컴퓨터 과학 연구 동향을 반영한 그래프 기반의 arXiv 데이터셋 구축

전주현 [1], 강윤석 [2], 김상욱 [3*]
[1] 한양대학교 지능융합학과
[2] 미시간대학교 정보대학
[3] 한양대학교 컴퓨터소프트웨어학과
jjh1012@hanyang.ac.kr, dyskang@umich.edu, wook@hanyang.ac.kr

Constructing a Graph-Based arXiv Dataset By Reflecting the Research Trend in Computer Science

Juhyun Jeon[1], David Y. Kang[2], Sang-Wook Kim[3*]
[1]Dept. of Intelligence and convergence, Hanyang University
[2]School of Information, University of Michigan
[3]Dept. of Computer Science, Hanyang University

요 약

컴퓨터 과학(CS) 분야는 다른 학문 분야에 비해 연구 동향이 빠르게 변하는 특성을 가지고 있다. 그래프 마이닝에서 활발히 사용되는 CS 분야 논문 데이터셋들(e.g., Cora, Citeseer, DBLP)은 오래된 논문을 중심으로 구성되어 있어 이러한 특성을 제대로 반영하지 못하는 한계가 있다. 따라서 본논문에서는 CS 분야의 최신 트렌드를 반영하는 논문 데이터셋을 제안한다. 이를 위해, 우리는 CS 분야 논문을 활발히 공개하는 플랫폼인 arXiv 에서 2007 년부터 2023 년까지 해당 플랫폼에서 공개된 논문들을 수집하고, 이를 기반으로 공저자 그래프 및 인용 그래프로 구축한다. 해당 데이터셋을 대상으로 폭넓은 분석을 통해, 우리가 구축한 데이터셋이 실세계 그래프 네트워크 특성을 잘 반영하고 있음을 보인다. 또한, 향후에 해당 데이터셋을 사용하려는 연구자들을 위해, 해당 데이터셋에서의 기존 그래프 기반 응용들의 노드 분류 성능을 제시한다.

1. 서론

컴퓨터 과학(CS) 분야는 다른 학문 분야에 비해 연구 동향이 빠르게 변하는 특성을 지닌다. 이에 따라 CS 분야는 논문 공개가 상대적으로 느린 저널(journal)보다 학회(conference)에 논문을 더 많이 제출한다 [1]. 그러나 학회의 수가 한정되어 있고, 학회에서의 논문 게재 비율이 매우 낮기 때문에 하나의 논문이 어떤 학회에 발표되기까지 많은 시간과 비용이 요구된다. 이에 따라 CS 분야 연구원들은 자신이 연구한 내용이 사장되는 것을 방지하기 위해 미리 웹에 공개를 하는데, 주로 arXiv 라는 플랫폼을 활용한다.

arXiv 는 미국 코넬 대학교에서 운용하는 온라인 논문 리포지토리로서, 연구원들이 최신 연구 결과를 빠르게 공유하고 피드백을 받는 플랫폼이다. arXiv 에 올린 논문은 학회에 정식으로 출판되기 전에 사전에 공헌을 선점하는 것으로 간주되기 때문에 CS 분야에서는 arXiv 가 매우 중요한 플랫폼으로 자리 잡았고, CS 분야 학회들도 arXiv 를 인정하여 해당 플랫폼에 올린 논문들도 심사받을 기회를 제공한다.

그래프 기반 응용들은 자신의 성능을 검증하기 위해 다양한 분야(e.g., 논문, 사회연결망 서비스, 온라인 쇼핑몰 등)의 그래프 데이터셋을 이용한다. 이 중 가장 많이 이용되는 그래프는 논문 저자들의 관계를 나타내는 공저자 그래프, 논문의 인용관계를 나타내는 인용 그래프이다. 특히 CS 분야의 논문들의 공저자 또는 인용 관계를 그래프로 표현한 데이터셋(e.g., Cora [2], Citeseer [3], DBLP [4])들이 최근 들어 많이 사용되고 있다. 그러나 이들은 2006 년 이전의 논문들로 구성되어 있으며, 데이터를 수집한 플랫폼이 운용이 종료된 곳도 있어 최근 논문들로 업데이트를 하지 못한다. 이에 따라 해당 데이터셋들은 위와 같은 CS 연구분야의 특성을 제대로 반영하지 못하는 한계가 존재한다. 이에 따라 그래프 기반 응용기술의 정확한 검증을 위해서 실세계 상황을 제대로 반영하는 논문

* 교신저자

데이터셋이 필요하다.

이를 위해 우리는 먼저 arXiv 플랫폼 내에서 07 년도부터 23 년까지의 CS 분야의 논문을 수집한다. 이후, 수집한 논문들을 대상으로 공저자(arXiv-coauth) 및 인용(arXiv-cocitation) 관계 그래프를 구축한다. 실험을 통해 우리는 구축한 데이터셋을 분석하고 기존 그래프 기반 응용 기술들의 노드 분류 성능을 제시한다. 이를 통해 우리가 구축한 데이터셋들이 현실성이 있음을 보인다.

2. 관련 연구

2.1 실세계 그래프 분석

기존의 많은 연구들이 실세계 그래프가 지닌 같은 특성들을 분석했고 [5], 이를 정리하면 다음과 같다. Degree distribution (DD)은 각 노드의 차수의 확률 분포이다. 실세계 그래프는 heavy-tailed 분포, i.e., 높은 차수의 노드가 적게 존재하고, 낮은 차수의 노드가 많이 존재하는 구조를 가진다. Giant connected component (GCC)는 노드가 서로 연결되어 있는 집합들 중 가장 큰 집합을 의미한다. 실세계 그래프는 GCC 의 크기가 매우 큰 경향을 보인다. Effective diameter (ED)은 모든 연결된 노드 쌍 중 90%의 노드 쌍들이 도달할 수 있는 최소 거리이다. 실세계 그래프는 대부분 작은 거리로 도달이 가능하다. Clustering coefficient (CC)는 특정 노드가 얼마나 그 주변 노드들끼리 연결되어 있는지를 측정하는 지표이다. CC 가 높을수록 해당 노드는 강한 연결성을 가진다. 실세계 그래프는 평균 CC 값이 높은 경향을 가지며, 이는 공통 이웃이 많이 존재함을 의미한다.

2.2 노드 분류 (Node classification)

노드 분류는 그래프 내의 노드에 레이블을 분류하는 대표적인 그래프 기반 태스크 중 하나이다. 이를 위한 그래프 기반 분류 방법은 다음과 같이 정리될 수 있다. 그래프 기반 분류 방법은 그래프 구조를 이용하는 것과 이용하지 않는 것으로 나눠진다. 그래프 구조를 이용하지 않는 방법들 중 하나인 MLP 는 각 노드의 특성을 입력으로 받아 여러 개의 은닉층을 거쳐 출력층으로 전달해 각 노드의 레이블을 예측한다.

그래프 구조를 이용하는 방법은 (1) 노드의 feature 를 추출할 때만 Node2vec [6]과 같은 임베딩 기법을 이용해 그래프 구조를 이용하고, 추출된 feature 를 이용해 분류기를 학습하고 분류하는 것과 (2) 노드의 feature 추출, 분류기를 학습할 때 그래프 구조를 이용하는 것으로 나뉜다. GCN [7], GraphSAGE [8]가 대표적인 방법들로, 이웃과의 관계를 고려해 노드 특성 및 분류기를 학습한다. 그리고 학습된 분류기를 통해 노드 레이블을 예측한다. 그래프 기반 노드 분류 방법들은 그래프의 구조를 활용해야 분류 성능을 향상되는 것으로 알려져 있다 [7, 8].

3. arXiv CS 데이터셋

3.1 구축 방법

본 장에서는 arXiv CS 데이터셋 구축 과정을 소개한다. 우리는 arXiv api 를 사용해 2023 년에 발행된 약 500 개의 논문을 샘플링했다. [9]와 같은 방식으로, 샘플링된 논문에서 추출된 참고 문헌과 저자 정보로 2-hop 까지의 이웃 논문을 수집했다. 논문 정보를 바탕으로 공저자(arXiv-coauth)와 인용(arXiv-cocitation) 그래프를 구축한다. 동적인 상황을 고려해 시간별로 훈련/검증/테스트 셋을 구성한다.

3.2 데이터 통계 및 분석

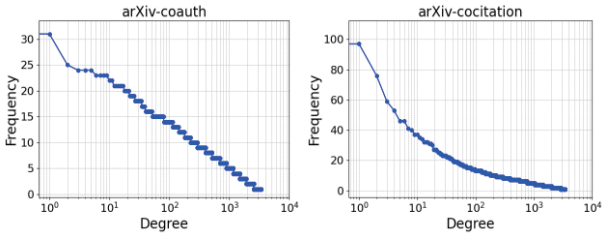
<표 1>은 arXiv-coauth, arxiv-cocitation 데이터셋의 통계를 나타낸다. arXiv-coauth 데이터셋은 공저자 그래프로서 3397 개의 노드(논문)과 7170 개의 엣지로 구성된다. 연도별 분할을 사용해 훈련(22 이전), 검증(22), 테스트(23)로 분할한다. arXiv-cocitation 데이터셋은 인용 그래프로 3433 개의 노드(논문)과 7260 개의 엣지로 구성된다. 마찬가지로 연도별 분할을 사용해 논문을 훈련(20 이전), 검정(20, 21, 22), 학습(23)로 분할한다.

<표 1> arXiv-coauth, arXiv-cocitation 데이터셋 통계

Dataset	arXiv-coauth	arXiv-cocitation
Nodes	3397	3433
Edges	7170	7260
Feature	1000	1000
Classes	7	7
Training node	0.52(22 이전)	0.63(20 이전)
Validation node	0.37(22)	0.27(20, 21, 22)
Testing node	0.11(23)	0.10(2023)

우리는 구축한 데이터셋이 현실성이 있는지 확인하기 위해, 우리 데이터셋이 기존의 특성을 잘 따르는지 분석한다. (그림 1)은 두 데이터셋(arXiv-coauth(a), arXiv-cocitation(b))의 degree distribution 을 나타낸다. 두 데이터셋 모두 heavy-tailed 분포를 따르는 것을 볼 수 있다. <표 2>는 두 데이터셋의 GCC, CC, ED 값을 나타낸다. <표 2>에서 볼 수 있듯이, GCC 비율이 두 데이터셋 모두에서 큰 비율을 차지하고 있다. 또한, ED 또한 기존의 논문 데이터셋(Cora: 7.0, citeseer: 12.8)과 비슷한 경향을 보인다. 반면, CC 는 arXiv-coauth 에서는 비교적 큰 값을 가지나 arXiv-cocitation 에서는 작은 값을 가진다. 빠르게 연구동향이 변하는 CS 분야 특성상 최근 논문들을 인용하는 경향이 있어 연도가 달

라지면 비슷한 연구라도 다른 논문이 인용될 가능성이 높다. 그래서 인용 그래프는 CC 가 낮은 것으로 생각되며, 기존의 인용 그래프도 낮은 CC 값을 가지는 것을 기존 연구에서 확인했다 [10].



(그림 1) arXiv-coauth(a), arXiv-cocitation(b)의 DD

<표 2> arXiv-coauth 와 arXiv-cocitation 의 세가지 특성

Dataset	arXiv-coauth	arXiv-cocitation
Clustering coefficient	0.42	0.09
Connected component	0.78	0.96
Efficient diameter	10	8

이를 통해 우리가 추출한 데이터셋이 기존의 논문 데이터셋과 비슷한 특성을 보이는 것을 확인할 수 있었고, 이는 우리의 데이터셋이 현실성이 있는 것을 의미한다.

3.3 노드 분류 성능 비교

다음으로 우리는 구축한 데이터셋에서의 그래프 기반 작업(노드 분류) 성능 결과를 제시한다. 이를 위해 우리는 2.1장에서 언급한 노드 분류 방법들 MLP, node2vec+MLP, GCN, GraphSAGE를 이용한다. <표 3>은 구축한 데이터셋에서의 노드 분류 결과를 나타낸다. 아래 <표3>에서 알 수 있듯이 GCN과 GraphSAGE의 정확도가 MLP나 Node2vec에 비해 높은 성능을 보이고, 그래프의 구조를 고려하지 않는 MLP는 성능이 낮음을 보인다. 이러한 결과는 기존의 논문 데이터셋에서 나온 결과와 유사한 경향이며, 이는 우리가 구축한 데이터가 현실적이라는 것을 다시 한번 더 입증한다.

<표 3> 노드 분류 정확도

Dataset	arXiv-coauth	arXiv-cocitation
MLP	38.80%	40.17%
Node2Vec	36.98%	41.29%
GCN	41.67%	57.30%
GraphSAGE	42.97%	50.28%

4. 결론 및 향후 연구

본 논문에서는 CS 분야의 최신 트렌드를 반영한 arXiv 의 논문 데이터셋인 arXiv-cocitation, arxiv-coauth 를 제공한다. 두 데이터셋은 arXiv 플랫폼의 CS 분야 논문들로 이루어져 있으며 각 논문은 세부 분야로 나눠지며 연도별로 분류하는 현실적인 데이터 분할을 사용한다. 폭넓은 실험을 통해, 우리가 구축한 데이터셋이 현실적이라는 것을 보인다. 또한, 우리는 노드 분류 정확도를 제시하고, 해당 실험을 통해 다시 한번 더 우리의 데이터가 현실성이 있음을 보인다. 향후 계획으로, 우리는 다양한 도메인(영화, 법안, 쇼핑몰 등)과 다양한 그래프 기반 태스크(edge prediction, graph classification)를 수행할 수 있는 통일된 데이터셋을 제공할 예정이다. 또한 노드 혹은 엣지에 text-attribute 을 추가해 언어 모델(language model)도 결합할 수 있는 데이터셋으로 확장하고자 한다.

사사

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2018R1A5A7059549, No.2022-0-00352, No.RS-2022-00155586, 실세계의 다양한 다운스트림 태스크를 위한 고성능 빅 하이퍼그래프 마이닝 플랫폼 개발 (SW 스타랩)).

참고문헌

[1] J. Kim, "Author-based analysis of conference versus journal publication in computer science," Journal of the Association for Information Science and Technology, Vol. 70, No. 1, pp. 71-82, 2019.

[2] A.K. McCallum, et al, "Automating the construction of internet portals with machine learning," Information Retrieval, Vol. 3, No.2, pp. 127–163, 2000.

[3] C.L. Giles, K.D. Bollacker, and S. Lawrence, "CiteSeer: an automatic citation indexing system," In Proc. of the third ACM conference on Digital libraries, 1998, p. 89-98.

[4] J. Tang et al, "ArnetMiner: extraction and mining of academic social networks," In Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 990-998.

[5] M. Tuan Do et al, "Structural patterns and generative models of real-world hypergraphs," In Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 176-186.

[6] A. Grover, J. leskovec, "node2vec: scalable feature learning for networks", In Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2016, pp. 855–864.

[7] T.N. Kipf, W. Max, "Semi-supervised classification with graph convolutional networks," In International Conference on Learning Representations, 2017.

[8] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive represen tation learning on large graphs," In Advances in Neural Information Processing Systems, 2017, pp. 1025–1035.

[9] J. Huang et al, "Can LLMs effectively leverage graph structural information through prompts, when and why," arXiv, 2023.

[10] W. Hu, et al, "Open graph benchmark: Datasets for machine learning on graphs. In Advances in Neural Information Processing Systems," 2020.