International Journal of Knowledge Based Computer Systems
7 (2), December 2019, 01-07
http://www.publishingindia.com/ijkbcs/

A Survey on Graph-Based Collaborative Filtering Techniques in Recommender Systems

Sanket Kamta[1*] and Vijay Verma[2]
[1]National Institute of Technology, Kurukshetra, Haryana, India. Email: iamsanketkamta@gmail.com
[2]National Institute of Technology, Kurukshetra, Haryana, India. Email: vermavijay1986@gmail.com
*Corresponding Author

Abstract: Recommender Systems (RS) are software tools which can be used in making useful predictions of items to users. RS has been an important research area since the mid-1990s, and there are a lot of RS tools built since then to improve user satisfaction. While building the RS tools, researchers face a lot of problems in the form of data sparsity, information overload, cold-start, scalability, lack of resources and time. These factors may reduce the accuracy of predictions. To overcome these problems, researchers model the rating data as graphs. Through graphs, we can explore the transitive associations and hidden information in our dataset. Particularly, this work focuses only on the graph-based collaborative filtering (GBCF) techniques introduced in the recommendation systems. We have studied and analyzed various GBCF articles published in the most popular online digital libraries during the last two decades. These approaches have been categorized into three broader categories: user-item, user-user, and item-item based on the model of the graph, which is used for the recommendation purpose. This survey provides an understanding of how graph-based CF has helped researchers and developers built a more efficient RS model in terms of different RS goals, such as accuracy.

Keywords: Cold-start, Scalability, Data sparsity, Graph-based collaborative filtering, Information overload, Recommender systems.

I. Introduction

A Recommender System is a software which is used for suggesting useful items to active users [1]. These recommendations can be in many forms, such as which games to play, which movies to watch or which places to visit. The growth of e-commerce websites has offered users with a lot of choices. RS helps those users to decide which item to select from the ocean of items available on the web. The prediction process done by the RS is dependent on the user's desire and constraints. RS collects the data about the users by analyzing their previous transactions and feedback for certain items.

RS has come up as an essential area of research since the mid-1990s [2] [3]. There has been a lot of advancement in the field of recommender system in today's world. RS is used in these famous internet sites like Amazon [4], YouTube [5], Netflix [6], IMDB [7] helping the users find a suitable item that best matches their needs. A lot of dedicated symposium and workshops are held in this field, ACM Recommender System (RecSys) [8] being one of them.

There are various RS techniques depending on the address domain, the knowledge used and the prediction algorithm [9]. Content-based filtering is used to recommend items to an active user based on the user's preferences made in the past. Collaborative filtering is used to recommend items to an active user based on the other users with whom it shares the same taste. CF, the most popular RS technique, is often said as "people-to-people correlation". Demographic filtering is used to recommend items to an active user based on a certain common personal attribute such as sex, age, country, etc. Community-based filtering is used to recommend items to an active user based on the choice of its social friend. Hybrid RS combines the above techniques to make recommendations such that the advantage of any technique is used to fix the disadvantage of other technique. A general classification of RS is shown in Fig. 1.

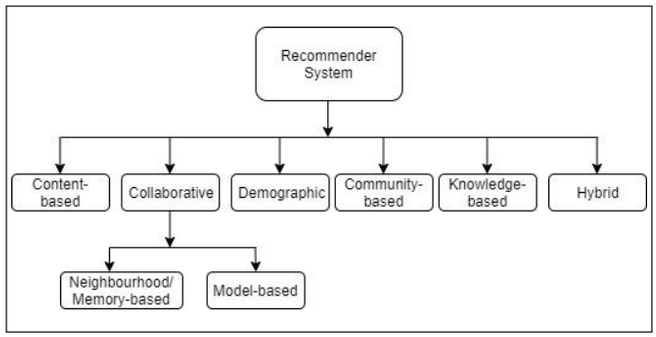

Fig. 1: General Classification of Recommender System

II. Literature Review

Out of the above-mentioned RS techniques, CF is still the most popular one as it is flexible across different domains and can produce more serendipitous recommendations [10]. While building the RS tools, researchers face a lot of problems in the form of data sparsity, information overload, cold-start, scalability, lack of resources and time. These factors may reduce the accuracy of predictions. To overcome these problems, researchers model the rating data as graphs. The transitive associations captured through the graphs can be very useful in recommending items as they can handle the problem of sparsity and limited coverage.

In the graph-based approach, the data is modelled in the form of graphs where the nodes are users or items or both, and the edges represent the similarities between users and items. In Fig. 2, we model the *user-movie* data in the form of a bipartite graph where the two sets of nodes represent *users* and *movies*, and the edge is connected between *user* and *movie* only if a *user* has rated a *movie.*

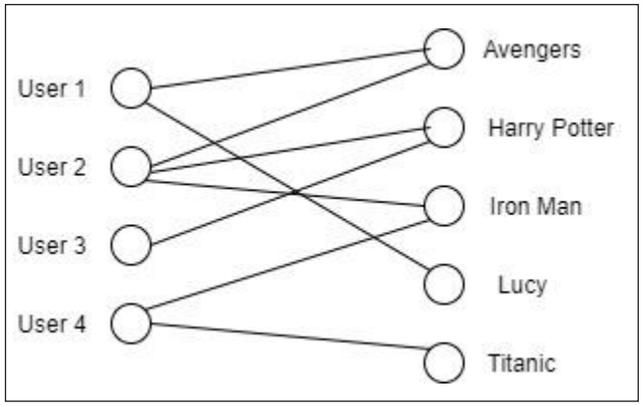

Fig. 2: Representation of Data in the Form of Graphs

Since 2000, there has been a lot of researches done and modification made in graph approaches to improve the quality of recommendations by improving the similarity between users/items. Some authors [11] [12] have conducted a comparative analysis of various structural similarity measures. While some authors [13] [14] [15] [16] tried to enhance the recommendation through dealing the data-sparsity problem while others made an approach [17] [18] [19] [20] to curb the problem of cold-start. A detailed analysis of various graph approaches made to enhance the similarity between users/items has been done through recent research papers and then arranged the graph approaches in a categorized manner. Through this paper, we aim to organize the graph approaches that has been quite influential in the field of RS over the last two decades. The remaining sections are organized as follows: Section III gives an idea about how we have executed our survey and divided the approaches into different categories. Section IV, V and VI define the categories and some important approaches that come under each category. Section VII ends the paper with some concluding notes.

III. Research Methodology

A. Survey Methodology

This comprehensive survey aims to explore the different graph-based collaborative filtering approaches introduced by various researchers to improve the quality of recommendation. We have analyzed various articles published over two decades in the six most popular online digital libraries (IEEE Xplore Digital Library, Springer Link, ACM Digital Library, Wiley Online Library, Taylor & Francis Online, Science Direct). We did an extensive advanced search in all the online digital libraries to retrieve the desired articles. The advanced search was done using a lot of combinations of the following search items – graph-based, collaborative filtering, similarity, data sparsity, cold-start, bipartite graph. This survey creates an understanding of how graph-based CF has helped researchers and developers built a more efficient RS model in terms of accuracy and coverage. We have categorized some important approaches in each category, and a summary is provided for each of the methods.

B. Categorization Methodology

We have categorized the graph-based CF techniques based on the model of the graph used for recommendation. We can model the graph into three categories User-item, user-user and item-item, as shown in Fig. 3. In the below sections, we will explore how these models are used in improving the quality of recommendations.

IV. User-Item Collaborative Filtering

User-Item CF analyses the user-item dataset and then identifies the relationship between the users and items. Using this relationship, it recommends items to users. In 2002, [21] tried to improve the scalability and performance of the recommender models without reducing the quality of recommendation. They used the Dense Bipartite Graph (DBG) based community extraction approach in finding a small group of neighbours for a given user and then provide a recommendation to the user by performing CF only on the small group of neighbours.

In 2004, [22] focused on dealing with the information overload problem on the web and improving the flexibility of the recommendation by building a graph model which can incorporate different recommendation approaches. Fig. 4 shows the design of the two-layered graph model in which the two layers of nodes represent the customers and the products. There are three kinds of links between the nodes: links between products to depict the similarity between two products, links between customers to describe the similarity between two customers, and links between customers and products to describe the purchase history. Based on this graph model,

three recommendation approaches were designed. In direct retrieval approach, the recommendation is made by extracting products which were earlier bought by the target customer and target customer's neighbours. In associative mining, the recommendation is made by generating association rules on customer's purchase patterns. In High-degree associative retrieval, the model recommends based the interaction power between a product and a customer.

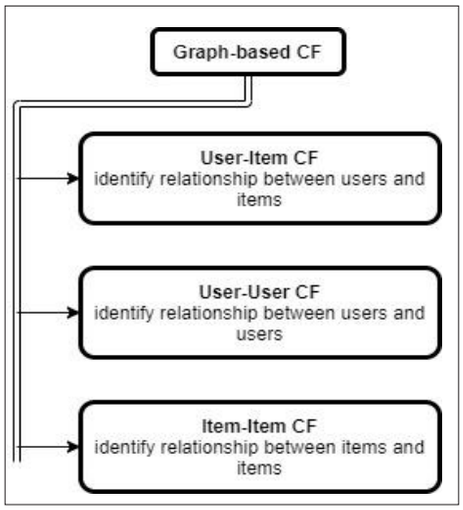

Fig. 3: Classification of Graph-Based CF

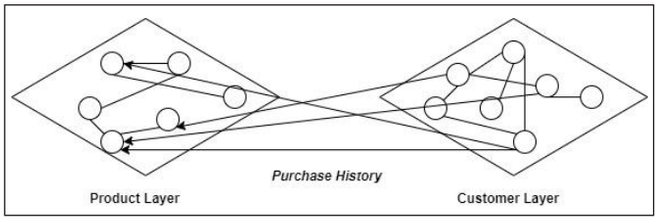

Fig. 4: Two-Layered Graph Model

The problem of data sparsity in a given dataset has been handled in [13] [14] [23] [24]. In [13], the user-item matrix is made dense by exploring the transitive interactions between the items and the users in a graph by applying the associative retrieval technique. On top of that, three spreading activation algorithms including a constrained Leaky Capacitor algorithm, a branch-and-bound serial symbolic search algorithm, and a Hopfield net parallel relaxation search algorithm were applied to make recommendations which proved to better than several other CF approaches.

In 2005, [14] also tried to solve the problem of data sparsity by applying a series of link prediction algorithms on the user-item graph to explore the unobserved links between nodes. Table I shows the six linkage measures listed in the paper. Experiments depict that Katz linkage measure outperforms other linkage measures in terms of performance.

TABLE I: LINKAGE MEASURES

	Linkage Measures	Algorithm
Neighbour-based measures	Common Neighbours	$\left\|\partial(x) \cap \breve{\partial}(y)\right\|$
	Jaccard's Coefficient	$\frac{\left\|\partial(x) \cap \breve{\partial}(y)\right\|}{\left\|\partial(x) \cup \breve{\partial}(y)\right\|}$
	Adamic/Adar	$\sum_{z \in \partial(x) \cap \breve{\partial}(y)} \frac{1}{\log\left\|\partial(Z)\right\|}$
	Preferential Attachment	$\left\|\partial(x)\right\| \cdot \left\|\partial(y)\right\|$
Path-based measures	Graph Distance	Length of the shortest path between pairs of nodes in Graph G
	Katz Measure	$\sum_{l=1}^{\infty} \beta^l . \left\|Paths_{x,y}^{\langle l \rangle}\right\|$ Where $Paths_{x,y}^{\langle l \rangle}$ is all *l* length paths from x to y

In 2012, [23] introduced an algorithm to deal with the problem of data sparsity by applying the dimensionality reduction approach on the bipartite graph. Firstly, the similarity calculation is done between user pairs and item pairs separately using a combination of Pearson Correlation and Jaccard Similarity. Next, clustering is done for users and items separately to reduce the data dimension. Finally, co-clustering is performed on the user cluster and item cluster to assemble similar ones together. A graphical analysis of the above algorithm is shown in Fig. 5.

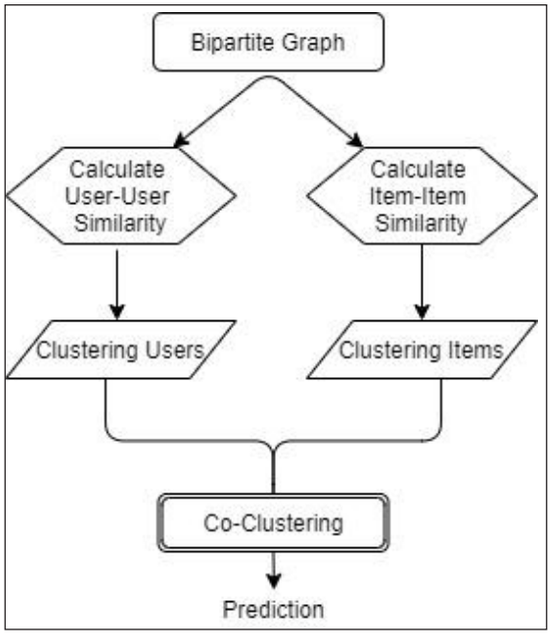

Fig. 5: Graphical Design of the Two-Level Clustering Approach

In 2016, [24] was introduced to tackle the data sparsity problem by improving the similarity measure using the GK-CF algorithm. The algorithm is divided into three phases. GraPA method selects similar users before calculating user similarity based on earlier ratings given by the users. The user similarity is then calculated for the user and his two-degree related users only. Finally, short PA method filters the items on the basis of shortest paths in the graph while maintaining the sorted list of received messages and accepted messages in every iteration.

In the modern world, the size of the dataset is quite large. Transforming the existing datasets in the form of graphs and applying CF on them may require a lot of resources. In 2017, [25] introduced a *divide-and-conquer* approach to handle this problem. In this algorithm which is implemented on Apache Spark employing Lenskit recommender toolkit for creating recommendations, the graph is partitioned to run CF parallelly on smaller graphs which helps in reducing the total execution time and memory requirement. Firstly, the *Social graph* is partitioned in groups of similar users, which, in turn, is used to best partition the *Bipartite graph*. Applying CF on the smaller graphs can limit the usage of resources. A graphical representation of this partitioning scheme is shown in Fig. 6.

[26] and [18] targets the problem of cold-start, which gives a user negative experience about the RS. In 2018, [26] aimed to solve this problem by introducing multi-layered model *SpectralCF* algorithm which is capable of extracting the rich and hidden information in the spectral domains by finding the deep connections between items and users. In 2019, [18] introduces a new model which uses the item-item and user-user similarity to enhance the link prediction in the graphs, and thereby curbing the problem of cold-start from the RS.

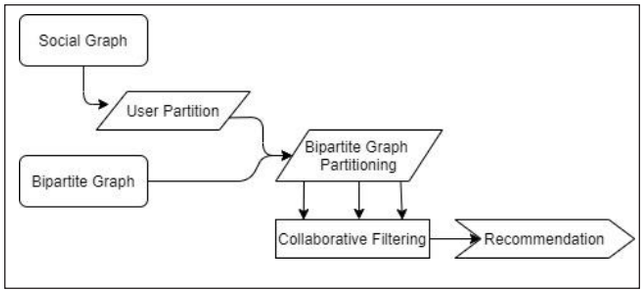

Fig. 6: Workflow of the Partitioning Algorithm

In this world of advertisement and social networking, it is quite important to understand which users will be influenced by which products. In [27], a new socially-aware neural graph CF model (S-NGCF) was designed, to find the set of new users who would adapt an item by learning from the past experience. This neural network model takes the high-order connectivity between users and items to learn feature representation between users and items and the social connections between users.

V. User-User Collaborative Filtering

User-User CF analyses the user-item dataset and then identifies the relationship between users and users. Using this relationship, it recommends items to users. User-User CF relies on the viewpoint of a group of similar users (nearest neighbours) to predict ratings. In 2002, [28] proposed an algorithm to handle the bigger search space. The algorithm makes use of the K-mean clustering technique to cluster similar users into groups. Thus, the recommendation is only performed on the small clusters instead of the whole data set, which eventually reduces the search space and improves the accuracy of the recommendation.

In 2016, [29] tried to address the problem of data sparsity in the Music Recommender System by introducing an algorithm which applies Spectral clustering after improving the similarity between users. The similarity between users is improved by connecting users that were not earlier connected using the random walk method. By applying Spectral clustering, the graph can be sub-divided into groups which gradually reduces the search space during recommendation.

[30] and [31] is focused more upon alleviating the cold-start problem and improving the accuracy of recommendation. [30] works on the idea that *Key-nodes* can make better prediction than aggregation on the set of similar users. In this algorithm, the grouping of users is done by modularity-based algorithms. Partitioning or Clustering algorithm is slightly different from modularity algorithms as overlapping of nodes is not permitted in clustering algorithms, whereas the overlapping of nodes is permitted in modularity algorithms. The main advantage of using modularity algorithms is that it is robust and takes less computation time. The algorithm is divided into five phases, as shown in Fig. 7. In the first phase, a similarity matrix has

been created after computing the similarity between users using the Jaccard similarity measure. The second phase deals with identifying the communities for users using one of the modularity-based algorithms, Louvain community detection algorithm [32]. The third phase handles the selection of key nodes using the *degree centrality* method. Fourth stage profiles the key nodes of every community and fifth stage provide predictions to users in the community.

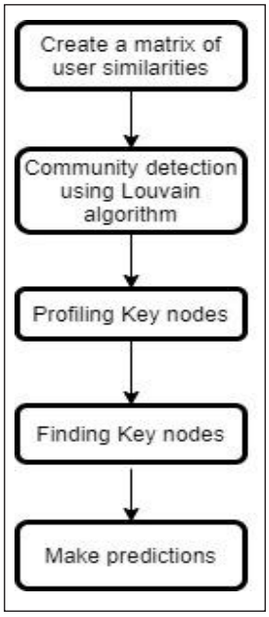

Fig. 7: The Workflow of the CF Model Based on Community Detection

[31] also works similar to the workflow shown in Fig. 7. In this paper, the second phase of the algorithm is tested with a lot of variety of other community detection methods in order to improve the algorithm in terms of accuracy and execution time. Well, after testing the other community detection algorithms on MovieLens and YOW datasets, AHC and K-means Clustering algorithm can be more useful over Louvain algorithm while handling the large datasets.

VI. Item-Item Collaborative Filtering

Item-Item CF analyses the user-item dataset and then identifies the relationship between the items and items. Using this relationship, it recommends items to users. Item-Item CF relies on the ratings given to similar items to predict ratings. In today's generation, in any e-commerce website or movie website, the list of the available items is quite static in comparison to the number of users. Thus, Item-Item collaborative turns out to better option in RS as the system gets freed from calculating similarities at regular intervals which in turn improves the scalability.

In 2012, [33] introduced an approach to tackle the problem of data sparsity and scalability and make prediction much faster than classic RS while maintaining the prediction quality. In the *incremental similarity update* stage, the whole process of calculating similarity is subdivided and calculated independently. The similarity value is updated once there is an addition or update in the ratings, which lets the system scale very well. The *local link prediction* stage handles the data sparsity problem by predicting the links not yet discovered using the Adamic-Adar Coefficient.

In 2013, to deal with the issue of limited resources and time, [34] introduces a new algorithm where clustering is done without depending on the number of users. The Markovian Chain Algorithm uses the random walk method to cluster the graph independent of the size of the dataset keeping the prediction quality unchanged. [35] and [36] made some changes in the similarity measures to build an item-based graph model much superior to the traditional CF model.

Most of the traditional item-item CF algorithms compute the similarity between all the items within the dataset, which increases the execution time of the algorithm to $O(n^2)$. In 2014, [37] introduced an algorithm which helps in reducing the execution time of recommendation to O(n). They presented a fast CF algorithm which uses the k-nearest neighbour graph constructed with greedy filtering. In this approach, nodes whose larger value dimensions do not match are filtered out, and TF-IDF scheme is applied before the construction of the kNN item graph. The execution time is gradually reduced as the recommendation is made on the filtered item graph.

VII. Conclusion

This paper presents a survey of the graph-based collaborative techniques that can be applied to a recommendation system. We categorized some important techniques in the RS literature and showed their implication in the field of RS. We can say that there is drastic development in the field of graph-based CF in the last two decades. Some of the graph-based techniques are being used by big firms like Amazon and Netflix to grow the number of customers on their platform. Well, we can expect more approaches in the field of graph-based CF, especially in the field of Item-Item Collaborative Filtering.

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2015.

[2] 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.

[3] J. Chakraborty, and V. Verma, "A survey of diversification techniques in recommendation systems," *Proceedings of 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pp. 35-40, 2016.

[4] "Amazon." [Online]. Available: https://www.amazon.in/

[5] "Youtube." [Online]. Available: https://www.youtube.com/

[6] "Netflix." [Online]. Available: https://www.netflix.com/in/

[7] "IMDB." [Online]. Available: https://www.imdb.com/

[8] "ACM RecSys." [Online]. Available: https://recsys.acm.org/

[9] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013.

[10] X. Su, and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, article ID 421425, 2009.

[11] V. Verma, and R. K. Aggarwal, "A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: Empirical and theoretical perspective," *Social Network Analysis and Mining*, vol. 10, article no. 43, 2020.

[12] T. Arsan, E. Köksal, and Z. Bozkus, "Comparison of collaborative filtering algorithms with various similarity measures for movie recommendation," *International Journal of Computer Science, Engineering and Applications*, vol. 6, no. 3, pp. 1-20, 2016.

[13] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 116-142, 2004.

[14] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," *Proceedings of ACM/IEEE Joint Conference on Digital Libraries, JCDL,* pp. 141-142, 2005.

[15] B. Chen, and H. Xie, "A context-aware collaborative filtering recommender system based on GCNs," In *2020 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pp. 703-706, 2020.

[16] Y. Lee, D. Park, J. Son, T. Kim, and S. Kim, "Graph-theoretic one-class collaborative filtering using signed random walk with restart," In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 98-101, 2020.

[17] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," *Proceedings of the 12th ACM Conference on Recommender Systems,* pp. 311-319, 2018.

[18] Z. Kurt, A. Bilge, K. Özkan, and Ö. N. Gerek, "On similarity measures for a graph-based recommender system," *Information and Software Technologies*, vol. 1078, pp. 59-75, 2019.

[19] Y. Shao, and Y. Xie, "Research on cold-start problem of collaborative filtering algorithm," In *Proceedings of the 2019 3rd International Conference on Big Data Research*, pp. 67-71, 2019.

[20] S. Das Barman, M. Hasan, and F. Roy, "A genre-based item-item collaborative filtering: Facing the cold-start problem," In *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, pp. 258-262, 2019.

[21] S. S. R. P. K. Reddy, M. Kitsuregawa, S. S. Rao, and P. Sreekanth, "A graph based approach to extract a neighborhood customer community for collaborative filtering," *DNIS '02: Proceedings of the Second International Workshop on Databases in Networked Information Systems*, pp. 188-200, December 2002.

[22] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 3, pp. 259-274, 2004.

[23] E. Hoseini, S. Hashemi, and A. Hamzeh, "A levelwise spectral co-clustering algorithm for collaborative filtering," *In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication,* ACM, p. 6, 2012.

[24] M. Huanyu, L. Zhen, W. Fang, and X. Jiadong, "Towards efficient collaborative filtering using parallel graph model and improved similarity measure," In *Proceedings of 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 182-189, 2016.

[25] C. Sardianos, I. Varlamis, and M. Eirinaki, "Scaling collaborative filtering to large-scale bipartite rating graphs using lenskit and spark," *Proceedings of 2017 IEEE 3rd International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 70-79, 2017.

[26] L. Zeng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," In *RecSys*, pp. 311-319, 2018.

[27] Y. Tsai, M. Guan, C. Li, M. Cha, and Y. Li, *Predicting New Adopters via Socially-Aware Neural Graph*, vol. 4. Springer International Publishing, 2019.

[28] T. H. Kim, Y. S. Ryu, S. I. Park, and S. B. Yang, "An improved recommendation algorithm in collaborative filtering," *Lecture Notes in Computer Science (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2455, LNCS, pp. 254-261, 2002.

[29] V. Gavriilidis, A. Tefas, C. Kotropoulos, and N. Nikolaidis, "Enhanced similarities for a music recommender system," *Proceedings of 2016 Digital Media Industry & Academic Forum (DMIAF)*, pp. 113-116, 2016.

[30] S. Bourhim, L. Benhiba, and M. A. J. Idrissi, "Towards a novel graph-based collaborative filtering approach for recommendation systems," *ACM Int. Conf. Proceeding Ser.*, 2018.

[31] S. Bourhim, L. Benhiba, and M. A. J. Idrissi, "Investigating algorithmic variations of an RS Graph-based collaborative filtering approach," *ACM Int. Conf. Proceeding Ser.*, 2019.

[32] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics Theory and Experiment,* vol. 2008, no. 10, pp. 1-12, 2008.

[33] X. Yang, Z. Zhang, and K. Wang, "Scalable collaborative filtering using incremental update and local link prediction," *ACM Int. Conf. Proceeding Ser.*, pp. 2371-2374, 2012.

[34] E. Cogo, and D. Donko, "Clustering approach to collaborative filtering using social networks," *ICEIEC 2013 - Proeedings of 2013 IEEE 4th International Conference on Electronics Information and Emergency Communication,* pp. 289-292, 2013.

[35] D. T. Lien, and N. D. Phuong, "Collaborative filtering with a graph-based similarity measure," *2014 Int. Conf. Comput. Manag. Telecommun. ComManTel 2014*, pp. 251–256, 2014.

[36] W. Shalaby, et al., "Help me find a job: A graph-based approach for job recommendation at scale," *Proceedings of 2017 IEEE International Conference of Big Data, Big Data January 2017*, vol. 2018, pp. 1544-1553, 2017.

[37] Y. Park, S. Park, S. G. Lee, and W. Jung, "Fast collaborative filtering with a k-nearest neighbor graph," *2014 International Conferene of Big Data Smart Computation, BIGCOMP 2014*, pp. 92-95, 2014.