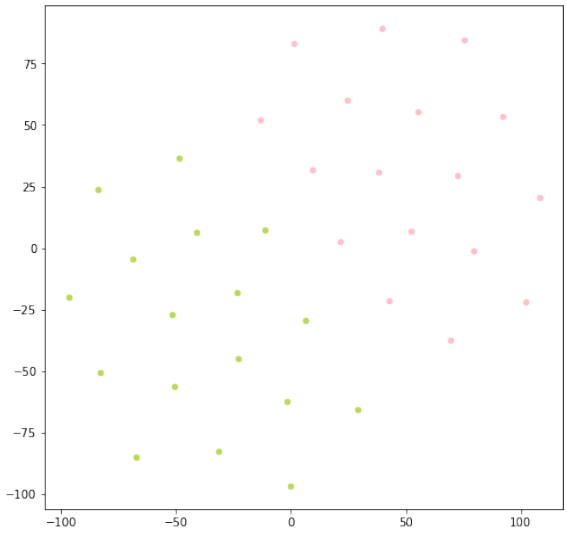


Figure 7 전체 그래프를 graphSAGE방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 1.0이다.

4.2. Cora

4.2.1. ICA

Cora데이터에 ICA를 적용시켜 노드를 분류했다. 이 때의 accuracy는 0.737이다.

4.2.2. Node2vec

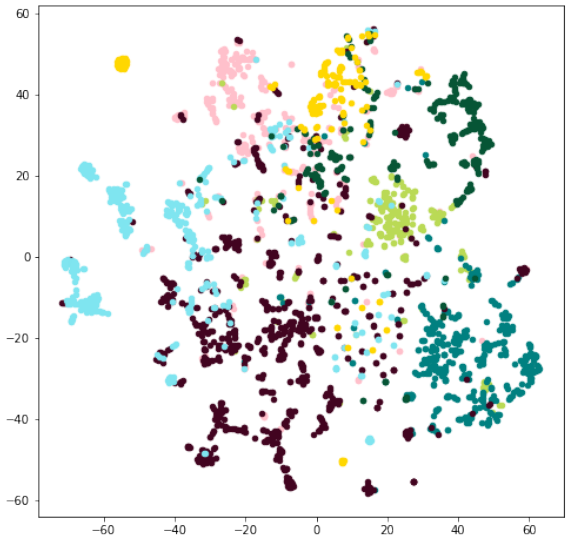


Figure 8 Cora 전체 그래프를 node2vec방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 0.730이다.

4.2.3. GCN

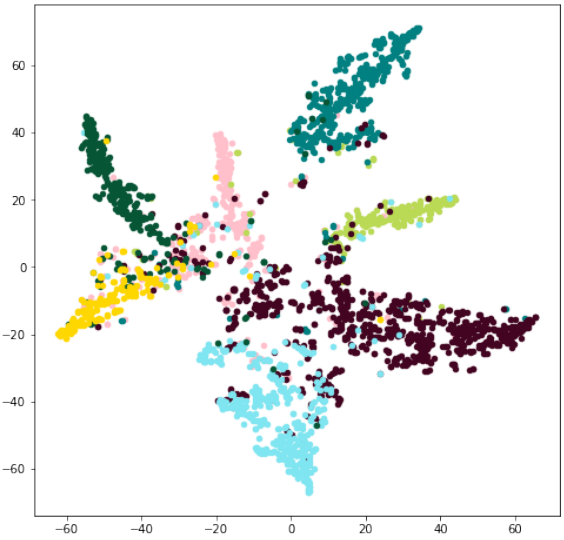


Figure 9 Cora 전체 그래프를 GCN 방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 0.807이다.

4.2.4. graphSAGE

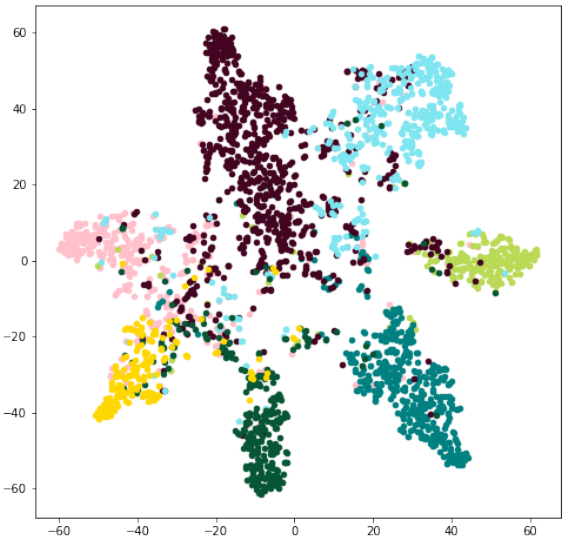


Figure 10 Cora 전체 그래프를 graphSAGE방식 적용하여 이차원 매핑시킨 결과

이 때의 accuracy는 0.798이다.

4.3. Methods 성능 비교 (Accuracy)

	Karate club	Cora
ICA	0.940	0.737
Node2vec	0.529	0.730
GCN	0.970	0.807
graphSAGE	1.0	0.798

Table 2 각 방식별 accuracy 측정 결과

5. Conclusion

그래프 이론은 다양한 영역에서 다양한 응용에 널리 사용된다. 커뮤니티를 탐색해 어떻게 하면 질병의 전파를 막을 수 있을지, 항공사나 고속도로를 결정할 수 있을지 등등을 알 수 있다. 그래프 신경망을 이용하여 그래프 데이터를 매우 높은 성능으로 분석할 수 있게 되었고 노드 분류와 같은 문제들을 다루기 쉽게 만들었다. 각각의 method가 최근 몇 년 사이에 만들어졌고 위에서 알 수 있듯이 매우 빠른 속도로 발전하고 있다. 앞으로 분류/예측이 매우 쉬워질 것으로 예상된다.

6. References

[1] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T., "Collective Classification in Network Data", AI Magazine, 2008

[2] Mikolov & et.al.,"Distributed Representations of Words and Phrases and their Compositionality", NIPS, 2013

[3] Mikolov & et.al., "Efficient Estimation of Word Representations in Vector Space", ICLR,2013

[4] Bryan Perozzi et al., "DeepWalk: Online Learning of Social Representations", KDD, 2014

[5] Aditya Grover and Jure Leskovec., "node2vec: Scalable Feature Learning for Networks", Knowledge Discovery and Data Mining, 2016.

[6] Scarselli, F., Gori, M., Tsoi, A., Hagenbuchner, M. & Monfardini, G., "The graph neural network model", IEEE, 2009

[7] Jie Zhou et al., "Graph Neural Networks:
A Review of Methods and Applications",

[8] Thomas N Kipf and Max Welling., "Semi-supervised classification with graph convolutional networks.", ICLR, 2017

[9] William L. Hamilton et al., "Inductive Representation Learning on Large Graphs", NIPS, 2017

[10] Zachary, W. W. , "An Information Flow Model for Conflict and Fission in Small Groups". Journal of Anthropological Research,1977.

[11] Zhilin Yang et al., "Revisiting Semi-Supervised Learning with Graph Embeddings", 2016