A review of recommendation system research based on bipartite graph

Ziteng Wu *, *Chengyun* Song, Yunqing Chen, and *Lingxuan* Li

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China

Abstract. The interaction history between users and items is usually stored and displayed in the form of bipartite graphs. Neural network recommendation based on the user-item bipartite graph has a significant effect on alleviating the long-standing data sparseness and cold start of the recommendation system. The whole paper is based on the bipartite graph. An review of the recommendation system of graphs summarizes the three characteristics of graph neural network processing bipartite graph data in the recommendation field: interchangeability, Multi-hop transportability, and strong interpretability. The biggest contribution of the full paper is that it summarizes the general framework of graph neural network processing bipartite graph recommendation from the models with the best recommendation effect in the past three years: embedding layer, propagation update layer, and prediction layer. Although there are subtle differences between different models, they are all this framework can be applied, and different models can be regarded as variants of this general model, that is, other models are fine-tuned on the basis of this framework. At the end of the paper, the latest research progress is introduced, and the main challenges and research priorities that will be faced in the future are pointed out.

1 Introduction

With the rapid development of information network technology, data information has exponentially increased [1]. If users want to dig out effective information from a large amount of information, they need to use recommendation tools. Recommendation technology is an effective method of information screening. The above alleviated the problem of data overload [2]. The core of the recommendation system is the recommendation algorithm, which constructs a preference model by analyzing user behavior information, user portraits, and item attributes, and pushes the items that best match the user's interest in the network to users, so that users can get rid of being surrounded by spam. The dilemma of not being able to find the target data increases the user's dependence and experience. At present, the application of the recommendation system is reflected in all aspects of life, Taobao's guess you like it, Douyin's video

* Corresponding author: 674606550@qq.com

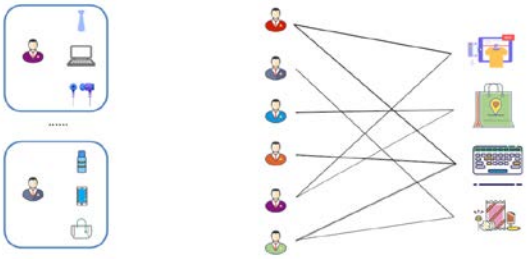
recommendation, QQ Music's daily playlist, Weibo's hot search list and WeChat look, etc. It can be said that life is more colorful due to the existence of the recommendation system.

Data in the field of machine learning is generally divided into European data (text, image, audio and video, etc.) and non-European data (manifold, node graph, molecular structure graph, cell graph, etc.) as shown in Figure 1. A graph is one of the basic data structures, usually represented by G=(V, E), that is, a graph is composed of a node and its adjacent edges, and most phenomena or scenarios can be used to capture the special relationship in the graph [3].There are a large number of user-item interaction lists in the recommendation field. These interaction histories can form a huge bipartite graph of network topology (Figure 2). The bipartite graph intuitively expresses the connection between users and items, and how to use user-item 2 It has become a research hotspot to bring more benefits to businesses.

(a) European space (b) Non-european space

Fig. 1. Examples of European and non-European data.

The strong learning ability of deep learning technology in images and text has attracted more and more researchers to apply deep learning methods to graph data, learn feature extraction and representation of graph structures, and graph neural networks (GNN) It came into being [4]. Graph neural network is a deep learning-based method running on the graph domain, which makes up for the problem that traditional deep models cannot generalize to graph data. The development of graph neural network enables node information and node-to-node relationship information in the recommendation system to be fully mined, bringing greater commercial value.



(a) List of User-item interactions (b) Bipartite graph

Fig. 2. User-item bipartite graph.

2 Introduction to recommendation system development

The recommendation system, as the user's preferred filter, is constantly updated with people's needs and technological development. The development of the recommendation system has experienced three generations of technical improvements.

The first generation is traditional recommendation technology. Among them, the recommendation based on collaborative filtering [5] is the most widely used in traditional recommendation. The item that best matches the user's explicit preference is recommended

to the user. This has a defect that the user's explicit preference matrix has a high dimension but a very large distribution. Sparse, cold start problems exist for new users or new projects.

The second generation is a recommendation technology based on a deep model. Many studies use deep learning techniques to complete recommendation tasks [6]. The recommendation system based on deep learning can already solve the problem of data sparseness, but the cold start problem of the recommendation system is still not effectively solved. In order to get more connections with new users or new projects, try to add auxiliary information to the relationship between users and projects, establish the relationship between projects and projects, etc. The topological bipartite graph structure data can naturally represent users and projects The relationship between time, project and project, user and user, even after adding auxiliary information, the graph structure can still be connected. Therefore, in the field of recommendation systems, industry and academia are increasingly inclined to use graph data.

The third generation is a recommendation system based on graph neural networks with continuous achievements in recent years. Graph neural network (GNN) is a general deep learning framework defined specifically for processing bipartite graph data structures (Figure 2), through Perform end-to-end recommendation modeling on graph data, learn more deep features of nodes, edges, and subgraphs [7], provide full scoring grades for those unobserved cross-complementarity, and then predict users' future behavior.

Graph neural network recommendation mainly solves the following problems in deep method recommendation:

- Node sensitivity, small differences in node input order have a great impact on the output of the deep model.
- Blocking the transfer of neighbor information, traditional deep models often fail to learn the high-level neighbor information of nodes when learning on the interactive graph.
- Weak interpretability. The traditional depth model faces the intuitive interactive graph structure and cannot be used for graph-based explanation and reasoning.

When the graph neural network processes two-part graph input, the input order of the nodes will not affect the output, so it is not necessary to consider the input order of the nodes in the propagation on each node. In addition, the graph neural network uses the edges between nodes to assist in propagation, aggregates the neighbor state and updates the hidden state of the current node, so that the structural information is integrated into the node representation to learn the hidden representation, which is expected to solve the problem in the recommendation system Sparsity issues. Three unique properties of graph neural network bipartite graph recommendation: interchangeability, that is, the model is not sensitive to the input order of nodes; multi-hop transferability, that is, long-distance multi-hop neighbors can also spread information layer by layer to the target node , At the same time, it can also promote the representation learning of inactive nodes; strong interpretability, that is, multi-layer communication on the user-item interaction graph can provide a basis for recommendation.

In the past three years, the optimal models recommended by bipartite graphs, NGCF[8], PUP[9], Bipar-GCN[10], have all been done in an end-to-end manner by encoding cooperative signals on bipartite graphs. To generate high-quality embeddings of users and items, although the entire neural information transmission framework is not the same, the model can be summarized as a three-layer general neural information transmission framework(Figure 3): embedding layer, propagation update layer and prediction layer.

2.1 Embedding layer

The embedding layer focuses on how to obtain the interaction graph and a set of node features, and uses this information to generate node embeddings, thereby generating

embeddings for the subgraph and the entire graph. The role of this layer is to learn a low-dimensional vector representation for the input of the graph neural network model. In the early recommendation system, the most widely used to extract user and item features is matrix factorization (MF), such as probability matrix factorization (PMF), bias matrix factorization (BiasedMF), neural network matrix factorization (NNMF) and so on. However, these methods have great shortcomings. Each decomposition costs high time complexity and space complexity, and cannot fully explore the implicit relationship between users and items, which leads to unsatisfactory recommendation effects. With the DeepWalk algorithm for the first time to introduce deep methods into the field of network representation learning, then embedding methods based on deep learning emerge in endlessly.

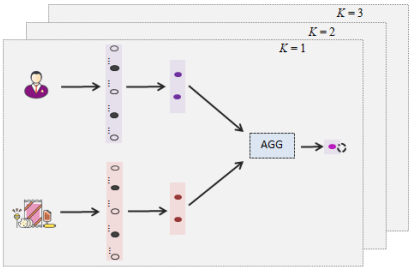


Fig. 3. Graph neural network recommendation general framework.

The simplest way is to use the shallow embedding method to generate a unique low-dimensional vector representation of the node, that is, in the initial step, each user and item is associated with an embedded ID, which is defined as the user ID embedding Means, similarly, is defined as the ID embedding of the project. The advantage of this design is that the model is interchangeable and not deformed. The invariance of interchangeability is that the model does not depend on the arbitrary order of rows/columns in the adjacency matrix, which makes up for the defect that the depth model is highly sensitive to the order of node input.

2.2 Propagation update layer

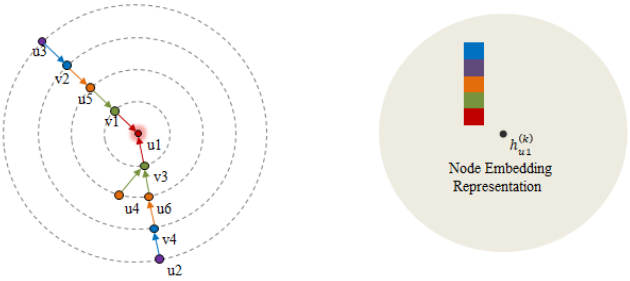
The propagation update layer is the core component of the graph neural network recommendation model, which includes two stages: neighbor node information transfer, aggregation and update (Figure 4). The embedding layer solves the problem that the graph data is difficult to efficiently input the recommendation algorithm. The next step is to solve the problem of the deep model message transmission path being blocked and the information dissemination problem of long-distance nodes [11], and then the neighbor information can be Together, it provides a feasible idea for solving the problem of data sparseness in the recommendation system. In each message passing iteration process, the hidden embedding corresponding to each node u is updated according to the information aggregated from the graph neighborhood N(u) of the user u, and the propagation update layer can be abstractly defined as:

$$h_u^{(k+1)} = UPDATA^{(k)}(h_u^{(k)}, AGGREGATE^{(k)}(\{h_v^{(k)}, \forall v \in N(u)\})) \quad (1)$$

$$= UPDATE^{(k)}(h_u^{(k)}, m_{N(u)}^{(k)})$$

$$m_{N(u)}^{(k)} = AGGREGATE^{(k)}(\{h_v^{(k)}, \forall v \in N(u)\}) \tag{2}$$

Among them, k represents the number of layers, $h_u^{(k)}$ and $h_v^{(k)}$ respectively represent the embedding representations of user u and item v after propagation in the k layer; N(u) represents the set of items adjacent to user u; UPDATE and AGGREGATE are arbitrary differentiable functions (neural network); $m_{N(u)}^{(k)}$ is a message aggregated from the graph neighborhood N(u) of u, using the superscript k to distinguish the embedding and function of message passing in different iterations.



(a) Information propagation (b) Embedding space

Fig. 4. Multi-hop propagation update layer.

According to the different information dissemination mechanism, it can be divided into: convolution aggregator, gate aggregator and attention aggregator.

2.2.1 Convolutional aggregator

The graph convolutional aggregator[12] uses the aggregation of the embedding representation of the central node and the embedding representation of its neighboring nodes as the new representation of the central node, and iteratively propagates and aggregates the embedded feature information from the node neighborhood, and only one convolution operation Transform and aggregate the embedding representation in the neighborhood of the first-order graph, spread the information of the distant multi-level neighborhood through the stacked multi-layer graph convolutional network, and finally update the embedding representation of the current node. Literature [13] proposed a general inductive framework GraphSAGE, which proposed a small batch aggregation algorithm, which first sampled the features in the node neighborhood, but did not use the entire set of neighbors for one-time update, but through unified Sample a fixed-size domain set to update the embedding representation of the current node, and use the updated final state for prediction and back propagation. The recommendation system of the PinSage framework [14] uses local graph convolution operations to aggregate information from node neighborhoods. Graph convolution aggregation can be abstractly defined as:

$$h_u^{(k)} = \sigma\left(W_{self}^{(k)} h_u^{(k-1)} + W_{neigh}^{(k)} \sum_{v \in N(u)} h_v^{(k-1)} + b^{(k)}\right) \tag{3}$$

$$m_{N(u)} = \sum_{v \in N(u)} h_v \tag{4}$$

$$\mathrm{h}_u^{(k+1)} = UPDATE(h_u, m_{N(u)}) = \sigma(W_{self} h_u + W_{neigh} m_{N(u)}) \tag{5}$$

Among them, $W_{self}^{(k)}, W_{neigh}^{(k)} \in R^{d(k) \times d(k-1)}$, W_{self} and W_{neigh} are trainable weight matrices that are used to perform feature conversion in each layer; $b^{(k)} \in R^{d(k)}$ are bias terms, but for simplicity, they are usually omitted; σ represents nonlinear activation functions, and superscript k is used to distinguish different layers of GNN Parameters, embeddings, and dimensions.

2.2.2 Gate aggregator

The classic long and short-term memory cyclic neural network system has been proven to be very effective for learning the dynamic characteristics of sequence data, among which the input and output gate mechanisms in GRU and LSTM can maintain long-term internal states. Gated Graph Neural Network (GGNN)[15] uses a gate recursive unit in the propagation process to update the hidden state of the aggregation node with a fixed number of steps. The gate aggregation can be abstractly defined as:

$$h_u^{(k)} = GRU(h_u^{(k-1)}, m_{N(u)}^{(k)}) \tag{6}$$

2.2.3 Attention aggregator

The attention mechanism has almost become a standard based on sequence tasks, and has been successfully applied to machine translation, and machine reading. The attention mechanism can distinguish the importance of different nodes. For example, in e-commerce recommendations, recently purchased products have more reference value than those purchased a few months ago; in social recommendations, others that interact with the target user the most Users naturally have a greater influence on the behavior of target users. Incorporating the attention mechanism in the propagation process can assign different attention weights to different neighbors when aggregating neighbor embedding representations, and then focus on aggregation according to the attention of each neighbor [16], and finally update the hidden state information In order to match users with high-quality recommendations. The attention aggregator can be abstractly defined as:

$$m_{N(u)} = \sum_{v \in N(u)} \alpha_{u,v} h_v \tag{7}$$

$$\alpha_{u,v} = \frac{\exp(a^T [W h_u \oplus W h_v])}{\sum_{v' \in N(u)} \exp(a^T [W h_u \oplus W h_{v'}])} \tag{8}$$

Among them, $\alpha_{u,v}$ represents the different weights of neighbor v when aggregating information at the user u, is a trainable attention vector, and W is a trainable weight matrix, $\oplus$ represents a splicing operation.

2.3 Prediction layer

After multi-layer propagation, the hidden state obtained from each layer is spliced (||) as the final feature of the user or item. In addition to splicing, other combination methods, such as weighted average, maximum pool, and LSTM, can also be applied. Studies have proved that the use of splicing is the simplest, because it does not involve other parameters to be learned. Finally, calculate the inner product between the user and the item to predict the user's future behavior:

$$h_u^* = h_u^{(0)} || ... || h_u^{(k)}, h_v^* = h_v^{(0)} || ... || h_v^{(k)} \tag{9}$$

$$\hat{f}(u,v) = h_u^{*T} h_v^* \tag{10}$$

$$L = \sum_{(u,v,v') \in G'} -ln\sigma(\hat{f}(u,v) - \hat{f}(u,v')) + \lambda ||\Theta||_2^2 \tag{11}$$

Among them, $G' = \{(u,v,v') | (u,v) \in R^+, (u,v') \in R^-\}$, R^+ represents the positive interaction between u and v, R^- represents the negative interaction between u and v; h_u^*, h_v^* respectively represents the user's ultimate embedding and the item's ultimate embedding; Θ is the parameter set of the model, often used to prevent overfitting; λ is the L2 regularization parameter.

3 Conclusion and future work

In this paper, we summarize the graph neural network Recommended general framework: embedding layer, propagation update layer and prediction layer. This general framework not only inherits the powerful learning capabilities of deep models, but also has the unique interchangeability, multi-hop transmission and strong interpretability of processing topological data; With the emergence of key knowledge graph technologies such as knowledge representation learning and knowledge path reasoning, there will be many challenges in using knowledge graphs to assist recommendation in the future, and it is also a major research focus.

In recent years, there have been endless researches on the application of graph neural networks in the field of recommendation systems, which is expected to solve the long-standing sparsity problem of recommendation systems. Based on the above discussion, it can be seen that it is not enough to use the information of the bipartite graph itself to improve the expressive ability of the model. The KGAT model proposed by Wang et al. [17] recently introduced knowledge graphs to help recommendation. It can be said that the KGAT model has created a precedent for the joint recommendation of the two-part collaborative knowledge map. In addition to combining the knowledge map with the recommendation, there are many other external structural information in the real world that can assist the recommendation. This direction It is worth exploring. Another point that can be studied in the future is to use graph structure for path reasoning. The KARN model proposed by Cao et al. [18] and the SCPR model proposed by Lei et al. [19] both perform recommendations by reasoning about the path between users and items. From recent research results, it can be seen that the path reasoning recommendation based on graph neural network is still in its infancy, and more extensive attempts are needed in the future.

This work has been supported by the Natural Science Foundation of China (No. 41804112), and Scientific Research Foundation of Chongqing University of Technology.

References

1. Marz N, Warren J. Big Data: Principles and Best Practices of Scalable Realtime Data Syatems.Greenwich,USA: Manning Publications Co.,(2015).
2. Adomavicius G, Tuzhilin A. Toward the next generation of recommender ayatems: A survey of the state-of-the-art and possible extensions. IEEE Transaction on Knowledge and Data Engineering, 17(6): 734-749, (2005).
3. Sanchez-Gonzalez, Alvaro, et al. “Graph Networks as Learnable Physics Engines for Inference and Control.” ArXiv Preprint ArXiv:1806.01242,(2018).
4. Bruna, Joan, et al. “Spectral Networks and Locally Connected Networks on Graphs.” ICLR 2014 : International Conference on Learning Representations (ICLR) 2014, (2014).
5. Ungar LH, Foster DP. Clustering methods for collaborative filtering. Proc Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08, Menlo Park, 1998: 84-88.
6. Liu Q,Wu S,Wang L. Multi-behavioral sequential prediction with recurrent log-bilinear model. IEEE Transactions on Knowledge and Data Engineering, 29(6): 1254-1267,(2017).
7. Grbovic M, Radosavljevic V, Djuric N, et al. E-commerce in your inbox: Product recommendations at scale//Proceedings of 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2015: 1809-1818.
8. Wang, Xiang, et al. “Neural Graph Collaborative Filtering.” Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174, (2019).
9. Jin, Yuanyuan, et al. “Syndrome-Aware Herb Recommendation with Multi-Graph Convolution Network.” 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 145–156, (2020).
10. Jin, Yuanyuan, et al. “Syndrome-Aware Herb Recommendation with Multi-Graph Convolution Network.” 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 145–156, (2020).
11. Jheng-Hong Yang, Chih-Ming Chen, Chuan-ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In RecSys. 140-144.
12. Wang H, Shi X, Yeung D Y. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks//Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 2016: 415-423.
13. Hamilton, William L., et al. “Inductive Representation Learning on Large Graphs.” Advances in Neural Information Processing Systems, pp. 1024–1034, (2017).
14. 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, pp. 974–983, (2018).
15. Sorokin, Daniil, and Iryna Gurevych. “Modeling Semantics with Gated Graph Neural Networks for Knowledge Base Question Answering.” Proceedings of the 27th International Conference on Computational Linguistics, pp. 3306–3317, (2018).
16. Veličković, Petar, et al. “Graph Attention Networks.” ICLR 2018 : International Conference on Learning Representations 2018, (2018).
17. Wang, Xiang, et al. “KGAT: Knowledge Graph Attention Network for Recommendation.” Proceedings of the 25th ACM SIGKDD International Conference

on Knowledge Discovery & Data Mining, pp. 950–958, (2019).
18. Zhu, Qiannan, et al. “A Knowledge-Aware Attentional Reasoning Network for Recommendation.” AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence, vol. 34, no. 4, pp. 6999–7006, (2020).
19. Lei, Wenqiang, et al. “Interactive Path Reasoning on Graph for Conversational Recommendation.” KDD 2020 : 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2073–2083,(2020).