SNA-Based Recommendation in Professional Learning Environments

Mohamed Amine Chatti, Peyman Toreini, Hendrik Thüs, Ulrik Schroeder

Informatik 9 (Learning Technologies)
RWTH Aachen University, Germany
Email: {chatti;schroeder;thues}@cs.rwth-aachen.de; peyman.toreini@rwth-aachen.de

Abstract—**Recommender systems can provide effective means to support self-organization and networking in professional learning environments. In this paper, we leverage social network analysis (SNA) methods to improve interest-based recommendation in professional learning networks. We discuss two approaches for interest-based recommendation using SNA and compare them with conventional collaborative filtering (CF)-based recommendation methods. The user evaluation results based on the ResQue framework confirm that SNA-based CF recommendation outperform traditional CF methods in terms of coverage and thus can provide an effective solution to the sparsity and cold start problems in recommender systems.**

Keywords–Professional Learning; Recommender Systems; Collaborative Filtering, Social Network Analysis

I. INTRODUCTION

Over the past years, companies and researchers are starting to recognize relationships and intersections between the professional learning and knowledge management fields and to explore the potential and benefits of their integration [1]. Professional learning is no longer regarded as an external online training activity separate from the work flow, but rather as a learner-controlled evolving activity embedded directly into work processes [2]. Professional learning environments are characterized by self-organization and networking. There is a wide agreemnt that effective professional learning environments need to follow a learner-centric model that supports the self-directed learning process by surrounding the professional learner with the environment that matches her needs best. Moreover, there is a crucial need to adopt a learning model that fosters continuous networking and collaborative knowledge creation by enabling professional learners to network, collaborate, and actively participate in a continuous learning process. One concern with a learner-centric model is information overload. Professional learners are lifelong learners who continuously try to create and update their own knowledge environments where they can pull knowledge that meets their particular needs from a wide range of knowledge sources. This is a highly challenging task, given the abundance of information and the complexity of the new knowledge-intensive working and learning environments. It thus becomes crucial to examine some mechanisms that would help professional learners to cope with the information overload problem. This is where recommender systems can play a crucial role to foster self-directed professional learning [3].

Recommender Systems (RS) is an effective solution to deal with the information overload problem. RS track user's actions and provide proper results by identifying information relevant to them, based on their individual interests and preferences. RS are important and successfully implemented in e-business applications. They assist people in different domains like bookshops, research papers, news and articles, web pages, movies, music, etc. In addition to the e-business applications, RS are increasingly being developed in the technology-enhanced learning (TEL) area. They offer a promising approach to facilitate both learning and teaching tasks, by identifying suitable learning resources from a potentially overwhelming variety of choices [4]. There is a large number of RS that have been deployed in TEL settings [5]. However, relatively little significant work around the application of RS in professional learning environments has been undertaken. Traditional RS need adaptation in order to meet the characteristics of professional learning environments. Professional learners rarely share the same or similar learning resources due to the fact that they follow their individual interests and preferences. Thus, recommendation in professional learning environments should rely on the interests of the professional learners and take into account the social networks in which they perform. In this paper, we explore the potentials of RS to support professional learning. We consider using the interest and network informations and incorporating them into traditional collaborative filtering recommendation methods in order to provide effective recommendation of collaborators in professional learning settings.

The remainder of the paper is organized as follows. Section 2 is a review of the related work. Section 3 provides the details of the conducted case study. We then present the study evaluation results in Section 4. And finally, we summarize our findings and outline perspectives for future work in Section 5.

II. FUNDAMENTALS AND RELATED WORK

RSs are used to solve the information overload problem and provide a personalized recommendation. They are software tools and techniques providing a suggestion for items that the user might like by identifying information relevant to users [6]. The relevant information of the users are collected by tracking their actions and follow user's profile in different environments. The first paper on collaborative filtering (CF), as one of the first recommendation techniques, published in the mid-1990s and since then this topic became popular for RS researchers in different fields [7]. One of the first implementation of RSs is Amazon's RS which is used to recommend relevant products to the users [8]. Another early application that focused on numerical scores and mathematical treatment was Movielens [9].

Technically, RS are classified into three main classes, based on how recommendations are made, namely collaborative filtering (CF), content-based recommendation (CB), and hybrid recommendation [7]. We will focus further only on the CF recommendation techniques as they are the main interest for this paper.

There exists different types and techniques to provide recommendation by means of RSs. In this article, we focus on investigating the performance of memory-based CF, Trust-based CF and two types of SNA-based RS. In the following sections, background of these techniques are discussed.

A. *Collaborative Filtering RS*

In Collaborative filtering (CF) RS, the user will be recommended items that people with similar tastes and preferences liked in the past. CF RS have two primary tasks, first rating prediction in which they predict the rating of the target user for the target item [10]. The second one is Top-N recommendation in which they predict the top-N highest-rated items among the items not yet rated by the target user [10].

CF RS can be classified in two categories: memory-based and model-based CF techniques. Memory-based CF is a RS method that makes a prediction about interests of a user by collecting information from many other users. This information includes the list of favorites and their ratings. In the memory-based version, similar users or similar items are detected and missing ratings are predicted based on those. If the similarity is computed based on users, it is called user-based CF and if it is based on items, it is called item-based CF. User-based methods predict the rating $r_{u,i}$ of a user u for an item i using the ratings given to i by other users who are most similar to u. These similar users are called nearest-neighbors and k-nearest-neighbors (k-NN) of u are k users with the highest similarity to u. There are different ways to calculate the correlation of users in memory-based CF such as Pearson, Cosine, Jaccard similarity. After computing the correlation between users and find k-NN neighbors of the user u, the prediction procedure starts. User-based CF algorithms can predict the rating of a user in an item that has not rated or recommend a list of items as top-N items.

Model-based CF is another type in CF RS. Algorithms based on this type learn a model by means of the collection of ratings and then make rating predictions. Memory-based CF algorithms are more popular than model-based versions as they are simple and do not have the complexity of the model-based algorithm. Furthermore, there are accurate when sufficient data is available. On the other hand, there are shortcomings for memory-based CF. These include [10]:

- Cold start: The cold-start problem refers to the problem of giving an accurate recommendation to a user who is new in the system. In user-based CF, it causes the problem with computing similarity between users, as some of them do not have any rating.
- Sparsity: The user-item matrix can be vast and sparse. The number of users who would have rated the same set of items would be quite less.
- Scalability: There can be millions of users in the database, memory-based CF system needs to compare the target user to all other users to find similar users. Doing recommendation is computationally expensive in this case.

To overcome the limitations of traditional CF recommendation, some research has been done recently in trying to generate recommendations by harnessing the user's social network information. Formally, these approaches introduced new type of RS, namely Trust-based CF and SNA-based RS.

B. *Trust-Based RS*

The trust has different meanings in different domains. In the field of RS, the trust-based approach assumes a trust network among users and makes recommendations based on the rating of the users that are directly or indirectly trusted by the active user. These approaches consider only the information provided by the user's trusted neighbors [11]–[14]. One of the conventional approaches of trust-based RSs is using trust over CF approach [15]. Traditional CF RS find neighbors by computing the similarity between users. By involving social relationships to the CF, the neighbors of the active user can be selected by trust relations instead of similarity [15]. The number of these neighbors is dynamic for different users as they have a different number of trust relationships. The recommendation results of these methods show that the performance of traditional RSs can be improved by utilizing trust relations [16]. Moreover, using trust-based CF will solve the problem of scalability in memory-based CF by means of clustering. However, the cold start and sparsity problems are not addressed by means of this method. In fact, it is still hard to recommend items to users without or with fewer ratings.

C. *SNA-Based RS*

One of the ways to improve the performance of CF RS is using social network analysis (SNA) [17]. SNA explores relationships among entities of networks and analyzes their features. There are several metrics available for SNA. These metrics help to identify the important node in a given social graph such as: Degree, Betweenness, Closeness, Eigenvector [18]. SNA-based RS algorithms apply the social influence of the users on the procedure of recommendation to improve its accuracy. This technique has the ability to provide the recommendation for cold start users as long as they are connected to the social network. Several SNA-based RS approaches have been proposed in the literature [19]–[30]. In general, these approaches use social networks as the source of information to generate recommendations. However, none of these approaches build social networks based on the user's interests.

III. STUDY

In this paper, we focus on leveraging network and interest information to generate effective recommendation of collaborators in professional learning environments. As a proof of concept, we addressed in this work academic networks for two reasons. First, academic researchers are lifelong professional learners who continuously network for research purposes. Second, interests can be easily gathered from researchers' publication activities. To note, however, that the concepts discussed in this paper are valid in other professional learning environments where it is possible to build interest-based social networks. We conducted a study to investigate SNA-based recommendation methods and compare their performance with both user-based CF and Trust-based CF.

A. PALM

This study was conducted based on a dataset generated within the PALM environment. PALM is a personal academic learner modeling service that combines the web, text, and interest mining techniques to create a learner model based on the collected publication information. It has three main tasks to do. First, mining interests of a researcher based on the researcher's published papers. Second, assigning a score to each stored interest of the researcher. This score is assigned based on the level of researcher's concern toward the subject. Third, storing publications of target researcher to identify co-authors in further steps. As a result, PALM provides a user-item matrix, where users are researchers around the world, items are research topics, and the scores show the levels of interest of the researchers in the topics. Additionally, since it finds publications of a researcher, co-authors of a particular researcher and information about them are also accessible [31].

The PALM dataset includes 754 researchers, 14.208 individual interests, and 40.674 publications. 169 of the stored researchers do not have any collected interests and can be seen as "cold start researchers". In this study, the minimum requirement to provide recommendations to the researchers is to have their co-authorship information.

B. Architecture and Implementation

The system architecture of the PALM recommender system is shown in Figure 1. As it can be seen, there are four primary components in this architecture. The first component is an application service with the core of Spring MVC [32]. Spring MVC has the responsibility of handling requests for recommendations. These requests are originated from an active researcher and are sent through RESTful services to Spring MVC. After that, Spring MVC starts to retrieve the required information by querying the PALM database based on the chosen recommendation method. The next step is sending a file that consists of the required information to the modules of the recommendation engine, namely Gephi Toolkit [33] and Apache Mahout [34]. The sent files have different contents based on the intended task. For the recommendation that applies user-based collaborative filtering, Mahout manages the whole recommendation procedure and sends the result back to Spring MVC. For SNA-based recommendations, Gephi computes the centrality measures and sends the result to the implemented services in Spring MVC for providing the final list of recommendations. These services are responsible for computing Jaccard similarity, add the result of SNA centrality measure to them, and offer a list of recommendation. Next, the outcome of recommendations is converted to JSON format by Spring MVC and is sent to the recommendation interface. The provided interface, as the fourth component, handles the task of visualizing recommendation data to the active researcher.

C. Recommendation Generation

In this study, we implemented four different CF recommendation algorithms: User-based CF, Trust-based CF and 2 different ways of SNA-based CF, as listed in Table I. In the following sections, we discuss the implementation of the proposed algorithms in some detail.

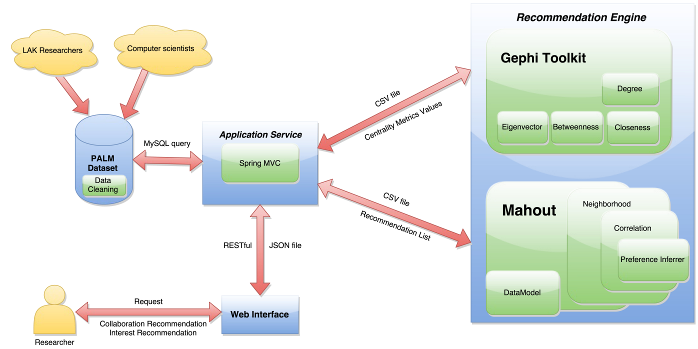

Figure 1. PALM Recommender System Architecture

TABLE I. IMPLEMENTED ALGORITHMS FOR COLLABORATION RECOMMENDATION

Rec. No	Rec. Method
1	User-based CF
2	Trust-based CF
3	SNA-based RS on Co-authorship Network
4	SNA-based RS on Interest Network

1) User-based CF: This algorithm uses the entire user-item matrix of the PALM dataset to find similar users. To do recommendation, first a file that represents information about users and their associated preferences for items is created. In this file, each line has the format of "userID, itemID, value" that is assigned to a particular researcher. UserID means the ID of a researcher, itemID shows the interest ID that is saved in the PALM database and value means the score assigned the researcher's interest. A service in Spring MVC is responsible for creating this file by a query to the MySQL database. To apply user-based CF for a researcher, the created file is used by Mahout to find 10 researchers who have similar interests with the active researcher. The recommended researchers are not from the list that the active researcher published a paper with them before.

2) Trust-based CF: In this algorithm, the trust information of co-authorship networks is used to implement trust-based CF recommendation. Here, researchers are clustered based on information of indirect co-authors in 2-depth co-authorship networks to identify the trust network of an active researcher. For a particular researcher, the first depth of the co-authorship network contains his direct co-authors and the second depth contains his indirect co-authors; i.e. the co-authors of his co-authors. To provide a recommendation based on this category, a file that represents information about indirect nodes in 2-depth co-authorship network of a particular researcher is created by a service in Spring MVC. This file represents information about researchers and their associated preferences for items in the same format "userID, itemID, value" as above. As compared to the user-based CF, only researchers from the trust network are listed in the file. This file builds the input for collaboration recommendation computation, which is done by Mahout. The recommendation of potential collaborators for an active researcher is then collected from the list of researchers who are in the second depth of the co-authorship network.

3) SNA-based RS on Co-authorship Network: The primary component of this algorithm is again the 2-depth co-authorship network of the active researcher. Two elements affect the result of this recommendation method. One is the degree network centrality measure and the other one is Jaccard similarity. We combine these elements to recommend potential collaborators, as defined in the formula below. $P(u, v)$ means the prediction value for recommending user v to the user u.

$$P(u, v) = C(v)_{degree} + Sim(u, v)_{Jaccard} \tag{1}$$

The procedure of recommendation for this algorithm starts by creating a file. This file includes the information of the 2-depth co-authorship network of the active researcher. This file is saved in a CSV format, and each line of this file contains two elements that are separated by a comma: the researcher ID and ID pairs showing the co-authorship relation between two researchers. This file build the input for the SNA tool Gephi. By default, graphs that are imported to Gephi from a CSV file are directed graphs but the co-authorship networks are logically undirected networks as if two researchers published a paper with each other, both of them has a connection to each other. Therefore, before computing centrality measures based on the imported graph by Gephi, the undirected graph option is selected.

The second step in this algorithm is to compute centrality metrics using Gephi toolkit. The implemented centrality metrics algorithm in Gephi finds the result of degree centrality measure for each node in the graph. After that, the result is normalized and is saved for further computation. The normalization for the degree is computed as the degree value divided by the number of all nodes in the network minus one.

The third step is to compute Jaccard similarity between the interests of the active researcher and those of the other researchers in the graph. Cold start researchers have the Jaccard similarity of zero or very low with all the other researchers in the network. At the end, the result of Jaccard similarity is normalized and is saved for further computation.

The computed degree centrality and Jaccard values are computed for each node in the graph. The fourth step is to add these two elements together and compute an individual value for each node in the graph. The result is then sorted from highest to lowest value.

At the end, top ten researchers in this graph are listed for collaboration recommendation. This list should be new for the active researcher. Therefore, the recommended list should be checked with the list of direct co-authors before recommending. If someone from the co-author list exists, then this item is replaced with the next top rated researcher.

4) SNA-based RS on Interest Network: The primary component of this algorithm is the interest network. This network is created based on two aspects, co-authorship relationships and correlation between interests. To create an interest network, first, the 3-depth co-authorship network of the active researcher is built and all nodes in this graph are detected. After that, the five top interests of each node are picked. The interests that occur together for a researcher are then connected to each other to form the final interest graph. An example of such graph can be seen in Figure 2. The first step for recommendations in this algorithm is to create a file that contains information about relations between interests in the created interest network. Each line of this file has the structure of "interestID,interestID" reflecting two interests that occur together. The interest network is logically an undirected graph. Therefore, Gephi is set to realize two-way connections between nodes in this graph. In this algorithm, first, a list of interests based on degree centrality in interest network is computed. The first ten interests that have the highest degree values in the graph and that do belong to the actual interests of the active researcher are selcted as potential interest list. The collaboration recommendation list is then selected from the indirect co-authors in the 3-depth co-authorship network of the active researcher based on the potential interest list. For each interest in the potential interest list, the indirect co-author who has the top score for this interest is recommended.

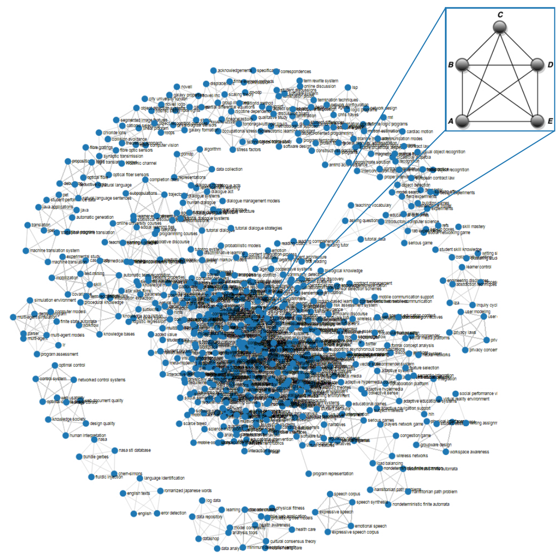

Figure 2. Example of Interest Network

IV. Evaluation Results

We conducted a user evaluation on the four recommendation methods, discussed in the previous sections. Eight users participated in this evaluation and all have a profile on PALM. The particpants in this study are Ph.D. students, professors and researchers at RWTH Aachen University. All of them have published at least one research and thus have a co-authorship network.

A. Evaluation Methodology

A questionnaire is prepared to do user evaluation based on the ResQue (Recommender systems quality of user experience) framework for user-centric evaluation of recommender systems [35]. A wide variety of questionnaire statements are provided in the ResQue framework. Eight sample questions from the suggested questionnaire statements are selected. Researchers are asked to assign their answer based on 1-5 Likert scale while answering questions. In this Scale, "Strong disagree" means number 1 and "Strong agree" means number 5. The category of questions can be seen in the following list.

1) Ability to recommend: The system can provide a recommendation for me (Coverage). (Y/N)
2) Accuracy: In my opinion, the system can recommend to me 1-3 / 4-6 / 7-10 relevant Collaborators.

3) Relative accuracy: The recommendation I received better fits me than what I may receive from a colleague
4) Novelty: The collaborators recommended to me are novel and interesting
5) Diversity: The collaborators recommended to me are diverse
6) Context Compatibility: The collaborators recommended to me considered my personal interests.
7) Perceived usefulness: I feel supporting to find proper new collaborators with the help of recommender.
8) Attitudes: Overall, I am satisfied with the recommender.

B. Discussion of The Results

A summary of the average scores per question and per algorithm are given in Figure 3. The first question investigates the sparsity and cold start problems by measuring coverage. The result of coverage performance checks the possibility of providing a recommendation list, not the quality of recommended items.

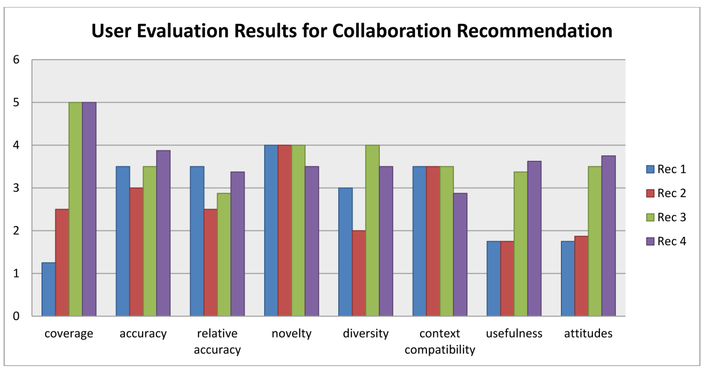

Figure 3. User evaluation average results for collaboration recommendation

The ranking of the examined algorithms based on coverage performance are presented in Figure 4. As it can be seen, SNA-based RS on co-authorship network and SNA-based RS on interest network have the highest coverage value and they could provide recommendations for most of the participants. These results show that the SNA-based RS methods can indeed provide a solution to the sparsity and cold start problems in CF RS.

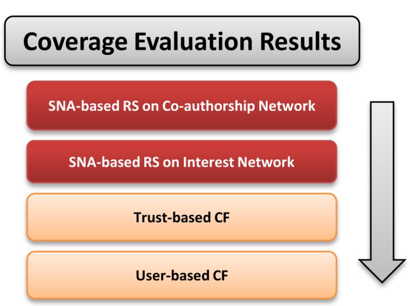

Figure 4. Ranking of Recommendation Algorithms based on Coverage

The evaluation results of accuracy, relative accuracy, diversity, and usefulness show that the SNA-based RS algorithms perform better than the trust-based CF while, in novelty and context compatibility, they performed with almost the same results. Moreover, the results of this evaluation show that not all SNA-based RS algorithms perform better than user-based CF algorithm. They performed equally or worse in accuracy, relative accuracy, novelty and context compatibility. The user-based CF algorithm performed equally or better than trust-based CF algorithm in all aspects except coverage and attitude.

In the attitude factor, the users were asked to evaluate their level of satisfaction based on previously asked factors (see Figure 5). The result shows that in general the evaluators were satisfied with the algorithm SNA-based RS on interest network, more than the other algorithms. SNA-based RS on 2-depth co-authorship network was evaluated far better than CF and trust-based CF and a little lower than SNA-based RS on interest network. Moreover, it can be seen that the level of satisfaction provided with the Trust-based CF algorithm achieved a slightly better result than the CF algorithm.

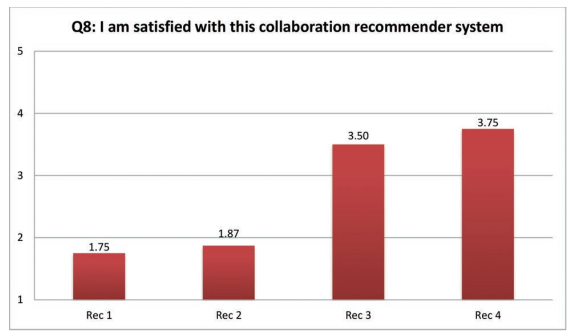

Figure 5. Results of Q8 - Satisfaction with the Recommender System

V. Conclusion and Future Work

In this paper, we addressed recommendation in professional learning settings. We incorprated the interest and network informations and leveraged social network analysis (SNA) methods to deal with the sparsity and cold start problems in traditional collaborative filtering (CF) recommeder systems. We conducted a study to investigate SNA-based recommendation methods and compare their performance with both user-based CF and Trust-based CF. The user evaluation results based on the ResQue framework confirmed that SNA-based CF recommendation outperform traditional CF methods in terms of coverage and thus can provide an effective solution to the sparsity and cold start problems. The evaluation further proved that using SNA metrics in recommender system could improve the performance better than using social networks as the source of trust information for CF.

While our early results are encouraging for generating effective recommendations in professional learning environments based on interest and network information, there are still a number of areas we would like to improve. This work focused on collaboration recommendation while other types of recommendation like interest recommendation can be explored as well. Moreover, an offline evaluation based on precision, recall and f-measure can be done and compared with the results of user evaluation. Additionally, in addition to degree centrality, we can inverstigate other types of network centrality metrics such as closeness, betweenness and eigenvector.

Furthermore, other types of networks, such as 3-depth co-authorship networks can be added as a source of information for trust-based and SNA-based CF recommendation.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the German Federal Ministry of Education and Research through the PRiME Project (https://prime.rwth-aachen.de/).

REFERENCES

[1] M. A. Chatti, H. Thüs, C. Greven, and U. Schroeder, "Layered knowledge networking in professional learning environments," in Proceedings of the 7th International Conference on Computer Supported Education (CSEDU 2015), vol. 2, 2015, pp. 363–371.

[2] M. A. Chatti, U. Schroeder, and M. Jarke, "Laan: convergence of knowledge management and technology-enhanced learning," IEEE Transactions on Learning Technologies, vol. 5, no. 2, 2012, pp. 177–189.

[3] M. A. Chatti, S. Dakova, H. Thüs, and U. Schroeder, "Tag-based collaborative filtering recommendation in personal learning environments," IEEE Transactions on Learning Technologies, vol. 6, no. 4, 2013, pp. 337–349.

[4] K. Verbert et al., "Dataset-driven research for improving recommender systems for learning," in Proceedings of the 1st International Conference on Learning Analytics and Knowledge, LAK 2011, 2011, pp. 44–53.

[5] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in Recommender Systems Handbook, L. R. P. B. Kantor, F. Ricci and B. Shapira, Eds. Springer, 2011, pp. 387–415.

[6] F. Ricci, L. Rokach, and B. Shapira, Introduction to recommender systems handbook. Springer, 2011.

[7] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, 2005, pp. 734–749.

[8] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," Internet Computing, IEEE, vol. 7, no. 1, 2003, pp. 76–80.

[9] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999, pp. 230–237.

[10] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.

[11] P. Massa and P. Avesani, "Trust-aware recommender systems," in Proceedings of the 2007 ACM conference on Recommender systems. ACM, 2007, pp. 17–24.

[12] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 397–406.

[13] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2009, pp. 203–210.

[14] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008, pp. 931–940.

[15] K. Kwon, J. Cho, and Y. Park, "Multidimensional credibility model for neighbor selection in collaborative recommendation," Expert Systems with Applications, vol. 36, no. 3, 2009, pp. 7114–7122.

[16] M. C. Pham, Y. Cao, R. Klamma, and M. Jarke, "A clustering approach for collaborative filtering recommendation using social network analysis." J. UCS, vol. 17, no. 4, 2011, pp. 583–604.

[17] M. Jamali and M. Ester, "Mining social networks for recommendation," Tutorial of ICDM, vol. 11, 2011.

[18] S. Wasserman and K. Faust, Social network analysis: Methods and applications. Cambridge university press, 1994.

[19] C. Lam, "Snack: incorporating social network information in automated collaborative filtering," in Proceedings of the 5th ACM conference on Electronic commerce. ACM, 2004, pp. 254–255.

[20] J. Li, F. Xia, W. Wang, Z. Chen, N. Y. Asabere, and H. Jiang, "Acrec: a co-authorship based random walk model for academic collaboration recommendation," in Proceedings of the companion publication of the 23rd international conference on World wide web companion. International World Wide Web Conferences Steering Committee, 2014, pp. 1209–1214.

[21] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Collabseer: a search engine for collaboration discovery," in Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries. ACM, 2011, pp. 231–240.

[22] T. Huynh et al., "Scientific publication recommendations based on collaborative citation networks," in Collaboration Technologies and Systems (CTS), 2012 International Conference on. IEEE, 2012, pp. 316–321.

[23] E. Spertus, M. Sahami, and O. Buyukkokten, "Evaluating similarity measures: a large-scale study in the orkut social network," in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. ACM, 2005, pp. 678–684.

[24] G. R. Lopes, M. M. Moro, L. K. Wives, and J. P. M. De Oliveira, "Collaboration recommendation on academic social networks," in Advances in Conceptual Modeling–Applications and Challenges. Springer, 2010, pp. 190–199.

[25] J. Palau, M. Montaner, B. López, and J. L. De La Rosa, "Collaboration analysis in recommender systems using social networks," in Cooperative Information Agents VIII. Springer, 2004, pp. 137–151.

[26] E. Davoodi, M. Afsharchi, and K. Kianmehr, "A social network-based approach to expert recommendation system," in Hybrid Artificial Intelligent Systems. Springer, 2012, pp. 91–102.

[27] E. Castillejo, A. Almeida, and D. López-de Ipiña, "Social network analysis applied to recommendation systems: Alleviating the cold-user problem." in UCAmI. Springer, 2012, pp. 306–313.

[28] B. Fan, L. Liu, M. Li, and Y. Wu, "Knowledge recommendation based on social network theory," in Advanced Management of Information for Globalized Enterprises, 2008. AMIGE 2008. IEEE Symposium on. IEEE, 2008, pp. 1–3.

[29] B.-R. Dai, C.-Y. Lee, and C.-H. Chung, "A framework of recommendation system based on both network structure and messages," in Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on. IEEE, 2011, pp. 709–714.

[30] C. Yang, J. Sun, and Z. Zhao, "Personalized recommendation based on collaborative filtering in social network," in Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on, vol. 1, Dec 2010, pp. 670–673.

[31] M. A. Chatti, D. Dugošija, H. Thüs, and U. Schroeder, "Learner modeling in academic networks," in Proceedings of The 14th IEEE International Conference on Advanced Learning Technologies (ICALT 2014), 2014, pp. 117–121.

[32] Spring framework. Last accessed March 14, 2016. [Online]. Available: http://spring.io/

[33] Gephi. Last accessed March 14, 2016. [Online]. Available: http://gephi.github.io/toolkit/

[34] Mahout. Last accessed March 14, 2016. [Online]. Available: http://mahout.apache.org/

[35] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in Proceedings of the fifth ACM conference on Recommender systems. ACM, 2011, pp. 157–164.