Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/73/20240375

A survey of recommendation system via Graph Neural Network and collaborative filtering

Mingkai Zhu

Computer Science department China University Of Geoscience, Wuhan, 430079

1351208255@qq.com

Abstract. Recommendation systems play a vital role in today's information-rich digital environment, where users are overwhelmed with a vast number of choices and content. Because of the limitation of formal recommendation system models, Graph representation learning has become the mainstream. Among all different approaches, GNN-based CF is a state-of-art method that shows great performance. This study aims to offer a comprehensive overview of the fundamentals of GNN-based CF. Specifically, The study provides a taxonomy according to Graph Construction, Neighborhood Aggregation, Information Update, and Final Node Representation. Moreover, this survey summarizes different ways to enhancing. Meanwhile, a close classification of datasets are mentioned. This survey seeks to deepen the understanding of GNN-based CF, offering a roadmap for researchers and practitioners to navigate this exciting field and push the boundaries of recommendation system in our data-rich world.

Keywords: Recommendation System, Graph Neural Network, Survey.

1. Introduction

One popular approach to recommendation systems is collaborative filtering(CF), which leverages the wisdom of the crowd to make personalized recommendations. However, traditional collaborative filtering approaches often face challenges when dealing with sparse data, cold-start and large-scale scenarios.

In recent years, the emergence of graph neural network(GNN) has revolutionized the field of recommendation systems by incorporating the structural information and dependencies among items or users. GNN provides a powerful framework for capturing complex relationships and dependencies encoded in graph structures, enabling more effective and accurate recommendations[1]. As a result, the combination of GNN and CF is a promising area of recent research and has shown great capability[2]. This paper offers comprehensive background information and a careful classification of existing methods of GNN-based CF.

Although previous research have achieved great result, there are some limitations. For example, Zhang et al.[3,4] discuss deep-learning based recommendation system while not mention GNN. Zhou J et al.[5] review all kinds of techniques and applications in GNN but ignore the recommendation system. Ko et al.[6] summarize all recommendation system approaches, however, it only regard GNN as one approach without thorough research. Besides, although Wu et al.[2] combine GNN and recommendation system, the research in GNN-based recommendation is not sufficient.

Triggered by the rapid growth of GNN-based CF in various area and the lack of intensive and comprehensive review in the area, this paper proposes a detailed summary of baseline models and make a comprehensive classification of existing models.

The survey presents a comprehensive overview of the GNN-based Collaborative Filtering theory and its foundational models. This includes highlighting the limitations of previous CF approaches and showcasing the advantages of GNN-based CF. Additionally, the paper delves into a thorough classification of current GNN-based CF models, categorizing them based on Graph Construction, Neighborhood Aggregation, Information Update, and Final Node Representation. Meanwhile, contrastive and graph augmentation methods are discussed. Moreover, variations in datasets and feedback are also discussed to provide a complete understanding of the landscape.

2. Backgrounds and Categorization

Before getting deep into GNN-based CF. In this section, the survey briefly introduces some basic conceptions about the collaborative filtering and GNN techniques. This paper discusses the principle of GNN and some classic GNN models which lay the foundation for the follow-up study. Furthermore, it explains the advantage of utilizing GNN in recommendation system using collaborative filtering.

2.1. Collaborative Filtering

First appeared in the 1990s, Collaborative filtering(CF) is a widely used technique in the field of recommendation systems and gradually became the cornerstone for later models. CF aims to provide personalized recommendations to users based on their past preferences and the preferences of like-minded user[1,7]. With its ability to harness the collective wisdom of users, collaborative filtering has found wide application in diverse domains, revolutionizing the way personalized recommendations are generated. It also made a significant impact in e-commerce service[8,9], social network service[10,11], health care service[12] and so forth.

Broadly speaking, traditional collaborative filtering can be divided into 2 categories: memory-based CF, and model-based CF. Memory-based CF is recommended based on neighborhood information and it can be further classified into user-based CF and item-based CF.

User-based CF is a recommendation method Based on User similarity. The method first calculates the similarity between users and then predicts the target user's interest in unrated items based on the behavior and ratings of similar users. Item-based CF is a recommendation method based on Item similarity to predicts the target user's interest in the unrated items. Common similarity calculation methods include cosine similarity, Pearson correlation coefficient, jaccard similarity, KNN [13,14]. In other words, memory-based CF uses either user or item similarity to make recommendations.

Memory-based CF is simple to implement and effective, Amazon quickly deployed it as it ensures less search effort for the user, bringing higher sale and stronger loyalty for the e-commerce company[15,16,17]. However, there are several main problems with the technique:

Data Sparsity: It refers to the scenario where the available user-item interaction data is sparse, meaning that most users have only interacted with a small subset of items or vice versa[18].

Cold Start: When new users or items join the system, there is limited or no historical interaction data available for them. This makes it challenging to generate accurate recommendations for these users or items[19].

Scalability: It refers to the difficulty of efficiently handling large amounts of data and executing complex computations in recommendation systems, especially as the user and item populations grow[15].

Some model-based techniques can alleviate these problems, for instance, in order to solve the sparsity problem, N. Srebro et al. proposed Maximum-Margin Matrix Factorization(MMMF)[20,21], By enforcing a margin, MMMF encourages a more discriminative and robust model that can make accurate predictions even for sparsely observed pairs. Other methods such as clustering CF algorithms[22] address the scalability problem. These methods can solve part of the problems above but they suffer from either a trade-off between prediction performance and scalability or the loss of important information[13].

Then, deep-learning-based CF was developed, the survey will not make a subtle distinction between all the models, instead, the focus is GNN-based CF, the mainstream of the current CF model.

In summary, the relations are described in figure1, which concludes the classification of CF.

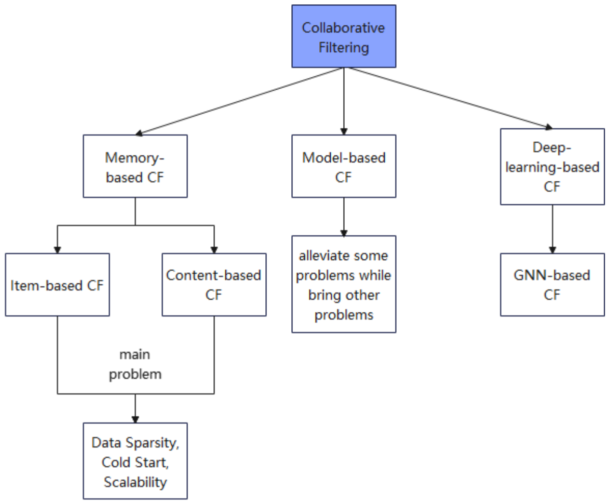

Figure 1. classification of CF models

2.2. Graph Neural Network

Graphs are mathematical structures that consist of nodes (representing entities) and edges (representing relationships or connections between nodes). The main idea of Graph Neural Networks (GNN) is to extend traditional neural network architectures to handle data structured as graphs. GNN leverage this graph structure to perform machine-learning tasks, such as node classification, link prediction, and graph classification.

The fundamental concept behind GNN is to learn node representations by aggregating and propagating information from neighboring nodes in the graph. GNN iteratively updates the node representations by incorporating information from local neighborhoods, gradually capturing complex relationships and patterns in the graph by getting low-dimensional vector embeddings of nodes[23,24].

Here, the paper introduces 3 classic GNN models from the perspective of aggregation and update method. All other GNN-based CF can be seen as the variant of these models.

GCN[25]: a semi-supervised learning method that consider a multi-layer Graph Convolutional Network (GCN) with the following layer-wise propagation rule:

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

where $\tilde{A}$ denotes A+I, I is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $W^{(l)}$is a layer-specific trainable weight matrix, $H^{(l)} \in R^{N\times D}$denotes the feature matrix in each layer. σ represents the non-linear function, which can be RELU, Softmax.

Aggregation method: get the average of all neighbors' feature vector(including itself).

Update method: Multiply the weight matrix and activate using the activation function.

GraphSAGE[26]: GraphSAGE introduces a neighborhood sampling strategy. Instead of considering all neighbors, GraphSAGE samples a fixed-size subset of neighbors for each node during each training iteration.

$$h^k_{N(v)} \leftarrow \ \mathrm{AGGREGATE}_K(\{h^{k-1}_u, \forall u \in N(v)\}) \tag{2}$$

Aggregation method: AGGREGATEK can be one of the three candidates:

Mean-pooling:

$$h_v^k \leftarrow \sigma(W \cdot \mathrm{MEAN}(\{h_u^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\}). \tag{3}$$

Max-pooling:

$$\mathrm{AGGREGATE}_k^{pool} = \max(\{\sigma(W_{pool}\, h_{u_i}^k + b), \forall u_i \in N(v)\}) \tag{4}$$

LSTM: larger expressive capability but not inherently symmetric.

Update method:

$$h_v^k \leftarrow \sigma(W^k \cdot \mathrm{CONCAT}(h_v^{k-1}, h_{N(v)}^k) \tag{5}$$

GAT[27]:
Firstly, calculate the Attention coefficient α_{ij}.

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakyReLU}(\vec{h_i} \| W\vec{h_j}]))}{\sum_{k \in N_i} \exp(\mathrm{LeakyReLU}(\vec{h_i} \| W\vec{h_k}]))} \tag{6}$$

Then use the coefficient to calculate the feature embedding of the next layer.

$$\vec{h}_i' = \|_{k=1}^K \sigma(\sum_{j \in N_i'} \alpha_{ij}^k W^k \vec{h_j}) \tag{7}$$

Aggregation method: Weighted average(using attention).
Update method: the same as GCN.

2.3. *Why GNN For Collaborative Filtering*

There are several major advantage for using GNN in CF, essentially, most of the information in the real world can be represented by graph and the closed connection between user, item, context can be learned through GNN. Firstly, the high performance motivates us to use GNN for CF [28,5]. Meanwhile, GNN are effective at capturing both explicit and implicit collaborative signals present in the user-item interaction graph even in complex graph structure[29] and the method has shown great potential in handling scalability problem. For example, R Ying et al. developed PinSage[30], the model is able to recommend in web-scale situation containing over 3 billion nodes and 18 billion edges. Moreover, data sparsity problem can be alleviated through the method[31].

3. Classification of GNN on CF

Here, this survey uses GCMC to illustrate the process of implementing GNN on CF and it also discuss the contrastive learning and graph enhancing methods that are used in existing models. Moreover, a classification of datasets is offered and the difference of implicit feedback and explicit feedback is illustrated.

3.1. *How To Implement GNN On Cf*

Basically, according to S Wu,et al.[32] there are four steps to implement GNN on cf, which is Graph Construction, Neighborhood Aggregation, Information Update, and Final Node Representation. Here, the paper introduces a classic model GCMC[33] as an example and makes comparison with other proposed models to illustrate the basic steps of GNN-based implementation.

Graph Construction: GNN-based cf treat user-item interaction as bipartite graph, where take user and item as node and their interaction as edge[34]. There are two kinds of graph construction, the first method is using direct original bipartite graph as input[34,35,36,37,38,39,40], the other way is adding extra information(adding either node or edge) to learn lament information and deal with cold start problem[41,42]. Meanwhile, in order to handle scalability problem, some models use sampling strategy such as random sampling[34,41] or random-walk sampling[31] to choose a subset from original graph information, making the model more scalable. GCMC uses the original bipartite graph to construct a graph.

Neighborhood Aggregation: Followed the idea of Kipf et al.[43] and Tian et al.[,44], GCMC revisit a graph encoder to efficiently use of location and edge information using weight sharing[45,46,47].

The graph convolution can be seen message passing from item j to user i is defined as(message passing from user i to item j can be represented in the same way):

$$\mu_{j\to i,r} = \frac{1}{c_{ij}} W_r x_j^v \tag{8}$$

Aggregation：accum() denotes an accumulation operation, this is a formal kind of neighborhood aggregation, which treats all the neighbors in the same way[33,36,40,48]

$$n_u^{(l)} = \mathrm{accum}\left(\sum_{j\in N_i(u_i)} \mu_{j\to i,1}, \dots \sum_{j\in N_R(u_i)} \mu_{j\to i,R}\right) \tag{9}$$

There are also other ways to aggregate neighborhood information, for example, insipired by GraphSage[26], PinSage[31] assign weight to each neighbor based on graph structure.

Information Update: in this model, the activation function σ is used to update the information, the formal idea of GCN contains σ and $w^{(l)}$ to do Non-liner transformation.

$$h_i^u = \sigma(n_u^{(l)}) \tag{10}$$

However, LR-GCCF[35] and LightGCN[36] states that liner transformation which simplify the network can lead to bett2er overall performance. Other models bring the node itself into account[38,49,50], which grab the intrinsic information of the node.

Final Node Representation: GCMC uses the representation of the last layer specifically and adopts non-liner transformation to get the final node embedding[30,37,39,48,51] :

$$z_i^u = \sigma(Wh_i^u) \tag{11}$$

Other models consider the importance of other layers, they use methods such as concatenating every layer[35,40,49] or calculating the weighted sum of each layer[42,51].

3.2. Contrastive Learning And Graph Enhancing

3.2.1. Contrastive Learning

Contrastive learning is a self-supervised learning method that aims to maximize the agreement between similar instances while minimizing the agreement between dissimilar instances. Typically, surveys use graph enhancing to generate multiple views and regard views generated by same node as similar instances[52]. The method is able to learn internal information among nodes as it extracts extra information from unlabeled data.

3.2.2. Graph Enhancing

Graph enhancing is a crucial method in GNN-based CF, which reduce the dependence on specific node/edge thus making the trained network more generalize. Also, it turns the unsupervised learning to supervised learning through generating multiple views. There are a few ways to do the graph enhancing:

Node Dropout: node dropout refers to randomly "dropping out" (i.e., temporarily disabling) some of the neurons or nodes in a neural network during training. This is done by randomly setting the outputs of the nodes to zero with a certain probability for each training sample. Many models adopt node dropout[33,49,52].

Edge Dropout: In edge dropout, individual edges (or connections) in the graph are randomly dropped during training with a certain probability[52,53].

Random Walk: random walk typically refers to the process of starting at a specific node in a graph and taking steps to neighboring nodes based on a stochastic rule[30,52].

DOI: 10.54254/2755-2721/73/20240375

However, J Yu et al.[54] using ablation study to compare node-dropout, edge-dropout and random walk with a graph-augmentation-free method. The result shows that graph augmentation are not necessary. Meanwhile, these methods are time-consuming and can even result in the loss of some valuable information. Instead, he proposes an effective simple graph contrastive learning method , which is able to regulate the uniformity in a smooth way.

3.3. Datasets

In table2, the author summarizes all the datasets that are used in GNN-based collaborative filtering.

Explicit feedback refers to direct and deliberate user feedback(i.e, user's rating on movies), and implicit feedback refers to undirect user feedback(i.e, click,view, time spenton an item). The explicit feedback can provide a straightforward interpretation of userintent. However, the kind of feedback may also suffer from data sparsity challenges asa lot of users will only provide ratings for a small subset of items. On the other hand, Implicit feedback is often collected passively, making its collection easier and less intrusive and since implicit signals are usually more abundant, the data sparsity challenges can be alleviated. However, since the feedback does not directly convey user preferences and can be influenced by factors other than user interest, it is difficult to interpret the data.

Table1. Datasets with different models.

Datasets	Information type	Model
MovieLens-100k	Explicit feedback	GCMC[33],STAR-GCN[48],MCCF[55],IGMC[56],AGCN[50],DHCF[57], DGCF[42]
MovieLens-1M	Explicit feedback	GCMC[33],SpectralCF[41],STAR-GCN[48],DGCN-BinCF[31],IGMC[56], HashGNN[39],DGCF[42],SMOG-CF[58]
MovieLens-10M	Explicit feedback	GCMC[33],DGCN-BinCF[31]
Amazon	Implicit feedback	SpectralCF[40],NGCF[49],Multi-GCCF[41],MCCF[55],LR-GCCF[35], LightGCN[36],AGCN[50],NIA-GCN[38],Disenhan[51],DGCF[42],GCM[59],SMOG-CF[58]
Gowalla	Implicit feedback	NGCF[49],Multi-GCCF[41],MCCF[55],LR-GCCF[35],LightGCN[36], Disenhan[51],HashGNN[39],DGCF[42],GraphSAIL[60],SMOG-CF[58],SGL-ED[52]
Yelp	Implicit feedback	DGCN-BinCF[31],NGCF[49],LightGCN[36],NIA-GCN[38],Disenhan[51], GCM[59],SGL-ED[52]
HetRec	Explicit feedback	SpectralCF[40]
Douban	Explicit feedback	STAR-GCN[48],IGMC[56]

3.4. Summary And Future Work

Inspired by Ko H et al.[6], the paper makes a classification based on four main issue: Graph Construction, Neighborhood Aggregation, Information Update, and Final Node Representation. The main innovation is that the survey focus specifically on GNN-based CF models, a potential area of research with no thorough study. Also, new taxonomy regarding contrastive learning and graph augmentation and different feedback are proposed to provide a full understanding of the method and the current research trend.

GNN-based CF are often viewed as black-box models, making it challenging to explain and interpret the recommendations. Future work can focus on developing explainable GNN models that provide transparent justifications or evidence for the recommendations. This can involve designing attention mechanisms, interpretability modules, or graph visualization techniques to enhance transparency and trust in GNN-based CF.

4. Conclusion

Recently, Graph neural networks have become an efficient and powerful tool for collaborative filtering. Inspired by the superiority, this paper introduces preliminaries behind GNN-based CF such as different categorizations of CF and current challenges and how GNN-based CF helps to alleviate these problems. Then, the paper explicitly discusses how to implement that model to make recommendations regarding four main steps. Moreover, the paper has a close discussion of contrastive learning and graph augmentation. Finally, a summation of all models and their datasets are proposed. The survey reviews the context of previous research about GNN-based CF and it also stands a chance to shed some light on future research.

References

[1] Goldberg, David, et al. "Using collaborative filtering to weave an information tapestry."Communications of the ACM35.12 (1992): 61-70.
[2] Wu S, Sun F, Zhang W, et al. Graph neural networks in recommender systems: a survey[J]. ACM Computing Surveys, 2022, 55(5): 1-37.
[3] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM computing surveys (CSUR), 2019, 52(1): 1-38.
[4] Batmaz Z, Yurekli A, Bilge A, et al. A review on deep learning for recommender systems: challenges and remedies[J]. Artificial Intelligence Review, 2019, 52: 1-37.
[5] Zhou J, Cui G, Hu S, et al. Graph neural networks: A review of methods and applications[J]. AI open, 2020, 1: 57-81.
[6] Ko, Hyeyoung, et al. "A survey of recommendation systems: recommendation models, techniques, and application fields."Electronics11.1 (2022): 141.
[7] Resnick, Paul, et al. "Grouplens: An open architecture for collaborative filtering of netnews."Proceedings of the 1994 ACM conference on Computer supported cooperative work. 1994.
[8] Huang, Zan, Daniel Zeng, and Hsinchun Chen. "A comparison of collaborative-filtering recommendation algorithms for e-commerce."IEEE Intelligent Systems22.5 (2007): 68-78.
[9] Jiang, Liaoliang, et al. "A trust-based collaborative filtering algorithm for E-commerce recommendation system."Journal of ambient intelligence and humanized computing10 (2019): 3023-3034.
[10] Kautz, Henry, Bart Selman, and Mehul Shah. "Referral Web: combining social networks and collaborative filtering."Communications of the ACM40.3 (1997): 63-65.
[11] Groh, Georg, and Christian Ehmig. "Recommendations in taste related domains: collaborative filtering vs. social filtering."Proceedings of the 2007 ACM International Conference on Supporting Group Work. 2007.
[12] Kaur, Harmanjeet, Neeraj Kumar, and Shalini Batra. "An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system."Future Generation Computer Systems86 (2018): 297-307.
[13] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques."Advances in artificial intelligence2009 (2009).
[14] Nilashi, Mehrbakhsh, Othman Ibrahim, and Karamollah Bagherifard. "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques."Expert Systems with Applications92 (2018): 507-520.
[15] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering."IEEE Internet computing7.1 (2003): 76-80.
[16] Hofmann, Thomas. "Latent semantic models for collaborative filtering."ACM Transactions on Information Systems (TOIS)22.1 (2004): 89-115.
[17] Ansari, Asim, Skander Essegaier, and Rajeev Kohli. "Internet recommendation systems." (2000): 363-375.
[18] Papagelis, Manos, Dimitris Plexousakis, and Themistoklis Kutsuras. "Alleviating the sparsity problem of collaborative filtering using trust inferences."International conference on trust management. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
[19] Wei, Jian, et al. "Collaborative filtering and deep learning based recommendation system for cold start items."Expert Systems with Applications69 (2017): 29-39.

DOI: 10.54254/2755-2721/73/20240375

[20] Srebro, Nathan, Jason Rennie, and T. Jaakkola. "Maximum-margin matrix factorization, in ‘Advances in Neural Information Processing Systems 17’." (2005): 1329-1336.

[21] Rennie, Jasson DM, and Nathan Srebro. "Fast maximum margin matrix factorization for collaborative prediction."Proceedings of the 22nd international conference on Machine learning. 2005.

[22] Chee, Sonny Han Seng, Jiawei Han, and Ke Wang. "Rectree: An efficient collaborative filtering method."Data Warehousing and Knowledge Discovery: Third International Conference, DaWaK 2001 Munich, Germany, September 5–7, 2001 Proceedings 3. Springer Berlin Heidelberg, 2001.

[23] Cao, Shaosheng, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information."Proceedings of the 24th ACM international on conference on information and knowledge management. 2015.

[24] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks."Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.

[25] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks."arXiv preprint arXiv:1609.02907(2016).

[26] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs."Advances in neural information processing systems30 (2017).

[27] Veličković, Petar, et al. "Graph attention networks."arXiv preprint arXiv:1710.10903(2017).

[28] Guo Q, Qiu X, Xue X, et al. Syntax-guided text generation via graph neural network[J]. Science China Information Sciences, 2021, 64: 1-10.

[29] Wang S, Hu L, Wang Y, et al. Graph learning based recommender systems: A review[J]. arXiv preprint arXiv:2105.06339, 2021.

[30] Ying, Rex, et al. "Graph convolutional neural networks for web-scale recommender systems."Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

[31] Wang, Haoyu, Defu Lian, and Yong Ge. "Binarized collaborative filtering with distilling graph convolutional networks."arXiv preprint arXiv:1906.01829(2019).

[32] Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey."ACM Computing Surveys55.5 (2022): 1-37.

[33] Berg, Rianne van den, Thomas N. Kipf, and Max Welling. "Graph convolutional matrix completion."arXiv preprint arXiv:1706.02263(2017).

[34] Wu, Ziteng, et al. "A review of recommendation system research based on bipartite graph."MATEC Web of Conferences. Vol. 336. EDP Sciences, 2021.

[35] Chen, Lei, et al. "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach."Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 01. 2020.

[36] He, Xiangnan, et al. "Lightgcn: Simplifying and powering graph convolution network for recommendation."Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.

[37] Li, Chong, et al. "Hierarchical Representation Learning for Bipartite Graphs."IJCAI. Vol. 19. 2019.

[38] Sun, Jianing, et al. "Neighbor interaction aware graph convolution networks for recommendation."Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020.

[39] Tan, Qiaoyu, et al. "Learning to hash with graph neural networks for recommender systems."Proceedings of The Web Conference 2020. 2020.

[40] Zheng, Lei, et al. "Spectral collaborative filtering."Proceedings of the 12th ACM conference on recommender systems. 2018.

[41] Sun, Jianing, et al. "Multi-graph convolution collaborative filtering."2019 IEEE international conference on data mining (ICDM). IEEE, 2019.
[42] Liu, Zhiwei, et al. "Deoscillated graph collaborative filtering."arXiv preprint arXiv:2011.02100(2020).
[43] Kipf, T. N., and M. Welling. "Variational graph auto-encoders. NIPS workshop on Bayesian deep learning." (2016)
[44] Tian F, Gao B, Cui Q, et al. Learning deep representations for graph clustering[C]//Proceedings of the AAAI conference on artificial intelligence. 2014, 28(1).
[45] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203, 2013.
[46] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[J]. Advances in neural information processing systems, 2016, 29.
[47] Duvenaud D K, Maclaurin D, Iparraguirre J, et al. Convolutional networks on graphs for learning molecular fingerprints[J]. Advances in neural information processing systems, 2015, 28.
[48] Zhang, Jiani, et al. "Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems."arXiv preprint arXiv:1905.13129(2019).
[49] Wang, Xiang, et al. "Neural graph collaborative filtering."Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 2019.
[50] Wu, Le, et al. "Joint item recommendation and attribute inference: An adaptive graph convolutional network approach."Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.
[51] Wang, Yifan, et al. "Disenhan: Disentangled heterogeneous graph attention network for recommendation."Proceedings of the 29th ACM international conference on information & knowledge management. 2020.
[52] Wu, Jiancan, et al. "Self-supervised graph learning for recommendation."Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021.
[53] Spinelli I, Scardapane S, Hussain A, et al. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning[J]. IEEE Transactions on Artificial Intelligence, 2021, 3(3): 344-354.
[54] Yu J, Yin H, Xia X, et al. Are graph augmentations necessary? simple graph contrastive learning for recommendation[C]//Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 2022: 1294-1303.
[55] Wang, Xiao, et al. "Multi-component graph convolutional collaborative filtering."Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 04. 2020.
[56] Zhang, Muhan, and Yixin Chen. "Inductive matrix completion based on graph neural networks."arXiv preprint arXiv:1904.12058(2019).
[57] Ji, Shuyi, et al. "Dual channel hypergraph collaborative filtering."Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020.
[58] Zhang, Hengrui, and Julian McAuley. "Stacked mixed-order graph convolutional networks for collaborative filtering."Proceedings of the 2020 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2020.
[59] Wu, Jiancan, et al. "Graph convolution machine for context-aware recommender system."Frontiers of Computer Science16.6 (2022): 166614.