Collaborative Filtering Recommendation model Based on Graph Neural Network and Attention Mechanism

Matt Dahl (mattdahl.tus@gmail.com)
Technical University of Sofia
Vessela Ivan
Technical University of Sofia
Dora Patko
Technical University of Sofia
Steve Georgi
Technical University of Sofia

Research Article

Keywords: Recommendation, Graph Neural Network, Attention Mechanism

Posted Date: August 23rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1971133/v1

Collaborative Filtering Recommendation model Based on Graph Neural Network and Attention Mechanism

Matt Dahl, Vessela Ivan, Dora Patko, Steve Georgi,

Abstract—With the popularity and development of the internet and mobile terminals, people can access a lot of information through them every day. Recommender systems have become one of the important technologies for various online platforms that aim to predict whether a user will interact with an item or not. Among them, collaborative filtering-based models have made effective progress in learning user and item representations by modeling historical user-item interactions. Recently, models based on GCN have been effective in recommendation, and the main function of GCN models is to improve the embedding representation of users and items by iteratively aggregating feature information from neighbors using graph connectivity to extract additional information. However, in previous works, dividing users into subgraphs without intersection only leads to a partial loss of information, ignoring the potential connections that may exist between different groups of users; and because only users are divided, the influence of commodity factors on the purchase outcome at the time of purchase is ignored in the learning process. Based on the above considerations, in this paper we propose a message-passing recommendation model. The model uses intersects users and items in separate subgraphs and uses an optimized attention mechanism to obtain the final node embedding to optimize the embedding representation by introducing multiple embedding propagation layers that encode higher-order connectivity relationships. We conduct extensive experiments to evaluate the proposed model. The results show that our model can effectively improve the performance of the recommendation.

Index Terms—Recommendation, Graph Neural Network, Attention Mechanism.

1 Introduction

With the popularity and development of the internet and mobile terminals, people can access a lot of information through them every day. It gradually becomes a very difficult thing for people to get the information they really need. Whether it's news, shopping or music, videos, etc. [1], these elements that can be found everywhere in life also put people in a state where it is difficult to get information of interest. A large amount of information is presented at the same time, and users are unable to get the information they are really interested in, which makes the information usage rate decrease, similar to this situation, which leads to the topic of information overload. Recommendation systems can proactively provide personalized services to users. In a recommendation system, it is not necessary for the user to provide a clear demand, but by going through the analysis of the user's historical behavior and modeling the user's interests, thus generating personalized recommendations for the user. In recent years, with its unique advantages, recommendation systems have been widely used in various fields, greatly improving the user experience and creating great value [2].

Currently, in the field of computer vision, compared with traditional algorithms, deep learning has better image processing results [3, 4, 5]. Compared with traditional machine learning, the advantages of deep learning are mainly in two aspects: first, the recognition or classification performance is higher, and the average effect of deep learning is better than that of traditional algorithms. Second, it is more applicable, and deep learning can fine-tune the model by fine-tune methods [6, 7] to get application-specific solutions for specific scenarios. In other words, based on deep learning, on the one hand, the accuracy of some algorithms is improved, on the other hand, some problems that cannot be better achieved by traditional machine learning methods can be achieved by deep learning [8]. Recommender systems have become one of the important technologies for various online platforms that aim to predict whether a user will interact with an item or not. Among them, collaborative filtering-based models have made effective progress in learning user and item representations by modeling historical user-item interactions [9, 10, 11]. Recently, models based on GCN [12] have been effective in recommendation, and the main function of GCN models is to improve the embedding representation of users and items by iteratively aggregating feature information from neighbors using graph connectivity to extract additional information. NGCF [13] exploits the higher-order connectivity of graphs to alleviate the sparsity problem in recommender systems, however, GCN suffers from the over-smoothing problem, as graph convolution operations can actually be seen as a special kind of graph Laplacian smoothing, which makes node representations indistinguishable after multi-layer graph convolution. To address these problems, Chen et al. [14] proposed the LR-GCN model to mitigate the oversmoothing effect by adding residual operations. He et al. [15] further pointed

*Matt Dahl is the corresponding author.

- *Matt Dahl, Vessela Ivan, Dora Patko, and Steve Georgi are with the Technical University of Sofia,Bulgaria. (e-mail: mattdahl.tus@gmail.com).*

out that the feature transformation and nonlinear activation in the original GCN model have almost no positive impact on the final performance, and even have more negative impact with training; based on this, the proposed LightGCN model only retains the neighborhood aggregation operation and removes the ”self-loop” in the aggregation operation to alleviate the oversmoothing problem. To reduce the propagation of negative information, DropEdge [16] randomly removes a certain number of edges from the input graph in each training cycle and theoretically demonstrates that DropEdge either reduces the convergence speed of oversmoothing or mitigates the information loss from oversmoothing. Wei et al. [17] focued on adaptively refining the structure of interaction graph to discover and prune potential false-positive edges. They devised a new GCN-based recommender model, Graph-Refined Convolutional Network (GRCN), which adjusts the structure of interaction graph adaptively based on status of model training, instead of remaining the fixed structure. The IMP-GCN [18] model proposed in the same period considers that not all information from higher-order neighbors is positive in reality, and by randomly dividing users into two subgraphs for learning, the association edges of users between the two subgraphs are cut off, thus filtering out the propagation of negative information in higher-order graph convolution operations. However, in IMP-GCN, dividing users into subgraphs without intersection only leads to a partial loss of information, ignoring the potential connections that may exist between different groups of users; and because only users are divided, the influence of commodity factors on the purchase outcome at the time of purchase is ignored in the learning process.

Based on the above considerations, in this paper we propose a message-passing recommendation model. The model uses intersects users and items in separate subgraphs and uses an optimized attention mechanism to obtain the final node embedding to optimize the embedding representation by introducing multiple embedding propagation layers that encode higher-order connectivity relationships. We conduct extensive experiments to evaluate the proposed model. The results show that the model can learn better node representations effectively improve the performance of the recommendation.

2 Related Work

Traditional recommendation algorithms can be divided into content-based recommendation algorithms and collaborative filtering-based recommendation algorithms [19]. Content-based recommendation algorithms mainly recommend similar items for users based on the items they have been interested in [20]. For example, if a music playing platform wants to recommend music for a certain user, it can recommend music with similar content for that user based on the user’s history of listening to music, based on the genre of those music, the artist and other information (content here can be understood as attributes, features. Therefore, the content-based recommendation algorithm has a certain degree of interpretability [21]. At the same time, each user’s user profile is obtained based on their own preference level for items, and even if new items are added to the system, they can be recommended based on the user profile. Of course, the drawbacks of this algorithm are particularly significant, such as the difficulty of extracting the characteristics of different items and the inability to complete the algorithm [22]. The disadvantages of this algorithm are particularly significant, such as the difficulty of extracting the characteristics of different items, and the inability to complete the recommendation for new users [23].

YehudaKoren et al. [24] proposed matrix decomposition in 2008, and its basic idea is mainly to decompose the use-Item rating matrix into two low-rank vectors representing the potential factors of the user, item, respectively. In 2008 Ajit et al. [25] proposed CMF to decompose multiple matrices simultaneously and share parameters between factors when an entity is involved in multiple relationships. In 2018 Fan et al. [26] proposed to use DMF for matrix filling. DMF is a deep structured neural network with low-dimensional unknown hidden variables as input and partially observed variables as output. In DMF, the inputs and parameters of the multilayer neural network are simultaneously optimized to minimize the reconstruction error of real data. The lost entries can be recovered more easily by propagating the latent variables to the output layer [27]. Collaborative filtering based recommendation algorithm can be effective in obtaining potential information about users and items, but the algorithm faces problems such as data sparsity and cold start [28, 29].

Deep neural network models [30, 31] have had a great impact on learning effective feature representations in several fields [32, 33] such as speech recognition [34], computer vision [35], and natural language processing [36]. Some recent studies have applied deep neural networks to recommendation tasks with good results [37]. In 2016 Cheng et al. [1] proposed the Wide&Deep model, using a logistic regression model as a Wide model to mine correlations in historical interaction behaviors to make the model have better memory capability and thus improve model accuracy. Then a deep neural network is used as the Deep model to improve the model generalization ability. However, this model is more dependent on the appropriate feature selection. Therefore, to address this problem of manual feature selection, Guo et al. [38] proposed DeepFM model in 2017, using a factorization machine model instead of the previous Wide model, in which the factorization machine uses additive and inner product operations to obtain linear and first-order interactions, and the multilayer perception is used to obtain higher-order linear signs. In recent years, graph neural networks have been shown to work well on graph-structured data, and the use of graph neural networks for semi-supervised classification was first proposed in the image domain by Kipf et al. in 2016 [39]. In the recommendation system task, the user’s rating data for items is an obvious graph-structured data. Pinsage et al. [10] proposed a random wandering graph neural network to learn the embedding of nodes in a web-scale graph. The GCMC model proposed by Den et al. [12] applied a graph convolutional network GCN, but it uses only one convolutional layer to compute the direct connections between users and items. Therefore, it

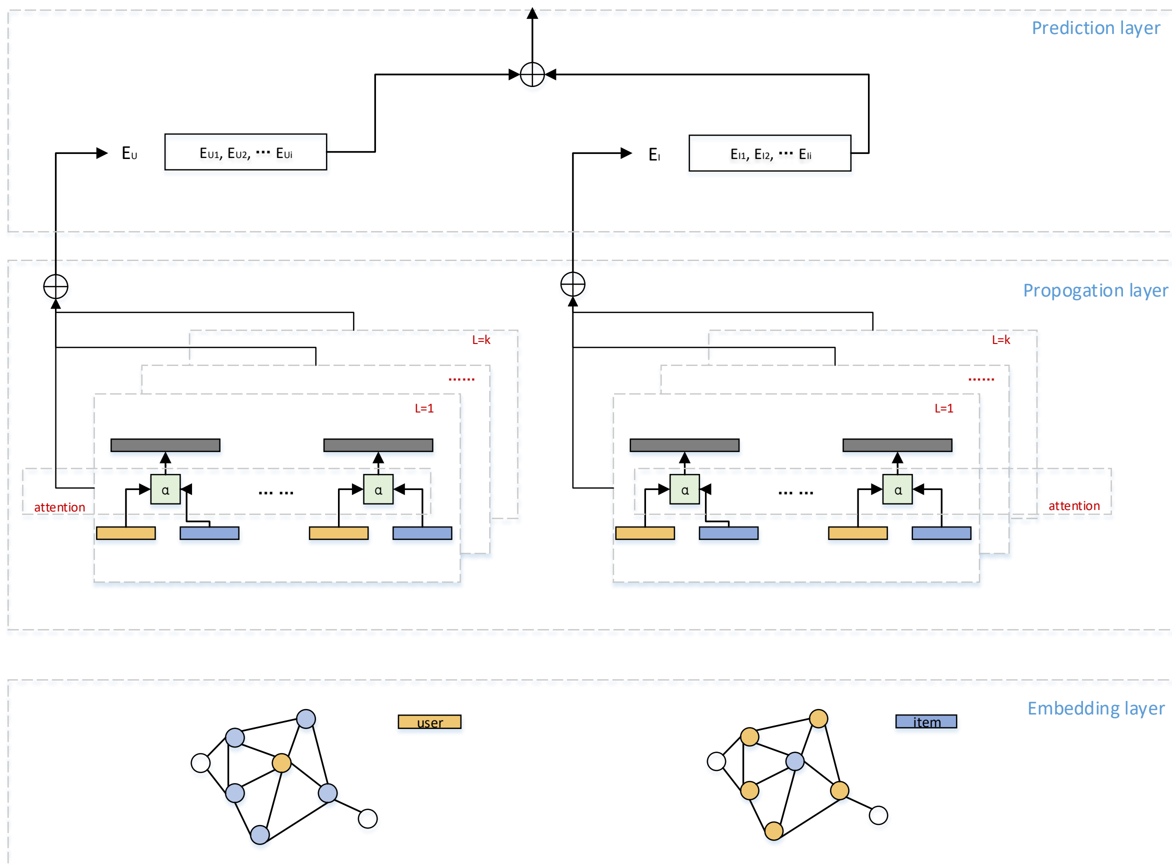

Fig. 1: Modelframework.

cannot reveal the collaborative filtering signal in higher-order connectivity.

3 METHODOLOGY

3.1 Embedding layer

User-item interactions can be modeled based on user embeddings and item embeddings. We describe a user U_a(item i_b) by an embedding vector $e_{ua} \in \mathbb{R}^d (e_{ib} \in \mathbb{R}^d)$, where d denotes the size of the embedding.

$$E = [e_{u1}, e_{u2}, \ldots, e_{um}, e_{i1}, e_{i2}, \ldots, e_{in},], \tag{1}$$

where $e_{u1}, e_{u2}, \ldots, e_{um}$ denotes user embedding, and $e_{i1}, e_{i2}, \ldots, e_{in}$ denotes item embedding.

3.2 Embedding propagation layer

Suppose $A \in R^{m \times n}$ is the adjacency matrix of the user-item interactions. A non-zero entry $a_{um} \in A$ indicates that user $u \in U$ has an interaction relationship with item $u \in I$. Otherwise, the entry is empty. Based on this relationship matrix, a user-item bipartite graph $\mathcal{G} = (W, E)$ can be constructed, where W is the set of nodes, E is the set of edges, and a non-zero entry $a_{um} \in A$ indicates that there is an edge between user u and item i. Using e_{u0} to denote the embedding of user u at level 0 and e_{i0} to denote the embedding of item i at level 0, the low-order aggregation operation is as follows:

$$e_{u1} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} e_{i0}, \tag{2}$$

$$e_{i1} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i||N_u|}} e_{u0}, \tag{3}$$

where e_{uk} and e_{ik} denote the embeddings of user u and item i, respectively, after propagation at k-th layer. N_u denotes the set of items interacting with user u and N_i denotes the set of users interacting with item i.

Only the user nodes and the item nodes adjacent to these user nodes are retained in the higher-order information propagation process, then n subgraphs are obtained. For each subgraph after k-th layers of propagation the items and users get the following low-dimensional representation results:

$$e_{u(k+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} e_{ik}, \tag{4}$$

$$e_{i(k+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i||N_u|}} e_{uk}, \tag{5}$$

where e_{ik} denotes the node representation of item i after k-layer convolution in subgraph s, and e_{uk} denotes the node representation of user u after k-layer convolution in subgraph s.

For a connected user-item pair, we define the message from i-u as:

$$h_a = \theta(w \cdot Aggre_i(\{x_{as} s \in C(a)\} + b), \tag{6}$$

where $C(a)$ denotes the set of items with which user u has interaction, x_{as} denotes the viewpoint-based embedding representation of the interaction between user u and item i, and $Aggre_i$ is the item aggregation function. In addition θ denotes the nonlinear activation function and w,b are the weights and biases of the neural network, respectively. Obviously, users can express their opinions by rating the interactions of the items and noting them as r. These opinions about the items capture how users prefer the items. x_{as} can be expressed as follows:

$$x_{as} = g_v([e_{is} \oplus e_r]), \tag{7}$$

where $\oplus$ represents the concat operations of two vectors. The aggregation function $Aggre_i$ is based on the mean operator, which takes the vector in $\{x_{as} s \in C(a)\}$ element mean values. The mean value based aggregation function is as follows:

$$h_a = \theta(w\{\sum_{s \in C(a)} \beta_a x_{as}\} + b), \tag{8}$$

The approach assumes that all interactions contribute equally to understanding the user. In fact, not every interaction has the same impact on the user's viewpoint-based embedding. So we allow various interaction behaviors to contribute differently to the user's latent factor.

3.3 Attention Mechanisms

However, this is achieved through the attention mechanism as a way to improve the β_a for the target user u. In particular, we parameterize item attention β_a using an attention network consisting of two layers of networks. The attention network is defined as follows:

$$\beta_a = w_2^T \theta([x_{as} \oplus e_u] + b_1) + b_2, \tag{9}$$

The attention mechanism is used to learn the weights of the two sets of node embeddings to obtain the final node representation. The model utilizes where the embedding learning approach takes the two node embedding representations obtained from the two classification approaches and splices the node embeddings at the k-th layer into the matrix U_i.

$$U_i = (E_{Ui}, E_{Ii}), \tag{10}$$

Calculation of weights on individual nodes using the self-attentive mechanism:

$$a_{i,U} = softmax(w_r^T, tanh(W_r, U_i))^T, \tag{11}$$

$$a_{i,I} = softmax(w_i^T, tanh(W_i, I_i))^T, \tag{12}$$

Then the representation vector of node v_i under the r-relationship is obtained:

$$E_{Ui} = U_i a_{i,U}, \tag{13}$$

$$E_{Ii} = U_i a_{i,I}, \tag{14}$$

$$E_I = E_{Ui} + E_{Ii}, \tag{15}$$

The learned embedding of the user node is denoted as eu and the embedding of the item node is denoted as ei. For a given user u and item i, the user's preference for the item can be calculated by inner product to obtain the score between the user and the item.

$$\hat{r}_{uv} = e_u^T e_i, \tag{16}$$

3.4 Optimization

The objective function is expressed as:

$$Loss = \frac{1}{2|O|} \sum_{a,b \in O} (r'_{ab} + r_{ab})^2, \tag{17}$$

where $|O|$ is the number of observed ratings, r_{ab} is the true value rating from u for i, and r'_{ab} is the predicted rating of item i by user u. A small batch Adam optimization prediction model is used to update the model parameters using a gradient descent approach.

4 EXPERIMENTS

4.1 Dataset

To evaluate the effectiveness of our model, two benchmark datasets from two popular websites, Ciao, and Epinions are used for the experiments. Each service allows users to rate, browse, and write reviews on items. As a result, they provide a large amount of rating information.

4.2 Baselines

To evaluate the model performance, we compared our model with three baselines.

- PMF [40]: Probability matrix decomposition, using only the user-item rating matrix, models the hidden factors of users and items through Gaussian distributions.
- GraphRec [41]: This model uses GNNs to integrate node information and topology to model user-item graphs, user-user graphs, and consistency across intensities.
- GCMC [12]: Graph Convolutional Matrix Completion utilizes a graph auto-encoder to learn the connectivity information of a bipartite interaction graph for latent factors of users and items.

4.3 Metrics

Two widely used recommended evaluation metrics were used for the experimental evaluation: Recall and normalized discount cumulative gain (NDCG).

$$Recall = \frac{N_{rs}}{N_r}, \tag{18}$$

where N_{rs} indicates the number of items in the recommendation list that are of interest to the user. N_r indicates the number of all items of interest to the user in the dataset.

$$NDCG = \frac{1}{log_2(1 + rank_p)}, \tag{19}$$

where $rank_{pos}$ indicates the location of the interested item.

TABLE 1: Statistical information of datasets.

dataset	Ciao	Epinions
users	7300	20818
items	5180	285549
rates	13300	287637

TABLE 2: Model Recall comparison on two datasets.

	Ciao	Epinions
PMF	6.35%	12.27%
GCMC	9.13%	14.55%
GraphRec	10.46%	15.81%
Ours	11.08%	16.26%

TABLE 3: Model NDCG comparison on two datasets.

	Ciao	Epinions
PMF	2.77%	5.85%
GCMC	4.38%	6.79%
GraphRec	6.45%	7.81%
Ours	7.03%	8.27%

4.4 Result analysis

The comparison of performance of baselines are shown in the Table2 and Table3.

As can be seen, our model surpasses these baseline models. By looking at Table 2 and Table3, it can be seen that over model improves recall by 5.57%, and 2,85% in the two datasets, and NDCG improves by 7.71%, and 5.89%, respectively. GraphRe outperforms GCMC. Both methods utilize only graph neural networks. However, GraphRec is based on the embedding propagation architecture, which illustrates the power of the embedding propagation model. The method in this paper outperforms all baseline methods. In contrast to GraphRec, the model in this paper utilizes attention mechanism in the user item graph to fuse both interaction behaviors of user items and user perspectives. This illustrates the importance of simultaneously fusing the interaction behaviors of user items and user perspectives.

The experimental comparison of the number of network layers, the performance of the model at different layers is shown in Figure. It is observed that the model does not show any performance degradation when the number of model layers is increased. On the contrary, the number of layers increases in favor of the performance of the model.

5 Conclusion

Recommendation systems can proactively provide personalized services to users. In a recommendation system, it is not necessary for the user to provide a clear demand, but by going through the analysis of the user's

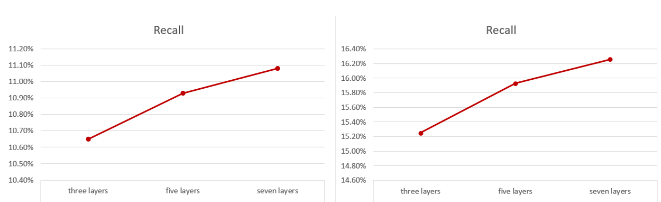

Fig. 2: The Recall of our model in different layers on two datasets.

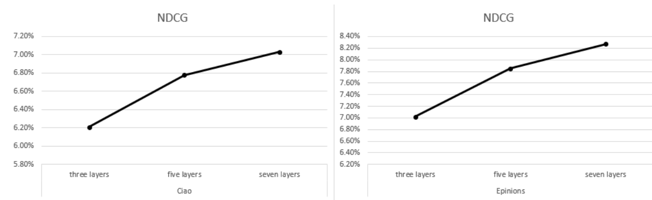

Fig. 3: The NDCG of our model in different layers on two datasets.

historical behavior and modeling the user's interests, thus generating personalized recommendations for the user. In recent years, with its unique advantages, recommendation systems have been widely used in various fields, greatly improving the user experience and creating great value. Currently, in the field of computer vision, compared with traditional algorithms, deep learning has better image processing results. Compared with traditional machine learning, the advantages of deep learning are mainly in two aspects: first, the recognition or classification performance is higher, and the average effect of deep learning is better than that of traditional algorithms. Second, it is more applicable, and deep learning can fine-tune the model by fine-tune methods to get application-specific solutions for specific scenarios. In other words, based on deep learning, on the one hand, the accuracy of some algorithms is improved, on the other hand, some problems that cannot be better achieved by traditional machine learning methods can be achieved by deep learning. Recommender systems have become one of the important technologies for various online platforms that aim to predict whether a user will interact with an item or not. Among them, collaborative filtering-based models have made effective progress in learning user and item representations by modeling historical user-item interactions. Recently, models based on GCN have been effective in recommendation, and the main function of GCN models is to improve the embedding representation of users and items by iteratively aggregating feature information from neighbors using graph connectivity to extract additional information. However, in previous works, dividing users into subgraphs without intersection only leads to a partial loss of information, ignoring the potential connections that may exist between different groups of users; and because only users are divided, the influence of commodity factors on the purchase outcome at the time of purchase is ignored in the learning process. Based on the above considerations, in this paper we propose a message-passing recommendation model. The model uses intersects users and items in separate subgraphs and uses an optimized attention mechanism to obtain the final node embedding to optimize the embedding representation by introducing multiple embedding propagation layers that encode higher-order connectivity relationships. We conduct extensive experiments to evaluate the proposed model. As can be seen from the experiment results, our model improves recall by 5.57%, and 2,85% in the two datasets, and NDCG improves by 7.71%, and 5.89%, respectively. These results show that the model can learn better node representations and effectively improve the performance of the recommendation.

REFERENCES

[1] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[2] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4041–4049.

[3] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2055–2068, 2017.

[4] L. Shao, Z. Cai, L. Liu, and K. Lu, "Performance evaluation of deep feature learning for rgb-d image/video classification," *Information Sciences*, vol. 385, pp. 266–283, 2017.

[5] M. Jian, S. Zhang, X. Wang, Y. He, and L. Wu, "Deep key frame extraction for sport training," in *CCF Chinese Conference on Computer Vision*. Springer, 2017, pp. 607–616.

[6] A. Khatami, M. Babaie, H. R. Tizhoosh, A. Khosravi, T. Nguyen, and S. Nahavandi, "A sequential search-space shrinking using cnn transfer learning and a radon projection pool for medical image retrieval," *Expert Systems with Applications*, vol. 100, pp. 224–233, 2018.

[7] F. Mahmood, R. Chen, S. Sudarsky, D. Yu, and N. J. Durr, "Deep learning with cinematic rendering: fine-tuning deep neural networks using photorealistic medical images," *Physics in Medicine & Biology*, vol. 63, no. 18, p. 185012, 2018.

[8] Y. Deldjoo, M. Schedl, B. Hidasi, Y. Wei, and X. He, "Multimedia recommender systems: Algorithms and challenges," in *Recommender systems handbook*. Springer, 2022, pp. 973–1014.

[9] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 635–644.

[10] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[11] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[12] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[13] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[14] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 27–34.

[15] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

[16] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.

[17] Y. Wei, X. Wang, L. Nie, X. He, and T.-S. Chua, "Graph-refined convolutional network for multimedia recommendation with implicit feedback," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3541–3549.

[18] F. Liu, Z. Cheng, L. Zhu, Z. Gao, and L. Nie, "Interest-aware message-passing gcn for recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 1296–1305.

[19] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[20] R. Devooght and H. Bersini, "Collaborative filtering with recurrent neural networks," *arXiv preprint arXiv:1608.07400*, 2016.

[21] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[22] X. Yu, T. Gan, Z. Cheng, and L. Nie, "Personalized item recommendation for second-hand trading platform," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3478–3486.

[23] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[24] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[25] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 650–658.

[26] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Networks*, vol. 98, pp. 34–41, 2018.

[27] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems." in *IJCAI*, vol. 17. Melbourne, Australia, 2017, pp. 3203–3209.

[28] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1633–1647, 2016.

[29] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5382–5390.

[30] F. Scarselli, S. L. Yong, M. Gori, M. Hagenbuchner, A. C. Tsoi, and M. Maggini, "Graph neural networks for ranking web pages," in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. IEEE, 2005, pp. 666–672.

[31] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[32] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[33] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[34] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[35] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3642–3649.

[36] H. Li, "Deep learning for natural language processing: advantages and challenges," *National Science Review*, 2017.

[37] Q. Wang, Y. Wei, J. Yin, J. Wu, X. Song, and L. Nie, "Dualgnn: Dual graph neural network for multimedia recommendation," *IEEE Transactions on Multimedia*, 2021.

[38] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[40] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, 2007.

[41] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.