Proceedings of the International Conference on Big Data, Cloud and Applications
Tetuan, Morocco, May 25 - 26, 2015

Improving Neighborhood-Based Collaborative Filtering by A Heuristic Approach and An Adjusted Similarity Measure

Yasser El Madani El Alami
Computer science department of FSDM
Sidi Mohammed Ben Abdellah University
Fez, Morocco
yasser.elmadanielalami@usmba.ac.ma

El Habib Nfaoui
Computer science department of FSDM
Sidi Mohammed Ben Abdellah University
Fez, Morocco
elhabib.nfaoui@usmba.ac.ma

Omar El Beqqali
Computer science department of FSDM
Sidi Mohammed Ben Abdellah University
Fez, Morocco
omar.elbeqqali@usmba.ac.ma

***Abstract*— "Collaborative filtering" is the most used approach in recommendation systems since it provides good predictions. However, it still suffers from many drawbacks such as sparsity and scalability problems especially for huge datasets which consist of a large number of users and items. This paper presents a new algorithm for neighborhood selection based on two heuristic approaches. The first of which is based on selecting users who rated the same items as the active user called "*intersection neighborhood*" while the second one builds the neighborhood using all users who rated one item at least as the active user called "*union neighborhood*". In addition, we employ an adjusted similarity measure that combines Pearson correlation with a set-similarity measure (such as Jaccard similarity) as a correction coefficient for .accurate similarities among users. Finally, experiments using FilmTrust dataset show that the proposed approaches give more predictions accuracy than the traditional collaborative filtering.**

Keywords—Collaborative filtering; Neighborhood selection;Spatial complexity; Recommender system; Similarity measure

1 Introduction

Ever since the 90s, the amount of information has been increased in exponential way. The Internet has played a key role in information growth. Mobile devices such as smart phones and tablets also contribute to this continuous expansion of information plethora. Thus, users are continually faced with information overload. It becomes difficult for them to distinguish relevant information from noise. In order to address this problem, there have been a great interest in recommendation. According to [1] recommendation systems have been considered as an effective means to reduce complexity in information retrieval. They promise to personalize the request based on the user's interest in a smart way [2]. As stated by [3] recommendation system helps users to deal with information overload and provides personalized recommendations, content and services to them. It suggests the appropriate items for each user according to his/her interests. Although, recommendation systems are largely used in both e-commerce applications such as Amazon [4] and academic researches such as MovieLens [5] [6], they are being extended to other domains such as digital libraries [7], e-learning [8], etc.

Authors in [3] show that there are three main categories of information filtering: content-based recommendations [9], collaborative filtering and hybrid recommendations where collaborative filtering methods are the most used in recommender systems [10]. They rely on users' evaluation (ratings) to identify "useful" items to these users. Unfortunately, many typical drawbacks are noticed in collaborative filtering approaches which weaken thereafter the quality of the recommendations such as sparsity and scalability problems.

Our work relies on using a heuristic approach in a preprocessing step for building users' neighborhood which relies on the well-known set operators; union and intersection. They induce a new ratings matrix with low dimension. This ratings matrix is less sparse than the ratings matrix of the whole users. Therefore, this method leads to a minimum of time consumption in the selection neighborhood phase. In parallel, we use a reformed similarity measure that combines the well-known Pearson correlation with set-similarity measures as adjustment coefficient which yields good results.

This paper is organized as follows: in section 2 we give an overview of the traditional collaborative filtering. In section 3 we mention some recent works conducted in collaborative filtering field. Section 4 describes our proposals. In section 5, we present the experiments and evaluation results of our proposals. At the end, we give some perspectives, and a conclusion.

2 Background

The term of collaborative filtering (CF) was introduced by David Goldberg in [11] where he proposed a mail system called Tapestry that filters documents based on users' interest in order to be used by other people. Collaborative filtering is based on mutual aid of users who share similar tastes and preferences to recommend the suitable items. According to [12], collaborative filtering relies on the following assumption: if users X and Y rate n items similarly or have similar behaviors, then, in the future, they will act (rating or behavior) on other cases similarly. As a result, CF based systems can predict a user's rating (or behavior) for an unknown item [11] or create a top-N list of recommended items for a target user (called active user) [13]. It is worth noting that the first work using CF was presented by Malone in [14] where he proposed stereotypes to build user models and use them to recommend relevant books to each user.

According to [15], we can distinguish between classes of CF algorithms: memory-based algorithm and model-based algorithms. In what follows we focus on the memory based approach

2.1 Memory based algorithm

Memory based approach builds predictions based on the whole set of ratings that users assigned to items before. Previous ratings are grouped in a matrix referred to as ratings matrix. It's the pillar input in this approach. It was the earliest approach adopted by many commercial systems thanks to its easiness and effectiveness [16].

Table 1 Example of ratings matrix

Items / Users	Item 1	…	**Item j**	…	Item n
User 1	1		2	2	3
…			2		1
User s		1	**?**		
…	3			3	5
User p		1	4		

As presented in table 1 above, also called ratings matrix, the cell r_{sj} refers to the rating given by user s to item j (on 1-5 rating scale). In most cases, this ratings' matrix is typically sparse [17] as most users do not rate viewed items regularly. Therefore, [18] argued that the sparsity can be an issue that can lead to weak recommendations. Besides, the most popular algorithm in memory based is neighbor-based algorithm which predicts ratings based on either users who are similar to the active user or similar items to the requested item. Generally, According to [19] there are three steps into processing a recommendation based on CF system: i) Representation, ii) Neighborhood formation, iii) Recommendation generation.

Neighbor-based approach is mainly divided in two analogous categories: users-based collaborative filtering [15] and item-based collaborative filtering [20]. For example, in what follows, we detail the User-Based CF.

2.1.1 User-based CF (UBCF)

User based recommendation relies on users similarity to the active user. In fact, it builds prediction and recommendation using the correlation between the active user and each other.

2.1.1.1 Representation

The first step in UBCF consists on building a rating matrix and assigning values to the unrated items to fill the porous ratings matrix. Two processes [21] can overcome sparsity and improve recommendation accuracy:

- Default rating: to set a same value for all unrated items.
- Pre-processing using average: to set an average rating based on user's votes for the missing rating-matrix entry $\bar{r}_s$

2.1.1.2 Neighborhood formation

The second step consists of measuring similarity between the other users. They are several similarity algorithms such as Pearson correlation, mean-squared difference [22] and Spearman correlation [23]. The most commonly used algorithm is the Pearson correlation. In fact, it has become a standard way of calculating correlation [19]. Using Pearson correlation, similarity between user u_a and u_b is calculated with the following formula:

$$sim_{a,b} = \frac{\sum_{j=1}^{n}(r_{aj}-\bar{r_a})(r_{bj}-\bar{r_b})}{\sqrt{\sum_{j=1}^{n}(r_{aj}-\bar{r_a})^2 \sum_{j=1}^{n}(r_{bj}-\bar{r_b})^2}} \quad (1)$$

where n is the cardinal of the set of items, r_{ai} is the rating given by user a to item j and $\bar{r_a}$is the average rating given by user a for all the items he rated. As an output, similarity process returns a user similarity matrix which determines correlation between pairs of users. Thus, building similarity between users allows forming the requested neighborhood. Two techniques have been employed [24]:

- Threshold-based: user is considered as neighbor when his/her user similarity exceeds a given threshold [25].
- K nearest users where k is given as input. Also, it can be computed for each user as proposed in [26].

2.1.1.3 Recommendation generation

This phase relies on generating predicted rating of user s to item i. It's calculated as aggregation of similarity between the active user and his neighborhood. It relies on both ratings matrix (input) and similarity matrix.

$$p_{s,i} = aggr_{s' \in S} r_{s',i} \quad (2)$$

Various aggregation functions are employed in predictions where the most used one is calculated as the weighted average of neighbors' ratings using their similarities as follows:

$$p_{s,i} = \bar{r}_s + \frac{\sum_{p=1}^{k}(r_{p,i}-\bar{r_p})}{\sum_{p=1}^{k}|sim_{s,p}|} \quad (3)$$

K represents the size of selected neighborhood.

Therefore, based on computed predictions, recommender system may select the top-N items as the recommendations list of unknown or new items that the active user has never seen before.

2.2 Performance measures

Performance measures are the result of a step of monitoring a proposed method or algorithm in real situation or approximating the reality with reliable data. In recommendation system, a great research effort has been made to deal with this influential task such as presented in [27]. In our case we are interested by prediction performance measures.

In this area, many indicators are used to measure the system performance. In most cases, they tend to evaluate the system accuracy. They measure the precision of computed predictions comparing to the real user ratings. A case in point is Mean Absolute Error (MAE). In fact, MAE is a common way used to measure accuracy based on statistical metric. It calculates the average absolute difference between predicted ratings and real ones:

$$MAE = \frac{\sum_{(s,i)}|p_{s,i}-r_{s,i}|}{N} \quad (4)$$

In the formula above $p_{s,i}$ is the predicted rating for user s to item i, $r_{s,i}$ is the real rating given by user s to item i and N corresponds to the number of predicted ratings calculated during the test phase.

In [27] authors argue that using different evaluation metrics leads to different conclusion concerning the recommendation system performance. Thus, most researches choose a single evaluation measure in order to demonstrate the effectiveness of their algorithms.

After giving a detailed overview of the background, and presenting the broad outlines of collaborative filtering, in the next section, we present some recent works and trends done in this area in order to overcome drawbacks and illnesses of the traditional approach.

3 RECENT WORKS

In the last decade, collaborative filtering approach motivated a larger number of works adding in each one an original concept and then opening a new perspective in order to deal with the problem of recommendation. Clustering method is one of the extensively used concepts in CF. One of the earliest works that have been done in this area was presented in [28] where the authors argued that clustering improves the performance of recommendation. [29] Proposes to group users' profiles into clusters of similar items and compose the recommendation list of items that match well with each cluster. Authors in [30] develop Eigentaste 5.0 recommender system that dynamically arranges the order of recommended items by integrating user clustering with item clustering. In [31] the author proposes a new probabilistic neighborhood-based approach as an improvement of the standard k-nearest neighbor algorithm. It's based on classical metrics of dispersion and diversity as well as on some newly proposed metrics. The author also proposes the concept of unexpectedness in recommender systems. He also fully uses it by suggesting various mechanisms for specifying the expectations of the users. Moreover, he proposes a recommendation method for providing the users with non-obvious but high quality personalized recommendations that fairly match their interests. This method is based on specific metrics of unexpectedness. In [32] the authors propose an adapted normalization technique called mutual proximity in the nearest neighbor selection phase to rescale the similarity space and symmetrize the nearest neighbor relation. They prove that removing hubs and incorporating normalized similarity values into the neighbor weighting step leads to increased rating prediction accuracy.

One of the major factors in collaborative filtering that greatly influences the recommendation accuracy is the selected similarity measure. In the literature, almost all works are based on the well-known Pearson correlation measure. As presented in formula (2), Pearson correlation measure doesn't take into account other decisive which provide meaningful information of how users' preferences are different. Sparsity is another consistent problem which contributes to generating incorrect recommendations. In fact, only a small number of the whole items are rated then the matrix rating becomes sparse. In addition, using a huge dataset required more time for computing similarities among users in order to build an effective neighborhood for the active user. Consequently, combining these factors increases mainly the margin of error and reduces the confidence interval which leads to inaccurate recommendations.

4 PROPOSED APPROACH

In order to limit the problems of sparsity and time computing problems, we propose a preprocessing step which relies on a heuristic approach of neighborhood selection (figure 1).

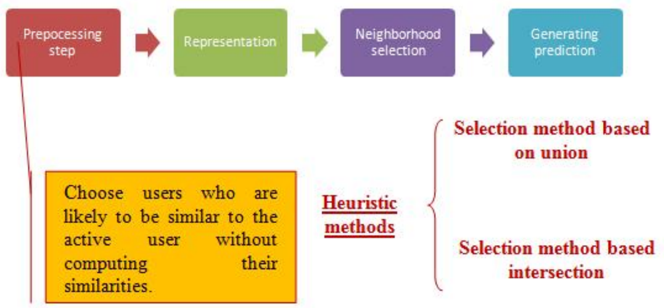

Figure 1 the proposed process of collaborative filtering

It can be done with two methods based on natural operators' sets without computing similarities among users. The first method is the intersection selection. It's based on selecting users who rate the same item as the active user. The second one is based on the union operator. It's based on building the neighborhood with users who rated one item in common at least. Obviously, time computing is reduced because of the reduction of the number of computed users' similarities. In fact our approach focuses on selecting neighbors who are likely to be reliable to the active user before starting similarities computation phase which is time-consuming if we compute the similarities for all system's users. As a result, the new ratings matrix is smaller than the ratings matrix of the system used in the traditional collaborative filtering methods, which leads to less sparseness in the matrix.

A. Intersection neighborhood

We call the first method of neighborhood selection the intersection neighborhood. It relies on selecting neighbors who rate the same items. Actually, it's rare to find two users who rate the same items. So, in order to deal with this point, we use a threshold of acceptance which corresponds to a minimum number of co-rated items. In what follows, we present the adopted algorithm:

1. For each active user.

2. Extract the list I_a of items that the active user U_a rated before. $I_a=\{i_1, i_2\ldots,i_p\}$ and card(I_a)=p.

3. Select users who rated the same list of items I_a as presented in the figure below (figure 2) or having more than the fixed threshold noted "TA".

4. Select the target item whose rating value is going to be predicted. As shown in the example below, we select the item $\mathbf{I_6}$.

		I_1	I_2	I_3	I_4	I_5	I_6
	U_1		3	1	3	5	4
Active user	U_2		4		5	2	?
	U_3	5	1		1	4	3
	U_4	2		3	4		

Figure 2 Basic ratings matrix

As presented in this example (figure 3), the neighborhood contains two users U_1 and $U_3$and the new adopted ratings matrix is:

		I_1	I_2	I_3	I_4	I_5	I_6
	U_1		3	1	3	5	4
Active user	U_2		4		5	2	
	U_3	5	1		1	4	3

Figure 3 Derived ratings matrix

5. Fill up the empty cells with the average of ratings of each user. As we can see, the new ratings matrix contains less empty cells than the first one.

6. Compute the similarity between the selected users and the active user using the new matrix of ratings.

7. Select the top n similar user N={u_1, u_2, … u_n}

8. Generate the recommendation based on the selected neighborhood.

B. Union neighborhood

The second method of neighborhood selection is called the union neighborhood. . It relies on selecting neighbors who rate one common item at least. The adopted algorithm is presented in what follows:

1. For each active user

2. We extract the list I_a of items that the active user U_a rated before. $I_a=\{i_1, i_2\ldots,i_p\}$ and card(I_a)=p

3. Select users who rated one item of the list I_a at least as presented in the figure 4 below.

4. Select the target item whose rating value is going to be predicted. For instance, as shown in the example below (figure 5) we select the item $\mathbf{I_6}$.

		I_1	I_2	I_3	I_4	I_5	I_6
Active user	U_1		3	1	3	5	4
	U_2		4		5	2	
	U_3	5	1		1	4	3
	U_4	2		3			4

Figure 4 Basic ratings matrix

As presented in this example (figure 5), the neighborhood contains two users U_1 and U_3. The new ratings matrix is:

		I_1	I_2	I_3	I_4	I_5	I_6
	U_1		3		3	5	4
Active user	U_2		4		5	2	
	U_3	5	1		1	4	3

Figure 5 Derived ratings matrix

5. Fill up the empty cells with the average of ratings of each user. As a direct result, the new ratings matrix contains more users than the intersection algorithm but more empty cells.

6. Compute the similarity between the selected users and the active user.

7. Select the top n similar users N={u1, u2, … u_n}

8. Generate the recommendation based on the selected neighborhood.

C. Similarity measure

In order to reduce the impact of the fallacious similarity on the computed recommendations, we propose to a modified version of Pearson correlation. In [33], authors argue that adding associated parameters of users x and y improve the similarity accuracy which lead to good predictions. Thus, the new similarity measure is presented as follows:

$$Sim_{xy} = \mathrm{S_{Jac}} * Corr_p \quad (5)$$

Where $\mathrm{Corr_p}$ represents the traditional Pearson correlation and $\mathrm{S_{Jac}}$ represents Jaccard coefficient used as adjustment coefficient:

$$\mathrm{S_{Jac}} = \frac{a}{a+b+c} \quad (6)$$

where

- ✓ a=|X∩Y| represents the number of attributes (the rated items) which are present in user X and user Y.
- ✓ b=|X-Y| represents the number of attributes (the rated items) which are present in user X and not in user Y.

- ✓ c=|Y-X| represents the number of attributes (the rated items) which are present in user Y and not in user X.

5 EXPERIMENTS AND RESULT

5.1 FilmTrust DataSet

For experiments we use the FilmTrust project dataset [34]. It is an academic research project being run by Jennifer Golbeck[1]. It's a movie recommendation website where users can rate and review movies. Users can give their opinion using a quantitative value on a rating scale from 0.5 to 4 stars where 0.5 means bad and 4 means excellent. The data is stored as semantic web annotations based on RDF[2] and FOAF[3]. It integrates semantic web-based on social networks with movie ratings so as to compute predictive movie recommendations. The collected dataset consists of 1856 users, 2092 movies and 759922 ratings. Thus, around 80.4% of the global ratings matrix is empty. It means that FilmTrust dataset represents a real situation of sparsity problem.

5.2 Experiments steps

Our test consists of two main experiments. The first experiment focuses on comparing the traditional approach with the union neighborhood selection method using the adjustment coefficient presented before. The second experiment focuses on comparing the comparing the traditional approach with intersection neighborhood selection method using the same adjustment coefficient.

The experiments respect the steps below:

1. From the dataset DS, for each user we build our ratings matrix based on the definition presented below.
2. We randomly select 30% of the ratings and set the value of those ratings to POSITIVE_INFINITY. In fact, we use this value to distinguish between the empty cells of the global ratings matrix which is represented by null value and the modified ones. Therefore we build two sets:

Set TG represents the training set (70% of the whole dataset) and Set T which represents the set of test with: T=DS-TG

3. For each user from the set DS.
   a. We select users according to the employed method (intersection or union)
   b. We build the rating matrix TG by filling up the missing ratings with the average ratings of each user.
   c. Use different size of neighborhood from 10 users to 100.
   d. Employ the prediction formula to predict the missing values.
   e. Repeat step d until predicting all the missing values
   f. Computing the MAE for each neighborhood size by comparing the predicted values with the observed ones.
4. Repeating this computation (from step 2) three times and then we give the average of the MAE for each user and neighborhood size.

5.3 Results and analysis

5.3.1 Experiment 1

We start our test with comparing the traditional collaborative approach method (TCFM) with the union neighborhood selection method (UNSM) by following the steps presented before. Even though the variation behavior of the union method is not regular, we can see (figure 6) that the union neighborhood selection method provides good results since the computed MAE is less than the traditional collaborative filtering one.

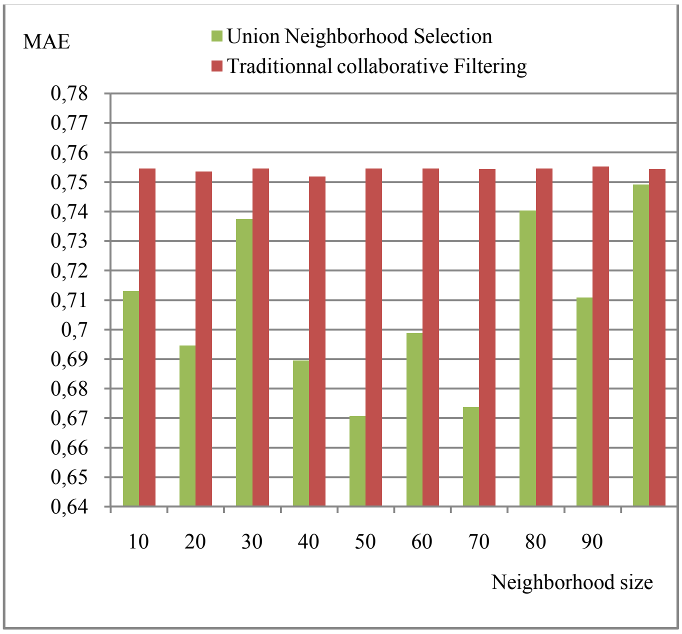

Figure 6 MAE comparison between TCFM and UNSM

5.3.2 Experiment 2

The second experiment compares the traditional collaborative filtering method (TCFM) with the intersection neighborhood selection method (INSM) by following the steps outlined beforehand. In addition, in order to take user y as a candidate neighbor of the active user x in intersection approach, we set 10 as a threshold of co-rated items. Then, user 'y' will be selected if he/she has10 co-rated items at least.

---

[1]https://www.cs.umd.edu/~golbeck/

[2] http://www.w3.org/RDF/

[3] http://www.foaf-project.org/original-intro

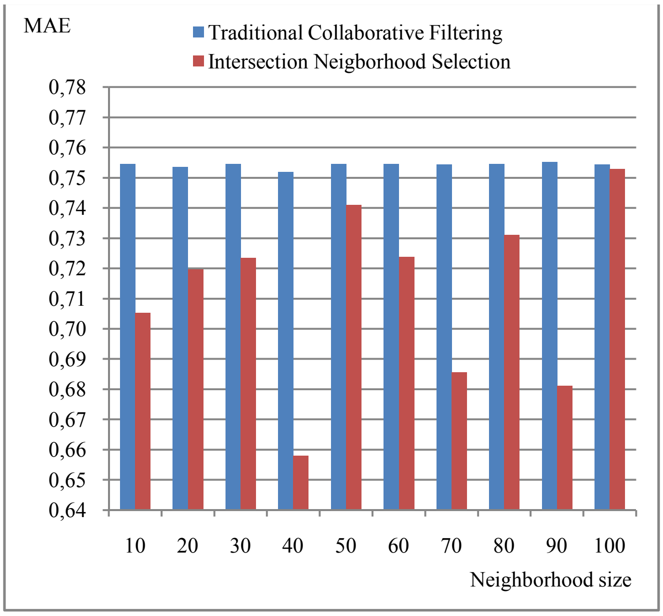

Figure 7 MAE comparison between TCFM and INSM

Even though the variation behavior of the intersection method is not regular (figure 7), we see that the intersection neighborhood selection method (INSM) performs better than the traditional collaborative filtering method (TCFM).

5.3.3 Comparative analysis

The last figure (figure 8) shows a comparison of the MAE between the traditional collaborative method and the two proposed methods: intersection and selection neighborhood disregarding the size of the neighborhood. As we can see, union neighborhood selection method (UNSM) gives the best result.

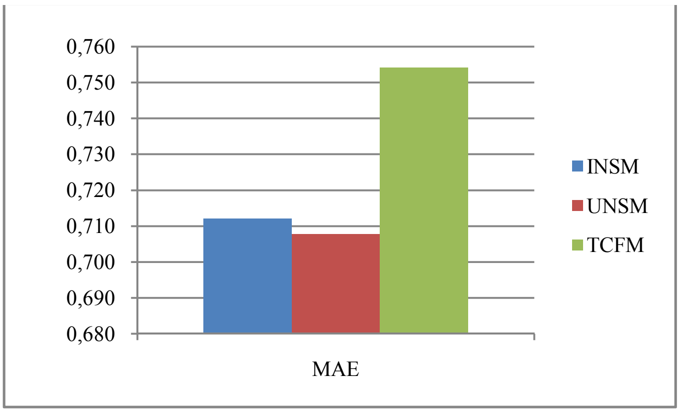

Figure 8 Synthesis of all approaches

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a preprocessing step which relies on two heuristic methods: the union neighborhood selection method and the intersection neighborhood selection. Both of them are combined with a reformed Pearson correlation similarity where we use a set-similarity measure as a correction factor (Jaccard similarity). As presented before, the two methods provide acceptable results comparing to the traditional collaborative filtering. As a result of these neighborhood selection methods, we reduce the rating-matrix dimension which leads, on one hand, to less sparseness in the induced matrix, and, on the other hand, to minimizing the consumed time in neighborhood selection phase. In addition, the prediction accuracy is improved.

Despite this, the two approaches are not efficient for cold start problem especially for intersection approach which needs a threshold of ratings to provide good recommendations. As future work, we will investigate incorporating social network data in order to deal with this problem. In fact, social networks offer many opportunities for recommendations since people generally use their social networks to obtain reliable and useful information.

REFERENCES

[1] A. K. Joseph, "Introduction to recommender systems: Algorithms and Evaluation," *ACM Transactions on Information Systems,* vol. 22, no. 1, pp. 1-4, 2004.

[2] Z. Zied, "Modèle multi-agents pour le filtrage collaboratif de l'information," Université du Québec, Montréal, 2010.

[3] A. Gediminas and T. Alexander, "Towards 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, p. 734–749, 2005.

[4] L. Greg, S. Brent and Y. Jeremy, "Amazon.com Recommendations: item-to-item collaborative filtering," *Internet Computing, IEEE,* vol. 7, no. 1, pp. 76 - 80, 2003.

[5] M. Bradley N., A. Istvan, L. Shyong K., K. Joseph A. and R. John, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System," in *the 8th international conference on Intelligent user interfaces*, Miami, Florida, 2003.

[6] B. N. Miller, "Toward a Personal Recommender System," Minneapolis, USA, 2003.

[7] A. Smeaton and J. Callan, "Personalization and recommender systems in digital libraries," *International Journal on Digital Libraries,* vol. 5, no. 4, pp. 299-308, 2005.

[8] N. Manouselis, D. Hendrik, R. Vuorikari, H. Hummel and R. Koper, "Recommender Systems in Technology Enhanced Learning," in *Recommender Systems Handbook*, Spring US, 2011, pp. 387-415.

[9] L. Pasquale, d. G. Marco and S. Giovanni, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, Spring US, 2011, pp. 73-105.

[10] P. L. Joel, L. Nuno, N. M. María, A. F. Ana and M. Constantino, "A hybrid recommendation approach for a tourism system," *Expert Systems with Applications,* vol. 40, no. 9, p. 3532–3550, 2013.

[11] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using collaborative fil- tering to weave an information tapestry," *Communications of the ACM,* vol. 35, no. 12, pp. 61-70, 1992.

[12] K. Goldberg, T. Roeder, D. Gupta and C. Perkins, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Information Retrieval,* vol. 4, no. 2, pp. 133-151, 2001.

[13] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Transactions on Information Systems (TOIS),* vol. 22, no. 1, pp. 143 - 177, 2004.

[14] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst and M. D. Cohen, "Intelligent information-sharing systems," *Communications of the ACM,* vol. 30, no. 5, pp. 390-402, 1987.

[15] J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *the Fourteenth conference on Uncertainty in artificial intelligence*, San Francisco, USA, 1998.

[16] M. D. Ekstrand, J. T. Riedl and J. A. Konstan, "Collaborative Filtering

Recommender Systems," *Foundations and Trends in Human-Computer Interaction,* vol. 4, no. 2, pp. 81-173, 2011.

[17] P. Melville and V. Sindhwani, "Recommender Systems," in *Encyclopedia of machine learning*, Springer US, 2010, pp. 829-838.

[18] E. Vozalis and K. Margaritis, "Analysis of Recommender Systems' Algorithms," in *HERCMA 2003*, Athens, Greece, 2003.

[19] B. Sameer, "Recommender System Algorithms," Toronto, Canada, 2008.

[20] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *the 10th international conference on World Wide Web*, New York, USA, 2001.

[21] E. Vozalis and K. Margaritis, "Analysis of Recommender Systems' Algorithms," in *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003)*, Athens, Greece, 2003.

[22] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating “word of mouth”," in *the SIGCHI Conference on Human Factors in Computing Systems*, New York, USA, 1995.

[23] M. Kendall and J. D. Gibbons, Rank Correlation Methods, Oxford: Oxford University Press, 1990.

[24] S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," *Journal of Software,* vol. 5, no. 7, pp. 745-752, 2010.

[25] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Sytems,* vol. 22, no. 1, p. 5–53, 2004.

[26] N. K. Lathia, "Evaluating Collaborative Filtering Over Time," London, UK, 2010.

[27] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research,* vol. 10, p. 2935–2962, 2009.

[28] B. M. 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*, 2002.

[29] M. Zhang and N. Hurley, "Novel item recommendation by user profile partitioning," in *The 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009.

[30] T. Nathanson, E. Bitton and K. Goldberg, "Eigentaste 5.0: constant-time adaptability in a recommender system using item clustering," in *In Proceedings of the 2007 ACM conference on Recommender systems*, 2007.

[31] P. Adamopoulos, "On Discovering non-Obvious Recommendations: Using Unexpectedness and Neighborhood Selection Methods in Collaborative Filtering Systems," in *Proceedings of the 7th ACM international conference on Web search and data mining*, New York, NY, USA, 2014.

[32] P. Knees, D. Schnitzer and A. Flexer, "Improving Neighborhood-Based Collaborative Filtering by Reducing Hubness," in *Proceedings of International Conference on Multimedia Retrieval*, New York, NY, USA, 2014.

[33] E. M. E. A. Yasser, N. El Habib and E. B. Omar, "An Adjustment Similarity Measure for Improving Prediction In Collaborative Filtering," in *The 2nd International Workshop on Software Engineering and Systems Architecture*, Tetouan, Morocco, 2014.

[34] [Online]. Available: http://trust.mindswap.org/FilmTrust/about.shtml. [Accessed 1 September 2013].