Dicle University
Journal of Engineering

https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Research Article

Applying graph convolution networks to recommender systems based on graph topology

Alper ÖZCAN[1*]

[1]Akdeniz University, Computer Engineering Department, alperozcan@akdeniz.edu.tr, Orcid No: 0000-0002-5999-1203

ARTICLE INFO

Article history:

Received 01 March 2022
Received in revised form 05 April 2022
Accepted 12 April 2022
Available online 28 June 2022

Keywords:

Recommender systems, graph convolution network, deep learning, graph modelling

Doi: 10.24012/dumf.1081137

* Corresponding author

ABSTRACT

The recommender systems are widely used in online applications to suggest products to the potential users. The main aim of recommender system is to produce meaningful recommendation to a potential user by monitoring user's purchasing habits, history, and useful information. Recently, graph representation learning methods based on node embedding have drawn attention in Recommender systems such as Graph Convolutional Networks (GCNs) that is powerful method for collaborative filtering. The GCN performs neighborhood aggregation mechanism to extract high level representation for both user and items. In this paper, we propose a recommendation algorithm based on node similarity convolutional matrices with topological property in GCNs where the linkage measure is illustrated as a bipartite graph. The experiments indicate the necessity of capturing user–item graph structure in recommendation. The experimental results show that node similarity-based convolution matrices and GCN-based embeddings significantly improve the prediction accuracy in recommender systems compared to state-of-art approaches.

Introduction

The modern era is quickly adopting technology in every aspect, especially e-commerce websites. Today, people like to shop on e-commerce websites as it provides a wide list of products. With the presence of such websites, consumers can select items with their preferences. However, listing plenty of products comes with the drawback of listing inappropriate products to the users. Providing appropriate and interesting items to the users makes a shopping experience more efficient and it enhances the customer satisfaction to the retailers. To enhance customer satisfaction and experience, even more, many retailers are recommending products to the users to increase their profits by keeping the customer loyal to their websites. The recommending process requires different analyses to keep the potential customer loyal to the retailer. E-commerce giants such as Amazon, AliExpress, Netflix, etc., are using recommender systems to suggest items to users.

The main purpose of the recommender system is to suggest items to the user. The items could be books, movies, music, news, etc. To recommend the best possible items to the user a series of information of a user is monitored such as demographic information, purchase behavior, rating behavior, previous history. In the past, these intelligent recommender systems helped Amazon to increase its profit to 35%, BestBuy to 24%, increased 75% views for Netflix, and 60% growth in views for YouTube [1]. Hence, designing a personalized recommender system is helping not only the e-commerce industry but other fields such as health care, food industry, academia, and more.

There are plenty of recommender systems but the standard algorithms such as collaborative filtering, content-based filtering model the user-item rating as a matrix and predict the rating for potential unrated items. Previously in the literature models like Collaborative Filtering (CF) use similarity index between the neighbors or items to recommend items. The model-based approaches such as Matrix Factorization (MF) are used to recommend products as a representation problem of the big user-item matrix. The main challenge these previous models face is the complexity of the user behavior towards multiple items. To resolve such issues neural network-based models are proposed [2, 3, 4]. However, these models are still limited to capturing the high-structured data.

However, among other divisions of the recommender systems, it can be divided into three stages: shallow models [5, 6, 7], neural models, and GNN-based models [8, 9, 10]. Recently, graphical networks based on deep learning concepts have shown promising results in recommendation systems. Graph Convolutional Networks (GCN) has brought new ideas to tackle some of the conventional artificial intelligence problems. It transforms the regular convolutional operation into a graphical structure and generates node representations using a graph aggregator. A simple graph aggregator calculates the node representation by aggregating its local neighbors and multiple layers of such aggregators are stacked non-linearly to build a deep neural network. Such architectures are also promising results in recommender systems. Monti et al. proposed a recommender system in which GCN was used to aggregate the user-user and item-item graphical information [11].

The user-item interaction can be represented as a bipartite graph. Different graphical algorithms have been used to recommend items to the users either as a simple random walks or more complex methods such as [9, 10, 12, 13, 14]. In such methods, the key concept is to fetch high order neighbors' information which captures the diverse information. The aim of these graphical methods is to capture more complex and diverse information about the user and its neighbors that could help for recommending better items.

In this study, we propose a GCN-based model that captures the topology of the bipartite network between user and item and verifies the efficiency of those topological measures in context of recommender system. To evaluate our model Amazon product purchasing network dataset is used. The main contributions of this paper are three-fold:

- We propose a GCN-based architecture for recommender systems to learn graph topology of users and items in bipartite network settings
- Local measures of node similarity metrics can provide effective features for GCN to propagate node embeddings across the network in a way that is useful for recommender system.
- To evaluate our model, we experimented on the real datasets and obtained results indicate that the proposed model produce much diverse and improved recommendations.

Literature Review

This section is divided into three parts. Firstly, a general overview of recommender system is discussed that contains traditional recommendation techniques. In the second part the concept of graph-based recommender system is presented by discussing state of the art approaches in this area. Finally, different graph neural networks approaches are mentioned to provide sufficient knowledge to the reader.

Recommender Systems

The main purpose of recommender system is to recommend the items to the users. The recommender systems have seen tremendous growth in the literature and still it is one of the hot topics that need exploration even though many mature and reliable algorithms have been developed. Techniques like collaborative filtering, content-based filtering, neighborhood methods are presented for a long time in the literature. The content-based recommender system requires items description that may include its genre, actor, producer, author of a book, etc., to construct the profile of a user [16]. The recommendation is generated based on the similarity between the user profile and specific item description [17].

On the other hand, collaborative filtering does not require a detailed description of the item [18]. It is based on the user's personalized recommendations that are predicted by using the user's past behavior. It takes the ratings given by the user into a matrix where rows represent the users and columns represent the items [19]. Each cell in this matrix represents the degree of preference of a user towards an item. Further, collaborative filtering methods are divided into model-based and memory-based models [20]. Model-based approaches are Principal Component Analysis, Singular Value Decomposition [21]. Memory-based models calculate the similarity between the nearest neighbors such as Pearson Coefficient, and cosine distance [22]. Moreover, deep learning techniques have shown improvements in the recommender system domain as they tackle more complex data.

Graph-Based Recommender Systems

Graphs are used to represent complex data and it is one of the best possible ways to interpret complex problems. Recently, graph-based techniques in recommender systems have shown promising improvements by analyzing complex and deep interaction between the users and items. The graphs convert the recommendation problem to the link prediction task which utilizes different link prediction techniques such as random walks to find the similarities in the user-user, user-item, or item-item data [23]. Generally, in the link prediction task, the aim is to classify the nodes, however, the recommendation task is a regression problem. So, the items are recommended based on how strong the relationship is between the user and item [24].

Li et al. proposed a graph-based recommender system that generates a bipartite graph between users and items. The kernel-based approach is used to generate the random walks between the user-item transaction. Two types of nodes for users and items are defined and the edge in the graph represents the transaction between the user-item node pair. Afterward, similarities are calculated by using random walks which can detect the more complex relationship between the user and item [25].

Zhang et al. proposed a music recommendation model in which the data is represented as a bipartite graph. The weight of the link between user and item was represented as a complex number that could determine the like and dislike of a user for a particular item [26, 27].

Generally, the graph-based models are used in heterogeneous networks in which the data is more complex. Graph-based recommender systems can effectively utilize such complex information and can find similarity measures between different nodes. By expanding the neighborhood of a particular node, it can effectively compute the proximity between the user and item [28].

Graph Neural Networks in Recommendation Models

The advancement in the graph neural network provides an oppourtunity to address the recommder sytem problem present in the literature. For example, graph neural networks provides the better aggregation for the neighborhood embeddings. The stacking of the propagation layer can access the high order neighbor's information. It dominates previous methods by gathering the more structural information in the network [29].

Since the user-item interaction can be illustrated as graph the Graph Neural Networks can be easily modeled to represent such graphs. Qui et al. [30] proposed a personalized recommender system by using graph neural network to find the interesting courses for the learners. It resolved the issue of explicit representation of the products in which structural information of the products were not learned precisely. Secondly, other methods obtain the general user preference for items by neglecting the recent items. This method uses user's sequence neighbors and use an attention mechanism to generate an item representation.

Zhang et al. proposed a STAcked and Reconstructed Graph Convolutional Networks (STAR-GCN) architecture that learns the nodes representations in the graphs to boost the performance of the recommender system [31]. Zheng et al. proposed a GCN based model that pushes the model from diversification towards the upstream candidate meaning that it rebalances the newly discovered neighbors and categorize them to achieve better results [32].

Methods

The real-world platforms scenarios the interaction between the user and item is the strongest signal that illustrate the user's preferences towards the item. Thus, in this study we construct a graphical representation of such scenario in which users and items are two kinds of nodes and the edge between these nodes represent the interaction. To achieve the collaborative filtering effect with such model, GCN is used on the top sub-section of the graph which fetch the features from the respective nodes by propagating the nodes back and forth.

Problem Statement

The recommender system problem is defined as follows: Given the set of users and items such as $\{User1, User2, \ldots., Usern\}$, $\{Item1, Item2, \ldots, Itemk\}$ respectively, the rating R given by user Ui towards item Ij, the goal of recommender system is to recommend a new item Ij to the existing user Ui that is not purchased before [33].

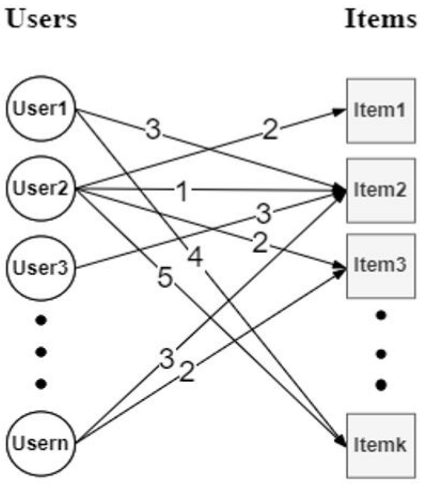

Figure 1. An example User-Item Bipartite Graph with ratings.

The recommender system can be illustrated as a bipartite graph as illustrated in Figure 1. The number on the edges between the links represents the weight between different nodes. A bipartite graph is containing exactly two types of nodes only an edge exists between two different node sets. A bipartite graph is defined as a graph G = (V1 U V2, E) where V1 and V2 are two different node sets and E is the edge between two sets.

As described earlier the bipartite graph illustrates the users and items. The edge between these two entities occurs only when a user purchases an item. Once this interaction system is modeled as a graph all the link prediction schemes such as [34, 35] are applicable to predict the future links. This study [23] applies the common neighbors' method with the length of two for the prediction of future links. Another study [27] extended this methodology towards a heterogenous environment which produces better outcomes with rich graphical structure.

Moreover, the link prediction method aims to estimate the existence of a link based on the observed nodes in the graph. The graphical perspective of the product recommendation problem can be viewed as a selection of unobserved links for each node and model such nodes as a link prediction problem [36]. So, the main goal is to predict the possible link in the user-item network by observing the link information.

Furthermore, to extract the features from the network topology different graphical representation techniques are used to embed the nodes into a multi-dimensional space. A network embedding is illustrated in the form of a matrix H $\in R^{|V| \times d}$ where $|V|$ and d are the parameters that define the number of dimensions in embedding space.

Node and Network Embedding using GCN

GCN is similar to the filter in convolution and denotes the definition of convolution from the regular grid to irregular structures like graphs [37]. The node embeddings are updated as follows:

]

$$H^{l+1} = \delta D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^l W^l \qquad (1)$$

Where $H^l \in R^{|V| \times d}$ represent the embedding matrix of $l-th$ layer convolution, d is the embedding dimension. A $\in R^{|V| \times |V|}$ is the adjacency matrix with self-loops, and $D_{ii} = \sum_j A_{ij}$ is the degree matrix. All these operations are making sure that all the neighbors are receiving an aggregated message from multiple hop neighbors. The weights W^l in the GCN are trained using gradient descent.

GCNs are the simplified version of Graph Convolutional Neural Networks (GCNNs). A typical GCN consist of three steps 1) feature propagation, 2) linear transformation, and 3) application of a non-linear activation function [38]. Feature propagation is achieved using convolutional matrix computed from graph topology. For the linear transformation different parameters are learned to minimize a loss function and for that typical activation functions such as sigmoid or ReLU are used.

Proposed approach: Recommender system in bipartite networks using GCN and Node Similarities

The implementation of graph convolutional neural network consists of two steps and illustrated in Figure 2. In the first step graph is constructed with the input data. For instance, the user-item interaction matrix is converted into a bipartite graph where user/item is represented as a node and interaction between them is represented as an edge. By transforming the data into graphical form, the traditional recommender problem can be represented as a link prediction problem.

Figure 2 illustrates the workflow of GCN encoder and decoder proposed in the study. Firstly, the GCN deep encoder is constructed for the user-item prediction model, and ReLu activation function is used between the hidden layers of the encoder. The input consists of neighborhood information and topological similarity measures of a graph. The generated output is the embedding vector for each node fed to the model.

Secondly, the decoder predicts the similarity score for each user-item pair by decoding the embedding vectors. The key process of decoding is to represent the user-item relation into new space by reconstructing edges from the embedding space information.

The proposed model is trained on all learnable parameters and after experimenting it shows greater improvements in the performance, as all the parameters receive gradient values from loss function.

To give more details, the second step involves the GCN layer applied to the constructed bipartite graph. In this architecture the GCN is different than other traditional methods because of its propagation layer. The propagation layer chooses the optimal path to find the high order similarity between the neighbors. Moreover, this layer can be parametric that assigns different weights to the nodes depending on the similarity and importance.

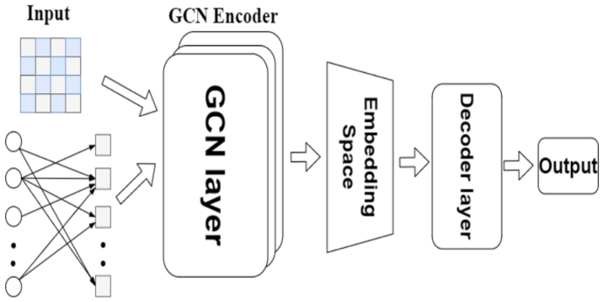

Figure 2. The overall procedure of implementing a GCN model: construct graphs from data, design GCN and prediction results with optimization.

Furthermore, various types of aggregation functions are used such as max, min, mean pooling, etc. Since there is no guarantee which aggregation function will perform better for different datasets, it is better to design a specific function that better suits the dataset. Also, these aggregation functions highly affect the performance of the model. For instance, the GNN-based recommendation system is used by using mean pooling that is computationally efficient. Meanwhile, these propagation/aggregation layers are stacked to achieve the high order neighbor hops to fetch diverse information. Moreover, modeling the too shallow or deep graphical structure will lead to poor recommendations. The shallow model cannot be modeled properly, and the deep model smoothens the node embeddings. Further, to achieve optimization traditional loss functions are turned into graphical learning losses. For example, the optimization log-loss is regarded as link prediction loss.

The proposed GCN model is composed of a graph convolutional layer and an embedding layer where the convolution layer serves the purpose of broadcasting and aggregation. Most of the studies in the literature use Laplacian-based convolutional matrix for network embeddings where the proposed study conjugates the convolution matrix with the single neural network layer for topological measures. After finding topological measures in the graph links are computed between each user-item pair for unrated items. These topological measures are helpful to assess the possible link scores between the entities. Further, Jaccard Index [23, 34] measures the direct number of neighbors for u and v.

To summarize the proposed approach, node embeddings are computed by using node similarity measures such as (CN, AA, JI, RA) between all nodes in the network and are fed to the GCN. These embeddings are used to predict and evaluate the links between the entities in a dataset.

Results and Discussion

Dataset

Amazon dataset (http://jmcauley.ucsd.edu/data/amazon/) for electronics, video games, movies, and tv is used for the evaluation of the proposed model. The dataset contains auxiliary information such as reviews, ratings, and item descriptions and illustrated in Table 1. Entities with at least 10 interactions are filtered to feed the model.

Table 1. Dataset detail.

Dataset name	#users	#items	#ratings
Electronics	67,345	45,398	2,719,345
Movies and TV	35,398	112,309	1,122,654
Video Games	10,123	34,643	729,332

The amazon dataset contains the user-item rating data, and these ratings are used to get the user-item interactions in the form of a graph. The proposed model utilizes these user-item ratings to create the embedding vectors to build a bipartite graph. Moreover, the split of 80% and 20% is used for training and testing the model respectively. Afterward, 5-fold cross-validation is used along with at least 10% of interactions to tune the parameters.

Experimental Settings and Evaluation Metrics

To evaluate the performance of the model AUROC, AUPR are used. Also, the following baseline models are compared to check the performance of our model. BPR-MF [34] which is a classical model to optimize the Bayesian personalized ranking loss, NeuMF [18] that learns the nonlinear interaction between user-item embeddings using multilayer perceptron's, and BiNE [14] uses biased random walks on bipartite graph to learn explicit and implicit relationship between user-item. For the baseline, all of the hyperparameters are initialized as presented in the original work and are carefully tuned to achieve optimal performance. Further, in proposed model the hyperparameters such as batch size, stopping criteria, learning rate are used for GCN.

Results

This section compares the result of proposed GCN model with the baseline models in terms of AUROC and AUPR. The experiments are performed on all three datasets. After comparing the outcomes our model performed better than other baseline models in all the three datasets. The results are illustrated in Table 2, Table 3, and Table 4.

Table 2. Test results for Amazon Electronics Dataset.

Models	AUROC	AUPR
GCN with CN	**0.92**	**0.23**
GCN with AA	0.90	0.21
GCN with JI	0.87	0.19
GCN with RA	0.88	0.20
BPR-MF	0.81	0.15
NeuMF	0.83	0.16
BiNE	0.87	0.19

Table 3. Test results for Movies and TV Dataset.

Models	AUROC	AUPR
GCN with CN	0.90	0.22
GCN with AA	**0.93**	**0.25**
GCN with JI	0.89	0.20
GCN with RA	0.86	0.21
BPR-MF	0.74	0.17
NeuMF	0.81	0.18
BiNE	0.86	0.18

Table 4. Test results for Amazon Video Games Dataset.

Models	AUROC	AUPR
GCN with CN	0.88	0.20
GCN with AA	**0.89**	**0.22**
GCN with JI	0.83	0.16
GCN with RA	0.85	0.18
BPR-MF	0.71	0.13
NeuMF	0.75	0.15
BiNE	0.81	0.16

The graph-based methods consistently performed better than baseline models, which represents the effectiveness of exploiting topological similarities between user-item interaction using bipartite graphs. Moreover, the achieved results indicate that convolutional matrices with topological similarities are more efficient and effective than Laplacian-based convolution in recommendation tasks. Moreover, the proposed model is tested on four different similarity measures that are Common Neighbor, Jaccard Index, Adamic Adar, Resource Allocation. The obtained results indicate that a single-layered GCN encoder based on node similarity performs better than the Laplacian-based convolution GCN.

Conclusion

This paper proposes a recommendation system that combines GCN with embedding vectors using similarity measures attained from graph topology. The traditional user-item interactions can convert into embeddings vectors with the usage of GCN. The achieved embedding vectors are used to construct a user-item bipartite graph and modeled as a recommender system. The proposed model is tested with three baseline models and achieves better results and performance using graph topology in the GCN model. In the future, the performance of the GCN can be improved by getting the semantic information of users such as their social information in the form of a graph.

References

[1] M. Chui, "Artificial intelligence the next digital frontier?", McKinsey and Company Global Institute, 47:3–6, 2017.

[2] H.Cheng, L. Koc, J. Harmsen, H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai and M. Ispir, "Wide & Deep Learning for Recommender Systems", In Proceedings of the 1st workshop on deep learning for recommender systems, pp. 7–10, 2016.

[3] H. Guo, R. Tang, Y. Ye, Z. Li and X. He, "DeepFM: a factorization-machine based neural network for CTR prediction", In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1725-1731, 2017.

[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering", In Proceedings of the 26th international conference on world wide web, pp. 173-182, 2017.

[5] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems”, Computer, 42(8), pp. 30-37, 2009.

[6] S. Rendle, “Factorization machines”, In 2010 IEEE International conference on data mining IEEE, pp. 995-1000, 2010.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback”, In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, pp. 452-461, 2009.

[8] X. He, K. Deng, X. Wang, Y.N. 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, pp. 639-648, 2020.

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

[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, pp. 974-983, 2018.

[11] F. Monti, M. Bronstein, and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks”, In NIPS, pp. 3697-3707, 2017.

[12] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, “Knowledge graph convolutional networks for recommender systems”, In the world wide web conference, ACM, pp. 3307-3313, 2019.

[13] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, “KGAT: Knowledge Graph Attention Network for Recommendation”, In KDD, pp. 950-958, 2019.

[14] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, “Session-based recommendation with graph neural networks”, In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. Pp. 346-353, 2019.

[15] C. Gao, Y. Zheng, N. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, “Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions”, arXiv, 2021.

[16] M. J. Pazzani and D. Billsus, “Content-based recommendation systems. The adaptive web: methods and strategies of web personalization”, Springer-Verlag, Berlin, Heidelberg, 325–341, 2007.

[17] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization”, In Proceedings of the Fifth ACM Conference on Digital Libraries, DL ’00, pages 195–204, ACM, 2000.

[18] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering”, IEEE Internet computing, 7(1):76–80, 2003.

[19] J. Leskovec, A. Rajaraman, and J. D. Ullman, “Recommendation Systems”, Cambridge University Press, 2 edition, 292–324, 2014.

[20] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems”, Computer, 42(8):30–37, 2009.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of recommendation algorithms for e-commerce”, In Proceedings of the 2Nd ACM Conference on Electronic Commerce, EC ’00, pages 158–167. ACM, 2000.

[22] B. Shams and S. Haratizadeh, “Graph-based collaborative ranking”, Expert Syst. Appl., 67(C):59–70, 2017.

[23] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks”, Journal of The American Society For Information Science and Technology, 58(7):1019–1031, 2007.

[24] L. Zhang, M. Zhao, and D. Zhao, “Bipartite graph link prediction method with homogeneous nodes similarity for music recommendation”, Multimedia Tools and Applications, pages 1–19, 2020.

[25] X. Li and H. Chen, “Recommendation as link prediction in bipartite graphs: A graph kernelbased machine learning approach”, Decision Support Systems, 54(2):880–890, 2013.

[26] L. Zhang, M. Zhao, and D. Zhao, “Bipartite graph link prediction method with homogeneous nodes similarity for music recommendation”, Multimedia Tools and Applications, pages 1–19, 2020.

[27] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, “New perspectives and methods in link prediction”, In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’10, pages 243–252. ACM, 2010.

[28] B. Shams and S. Haratizadeh, “Graph-based collaborative ranking”, Expert Syst. Appl., 67(C):59–70, 2017.

[29] T. J. Lakshmi and S. D. Bhavani, “Temporal probabilistic measure for link prediction in collaborative networks”, Applied Intelligence, 47(1):83–95, Jul 2017.

[30] R. Qiu, J. Li, Z. Huang, and H. Yin, “Rethinking the item order in session-based recommendation with graph neural networks”, In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 579-588, 2019.

[31] J. Zhang, X. Shi, S. Zhao, and I. King, “STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender System”, IJCAI, 2019.

[32] Y. Zheng, C. Gao, L. Chen, D. Jin, and Y. Li, “DGCN: Diversified Recommendation with Graph Convolutional Networks”, In Proceedings of the Web Conference, 401-412, 2021.

[33] D. Liben-Nowell, J.M. Kleinberg, “The link-prediction problem for social networks”, Journal of the American Society for Information Science and Technology , 58, 1019-1031, 2007.

[34] T. Zhou, L.-Y. Lu, Y.-C. Zhang, “Predicting Missing Links via Local Information”, arXiv: 0901.0553, 2009.

[35] Z. Huang, D. D. Zeng, H. Chen, “Analyzing Consumer-Product Graphs: Empirical Findings and Applications in Recommender Systems”, Management science. Vol. 53, No. 7, pp. 1146–1164 2007.

[36] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. Data Mining and Knowledge Discovery, 5(1):115–153, 2001.

[37] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis 30, 2 (2011), 129–150.

[38] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 32, 1 (2020), 4–24.