Insights on Social Recommender System

Wolney L. de Mello Neto
Vrije Universiteit Brussel
CoMo Lab
Brussels, Belgium
wdemello@vub.ac.be

Ann Nowé
Vrije Universiteit Brussel
CoMo Lab
Brussels, Belgium
ann.nowe@vub.ac.be

ABSTRACT

Recommender Systems (RS) algorithms are growing more and more complex to follow requirements from real-world applications. Nevertheless, the slight improvement they often bring may not compensate the considerable increase in algorithmic complexity and decrease in computational performance. Contrarily, context aspects such as social awareness are still not much explored. In view of that, this paper proposes insights on how to possibly achieve more efficient and accurate predictions for recommendations by exploring multiple dimensions of a RS architecture. A framework is designed, comprised of a Facebook application called *MyPopCorn* and some scenarios of user neighborhood RSs are proposed. The first one investigates how to recommend movies based on a narrowed subset of collaborative data, extracted from the social connections of the active user. Secondly, connections between users enable a solution for the cold-start problem. Preferences from social connections are aggregated, producing a temporary profile of the new user. Finally, a third dimension is explored regarding evaluation metrics. Results from traditional evaluation by offline cross-validation are compared to measuring prediction accuracy of online feedback data. These insights propose how community-based RS designs might take advantage of social context features. Results show that all three proposed solutions perform better assuming some conditions. Social neighborhoods can often provide representative data for collaborative filtering user-neighborhood techniques, improving a lot the RS performance in terms of computational complexity metric without compromising prediction accuracy. Assuming a user has a dense social network, the cold-start problem can be easily tackled. Finally, rating prediction accuracy performs better when evaluated online than by offline cross-validation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Collaborative Filtering; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Recommender System, Collaborative Filtering, Social Recommenders, Cold-Start Problem, Evaluation Metrics

1. INTRODUCTION

Our generation faces several tough challenges within the current *peta-*, *exa-* or even *zettabyte* information era. Every day we deal with huge amounts of information whose manipulation and storage struggles even on high-end computer technologies. Shifting from the point of view of computer capacity to an average single person, the problem gets even worse due to human being limitations. Online services are examples of big data resources with increasing importance in our lives. About two years ago, Google's search engine used to process approximately half of the entire written works of mankind per day [6]. Nowadays, it is impossible to avoid such reality while working, studying, and entertaining yourself. Perhaps this information overload comes with high cost, nevertheless, high benefit as well.

Movie domain is a great context where information overload is a high potential pain point to be explored. Moreover, Netflix movie streaming service is a good motivation for this work due to two main reasons. Firstly, figures disclosed in [1] mention 75% of their sales come from recommendations. Secondly, [1] reveals the decision of not implementing commercially the algorithm with around 10% improvement in prediction accuracy, winner of US$ 1 million prize[8]. Taking these facts into account, what would be the most potential path to explore within the field of RSs? Is accuracy the most important metric to take into account? What about computational complexity and transparency? What about online instead of offline evaluation methods?

Rather than building upon complex RS methods, this paper investigates a social framework for developing state-of-the-art RS. Aiming at current main challenges, this paper proposes contributions on how to tackle some of its most relevant issues based on possibilities enabled by social context information. The three explored RS challenges are: *(i)* performance issues related to scalability of recommender systems; *(ii)* lack of knowledge about new users, known as cold start problem; and *(iii)* definition of good evaluation methods.

Some insights are discussed based on how social-graph data enable a good implementation of a user neighborhood RS algorithm, focusing not only on prediction accuracy but also on other metrics such as scalability, computational complexity and transparency. These insights lead to 3 hypotheses listed below:

i. A user's social neighborhood is sufficiently representative to provide efficient, in the sense of computational complexity, and effective recommendations, in terms of prediction accuracy;

ii. Social neighborhood connections can derive assumptions about new users taste, avoiding the cold-start problem;

iii. Online evaluation of transparent recommendations should be a valid metric within social RSs.

2. RELATED WORK

In the introduction of the latest survey in RS field, [15] highlights current challenges for RSs. Some of them are investigated hereby, such as follows:

Scalability In real-world applications, the number of instances might often steeply increase in multiple dimensions such as number of users, items and, in turn, user-item preference signals. Despite being a good scenario for some RS algorithms to achieve better accuracy, bigger datasets may lead to a great increase in computational complexity.

[7] proposes an evaluation of top-N recommendation algorithms. Item-based RS is proposed as an alternative for non-scalable user-based recommenders, since it performs better when there are many more users than items. Some other item-based RSs avoiding scalability problems within memory-based CF algorithms are compared in [16].

Regarding model-based CF techniques, [17] follows a reasoning that is similar to the solution presented in Section 4.1, since both look for a narrowed neighborhood which does not to compromise general performance. Whereas the cited papers are based on clustering techniques, our heuristic consists of narrowing the database to a subset of user social-graph connections. Although scalability is an intrinsic disadvantage to user-based RS, the proposition of a local neighborhood might overcome this drawback. User-based RS is adopted since it enables some features related to the social RSs, such as transparent explanations for each recommendation;

Data Sparsity It is among the main bottlenecks for RSs. The lack of information is a big problem, especially during first interactions of a new user. This scenario is defined as the cold start or new user problem, which is traditionally solved by requiring initial user information before any recommendation is given. Nevertheless, this interaction is time consuming, since the user has to look for a couple of items to rate. To improve that, [14] has compared 6 techniques to generate this first list of items, aiming to maximize the percentage of rated items out of all items presented to a new user.

Besides requiring this first interaction with the RS, one could think of a temporary user profile in order to enable initial recommendations. [11] explores trust networks and propose the incorporation of preferences from trusted users. Nevertheless, the new user still has to explicitly provide information about who are his/her trusted users. Our work retrieves implicit information from social networks, regardless trust measurements. The method consists of retrieving social connections and building a virtual profile based on aggregation methods, originally proposed for group RSs. [13] describes 10 aggregation methods and empirically concludes that social-based think is the best basis for generating an artificial preference profile. The author claims that *Least Misery*, *Average* and *Average without Misery* are the most human-like reasoning techniques, achieving very good results.

Transparency Users eventually question themselves about the reasoning behind a recommendation. They are more inclined to accept and evaluate better once they understand how an item has been suggested to him or her. Nevertheless, it is not always possible to provide such a transparent explanation. [9] presents a survey on content-based RS and compares them to CF techniques also in terms of transparency. The authors claim CF techniques are a black box, and it is indeed the truth for most cases. In the case of user-neighborhood RSs, although RSs could tell to the active user about people with close taste that influenced the recommendation, privacy issues may not allow such transparency. In view of this challenge, this paper counteracts the affirmation made by the previously cited survey. It is possible to give explanation on user-based collaborative filtering technique once one assumes not having privacy issues, a tractable scenario within social networks, where connections previously agree on sharing some information. Besides this proposal, some solutions to tackle CF limitations related to transparency are proposed in [4].

Evaluation One of the main modules of a RS design, evaluation strategy is a critical and subjective aspect to be shaped throughout the whole process of building and maintaining a RS. Even though most papers adopt accuracy as the most important metric, one should consider many other evaluation criteria, as presented in [5]. Computational complexity is one metric highlighted in the insight presented in Section 4.1. Transparency is enabled by social context, as discussed in Section 3.1.3. Besides exploring metrics, this paper also focus on questioning methods (see Section 4.3). Offline and online methods should be compared while measuring rating prediction accuracy.

2.1 Social Recommenders

In view of all issues previously listed and the fact some state-of-the-art architectures might not be that attractive for commercial purposes, this paper dives into a RS design that is gaining special attention: Social RSs. Also called community-based recommenders, the basic architecture embeds context data into either collaborative filtering or content-based algorithms, improving the RS performance. According to [15], community-based paradigm is still a hot topic and it is not possible to find a consensus about whether social recommenders have better performance. [19] presents a broad survey on social recommenders. One could see social data in two ways: *(i)* unweighted social graph; *(ii)* or a more complex weighted social-graph. The former has been selected for this paper experiments based on empirical conclusions made by [2] while comparing CF and Social Filtering. Similarities between friends were in average higher than the same correlation measurement between non-connected users. Moreover, both weighted and basic social RSs performed the same or better than pure collaborative filtering RSs for the referred case.

Further than looking at social connections, the latter is

Table 1: **MyPopCorn and GroupLens datasets.**

	Users	Ratings	Movies
MyPopCorn	129	14k	3k
GroupLens	72k	10M	10k

a trust-based RS that focuses on weighted relationships. A clear comparison between social RS and trust-based RS is defined in [10]. Moreover, [3] highlights the possibility of explaining recommendations based on social connections and the fact active users rate better the RS in case of existing such transparency. Finally, the social RS described hereby profits from an unweighted social graph.

3. FRAMEWORK

As claimed in [15, pg 15], the context in which a RS is developed and its expected features determine the optimal algorithm to be adopted. Parameters such as movie domain, social community context, rating strategy and sparse data were definitely crucial to come up with the final architecture described hereby. A Facebook application called *MyPopCorn*[1], the RS front-end, and a social based implementation of user neighborhood CF algorithm compose the current framework, to be presented in the two following sections.

3.1 MyPopCorn, a Facebook app as Front-End

The idea of building this movie recommender system and making it available on a social network is due to the fact social graph enables proposed recommendation experiments based on social neighborhoods. Moreover, the capability of recommending to an active user and receiving an online feedback on rating prediction accuracy on recommended items is decisive to benchmark the implemented algorithms.

MyPopCorn is a web movie recommender system. Some of its interfaces are composed as follows:

First screen presents a brief description of the main features before the user joins the application. After that, an active user can check statistics about top users and friends;

MyTaste is where a user can rate movies. Recommendation-wise, this is one of the main interactions with the user, in which RS collects data;

My Friends' Taste presents a list of friends and their respective number of ratings. The more ratings each friend has, the bigger his or her basket gets.

3.1.1 Social-Graph Data

The first collaborative data with ratings over movies were taken from GroupLens 10M dataset. From that point, the database was increased with ratings from users of *MyPopCorn*. Information about users, friendships are also made persistent into the same database. The dataset used for the experiments is summarized in Table 1.

In a very short timeframe, the application was accepted by a good number of users. Almost 130 active users have been exploring the application during 2 months time. Figure 1 illustrates all users who contributed for the experiments carried out into this paper. The more movies a user rates, the bigger the node is represented in the social graph. The average degree of connections in this graph was 10.543.

[1]http://mypopcorn.info/

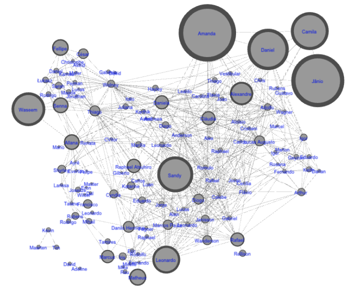

Figure 1: Social Graph representation of *MyPopCorn* database.

3.1.2 Rating Strategy

In *MyPopCorn*, the user can choose a rating from 1 to 5 'stars'. Asymmetric labels were defined for each of the 5 stars to achieve a more homogeneous judgment, namely *Bad*, *Regular*, *Good*, *Great* and *Masterpiece*. Test users reported good feedback on the proposed rating strategy claiming this discrete labeled design is certainly more intelligible, where users can have a hint of what each rating value may represent. While following such design, this research aims at reducing subjectivity that is intrinsic to rating process, the core interaction responsible for obtaining the main input of a Collaborative Filtering RS. This strategy also prevents the necessity of the RS to normalize user ratings.

3.1.3 Recommendation Strategy

Recommendations are generated from two implementations of user neighborhood recommenders, such as follows:

- Provided by a traditional user-based RS. The neighborhood calculated among all users in the database;
- Provided by a social-graph user-based RS. A social neighborhood is based on the set of active user friends, to be described in more details in the next section.

A shuffled list of recommendations generated by both RS implementations is presented to the user. Movie description and a continuous predicted value is presented. Therefore, recommendations are seen as a regression and not a classification problem within this framework. Finally, at the bottom of the frame one can see the explanation about each recommendation(see Figure 2). In the first example on light blue background, a message informs the recommendation was *"Based on all MyPopCorn database"*. Alternatively, the second message informs that is was *"Based on friends with closest taste"*, followed by the list of users *Friend X* and *Friend Y*.

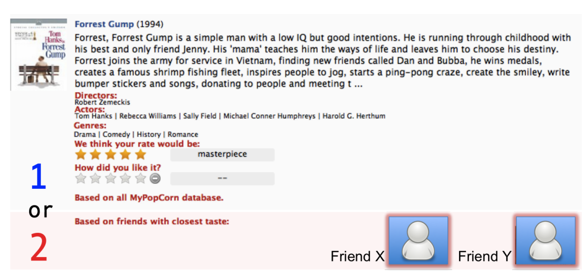

Figure 2: Recommendation strategy in *MyPopCorn*.

This system is designed to give the most transparent recommendations possible. In view of that, the reasoning behind the RS can be better understood by presenting the real number as predicted rating value. Furthermore, explaining the recommendation with a list of users will transform a formerly impersonal recommendation into a social passive interaction between friends. Due to privacy issues, presenting this list is only possible for the social neighborhood approach, where content sharing among users is agreed in advance.

3.2 Movie RS Back-End

The final architecture of the social-graph recommender was developed on top of the user-based RS implementation provided in Mahout[2]. User neighborhood CF paradigm has close reasoning to social user behavior, being the most relevant criterion that influenced this design choice. In possession of information about users taste, this user-centered method focus on comparing similarity among users. Furthermore, friendship data will be essential to enable modifications on the original algorithm. Insights on how to profit from social context information in different dimensions will be addressed below.

4. INSIGHTS ON RS CHALLENGES

As the title suggests, solutions to the current RS challenges listed in Related Work are described in this section. Each of the following implemented scenarios tackle three main challenges previously mentioned, namely *computational complexity* issues of scalable user-neighborhood RSs; *sparse data* about new users, known as cold start problem; and definition of *optimal evaluation methods* for transparent and non-transparent recommendations.

4.1 Social Neighborhood

The idea of narrowing the dataset to a subset of users aims to tackle scalability constraints and increase real-time performance, two issues that are intrinsic to user-based RS [7]. Assuming that calculating an active user's neighborhood (comprised of k similar users) among his or her social connections might be representative enough, good recommendations could be achieved without the necessity of comparing a user preference vector with all other users in the database. This hypothesis is based on a related work comparing the correlation between users similarity and the binary fact of being or not being friends[2]. It was observed that similarities between friends are in average higher than the same correlation measurement between non-connected users.

Experiments were performed in order to investigate the three insights proposed above. A standard user-based neighborhood RS setup is incrementally modified from the current insight until the third one. This scenario focus on predicting ratings contained in a training set comprised of 5% of all 14.367 ratings provided by *MyPopCorn* users. The reason for not adding any rating from GroupLens into the training set of the standard neighborhood is allow a fair comparison between both neighborhoods. By applying two strategies, namely *Standard* full neighborhood and hereby proposed *Social* one, some hypotheses are tested: *(i)* Real-time recommendation performance will become much more efficient while adopting social neighborhood; *(ii)* Rating prediction accuracy from social neighborhood recommendations will be as much precise as in the standard method.

For the proposed experiment methods, standard neighborhood RS performs around 70k calculations, the number of all users in the merged dataset. In the case of social neighborhood, the number of comparisons is relative to the degree of each node (user) in the social graph, which varies from 0 to 49 for *MyPopCorn* dataset with an average degree of 10.543. Concerning average runtime, whereas prediction process for one rating takes around 950.55 ms for standard neighborhood, after narrowing the search space to the set of social connections, it takes in average 69.975 ms, 92.63% lower. Regarding accuracy, Figure 3 presents prediction accuracy error for this new neighborhood compared to the standard implementation. Both implementations were compared by varying the size of the neighborhood k while experimenting two values of threshold t=1 and t=2. This threshold defines the minimum number users in the neighborhood that rated a same candidate item. When t=2, the items rated by only one user in the neighborhood are not taken into account.

Figure 3: Standard and Social Neighborhoods prediction accuracy (*RMSE*).

The minimum $RMSE = 0.8385664$ was obtained by Standard neighborhood (k=3,t=2). Besides that, Social (k=2,t=2) achieved $RMSE = 1.018598$. Surprisingly, rating prediction accuracy also improved. Except for values of k neighbors equal to 2 and 3, Social Neighborhood outperforms, in average, the standard method, confirming the first hypothesis for this scenario. Besides that, the value of threshold t=2 performs better. The fact of accepting only items rated by at least two users might have increased the confidence on preference data, achieving better accuracy results. On the contrary, hypothesis 2 was surprisingly refuted. Instead of performing almost the same as in the original approach, Social Neighborhood can *significantly* outperform prediction accuracy for $k > 3$. While increasing the value of k, such social neighborhood enables a more accurate predictions and, probably, reaching higher serendipity.

REMARK: This approach is not available for people with no or few friends, suffering from the cold start problem, to be solved next.

[2]Apache Mahout machine learning library

4.2 Social Aggregation for Cold-start Problem

One of the main issues related to RS, the cold-start problem or new-user problem prohibit some active users to receive recommendations. In the dataset used for all experiments, 21 users out of 129 have rated less than 10 movies, while others more than a thousand. These users with few ratings are almost unable to receive any recommendation.

Instead of adopting the classic approaches such as content-based or presenting a list to be rated as from the first user interaction, this paper proposes a solution based on social-graph information. It is based strategy from group RS based on aggregating user profiles. One could see this problem following the quote *"Tell me who your friends are and I will tell you who you are"*. This reasoning is also motivated by the work carried out in [2], where social filtering is explored and conclusions reinforce the suggested heuristic. Likewise, [**?**] developed a probabilistic RS and achieved good results in experiments where active users were recommended items based on the preferences of his or her social connections. On the contrary, the idea presented in this paper follows the same reasoning of absorbing social context data into the system to solve the cold-start problem, nevertheless, by different means (based on group RS) and in a different RS implementation technique (user neighborhood RS).

Among some aggregation techniques mentioned in the Related Work, *Average without Misery* is adopted, since it finds a balance between the *Least Misery* and *Average*. It preserves the main advantages of both aggregation strategies originally applied to group RS and now reflected in the aggregated virtual profile to be considered by our single-user RS. It follows the human-like reasoning in which a group of people tend to select items that please, in average, most persons involved. Moreover, it excludes items once rated below a defined threshold, as described by [13]. The same author proposed such aggregation for solving the cold-start problem in [12], although in a different RS paradigm. Experiments were run in order to test the following hypothesis: *(i) Recommendation accuracy for aggregated virtual social profile performs not much worse than cross-validation of real ratings. Hence, it would be a feasible solution to the cold-start problem.*

The social neighborhood method was adopted with parameters k=4 and t=1, so that the most number of predictions are enabled. The idea here is to investigate how many active users had the cold-start problem, meaning their neighborhoods were empty. While repeating the experiments from last section in 5% of MyPopCorn ratings dataset, around 103 users were in the testset. Nevertheless, RS could not estimate any rating for 13 users due to empty neighborhood issue. 6 users had no social connections, what can not be solved by the method proposed here. The remaining 7 users had their ratings predicted with accuracy error of $RMSE = 1.69588$.

One should raise the question that this is not much data, referring to the tiny set of 7 users. In view of that, another experiment has been run on 50% of ratings in *MyPopCorn* dataset. Ratings of 44 users experiencing the cold-start problem were hidden iteratively in order to be predicted by the RS. Foreach of the 44 users, the RS generated a virtual profile based on aggregating all ratings from their friends, including those removed in order to artificially cause the cold-start problem. Only 8 new users(18%) could not be helped by this method of aggregation due to the fact of having no social connections. Prediction accuracy error was $RMSE = 1.37461$.

Compared to the accuracy evaluated in the experiments of previous sections ($RMSE = 1.173435$ for k=4, t=1), this proposed solution to the cold-start problem has decreased performance in around 20%, considering the $RMSE = 1.37461$. In view of that, the proposed solution is considered to be a good alternative for social RSs. Besides not compromising the prediction accuracy significantly, this method should be considered in terms of how efficient the RS can deal with new users that are not interested in providing many ratings as from the first interaction. Despite not being an objective metric, the ability of solving the cold-start should be incorporated into RS evaluation.

4.3 Comparison of Evaluation Methods

While the first insight focuses on the two objective evaluation metrics, namely prediction accuracy and computational complexity, this insight focuses on transparency, a subjective metric, and evaluation methods. The most popular evaluation metric throughout RS state-of-the-art, prediction accuracy benchmark is often based on offline cross-validation and error calculation over *Root Mean Squared Error - RMSE*. In view of that, this third and last section compares offline and online methods of calculating estimation accuracy together with more transparent recommendations based on social explanation. One hypothesis is that this online method might make offline approach suboptimal for the context of social recommenders. Instead of cross-validation, one should consider the social factor involved within online evaluation. Due to the strategy of recommending a list of movies whose predicted ratings might not be always high and to make it more transparent, the predicted value is presented to the active user. Assuming that not many people tend to converge with the RS prediction, this strategy will not bias the comparison. Actually, we believe there are people who also try to diverge from what has been predicted.

The current experiment intends to test the effect of explained recommendations, as previously described in [18], but now in the context of social RSs, as defined in the following hypothesis: (i) Assuming social RSs where recommendations based on social connections are explained, rating estimation accuracy achieve better results if evaluated online, instead of offline.

Besides *RMSE*, metrics such as novelty or serendipity were taken into account while choosing higher values of k other than the ones that reached minimum accuracy, shown in Figure 3. Although the same number of recommendations with standard and social neighborhood were generated, active users gave more feedback on the social ones. 119 online feedbacks were provided, as presented in Table 2 in comparison with the traditional offline method.

As Table 2 shows, Standard Neighborhood method achieved a prediction accuracy of 1.0646 and Social Neighborhood RS setup achieved better rating prediction accuracy of $RMSE$ = 0.9952. Both of them presented an improvement when evaluated online other than offline. The decrease in $RMSE$ was of 14.16% and 6.64%.

Hypothesis was confirmed by the numbers shown in Table 2. Surprisingly, online evaluation accuracy with Standard Neighborhood improved better (14.16%) than 6.64% gain achieved by Social Neighborhood strategy. Finally, results have shown that, in average, RSs tend to present better

Table 2: Online evaluation of social and standard neighborhood.

	Std. N.	**Social N.**
Setup	k=8, t=2	k=4, t=2
OFFLINE *RMSE*	1.240429	1.066049
ONLINE *RMSE*	1.064686	0.995211
Improvement	14.16%	6.64%

accuracy results in online evaluations than offline for both explained and non-explained recommendations.

5. CONCLUSIONS

This paper first discussed the computational requirements intrinsic to user neighborhood RS, by nature a non-scalable algorithm. Based on the two most important evaluation metrics, state space reduction enabled a decrease of *92.63%* in computational complexity, while not compromising accuracy. Instead, the latter also improved.

Social graph was essential to enable a solution to the cold-start problem. Tested with success in group RS, *Average without Misery* enabled creation of virtual profiles based on active users network. Results confirmed the proposed hypothesis, indicating this solution as a good alternative to this issue while presenting a decrease on prediction accuracy of only *20%* by cross-validation.

Another important achievement was caused by transparent recommendations. Results from the third insight turn prediction accuracy by cross-validation an even more questionable benchmark method. Both neighborhood formation methods presented a considerable improvement of *6.64%* and *14.12%*. While choosing online evaluation methods, one could have better conclusions about the RS quality.

6. ACKNOWLEDGMENTS

This research is part of a master studies sponsored by *Monesia: **MO**bility **N**etwork **E**urope-**S**outhamerica: an **I**nstitutional **A**pproach*, an Erasmus Mundus External Cooperation Window.

Thanks Lucas Carvalho, researcher at Federal University of Sergipe - Brazil, for cooperating on the development of the Facebook application named MyPopCorn.

7. REFERENCES

[1] X. Amatriain and J. Basilico. Netflix recommendations: Beyond the 5 stars, 2012.

[2] G. Groh and C. Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 127–136. Citeseer, 2007.

[3] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[4] J. Herlocker and J. Konstan. Explaining collaborative filtering recommendations. *of the 2000 ACM conference on*, pages 241–250, 2000.

[5] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[6] D. Infographic. Visualizing the petabyte age, 2010.

[7] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM, 2001.

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

[9] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105, 2011.

[10] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[11] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508, 2004.

[12] J. Masthoff. Modeling the multiple people that are me. *User Modeling 2003*, pages 146–146, 2003.

[13] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, 2004.

[14] A. Rashid, I. Albert, D. Cosley, S. Lam, S. McNee, J. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134. ACM, 2002.

[15] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. *Recommender Systems Handbook*, pages 1–35, 2011.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the Fifth International Conference on Computer and Information Technology*, pages 158–167, 2002.

[18] N. Tintarev and J. Masthoff. Designing and evaluating explanations for recommender systems. *Recommender Systems Handbook*, pages 479–510, 2011.

[19] P. Victor, M. Cock, and C. Cornelis. Trust and recommendations. *Recommender Systems Handbook*, pages 645–675, 2011.