Modeling Trust and Distrust Information in Recommender Systems via Joint Matrix Factorization with Signed Graphs

Dimitrios Rafailidis
Department of Informatics
Aristotle University of Thessaloniki, Greece
draf@csd.auth.gr

ABSTRACT

We propose an efficient recommendation algorithm, by incorporating the side information of users' trust and distrust social relationships into the learning process of a Joint Non-negative Matrix Factorization technique based on Signed Graphs, namely JNMF-SG. The key idea in this study is to generate clusters based on signed graphs, considering positive and negative weights for the trust and distrust relationships, respectively. Using a spectral clustering approach for signed graphs, the clusters are extracted on condition that users with positive connections should lie close, while users with negative ones should lie far. Then, we propose a Joint Non-negative Matrix factorization framework, by generating the final recommendations, using the user-item and user-cluster associations over the joint factorization. In our experiments with a dataset from a real-world social media platform, we show that we significantly increase the recommendation accuracy, compared to state-of-the-art methods that also consider the trust and distrust side information in matrix factorization.

CCS Concepts

•Information systems → Collaborative and social computing systems and tools; Data mining;

Keywords

Recommender systems; matrix factorization; signed graphs

1. INTRODUCTION

Collaborative filtering is a widely used strategy in recommender systems. The main idea is that users who rate similar items tend to get similar recommendations [25]. Following the collaborative filtering strategy, several matrix factorization methods have been proposed, capable of revealing the latent associations between users and items [8]. However, an important issue that the matrix factorization methods face is that data sparsity negatively affects/degrades the recommendation accuracy. To handle sparsity several methods incorporate side information into the learning process of matrix factorization techniques, such as, users' demographics [22, 23], media content [14, 28], tag information [7], or any combination of them, such as performing personalized tag propagation based on media content [20].

In addition, towards handling the sparsity problem, various matrix factorization techniques have been proposed by exploiting the trust relationships among users, that is, social friends, such as the works of [18, 24]. The key idea in these works is that if two users have a friendship relation, then the recommendation from her/his friends probably has higher trustworthiness than the rest of users. Therefore, the challenge is how to combine, for example, a user-item rating matrix with the social/trust network of a user, so as to solve the sparsity problem and boost the recommendation accuracy. For example, Ma et al. [17] relied on the fact that the distance of users in vectorial representations in the latent space should be minimized, on condition that they have social trust relationships. Meanwhile, in the same work, authors showed that the distrust relationships between users can also be exploited, assuming that their corresponding latent features would have a large distance.

Recently, few works have been introduced to explicitly incorporate both the trust and distrust social relationships in matrix factorization [4, 27]. For example, in [4] the Matrix Factorization strategy with Trust and Distrust social relationships (MF-TD) has been proposed. The key idea is that the trust/distrust relationships between users are considered as similarity/dissimilarity in their preferences. To account this key factor, the latent features are computed in a manner such that the latent features of users who are distrusted by a certain user have a guaranteed minimum dissimilarity gap from the worst dissimilarity of users who are trusted by this user. This means that when the user agrees on an item with one of his trusted friends, he will probably disagree on the same item with his distrusted friends, assuming a minimum predefined margin.

1.1 Motivation

What is missing from the aforementioned matrix factorization strategies, is that users can be grouped based on their

SAC 2016, April 04-08, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2851613.2851697

trust and distrust relationships, considering that users in the same group should have trust relationships, while users in different groups should have distrust relationships. By performing user clustering, the sparsity problem can be further solved, because trusted and distrusted clusters can be formed, instead of considering explicitly trust and distrust social relationships in matrix factorization.

Meanwhile, several approaches consider both the trust and distrust relationships, when forming user groups, also known as community detection algorithms; for example, Lo et al. [15] analyze trust networks to extract direct antagonistic communities; Liu et al. [12] consider trust and distrust relationships as positive and negative links, respectively, and then they transform the problem of community detection into a multiobjective problem; Chiang et al. [3] exploit local and global aspects of social balance theory for predicting trust or distrust relationships, also known as the sign prediction problem, as well as for solving the clustering problem. Nonetheless, none of the aforementioned studies focus on the recommendation problem. In addition, there are some studies that incorporate user clusters into the learning process of factorization techniques, such as the study of [21]; however this study does not consider social trust and distrust relationships, when generating the clusters.

1.2 Contribution and Outline

We model trust and distrust information in recommender systems via **J**oint **N**on-negative **M**atrix **F**actorization with Signed Graphs, namely JNMF-SG. In the first step, we model trust and distrust relationships into Signed Graphs, where positive and negative edge weights are allowed. Then, using a spectral clustering approach for signed graphs, we extract user clusters. In the second step, we incorporate the extracted clusters into a Joint Non-negative Matrix Factorization (JNMF) framework in order to generate the final recommendations. As we will experimentally show the computed user clusters in the JNMF framework can boost the recommendation accuracy, by efficiently solving the sparsity problem.

The remainder of the paper is organized as follows, in Section 2 we present the proposed approach, in Section 3 we perform our experimental evaluation and Section 4 concludes this paper and provides interesting future directions in generating recommendations based on trust and distrust relationships.

2. PROPOSED APPROACH

2.1 Problem Formulation

In our setting, we consider n users, p product-items, as well as the sets $\mathcal{T}$ and $\mathcal{D}$ of trust and distrust social relationships, respectively. The social relationships $< i,j > \in \mathcal{T}$, with $i,j \in 1 \dots n$, or $< i,j > \in \mathcal{D}$, express the trust and distrust relationships between users i and j, respectively. Given a training set $\mathcal{S}^{train}$, we consider that users rate product-items, stored in a matrix $R \in \Re^{p \times n}$, where R_{wi} denotes the user i rating for product-item w. The goal is to *predict the ratings of a test set* $\mathcal{S}^{test}$*, stored in a matrix* $\hat{R} \in \Re^{p \times n}$ *by factorizing the matrix* R.

The proposed JNMF-SG method consists of two steps. In the first step, we model the trust and distrust social relationships into a signed graph, and then we generate k clusters, by trying to assign users with trust social relationships to the same cluster, and users with distrust ones to different clusters. Following a spectral clustering approach of signed graphs, the outcome of the first step is a cluster-user matrix $C \in \Re^{k \times n}$, where k is the number of clusters and C_{qi}=1 denotes that user i belongs to cluster q. In the second step, given matrices C and R, we perform Joint Non-negative Matrix Factorization (JNMF) to compute the final factorized matrix $\hat{R}$. The remainder of this Section details each step of our approach.

2.2 Spectral Clustering in Signed Graphs

Spectral clustering preliminaries: According to [16], given the vertices V of a graph G, spectral clustering methods differ in how they define and construct the Laplacian matrix L of the adjacency matrix A and, thus, which eigenvectors are selected to represent the graph, aiming to exploit special properties of different matrix formulations. Without loss of generality, in our approach, we focus on the Ratio Cut spectral clustering technique [16], which tries to minimize the total cost of the edges crossing the cluster boundaries. Ratio Cut also normalizes the k clusters by their sizes, to encourage balanced cluster sizes. Alternatively, we could use other spectral clustering techniques, such as Normalized Cut (NCut), which has different objective criterion in the clustering task than Ratio Cut [16]. In the ordinary case of unsigned graphs, Ratio Cut tries to find the cuts of a graph $G = (V, E, A)$, with V being the sets of nodes, E the sets of edges including both trust and distrust relationships and A being the adjacency matrix. A cut of G is a partition of the vertices $V = V_1 \cup V_2$. The weight of the cut is calculated by:

$$\mathrm{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij} \quad (1)$$

$\mathrm{cut}(\cdot)$ measures how well the two clusters are connected. To normalize the cuts by the size of the clusters, denoted by $|V_1|$ and $|V_2|$, the $\mathrm{RatioCut}(\cdot)$ is equal to:

$$\mathrm{RatioCut} = \mathrm{cut}(V_1, V_2) \times \left(\frac{1}{|V_1|} + \frac{1}{|V_2|} \right) \quad (2)$$

The clustering problem of the Ratio Cut spectral approach is how to minimize the $\mathrm{RatioCut}(\cdot)$ of the respective partitions into k clusters. The minimization of the optimization problem of RatioCut is thoroughly examined in [16].

Modeling trust and distrust information and signed graph clustering: In our approach, the trust and distrust social relationships are modeled into the adjacency matrix $A \in \Re^{n \times n}$ of a signed graph, where trust/distrust social relationships have positive/negative weights, denoted by $A_{ij} > 0$ and $A_{ij} < 0$, respectively. We construct the signed Laplacian matrix $\bar{L} = \bar{D} - A$, with $\bar{L} \in \Re^{n \times n}$. $\bar{D} \in \Re^{n \times n}$ is the signed degree matrix, which is a diagonal matrix, having entries only in the main diagonal. We would like to mention that in this paper we consider the $\bar{X}$ symbol of a matrix X, as a signed matrix, which differs from the ordinary notation of unsigned graphs, as in the case of signed graphs negative weights of edges are allowed. According to [6], we calculate the signed degree matrix $\bar{D}$ as follows:

$$D_{ii} = \sum_{j=1}^{n} |A_{ij}| \tag{3}$$

Let A^+ and A^- be the adjacency matrices with only positive and negative edges, respectively. Following the Ratio Cut approach in Eq. (1), the positive and negative cuts that only count positive and negative edges, respectively, are defined as follows:

$$\begin{aligned} \mathrm{cut}^+(V_1, V_2) &= \sum_{i \in V_1, j \in V_2} A_{ij}^+ \\ \mathrm{cut}^-(V_1, V_2) &= \sum_{i \in V_1, j \in V_2} A_{ij}^- \end{aligned} \tag{4}$$

According to [9], the signedCut of a partition of two sets of vertices V_1 and V_2 is defined as:

$$\begin{aligned} \mathrm{signedCut}(V_1, V_2) &= 2 \times \mathrm{cut}^+(V_1, V_2) \\ &+ \mathrm{cut}^-(V_1, V_1) + \mathrm{cut}^-(V_2, V_2) \end{aligned} \tag{5}$$

Similar to Eq. (2), the signedRatioCut equals:

$$\mathrm{signedRatioCut} = \mathrm{signedCut}(V_1, V_2) \times \left(\frac{1}{|V_1|} + \frac{1}{|V_2|} \right) \tag{6}$$

Hence, as presented in Eq. (6) similar to the case of Ratio Cut in unsigned graphs, the goal is to minimize the signedRatioCut$(\cdot)$ of the respective partitions to generate the final k clusters. In our approach, we followed the optimization algorithm of the Ratio Cut approach in [16], in order to generate the final k clusters. Finally, the cluster-user memberships are stored in the matrix C_{qi}, with $q \in 1 \dots k$ and $i \in 1 \dots n$.

2.3 Joint Non-negative Matrix Factorization

The inputs of this step are the rating matrix R and the user-cluster matrix C. In the case of ordinary Non-negative Matrix Factorization (NMF) [11] we have to perform the following eigen-decompositions of both matrices, separately:

$$\begin{aligned} \hat{R} &\approx U^{(R)} {V^{(R)}}^T \\ \hat{C} &\approx U^{(C)} {V^{(C)}}^T \end{aligned} \tag{7}$$

where the rows of $U^{(R)} \in \Re^{p \times d}$ and $U^{(C)} \in \Re^{k \times d}$ are the latent features of product-items and clusters, based on R and C, respectively, with d being the number of latent factors, and the rows of $V^{(R)} \in \Re^{n \times d}$ and $V^{(C)} \in \Re^{n \times d}$ being the latent features of users based on the matrices R and C, respectively. In addition, in the NMF approach the following constraints must be satisfied: $U^{(R)}, V^{(R)}, U^{(C)}, V^{(C)} \geq 0$. Our goal in this step is *to compute the factorized matrix* $\hat{R}$*, that is, the product of the* $U^{(R)}$ *and* $V^{(R)}$ *matrices of Eq. (7), by simultaneously considering the factorization matrix* $\hat{C}$ *of the cluster-user matrix* C. According to [13], in a JNMF framework, we have to compute a consensus matrix $V^* \in \Re^{n \times d}$, which forces the factorization of both matrices towards a common consensus. In particular, the goal is to compute matrices $V^*, U^{(R)}, V^{(R)}, U^{(C)}, V^{(C)}$, by minimizing the following loss function $\mathfrak{L}$:

$$\begin{aligned} \mathfrak{L}(V^*, U^{(R)}, V^{(R)}, U^{(C)}, V^{(C)}) = ||R - U^{(R)} {V^{(R)}}^T||_F^2 \\ + \lambda_1 ||V^{(R)} - V^*||_F^2 \\ + ||C - U^{(C)} {V^{(C)}}^T||_F^2 \\ + \lambda_2 ||V^{(C)} - V^*||_F^2 \end{aligned}$$

where $||\cdot||_2$ denotes the Frobenius norm. In the loss function $\mathfrak{L}$, the first and the third terms denote the approximation errors of the factorized matrices $\hat{R}$ and $\hat{C}$ of the matrices R and C, respectively, while the second and the fourth terms are the disagreement measurements, that is the regularization terms, between the matrices $V^{(R)}$,$V^{(C)}$ and the consensus matrix V^*, respectively. Note that $V^{(R)}$,$V^{(C)}$ and V^* are comparable, as they have the same dimensionality ($n \times d$), with the users latent factors. The regularization parameters λ_1 and λ_2 control the impact of the second and fourth regularization terms on $\mathfrak{L}$[1]. To calculate the consensus matrix V^* and the decomposition matrices $U^{(R)}, V^{(R)}, U^{(C)}, V^{(C)}$, we have to minimize the loss function $\mathfrak{L}$. In our approach, we solve the minimization problem of $\mathfrak{L}$, using the "multiplicative rules" [13]. The key idea is to solve the minimization problem by using an iterative update procedure, where in each iteration we fix V^* and minimize $\mathfrak{L}$ over the rest of matrices, and then we fix the rest of matrices and minimize $\mathfrak{L}$ over V^*. The iterations keep updating the matrices until convergence. The proof that the "multiplicative rules" are convergent can be found at [13]. Finally, having computed $U^{(R)}, V^{(R)}$, according to Eq. (7) we calculate the factorized matrix $\hat{R}$.

3. EXPERIMENTS

3.1 Experimental Setup

Data set: We use a real-world dataset from Epinions [5], also used in [4]. This dataset contains discrete value ratings from 1 to 5, containing trust and distrust social relationships. For comparison reasons, we conduct our experiments on a similar dataset, at the same scale of the evaluation dataset, as presented in [4]. To achieve this, we sampled a subset of the Epinions dataset with n=119,867 users and p=676,436 product-items, $|\mathcal{T}|$=452,123 trust, and $|\mathcal{D}| = 92,417$ distrust relationships. The total ratings are 12,328,927, that is, the non-zero elements in the rating matrix R.

Evaluation protocol: The data set was split into training and test sets, denoted by $\mathcal{TS}^{train}$ and $\mathcal{TS}^{test}$, respectively. We preserved a percentage of ratings of R in $\mathcal{TS}^{train}$ and the rest of ratings were hidden and stored in $\mathcal{TS}^{test}$. Given the ratings of R in $\mathcal{TS}^{train}$, the goal of each examined model (Section 3.2) is to predict the hidden ratings in $\mathcal{TS}^{test}$. We measure the performance of the examined methods in terms of Mean Absolute Error (MAE) and Root Mean Squared

[1]The tuning of the regularization parameters λ_1 and λ_2 was performed based on 5-fold cross validation, where we concluded in λ_1=λ_2=0.01.

Error ($RMSE$) [4, 17, 18]. Given the ratings to be predicted in the test set $\mathcal{TS}^{test}$, the actual rating values in R, and the predicted values in the factorized matrix $\hat{R}$, then MAE and $RMSE$ are defined as follows:

$$MAE = \frac{\sum_{(i,j)\in\mathcal{TS}^{test}} |R_{ij} - \hat{R}_{ij}|}{|\mathcal{TS}^{test}|} \tag{8}$$

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\mathcal{TS}^{test}} (R_{ij} - \hat{R}_{ij})^2}{|\mathcal{TS}^{test}|}} \tag{9}$$

where MAE considers every error of equal value, while $RMSE$ emphasizes larger errors [4, 17, 18]. In our experiments, we used different sizes $|\mathcal{TS}^{train}|$ of training sets, that is, 90%, 80% , 70%, and 60% of the total ratings in R, while the remaining ratings were used as test sets. As we performed random selection of the training and the respective test sets, we repeated our experiments five times. Hence, in our experiments, we report mean values and standard deviations of MAE and $RMSE$.

3.2 Compared Methods

- *Baseline* [11] is a baseline Non-negative Matrix Factorization (NMF) method, without using the additional information of trust and distrust social relationships. In this method, we used only the first part of Eq. (7) $\hat{R} \approx U^{(R)}V^{(R)^T}$, subject to $U^{(R)}, V^{(R)} \geq 0$, in order to calculate the low rank approximation $\hat{\bar{R}}$ of the initial user-product matrix R.

- *Baseline+Trust* [17] is a matrix factorization strategy that incorporates the trust social relationships, which assumes that the distance between the latent features of users who trust each other must be minimized. Without loss of generality, we set $\lambda_U = \lambda_V$, with λ_U and λ_V being the regularization parameters for the user and product matrices in the objective function of Baseline+Trust, respectively. In our implementation, we varied the parameters in [10^{-3} 10^3]. Using 5-fold cross validation, we concluded in λ_U=λ_V=10. In addition, we set $\alpha = 1$, with α controlling how much the Baseline+Trust method should use the information of trust information, where a selection of large values of parameter α indicates that the trust information will dominate the learning process of the low-rank approximation $\hat{R}$.

- *Baseline+Distrust* [17] is a matrix factorization strategy that incorporates the distrust information, with the latent features of users who are connected with a distrust social relationship, having large distance. Here, we set the regularization parameters as in the Baseline+Trust method. We also set $\beta = 10$, with β expressing how much Baseline+Distrust exploits the distrust relationships in the learning process.

- *MF+TD* [4] is the most competitive method, as it models distrust relations into the matrix factorization problem along with trust relations at the same time (Section 1). Regarding the parameter tuning in the implementation of MF+TD, we used 5-fold cross validation, where we concluded in the following regularization parameters of the MF+TD's objective function, λ_S=λ_U=λ_V=10, by varying the parameters in [10^{-3} 10^3], as in the Baseline+Trust and Baseline+Distrust methods. Accordingly, λ_U and λ_V are the regularization parameters for user and product matrices, respectively in the objective function of MF+TD, while λ_S controls how much the MF+TD should incorporate the information of the social network (trust and distrust relationships) in computing the low-rank approximation $\hat{R}$.

For making fair comparison, in all the aforementioned matrix factorization techniques, including the proposed JNMF-SG method, we set the number of latent factors d equal to ten. At this point we must mention that several nearest neighbors-based techniques have been introduced to solve the recommendation problem, such as the study of [27]. The goal of nearest neighbors-based methods is to evaluate a user's preference for a product based on the ratings of "neighboring" items by the same user. A product's neighbors are other products that tend to get similar ratings when rated by the same user. However, it has been observed that matrix factorization models are superior to classic nearest neighbors-based techniques [4, 8]; thus, a comparison against such techniques has been omitted in the experimental evaluation.

All experiments were performed on a 14-core two processor Intel Xeon E5 v3 2.60GHz machine, with 128GB RAM, and 223GB Hard Disk.

3.3 Results

In Figures 1 and 2, we report the experimental results, by comparing the examined methods for different training set sizes, corresponding to different levels of sparsity. We observe that, when the training set is smaller, all methods have higher errors in the rating predictions. This happens because the sparsity is increased, when reducing the training set. In contrast to the Baseline method, the rest matrix factorization techniques achieve lower errors, by exploiting the side information of the trust (Baseline+Trust), distrust (Baseline+Distrust) relationships, or even both types of relationships (MF-TD and JNMF-SG). Especially, the proposed JNMF-SG method and MF-TD have significantly lower errors, as both methods exploit the trust and distrust relationships at the same time. However, the proposed JNMF-SG method outperforms MF-TD for all the different training set sizes. This occurs, because the matrix factorization technique of MF-TD tries to explicitly compute the latent features, such that the latent features of users who are distrusted by a certain user i have a minimum dissimilarity gap from the worst dissimilarity of users who are trusted by user i; whereas, in our approach, the compact cluster-user form of matrix C, generated by the trust and distrust social relationships according to the signed graph clustering method (Section 2.2), can help the Joint Non-negative Matrix Factorization framework of JNMF-SG to handle the sparsity problem, thus significantly minimizing the rating prediction errors.

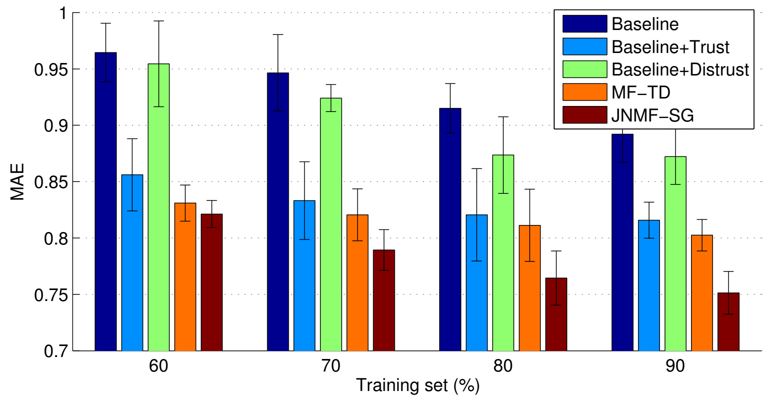

Figure 1: Methods comparison in terms of MAE.

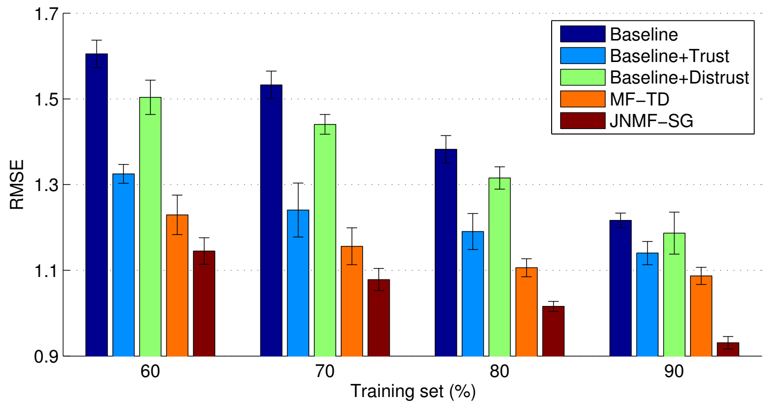

Figure 2: Methods comparison in terms of $RMSE$.

4. CONCLUSION

In this paper, we presented JNMF-SG, an efficient two-step matrix factorization framework, by exploiting the side information of the trust and distrust social relationships. In the first step, we model the trust and distrust relationships into signed graphs, and then we generate clusters, by utilizing a spectral approach. The generated clusters are incorporated into the learning process of a Joint Non-negative Matrix Factorization framework, in order to solve the sparsity that usually occurs in the user ratings. In our experiments, we demonstrated that the proposed JNMF-SG method can significantly lower the prediction errors for different levels of sparsity, compared to state-of-the art matrix factorization techniques, that either exploit the trust or the distrust social relationships, separately, or exploiting both types of relationships, simultaneously.

Recent research has started to incorporate temporal effects into matrix factorization models, by observing "drifts" in the rating behavior and modeling the way user and product characteristics change over time [2]. An open challenge is to extend our method for Incremental Matrix factorization techniques [19, 26], in order to capture how users change their rating behaviour over time. In addition, existing approaches for the recommendation problem do not take into account the fact that in several applications users interact with products over time [10], or unlike [10] that focuses on explicit feedback in the form of ratings, implicit quantitative feedback may be provided by users, such as the number of times they listened to a song or downloaded a video within a time period. This problem is called the *"repeat consumption"* recommendation problem and recently a few methods have been introduced to solve it, such as the studies of [1, 22, 23]. Our future plans include to examine how much trust and distrust relationships influence the learning process of factorization strategies for the repeat consumption problem.

5. REFERENCES

[1] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii. The dynamics of repeat consumption. In *ACM International Conference on World Wide Web WWW'15*, pages 419–430, 2014.

[2] D. G. F. M. Bollen, M. P. Graus, and M. C. Willemsen. Remembering the stars?: effect of time on preference retrieval from memory. In *ACM Conference on Recommender Systems, RecSys '12*, pages 217–220, 2012.

[3] K. Chiang, C. Hsieh, N. Natarajan, I. S. Dhillon, and A. Tewari. Prediction and clustering in signed networks: a local to global perspective. *Journal of Machine Learning Research*, 15(1):1177–1213, 2014.

[4] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans. Inf. Syst.*, 32(4):17:1–17:38, 2014.

[5] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *ACM International Conference on World Wide Web*, WWW '04, pages 403–412, 2004.

[6] Y. Hou. Bounds for the least laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4):955–960, 2005.

[7] B. S. Kim, H. Kim, J. Lee, and J.-H. Lee. Improving a recommender system by collective matrix factorization with tag information. In *Soft Computing and Intelligent Systems*, SCIS '14, pages 980–984, 2014.

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

[9] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. D. Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *SIAM International Conference on Data Mining*, SDM '10, pages 559–570, 2010.

[10] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *ACM SIGIR International Conference on Research and Development in Information Retrieval, SIGIR'10*, pages 210–217, 2010.

[11] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[12] C. Liu, J. Liu, and Z. Jiang. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Transactions Cybernetics*, 44(12):2274–2287, 2014.

[13] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *SIAM International Conference on Data Mining*, SMD '13, pages 252–260, 2013.

[14] N. N. Liu, L. He, and M. Zhao. Social temporal collaborative ranking for context aware movie recommendation. *ACM Transactions on Intelligent Systems and Technology*, 4(1):15:1–15:26, 2013.

[15] D. Lo, D. Surian, K. Zhang, and E. Lim. Mining direct antagonistic communities in explicit trust networks. In *ACM International Conference on Conference on Information and Knowledge Management*, CIKM'11.

[16] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[17] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *ACM Conference on Recommender Systems*, RecSys '09, pages 189–196, 2009.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *ACM Conference on Information and Knowledge Management*, CIKM '08, pages 931–940, 2008.

[19] P. Matuszyk, J. a. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. a. Gama. Forgetting methods for incremental matrix factorization in recommender systems. In *ACM Symposium on Applied Computing*, SAC '15, pages 947–953, 2015.

[20] D. Rafailidis, A. Axenopoulos, J. Etzold, S. Manolopoulou, and P. Daras. Content-based tag propagation and tensor factorization for personalized item recommendation based on social tagging. *ACM Transactions on Interactive Intelligent Systems*, 3(4):26, 2014.

[21] D. Rafailidis and P. Daras. The TFC model: Tensor factorization and tag clustering for item recommendation in social tagging systems. *IEEE Transactions Systems, Man, and Cybernetics: Systems*, 43(3):673–688, 2013.

[22] D. Rafailidis and A. Nanopoulos. Modeling the dynamics of user preferences in coupled tensor factorization. In *ACM Conference on Recommender Systems, RecSys'14*, pages 321–324, 2014.

[23] D. Rafailidis and A. Nanopoulos. Repeat consumption recommendation based on users preference dynamics and side information. In *ACM International Conference on World Wide Web Companion, WWW'15 - Companion Volume*, pages 99–100, 2015.

[24] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Conference on Neural Information Processing Systems*, NIPS '07, pages 1257–1264, 2007.

[25] 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*, WWW '01, pages 285–295, 2001.

[26] Q. Song, J. Cheng, and H. Lu. Incremental matrix factorization via feature space re-learning for recommender system. In *ACM Conference on Recommender Systems*, RecSys '15, pages 277–280, 2015.

[27] P. Victor, N. Verbiest, C. Cornelis, and M. D. Cock. Enhancing the trust-based recommendation process with explicit distrust. *TWEB*, 7(2):6, 2013.

[28] H. Zhang, Z. Li, Y. Chen, X. Zhang, and S. Wang. Exploit latent dirichlet allocation for one-class collaborative filtering. In *ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1991–1994, 2014.