Enhanced Similarities for a Music Recommender System

Vasileios Gavriilidis, Anastasios Tefas, Constantine Kotropoulos, Nikos Nikolaidis
Department of Informatics, Aristotle University of Thessaloniki
54124 Thessaloniki, Greece,
{tefas, costas, nikolaid}@aiia.csd.auth.gr

***Abstract*—Music is an essential part of human life. It is a way to express ourselves. Its importance gave rise to music Recommender Systems (RS) through e–commerce applications. The most commonly used technique in such applications is Collaborative Filtering (CF) that uses the User–Item (UI) matrix. The latter codes the users' preferences for items. The advantage of using CF methods is their simplicity and their relatively high efficiency. However, data sparsity deteriorates their efficiency. In this paper, we propose a process that can enhance the similarities among users in a way that can address the sparsity problem. The proposed user based CF algorithm alters the values of similarity matrix between users by incorporating a graph based method, improving the performance of the RS. We also propose a method that groups users using spectral clustering and together with the graph–based similarity method yield even more accurate predictions. Experiments have been conducted on a music dataset, highlighting the superiority of the proposed method against typical user–based CF.**

I. Introduction

Music recommender systems (RS) are designed in order to predict a user's taste for a song based on the user's previous listening relationships [1]. Recommendation has become an interesting topic, mainly because of its many practical applications. Music is just one area where RS are useful. An overview of RS can be found in [2].

The most successful recommendation technique area is collaborative filtering [3], [4]. The notion behind collaborative filtering technique is that similar users will likely have the same opinion about a song in the future [5]. This assumption is generally true. However, the process of finding similar users, which is the key element to algorithm performance, suffers from data sparsity [6]. Collaborative filtering algorithms can be categorized in to memory based and model based approaches [7]. In model based approaches prediction models are trained, while in memory based ones predictions are made by aggregating other known ratings. While the UI matrix is used in both approaches here we focus our research on the memory based category.

Memory based CF methods are further classified into, user based [8] and item based ones [9]. In both subcategories a similarity matrix is created. In user based CF methods, similarities are calculated between the users, while in the item based ones similarity between items are calculated. There are several similarity metrics that have been used in CF; Pearson correlation, cosine similarity, Spearman correlation. The similarity metric selection is a very important step in memory based CF, since the performance is depends on it. Another matter that deteriorates performance is the similarity between users is evaluated by using the items that have been rated by both users. So, there is no information about users that have zero co–rated items. Moreover, similarity that have few co–rated items by a pair of users can be noisy.

Usually in real world applications, users are much fewer than the items. Typically the users are in order of thousands while the items can be in the order of millions. Hence, data sparsity is likely to occur. That is the available number of ratings is very small compared to the number of ratings to be predicted. The immediate consequence is that the user similarity matrix can be distorted so that is does not reflect the true similarity between user pairs. To cope with this problem we incorporate graph theory in order to build the similarities between users [10]. Graph theory offers methods to analyse the data and has already been exploited in RS [11], [12].

Here, we propose a method that creates a graph based similarity matrix that reflects the general structure of the data. The proposed method performs random walks on a graph. It can alleviate sparsity and is able to find representative similarities between users. This similarity method is combined with spectral clustering techniques [13]. Grouping people has helped in the past to cope with the sparsity of the data [14].

The structure of the paper is organized as follows: In Section II, the problem to be solved is stated. We, then, introduce the proposed method of enhanced similarity matrices Section III, together with an extension that employs spectral clustering. In Section IV, we explain how experiments were conducted on a music dataset and present top N–recommendation results. Finally, concluding remarks and a discussion for future work are given in Section V.

II. Problem Statement

Usually, in CF there is a set of N users, and a set of M items. The preferences of users for individual items are gathered in a User–Item Matrix, R of size $N \times M$ and each value, $R_{u,i}$ denotes the preference of user u to item i. Such preferences are the ratings provided by users i and therefore depend on high-quality explicit feedback by this user. Explicit ratings are not always available, because the user has to spend time to rate items however, explicit feedback can be replaced by an implicit method by examining the behaviour of the

user [15]. A comparison between these two types of user preferences can be found in [16].

In all memory based methods of CF a similarity between users is created. There are several metrics for measuring similarities, but the Pearson correlation has been found to provide the best results [8], [17]. Pearson correlation between two users, u and v, is defined as:

$$S_{u,v} = \frac{\sum\limits_{i \in I_u \cap I_v} (R_{u,i} - \bar{R}_u)(R_{v,i} - \bar{R}_v)}{\sqrt{\sum\limits_{i \in I_u \cap I_v} (R_{u,i} - \bar{R}_u)} \sqrt{\sum\limits_{i \in I_u \cap I_v} (R_{v,i} - \bar{R}_v)}}, \quad (1)$$

where $\bar{R}_u$ and $\bar{R}_v$ are the average ratings for users u and v respectively. Moreover, I_u and I_v denote the set of items rated by users u and v, respectively. A summary of the symbols used in the paper and their definitions is provided in Table I. Note that Pearson correlation, like any other method of computing similarity between users in CF, performs the comparison between two users, using the items that have been rated by both users. Hence, if the intersection of the items rated by the two users is empty, then the similarity is zero.

Overall, the goal of a RS is to predict a preference value of user for an item that hasn't been rated yet.

III. Proposed Method

In RS the UI matrix is often sparse. The number of items is large relative to the number of users. Accordingly the profile of the user is not fully captured by the UI matrix. The Pearson correlation (1) is not efficient, when there are few co–rated items. We propose a method that can diffuse the similarity of users so that the similarity between two users can reflect better the structure of data using the whole UI similarity matrix.

A. Random Walks on User Similarity Matrix

Let $\mathbf{S}$ denote the user similarity matrix, Hence it is of size $N \times N$. Let also users be represented as nodes in a graph and an edge between two users indicate their similarity. So matrix $\mathbf{S}$ can be seen as the graph adjacency matrix, where $S(u,v) = 1$ if nodes u and v are connected and $S(u,v) = 0$ otherwise. A property of graph theory states that the p–th power of the adjacency matrix, $S^p(u,v)$, gives the number of paths of length p between nodes u and v. This notion can be applied to either directed or undirected graphs and can also be extended to weighted graphs, $S(u,v) \in [0, \inf]$. As explained, the property requires values in the range $[0,1]$, so the Pearson values are normalized accordingly.

A similarity matrix defined as:

$$\mathbf{S}^p = \underbrace{\mathbf{S}\mathbf{S}\dots\mathbf{S}}_{\text{p times}}, \quad (2)$$

manages to connect users that previously were not connected, exploiting the fact that, two users can be connected through other users, and this way similarity paths can be revealed. As said previously, Pearson correlation fails to compute a value where two users have zeros co–rated items. To the opposite, the proposed method finds connections between the two users by using the information of intermediate users, whose number is defined by the parameter p.

The matrix $\mathbf{S}^p$ can also be interpreted as performing random walks on a graph defined by the similarity matrix of users. There is a study for RS employing graph analysis [18] where a bipartite graph is created from the UI matrix. Then random walks are performed directly on the bipartite graph that suggests that users randomly select items with a specific probability. Contrary to [18], our method performs random walks between users, using the user similarity matrix, which suggests a user being similar to another with a given probability. Having computed similarity matrix, the prediction of user, u, for an item, i, is calculated as follows:

$$\hat{R}_{u,i} = \bar{R}_u + \frac{\sum\limits_{v \in V_k(u)} S^p_{u,v}(R_{v,i} - \bar{R}_v)}{\sum\limits_{v \in V_k(u)} S^p_{u,v}}, \quad (3)$$

where $V_k(u)$ is the subset of users that belong to the k most similar users of u after performing random walks on the graph defined by the similarity matrix, and have rated the item for which the prediction is required. The number k is an arbitrary number although it has been found empirically that choosing the 50 close neighbours yields the best results [8], [17].

Moreover, raising a matrix to a high power can create relatively large values so we normalise the similarity matrix as if it were a Kernel matrix using the following equation:

$$\tilde{S}^p_{u,v} = \frac{S^p_{u,v}}{\sqrt{S^p_{u,u} S^p_{v,v}}}. \quad (4)$$

B. Spectral Clustering for RS Fused with Random Walks

Clustering users may alleviate data sparsity. In the past, clustering has been performed to users' attributes [19] and more recently in [20]. Our proposal is focused on the spectral domain, when the clustering is performed on the underlying structure of data. There is a similar work in [21], although our proposal differs from [21] because we perform random walks in the similarity matrices of users belonging to each class. At the end, the final similarity matrix takes advantage of both the initial structure of all the users and the structure of each cluster of users. The overall algorithm is summarized as follows.

1) Compute Similarities $\mathbf{S}$ of size $(N \times N)$
2) Group people using spectral clustering on $\mathbf{S}$ matrix into C groups, thus generating C similarity matrices within each group of size $(N \times N)$. Let $\underset{l}{\mathbf{S}}$ denote the similarity matrix for cluster l where $(1 \le l \le C)$. $\underset{l}{S}_{u,v} \neq 0$ if and only if, users u and v belong to the l cluster.
3) Compute $\mathbf{S}^p$ and $\underset{1}{\mathbf{S}}^p$,$\underset{2}{\mathbf{S}}^p$,..., $\underset{C}{\mathbf{S}}^p$.
4) The final similarity matrix used is given by:

$$\tilde{S}_{u,v} = \begin{cases} \alpha S^p_{u,v} + (1-\alpha)\underset{l}{S}^p_{u,v}\text{: if } u,v \text{ belong to same} \\ \qquad\qquad \text{cluster, } l,\ 0 \le \alpha \le 1 \\ S^p_{u,v} \qquad\qquad \text{: otherwise} \end{cases}$$

TABLE I
SUMMARY OF SYMBOLS AND DEFINITIONS

Symbols	Definitions
$\mathcal{U}$	The set of users
N	Number of users
U_i	The i–th user $(1 \leq i \leq N)$
$\mathcal{I}$	the set of items
M	Number of items
I_u	The set of items that u user has rated
$\mathbf{R}$	The User–Item Matrix
$R_{u,i}$	The preference of user u for item i
$\bar{R}_u$	The average rating of user u
$\hat{R}_{u,i}$	The prediction of u user preference for i–th item
$\mathbf{S}$	Similarity matrix
$S_{u,v}$	Similarity value between users u and v
$S^p_{u,v}$	Similarity value between users u and v to the power p
S^p_u	The set of most similar users of user u to the power p
C	number of clusters
$\underset{l}{\mathbf{S}}$	Similarity matrix between users in the l–th cluster $(1 \leq l \leq C)$
$\underset{l}{\mathbf{S}^p}$	Similarity matrix between users in the l–th cluster to the power p $(1 \leq l \leq C)$
$\underset{l}{S^p_{u,v}}$	Similarity value between users u and v belonging to the l–th cluster to the power p $(1 \leq l \leq C)$

Again, we normalise the elements of the final similarity matrix using (4).

5) The recommendation is calculated as:

$$\hat{R}_{u,i} = \bar{R}_u + \frac{\sum\limits_{v \in V_k(u)} \tilde{S}_{u,v}(R_{u,i} - \bar{R}_v)}{\sum\limits_{v \in V_k(u)} \tilde{S}_{u,v}}$$

Notice that α is a scalar parameter that weigh the final values between the whole similarity matrix and the similarity that contains samples of one cluster, thus it provides a balance about how much the general structure matters against the structure of each group.

IV. EXPERIMENTS

In this section, we demonstrate the results of the experiments performed on a lastfm[1] dataset. The form of listening relationships are encoded in a UI matrix with total of 955 users and 12552 artists. Originally, they were more users and artists but we removed users that have not listened to at least 10 different artists and also removed artists that were not listened by at least 10 different users. Moreover, the UI matrix is very sparse, specifically only 4.86% of listening relationships are filled with a value.

The protocol to evaluate the proposed methods is as follows: split the UI matrix into a training UI matrix by obtaining the UI matrix that contains 80% of each user's listening relationships and the remaining 20% was used for evaluating the performance of the proposed methods against the typical CF. We performed 10 randomly selected splits, and the final performance is averaged.

The evaluation of a recommendation system is an area of great study [22]. To evaluate the methods, we used Kendall tau. This metric counts the number of pairwise disagreements between two ranking lists, hence the smaller the better. Finally, we are not interested in how close the rating prediction has been to reality, but whether the order of ratings is correct. At the end, the final performance is averaged for all users.

In Tables II and III, the performance of the proposed method is compared against that of typical CF approach. Each column denotes the number of top artists to be predicted and next to that the number of users that actually have enough artists. For example, in column 5, top 20 (949), there are only 949 out of 955 users that have rated at least 20 artists, thus we average the performance across these 949 users. In each column, the best result is denoted in boldface. We see that in all columns the proposed RS method outperforms the typical CF.

Using the best power, we observe that performing random walks in the similarity matrix between users can enhance the informations and provide more accurate predictions in all cases of top–N recommendations. Furthermore, by combining the random walks with spectral clustering techniques the performance can be further improved in some cases.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have enhanced known similarity matrices between users for CF, using random walks we have demonstrated that by incorporating implicit feedback by more users thanks to the aforementioned graph theoretic concept the performance of RS is benefited. We further combined the proposed method with spectral clustering techniques and have demonstrated further improvement.

Future work can be focused on performing experiments in more datasets and include content based information.

ACKNOWLEDGMENT

This research has been co–financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS–UOA–ERASITECHNIS MIS 375435.

[1] http://www.last.fm

TABLE II
KENDALL TAU EVALUATION OF RS

RS method	top 5 (955)	top 10 (953)	top 15 (951)	top 20 (949)	top 25 (947)	top 30 (944)
typical CF	0.4506	0.4440	0.4375	0.4336	0.4309	0.4298
power 2	0.4514	0.4444	0.4372	0.4318	0.4290	0.4277
power 3	0.4545	0.4468	0.4391	0.4333	0.4304	0.4292
best power	**0.4481**	**0.4430**	**0.4356**	0.4310	0.4289	0.4273
$C = 2$, $\alpha = 0.3$	0.4507	0.4436	0.4367	0.4306	0.4286	0.4277
$C = 2$, $\alpha = 0.5$	0.4509	0.4437	0.4369	**0.4302**	0.4281	0.4272
$C = 2$, $\alpha = 0.7$	0.4517	0.4440	0.4369	0.4302	0.4281	0.4270
$C = 3$, $\alpha = 0.3$	0.4508	0.4448	0.4372	0.4309	0.4285	0.4272
$C = 3$, $\alpha = 0.5$	0.4507	0.4443	0.4367	0.4305	**0.4281**	**0.4269**
$C = 3$, $\alpha = 0.7$	0.4506	0.4445	0.4367	0.4302	0.4281	0.4269

TABLE III
KENDALL TAU EVALUATION OF RS

RS method	top 35 (939)	top 50 (935)	top 60 (932)	top 70 (929)	top 80 (922)	top 100 (917)
typical CF	0.4285	0.4219	0.4206	0.4200	0.4193	0.4186
power 2	0.4264	0.4195	0.4179	0.4173	0.4168	0.4161
power 3	0.4279	0.4210	0.4197	0.4192	0.4189	0.4185
best power	**0.4257**	0.4196	0.4179	0.4172	0.4166	0.4160
$C = 2$, $\alpha = 0.3$	0.4266	0.4183	0.4169	0.4174	0.4169	0.4154
$C = 2$, $\alpha = 0.5$	0.4262	0.4180	0.4167	0.4172	0.4165	0.4149
$C = 2$, $\alpha = 0.7$	0.4257	**0.4178**	0.4164	0.4168	**0.4163**	**0.4148**
$C = 3$, $\alpha = 0.3$	0.4261	0.4180	0.4167	0.4173	0.4169	0.4152
$C = 3$, $\alpha = 0.5$	0.4259	0.4179	0.4165	0.4171	0.4167	0.4151
$C = 3$, $\alpha = 0.7$	0.4260	0.4179	**0.4162**	**0.4168**	0.4164	0.4148

REFERENCES

[1] Y. Song, S. Dixon, and M. Pearce, "A survey of music recommendation systems and future perspectives," 2012.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, no. 0, pp. 109 – 132, 2013.

[3] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 3:1–3:45, May 2014.

[4] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 81–173, Feb. 2011.

[5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[6] M. Grar, D. Mladeni, B. Fortuna, and M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *Advances in Web Mining and Web Usage Analysis*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4198, pp. 58–76.

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

[8] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," Microsoft Research, Tech. Rep. MSR-TR-98-12, May 1998.

[9] 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*, ser. WWW '01. New York, NY, USA: ACM, 2001, pp. 285–295.

[10] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Dec. 1996.

[11] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *J. Am. Soc. Inf. Sci. Technol.*, vol. 55, no. 3, pp. 259–274, Feb. 2004.

[12] C. wei Hang and M. P. Singh, "Trust-based recommendation based on graph similarity," in *in AAMAS Workshop on Trust in Agent Societies*, 2010.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 849–856.

[14] L. Ungar, D. Foster, E. Andre, S. Wars, F. S. Wars, D. S. Wars, and J. H. Whispers, "Clustering methods for collaborative filtering." AAAI Press, 1998.

[15] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *In IEEE International Conference on Data Mining (ICDM 2008)*, 2008, pp. 263–272.

[16] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, ser. HetRec '10. New York, NY, USA: ACM, 2010, pp. 47–51.

[17] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, Oct. 2002.

[18] B. J. Mirza, B. J. Keller, and N. Ramakrishnan, "Studying recommendation algorithms by graph analysis," *J. Intell. Inf. Syst.*, vol. 20, no. 2, pp. 131–160, Mar. 2003.

[19] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5-6, pp. 393–408, Dec. 1999.

[20] J. Rongfei, J. Maozhong, and L. Chao, "A new clustering method for collaborative filtering," in *Networking and Information Technology (ICNIT), 2010 International Conference on*, June 2010, pp. 488–492.

[21] A. Bellogin and J. Parapar, "Using graph partitioning techniques for neighbour selection in user-based collaborative filtering," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ser. RecSys '12. New York, NY, USA: ACM, 2012, pp. 213–216.

[22] G. Shani and A. Gunawardana, "Evaluating recommender systems," Tech. Rep. MSR-TR-2009-159, November 2009.