Thin Graph Convolution Network in Recommendation Systems

Sang-min Lee, Kim Heechan, Jeon Janggun and Namgi Kim
Department of Computer Science, Kyonggi University
Suwon, South Korea
[e-mail: d9249@kyonggi.ac.kr, train1212@kyonggi.ac.kr, 201812285@kyonggi.ac.kr, ngkim@kyonggi.ac.kr]
*Corresponding author: Namgi Kim

Abstract

Recommender systems help users quickly find the content they are looking for on the Internet, which is overflowing with information. A popular algorithm for efficiently implementing such RISs is the graph convolutional network (GCN) algorithm. In this work, we improve the structure of LightGCN, which is often used as a backbone network in research on GCN-based recommender systems, to show better recommendation performance, and demonstrate through experiments that the proposed ThinGCN model with WF methodology has improved accuracy over the existing LightGCN model.

***Keywords*:** Graph Convolution Network, learning method, Recommendation Systems

1. Introduction

Recommendation systems are a key tool in providing users with the information and content they want in a world of information overload. Recommendation systems analyze a user's past behavior, preferences, and interests to effectively present the most relevant information or products. Recent studies on these recommendation systems utilize convolutional networks (GCNs) to improve their performance, and in subsequent studies, the LightGCN model is often used as a backbone network, and improvements to the LightGCN model can advance the overall recommendation system research methodology. Therefore, in this paper, we propose an improvement method for combining layers of the LightGCN model.

2. Proposal methods

2.1 LightGCN Layer Combination

Equation 1 is the propagation rule for LightGCN, where $\mathcal{N}_i$ and $\mathcal{N}_u$ are the connected nodes of user u and item i, respectively. Then, a symmetric normalization term ($=1/\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}$) is performed to obtain the next embedding layer, and a weighted sum of Equation 2 is performed to obtain the final user-item embedding, and the recommendation score is calculated using the inner product of these embeddings.

$$e_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(k)};$$
$$e_i^{(k+1)} = \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{(k)} \qquad (1)$$

$$e_\mathrm{u} = \sum_{k=0}^{n} \alpha_k e_\mathrm{u}^{(k)};\ e_\mathrm{i} = \sum_{k=0}^{n} \alpha_k e_\mathrm{i}^{(k)} \qquad (2)$$

However, this propagation rule captures the loss of embedding values, and to overcome this, the Weight Forwarding(WF) methodology and the weighted summation of embeddings used by it can be omitted.

(Support note) This research was supported by a research grant from NRF(National Research Foundation)…..

2.2 Proposed ThinGCN

The proposed WF methodology scalarly multiplies the previous embedding by its weight and passes it to the next layer, as shown in Equation 3. In the process, the information from the previous layer is more strongly conveyed, allowing us to make recommendations using only the last layer's embedding, as shown in Equation 4.

$$e'^{(k+1)}_{u} = w_k \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e'^{(k)}_{i};$$
$$e'^{(k+1)}_{i} = w_k \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e'^{(k)}_{u} \quad (3)$$

$$e'_{\mathrm{u}} = e'^{(k)}_{\mathrm{u}};\ e'_{\mathrm{i}} = \mathrm{e}'^{(k)}_{\mathrm{i}} \quad (4)$$

3. Experiments

As experimental data, we used Yelp2018, Douban-book, and MovieLens-1M, which are frequently used in existing research. The composition of the dataset is shown in Table 2, and the experimental results of the existing method and the proposed initialization method are shown in Table 3.

Table. 2. Experiments data

	Yelp2018	
	Train	Train
User	31,668	31,668
Item	38,048	38,048
Interaction	1,237,259	1,237,259
Sparsity	99.90%	99.90%
	Douban-book	
	Train	Train
User	12,638	10,882
Item	22,222	19,075
Interaction	478,730	119,690
Sparsity	99.83%	99.94%
	MovieLens-1M	
	Train	Train
User	6,038	5,989
Item	3,492	3,190
Interaction	460,359	114,922
Sparsity	97.82%	99.40%

Table. 3. Experiments result

	Yelp2018		
	Weight	recall	NDCG
LightGCN	-	0.05949	0.04887
ThinGCN	8	0.06873	0.5656
	Douban-book		
	Weight	Recall	NDCG
LightGCN	-	0.14783	0.12508
ThinGCN	16	0.16701	0.14632
	MovieLens-1M		
	Weight	Recall	NDCG
LightGCN	-	0.27196	0.30274
ThinGCN	128	0.27748	0.30830

4. Conclusions

This paper, we proposed a ThinGCN model using WF methodology, and experimental results show that both recall and NDCG show performance improvement, which proves the superiority of ThinGCN utilizing only the last layer.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) Basic Science Research Program funded by the Ministry of Education (No. 2020R1A6A1A03040583) and this research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation) and this work was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0008691, HRD Program for Industrial Innovation)

References

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang *et al.*, “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation,” in *Proc. Special Interest Group in Information Retrieval (SIGIR)*, China, pp. 639–648, 2020.