그래프 합성곱 신경망을 위한 특징 노드를 활용한 그래프 확장 기법*

이동진°, 황혜수
서울시립대학교 컴퓨터과학부
ehdls1303@uos.ac.kr, hwang@uos.ac.kr

Graph Augmentation Technique for Graph Convolution Network Using Feature Nodes

Dongjin Lee°, Heasoo Hwang
Department of Computer Science and Engineering, University of Seoul

요 약

예측 시스템에 사용되는 LightGCN을 비롯한 그래프 합성곱 신경망 모델들은 주로 그래프 구조 정보를 활용하여 학습을 진행한다. 그러나, 이들은 노드의 특징 정보가 주어져도 이를 바로 모델 학습에 활용할 수 없는 경우가 많다. 본 논문에서는 기존의 모델을 수정하는 대신, 주어진 그래프 구조 정보에 노드들의 특징 정보를 담아내어 그래프 구조를 확장함으로써 그래프 합성곱 신경망 모델의 변경 없이 특징 정보를 학습에 활용하는 방법을 제안한다. 이를 위하여 노드의 개별 특징을 가상의 특징 노드(feature node)로 정의하고 이를 주어진 그래프에 추가한다. 이때 해당 특징과 강하게 관련된 노드들을 특징 노드와 연결함으로써 동일한 특징을 가진 노드들이 해당 특징 노드를 통해 서로 연결될 수 있고 결과적으로 전체 그래프의 연결성이 특징 정보를 반영하는 방향으로 강화된다. 실제 데이터셋 상의 실험을 통하여 그래프의 연결성이 약한 데이터셋에서 제안 방법의 추천 성능 향상이 더 두드러짐을 확인하였다.

1. 서 론

그래프 합성곱 신경망(Graph Convolution Network, GCN) 모델들은 그래프 구조를 바탕으로 인접 노드를 종합(Aggregation)하는 방식으로 학습을 진행한다. 하지만 대체로 그래프 구조 학습에 집중하기 때문에 각각 노드의 특징(feature) 정보는 바로 활용하지 못하는 경우가 많다. 대표적인 그래프 합성곱 신경망 모델인 LightGCN[1]은 NGCF[2]의 불필요한 복잡성을 줄여 성능과 속도 면에서 모두 개선하였지만, 노드의 특징(feature) 정보를 활용하지 못한다는 문제점은 해결하지는 못하였다. LightGCN에서 구조를 단순화하여 성능을 개선시킨 UltraGCN[3]도 동일한 문제점을 가지고 있다. 또한 이웃을 선별하여 종합하는 모델인 GraphSAGE[4]와 PinSAGE[5]에서도 노드의 특징을 선별에 활용하지 않는다.

그렇다면 왜 노드의 특징을 활용하는 것이 중요할까? 실제로 우리가 친구에게 어떤 식으로 영화를 추천해 주는지 생각해보자. 친구가 A 감독의 영화를 좋아했다면, A 감독의 다른 영화를 추천해 주었을 때 좋아할 확률은 상당히 높을 것이다. 그러나 A 감독의 영화들이 A 감독이라는 동일한 특징으로 연관되어 있음을 모르는 상태에서는 이와 같은 추천이 불가능하다. 본 논문에서는 그래프 합성곱 신경망 모델의 변경 없이 특징 정보를 활용하여 그래프 자체를 확장함으로써 추천 성능을 향상하고자 한다.

MultiSAGE[6]는 핀(pin)-보드(board) 그래프에서 모델 학습을 진행한다. 보드는 핀들을 모아두는 역할을 하여, MultiSAGE는 보드를 맥락(context)으로 활용하여 동일한 핀을 다양한 맥락에 따라 다르게 추천한다. 그래프 합성곱 신경망 모델들에서는 아이템(item)-유저(user) 그래프의 구조를 학습하여 각 유저 별 추천을 수행한다. MultiSAGE의 핀-보드 그래프와 비교했을 때, 우리는 유저 또한 관련 아이템을 모아두는 보드(또는 맥락)로 간주할 수 있음에 주목하였다. 영화 평점 데이터(영화-유저)그래프에서 생각해보자. 공상과학(SF) 영화를 좋아하는 매니악(maniac)한 유저는 수많은 공상과학 영화 아이템들과 상호작용(interaction)했을 것이다. 이 유저는 공상과학 영화를 모아둔 보드(또는 맥락)로 보아도 무방하다. 이러한 유저와 같은 가상의 노드를 그래프 상에 추가할 수 있다면, 동일한 특징을 가진 노드들이 연결되면서 전체 추천 성능을 향상 시킬 것이다.

즉, 그래프 합성곱 신경망 모델이 노드의 특징 정보를 반영하지 못한다면, 특징 정보를 반영하도록 가상의 노드를 추가하여 그래프를 확장함으로써 기존 모델의 변경 없이 추천 성능을 향상하고자 한다. 이를 위해 본 논문에서는 노드 특징 정보를 기존의 아이템-유저 그래프 상에 담기 위해 특징 노드(feature node)를 추가하고 관련도 높은 노드들과 연결하여 그래프를 확장할 것을 제안한다.

2. 제안 방법

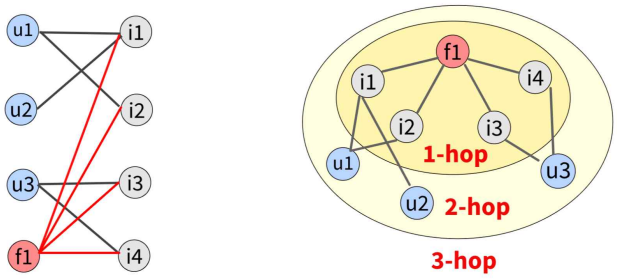
2.1. 특징 노드

특징 정보를 아이템-유저 그래프 구조에 반영하기 위해서, 특징 노드라는 가상의 노드를 그래프에 추가한다. 특징 노드는 동일한 특징을 가진 노드들을 연결하는 노드로 정의한다.

예를 들어 A 감독의 영화를 정말 좋아하는 유저 B가 있다면, A 감독의 거의 모든 영화 아이템과 상호작용을 가지고 있을 것이다. 이 유저 B는 그래프 상에서 A 감독의 많은 영화 아이템을 연결하는 역할을 하므로, A 감독의 영화를 몇 개 본 유저들은 유저 B를 통하여 A 감독의 다른 영화들로 연결되므로 이를 통해 추천 성능이 향상될 수 있다. 하지만 만약 이러한 특징적인 유저가 기존의 모든 그래프 데이터셋에 존재할까? 노드

* 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2022R1A2C1011637).

간 연결이 희박한(sparse)한 그래프일수록 이런 특징적인 유저가 존재하기 힘들다. 따라서 본 논문에서는 특징이 같은 아이템들을 연결하기 위하여 가상의 유저 노드에 해당하는 특징 노드를 추가하여, 유저 B 노드와 동일한 역할을 수행하도록 한다.



유저-아이템-특징 노드 그래프 f1 노드의 연결 구조

그림 1. 특징 노드 f1

[그림1]의 그래프를 유저-영화 데이터라고 하면, 같은 감독의 영화 i1, i2, i3, i4를 가상의 특징 노드인 f1로 연결한다. f1은 상호작용이 없던 i1, i2와 i3, i4를 연결하여 강하게 연관시킬 수 있다.

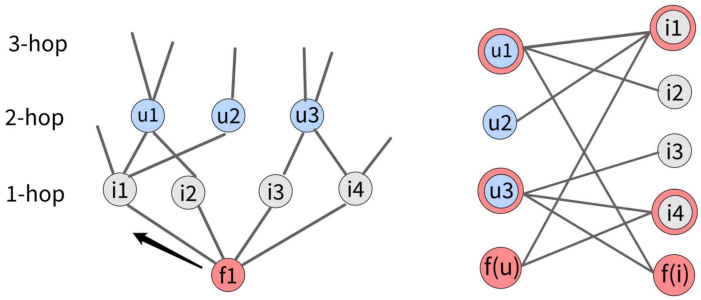
특징 노드는 노드의 특징 정보를 기반으로 생성된다. 예를 들어 아이템이 영화라면 장르(genre), 감독(director), 배우(actor), 아이템이 제품이라면 브랜드(brand), 카테고리(category)와 같은 특징 정보를 바탕으로 특징 노드를 생성할 수 있다. 이처럼 같은 특징의 아이템을 연결할 수도 있지만, 나이, 직업과 성별과 처럼 같은 특징을 가지는 유저를 연결하는 방법도 있다.

하지만 특징 노드와 해당 특징을 가진 노드를 단순히 모두 연결했을 때는 선택된 특징에 따라 문제가 있을 수 있다. movieLens latest[1] 데이터셋에서는 같은 감독의 모든 영화를 특징 노드로 연결시켰을 때 오히려 추천 성능이 낮아졌다. 원인 분석해 본 결과, 데이터셋에서 특정 감독의 영화들을 많이 시청하는 유저는 소수에 불가했다. 즉, 감독 기반의 추천이 모든 유저에게 효과적인 것이 아니다. 따라서 동일한 특징의 모든 노드를 연결하는 방식이 항상 추천의 성능을 높이지 않을 수 있다. 하지만 소수의 매니악한 유저나 중요한 아이템처럼, 해당 특징과 강하게 연관된 노드들만을 선별하여 연결시켜준다면 모든 특징을 활용할 수 있다.

2.2 특징 노드를 활용한 그래프 확장 기법

특징 노드와 연결할 노드를 선별하기 위하여 랜덤 워크(random walk)를 이용한다. 랜덤 워크는 그래프 상에서 주어진 시작 노드로부터 확률적으로 링크를 선택하여 주변 노드로 방문하는 방법이다. 노드 방문 횟수를 통해 시작 노드와 연관성이 높은 노드들을 찾아낼 수 있다. 이때 가장 많이 방문된 상위 K개의 노드를 Top-K 노드라고 하자.

일단 아이템을 연결시키는 가상의 특징 노드를 만들고, 만들어진 특징 노드로부터 랜덤 워크를 진행한다. 아이템을 연결시키는 특징 노드는 가상의 유저 타입 노드이기 때문에, [그림1]에서 확인할 수 있듯이 아이템-유저 이분 그래프 상에서는 1-hop 떨어진 노드는 아이템 타입, 2-hop 떨어진 노드는 유저 타입이다. 따라서 특징 노드로부터 랜덤워크를 진행하여 1-hop 떨어진 아이템 노드 중에서, 가장 많이 방문된 Top-K 노드는 해당 특징과 연관도가 높은 아이템 노드에 해당한다. [그림 2]에서 이를 연결하는 유저 노드를 f(u) 노드로 표현하였다. f(u)은 아이템을 연결하므로 그래프상에서는 유저 타입 노드로 삽입된다. 동일하게 특징 노드로부터 랜덤 워크를 진행하여 2-hop 떨어진 유저 노드 중에서, 가장 많이 방문된 Top-K 노드는 해당 특징과 연관도가 높은 유저 노드에 해당한다. [그림 2]에서 이를 연결하는 특징 노드를 f(i) 노드로 표현하였다. f(i)는 유저를 연결하므로 그래프 상에서는 아이템 타입 노드로 삽입된다. 모든 선별 작업이 끝나면, 랜덤 워크를 시작한 특징노드 f1은 그래프에서 제거한다. 이와 동일한 방법으로 유저의 특징을 활용해, 동일한 특징을 가지는 유저 노드를 특징 노드로 연결시키고, 랜덤워크로 선별을 진행할 수도 있다.



특징 노드로 부터 랜덤 워크 선택된 Top-K를 연결

그림 2. 랜덤 워크로 선택된 Top-K 노드를 연결

이러한 랜덤 워크 기반 노드 선별을 통해서 단순히 특징과의 연관도가 낮은 노드들을 포함한 모든 노드를 연결함으로써 발생하는 과도한 연결에서 오는 문제점을 개선할 수 있다. 이처럼 새로 생성된 특징 노드를 중심으로 그래프에 상호작용 데이터를 추가하면, 전체 그래프의 연결성이 노드의 특징 정보를 반영하는 방향으로 강화되어 기존의 그래프 구조 기반 합성곱 신경망 모델의 추천 성능이 향상될 수 있다.

3. 실험

실험 대상이 된 3가지 데이터셋의 정보는 표1) 에 정리하였다. Amazon Kindle Store와 Amazon Electornic 데이터셋은 상호작용이 10개 이상 존재하는 유저와 아이템 노드만을 선별하는 10-core-setting으로 전처리한 후 실험에 사용하였다.

표 1. 데이터셋 정보

데이터셋	#users	#items	#edges[2]	density[3]	epochs	특징
Amazon kindle store [4]	14357	15886	347392	0.000759	50	brand (작가)
Amazon Electronics[5]	20248	11590	367477	0.000725	60	category
MovieLens 1M[6]	6040	3952	1000207	0.020036	40	genre, director

표 1)의 마지막 열의 특징은 각 데이터셋 별로 가용한 아이템 특징 정보 중에서, 특징 노드를 만들기 위해 본 실험에서 사용한 특징을 나타낸다.

3.1. 실험 환경

특징 노드의 성능 향상 여부를 평가하기 위하여 본 실험에서는 추천에서 흔히 사용되는 그래프 합성곱 신경망 모델인 LightGCN[7]을 사용하였다. 학습데이터와 테스트데이터 비율은 8:2로 고정하고, 특징 노드 생성 시 훈련 데이터만을 사용하였

1) https://grouplens.org/datasets/movielens/

2) 유저-아이템 간 상호작용의 개수
3) 유저-아이템 그래프의 인접 행렬 밀집도
4) https://jmcauley.ucsd.edu/data/amazon/
5) https://jmcauley.ucsd.edu/data/amazon/
6) https://grouplens.org/datasets/movielens/
7) https://github.com/microsoft/recommenders

다. 실험 시 모두 동일하게 Nvidia A10 GPU와 Intel Xeon Gold 6348R CPU를 사용하였다.

3.2. 실험 결과

실험 결과는 모델 학습을 10번씩 진행하고, Top-20 추천 성능 점수의 평균을 낸 결과를 담았다. 실험에서 중요한 파라미터는 랜덤 워크로 선택되는 Top-K 노드의 개수(K)이다. 본 실험에서는 K를 해당 특징 노드와 연결된 노드의 개수(N)에 비례하도록 설정하였다. N개를 모두 선택하면 K는 100%이고, N개 중에서 20%만 선정한다면 K는 20%이다. 예를 들어 [그림 2]의 f1과 연결된 노드는 4개이므로 N은 4이다. 이로부터 생성되는 f(u)는 아이템을 연결하므로 유저 타입의 특징 노드이고, N(4)개 중에서 Top-2 노드를 선택하였으므로 K는 50%이다. 표 2-5의 실험 결과는 여러 비율로 K를 설정해 실험을 진행한 후, 그 중 가장 추천 성능이 높은 결과를 표시하였다.

표 2. Amazon Kindle Store - brand(작가) 대상 성능 결과

	#edges	map	NDCG	precision	recall
LightGCN	130208	0.0758	0.1393	0.0421	0.2018
With User Top 20%아이템 타입 특징 노드 K=20%	**+279**	**0.0760**	**0.1400**	**0.0424**	**0.2028**
improve	+ 0.21%	+0.26%	+ 0.50%	+ 0.71%	+ 0.49%
유저 타입 특징 노드 K=100%	**+2659**	**0.0784**	**0.1435**	**0.0431**	**0.2068**
improve	+2.04%	+3.43%	+ 3.01%	+ 2.37%	+ 2.47%

표 3. Amazon Electronics - category 대상 성능 결과

	#edges	map	NDCG	precision	recall
LightGCN	278386	0.0189	0.0406	0.0111	0.0675
아이템 타입 특징 노드 K=11.1%	**+2188**	**0.0192**	**0.0412**	**0.0112**	**0.0681**
improve	+ 0.78%	+1.58%	+ 1.47%	+ 0.90%	+ 0.88%
유저 타입 특징 노드 K=20%	**+6011**	**0.0197**	**0.0418**	**0.0113**	**0.0690**
improve	+2.15%	+4.23%	+ 2.95%	+ 1.80%	+ 2.22%

표 4. MovieLens 1M - 장르 대상 성능 결과

	#edges	map	NDCG	precision	recall
LightGCN	800194	0.1192	0.3673	0.2829	0.2431
아이템 타입 특징 노드 K=2.5%	**+297**	**0.1198**	**0.3685**	**0.2835**	**0.2439**
improve	+ 0.03%	+0.50%	+ 0.32%	+ 0.21%	+ 0.32%
유저 타입 특징 노드 K=2.0%	**+236**	**0.1197**	**0.3685**	**0.2836**	**0.2446**
improve	+ 0.02%	+0.41%	+ 0.32%	+ 0.24%	+ 0.61%

표 5. MovieLens 1M - 감독 대상 성능 결과

	#edges	map	NDCG	precision	recall
LightGCN	800194	0.1192	0.3673	0.2829	0.2431
아이템 타입 특징 노드 K=50%	**+663**	**0.1197**	**0.3681**	**0.2832**	**0.2436**
improve	+ 0.08%	+0.41%	+ 0.21%	+ 0.10%	+ 0.20%
유저 타입 특징 노드 K=20%	**+195**	**0.1194**	**0.3674**	**0.2824**	**0.2430**
improve	+ 0.02%	+0.16%	+ 0.02%	- 0.17%	- 0.04%

실험 결과, 특징 노드와 함께 추가되는 상호작용(interaction)의 비율이 높은 데이터셋인 Amazon Kindle Store나 Amazon Electronics에서는 LightGCN에서의 성능향상이 4개 성능지표에서 모두 일관되게 나타난다.

MovieLens 1M에서의 성능향상정도는 다른 두 데이터셋에 비해 낮다. 그 이유는 LightGCN에서는 인접 노드를 종합할 때 인접 노드의 개수에 따른 정규화를 진행하기 때문에, 인접 노드가 많을수록 각각의 노드로부터 받는 영향이 낮아진다. 따라서 두 가지 Amazon 데이터셋에 비해 밀도(density)가 월등히 높은 MovieLens 1M에서는 특징 노드의 영향이 낮을 수밖에 없다. 그러므로 희소한 데이터셋일수록 특징 노드를 통한 그래프 확장의 효과가 클 것으로 예상된다.

4. 결론

본 논문에서는 그래프 합성곱 신경망 모델에서 노드의 특징 정보를 모델 변경 없이 활용하기 위해 특징 노드를 활용하여 그래프를 확장하는 방법을 제안하였다. 그래프에 특징 노드 및 관련 상호작용을 추가한 그래프 상에서 LightGCN을 실행했을 때, Amazon Kindle Store와 Amazon Electronics 데이터셋은 뚜렷한 성능 향상을 보였고, 밀집도가 높은 MovieLens 1M 데이터셋에서는 성능 향상이 두드러지지는 않았다.

실험에서 아이템 특징만을 활용해 특징 노드를 생성하였지만, 나이와 성별과 같은 유저의 특징으로도 특징 노드를 생성할 수 있다. 이렇게 다양한 특징 노드를 그래프에 복합적으로 추가한다면 더 큰 성능 향상을 기대할 수 있을 것이다.

5. 참고문헌

[1] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang and Meng Wang, “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation”, Special Interest Group on Information Retrieval(SIGIR), 2020

[2] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng and Tat-Seng Chua, “Neural Graph Collaborative Filtering”, Special Interest Group on Information Retrieval(SIGIR), 2019

[3] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang and Xiuqiang He, “UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation”, Conference on Information and Knowledge Management(CIKM), 2021

[4] William L. Hamilton, Rex Ying and Jure Leskovec, “Inductive Representation Learning on Large Graph”, Neural Information Processing Systems(NIPS), 2017

[5] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton and Jure Leskovec, “Graph Convolutional Neural Networks for Web-Scale Recommender Systems”, Knowledge Discovery in Databases(KDD), pp. 974-983, 2018

[6] Carl Yang, Aditya Pal, Andrew Zhai, Nikil Pancha, Jiawei Han, Charles Rosenberg and Jure Leskovec, “MultiSage: Empowering GCN with Contextualized Multi-Embeddings on Web-Scale Multipartite Networks”, Knowledge Discovery in Databases(KDD), pp. 2434-2443, 2020