

A Regularization Method with Inference of Trust and Distrust in Recommender Systems

Dimitrios Rafailidis¹ and Fabio Crestani²

¹ Department of Computer Science, University of Mons, Belgium
`dimitrios.rafailidis@umons.ac.be`

² Faculty of Informatics, Università della Svizzera italiana, Switzerland
`fabio.crestani@usi.ch`

Abstract. In this study we investigate the recommendation problem with trust and distrust relationships to overcome the sparsity of users' preferences, accounting for the fact that users trust the recommendations of their friends, and they do not accept the recommendations of their foes. In addition, not only users' preferences are sparse, but also users' social relationships. So, we first propose an inference step with multiple random walks to predict the implicit-missing trust relationships that users might have in recommender systems, while considering users' explicit trust and distrust relationships during the inference. We introduce a regularization method and design an objective function with a social regularization term to weigh the influence of friends' trust and foes' distrust degrees on users' preferences. We formulate the objective function of our regularization method as a minimization problem with respect to the users' and items' latent features and then we solve our recommendation problem via gradient descent. Our experiments confirm that our approach preserves relatively high recommendation accuracy in the presence of sparsity in both the users' preferences and social relationships, significantly outperforming several state-of-the-art methods.

Keywords: Recommender systems, collaborative filtering, social relationships, regularization.

1 Introduction

Recommender systems have widely followed the collaborative filtering strategy, where similar-minded users tend to get similar recommendations [8]. In real-world scenarios, the main limitation of collaborative filtering strategy is the data sparsity of users' preferences, significantly degrading the recommendation accuracy. To overcome the data sparsity problem and generate trust-based recommendations, several models exploit the selections of trust friends [4, 12]. In trust-based recommendations, models consider that people tend to rely on recommendations from their friends [6, 10]. In addition, in online networks users may establish both trust and distrust relationships, while the vast majority of users have unknown relationships. Epinions³, an e-commerce site for reviewing

³ <http://www.epinions.com/>

and rating products, allows users to evaluate others based on the quality of their reviews, and form trust and distrust relations with them. In Slashdot⁴ users post news and comments, and tag other users as friends or foes. The analyses in related studies conclude that users might accept recommendations from their trusted friends, but will certainly exclude recommendations from their distrusted foes [20, 21, 23]. In this respect, more recently a few attempts have been made to exploit both trust and distrust relationships at the same time in recommender systems [1, 2, 15, 19]. Following the collaborative filtering strategy, these models assume that the latent features of trust/distrust users should be as close/far as possible. However, such models exploit trust and distrust relationships in an *explicit* manner, that is users' direct friends or foes, and do not *infer implicit (indirect)* social relationships that users establish in recommender systems. Due to the presence of sparsity in social relationships, models that use trust and distrust relationships exploit a limited number of social relationships, thus not performing well as we will show in our experiments in Section 5.

Inferring social relationships of trust and distrust users is a challenging task [20]. Given explicit social relationships, the goal is to infer the indirect relationships of trust and distrust users. Trust relationships show strong transitivity, which means that inferring trust relationships can be computed in a network of trust users, mainly because if two users a and b are friends and a third user c is friend with a , then user c might be a friend of b as well. However, recent studies showed that distrust is certainly not transitive [3, 21, 23]. Therefore, distrust cannot be considered as the negative of trust when inferring users' distrust relationships. Accounting for the transitivity of trust relationships, a few prediction models have been proposed to infer the implicit trust relationships, while exploiting explicit distrust relationships in their predictions [7, 20]. Nonetheless, these models are designed to predict missing-trust relationships and not to generate recommendations. Therefore, a pressing challenge resides on how to infer trust relationships of users with their distrust relationships to boost the recommendation accuracy.

1.1 Contribution

To overcome the shortcomings of existing methods our contributions are summarized as follows, first we infer implicit trust relationships with multiple random walks while considering the users' explicit distrust relationships during the inference. By significantly enhancing users' relationships and reducing the number of unknown relationships, we formulate a social regularization term to weigh users' trust and distrust degrees and capture the correlations of users' preferences with those of their friends and foes. We introduce a regularization method and design an objective function as a minimization problem with respect to the latent features, and solve our recommendation problem via gradient descent. Our experiments show that the proposed approach is superior to all the competitors

⁴ <https://slashdot.org/>

at different levels of sparsity in users' preferences, as well as in users' social relationships.

The remainder of the paper is organized as follows, Section 2 reviews the related work and in Section 3 we formally define our problem. Section 4 details the proposed model, Section 5 presents the experimental results and Section 6 concludes the study.

2 Related Work

In trust-based recommender systems, many different strategies have been introduced to exploit the selections of trust users. Jamali and Ester [6] extend the probabilistic matrix factorization of [12] by weighting the user latent factors based on their trust relationships. In [10], a trust-based ensemble method is presented to combine matrix factorization with a social-based neighborhood model. In [5], a random walk model is introduced to incorporate a trust-based approach into a neighborhood-based collaborative filtering strategy. In this study, authors run random walks on the trust network, formed by the trust relationships, and then perform a probabilistic item selection strategy to generate recommendations. The random walk performs the search in the trust network and the item selection part considers ratings on similar items to avoid going too deep in the network. However, TrustWalker does not infer implicit (indirect) trust relationships, as it performs random walks to generate recommendations using a neighborhood-based model. Guo et al. [4] extend SVD++ [8], to learn both the user preferences and the social influence of her friends. However, all the above studies ignore users' distrust relationships when generating recommendations, an important factor to boost the recommendation accuracy [23]. Ma et al. [11] present a social regularization method by sharing a common user-item matrix, factorized by ratings and social relationships. This work introduces a trust-based as well as a distrust-based model to exploit trust and distrust relationships in each model separately. The goal of the trust-based model is to minimize the distances of latent features between trust users, while the distrust-based model tries to maximize the latent features' distances between distrust users.

Recently, a few attempts have been made to exploit both trust and distrust relationships at the same time in recommender systems. Forsati et al. [2] incorporate trust and distrust relationships into a matrix factorization framework, by formulating a hinge loss function. This method assumes that the trust/distrust relationships between users are considered as similarity/dissimilarity in their preferences, and then the latent features are computed in a manner such that the latent features of foes who are distrusted by a certain user have a guaranteed minimum dissimilarity gap from the maximum dissimilarity of friends who are trusted by this user. This means that when the user agrees on an item with one of her trusted friends, she will probably disagree on the same item with her distrusted foes, assuming a minimum predefined margin. In [1], a recommendation strategy is introduced to rank the latent features of users, based on the users' trust and distrust relationships. This method also considers the neutral

relationships (of users who have no relation to a certain user), aiming to rank the user latent features after the trust relationships and before the distrust ones. In [15] a signed graph is constructed, considering positive and negative weights for the trust and distrust relationships, respectively. Then, a spectral clustering approach is presented to generate clusters in the signed graph. The clusters are extracted on condition that users with positive connections should lie close, while users with negative ones should lie far. Following a joint non-negative matrix factorization framework the final recommendations are generated, by co-factorizing the user-item and user-cluster associations. Tang et al. [19] form a signed graph with trust and distrust relationships and capture local and global information from the graph. Local information reveals the correlations among a user and her friends/foes, and the global information reveals the reputation of the user in the whole social network, as users tend to trust users with high global reputation. Then, they exploit both local and global information in a matrix factorization technique to produce recommendations.

3 Problem Formulation

Let n and m be the numbers of users and items. Given a rating matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, each entry \mathbf{R}_{iq} corresponds to the rating that user i has assigned to item q , with $i = 1 \dots n$ and $q = 1 \dots m$. If a user i has not rated an item q , then we set $\mathbf{R}_{iq} = 0$. In our approach we follow the collaborative filtering strategy of matrix factorization. This means that by factorizing the matrix \mathbf{R} , the recommendations are in its low-rank approximation $\hat{\mathbf{R}} \in \mathbb{R}^{n \times m}$. Matrix $\hat{\mathbf{R}}$ can be calculated as $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n \times d}$ is the user factor matrix, and $\mathbf{V} \in \mathbb{R}^{m \times d}$ is the item factor matrix with d the number of latent dimensions. To consider users' trust and distrust relationships, we also form a graph \mathcal{G} with $n = |\mathcal{V}|$ nodes and $i, j \in \mathcal{V}$. Two nodes are connected with edges in the form $(i, j) \in \mathcal{E}$. The edges are considered weighted, and in our setting we consider positive and negative weights to express trust and distrust relationships, respectively. Both positive and negative weights are stored in an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. In our approach we generate two different graphs, a graph G_+ which contains only the positive edges and a second graph G_- with the negative ones. Given $\mathcal{E} \equiv \mathcal{E}_+ \cup \mathcal{E}_-$, we compute two different adjacency matrices $\mathbf{A}_+ \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_- \in \mathbb{R}^{n \times n}$, corresponding to the weights of the positive $(i, j)_+ \in \mathcal{E}_+$ and negative $(i, j)_- \in \mathcal{E}_-$ edges/relationships. In addition, $\forall (i, j)_- \in \mathcal{E}_-$ we set $(\mathbf{A}_-)_{ij} = |\mathbf{A}_{ij}|$, storing the absolute values of the negative weights. With this setting, the goal of the proposed approach is formally defined as follows:

Definition 1 (Problem). *“Given (i) the rating matrix \mathbf{R} and (ii) the adjacency matrices \mathbf{A}_+ and \mathbf{A}_- with the trust and distrust relationships, the goal of the proposed approach is first to infer the implicit trust relationships based on users' explicit trust and distrust relationships, and then compute the low-rank approximation $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$ to generate the final recommendations.”*

4 Proposed Approach

4.1 Social Inference via Multiple Random Walks

To infer the implicit trust relationships, we perform random walks on the n nodes in graph \mathcal{G}_+ , by taking into account the distrust relationships in graph \mathcal{G}_- during the inference. In particular, the proposed approach runs multiple random walks on the graph \mathcal{G}_+ with the trust relationships and then filters out the inferred trust relationships by considering the distrust relationships in graph \mathcal{G}_- . The main reason that we avoid to perform random walks on graph \mathcal{G}_- is that *distrust is not transitive*, as opposed to trust [7, 20, 21, 23]. Next, we present the case of performing a single random walk on graph \mathcal{G}_+ and we show how to perform multiple random walks to better infer the implicit trust relationships.

Single Random Walk. Given a source node sou and a target node tar in graph \mathcal{G}_+ , with $(sou, tar) \notin \mathcal{E}_+$, the goal is to start a random walk from sou to reach tar to infer their trust relationships, denoted as $(\mathbf{A}_+)_{sou, tar}$. We assume that the walk moves from one node to a neighbourhood node at each step, and at time t has moved to node i . The walk chooses whether to move to another node with probability ξ_t or terminate the walk with probability $1-\xi_t$. In the case of terminating the walk, the value $(\mathbf{A}_+)_{sou, tar}$ is returned only if edge $(i, tar) \in \mathcal{E}_+$, and 0 otherwise. The *transition probability* of moving from a current node i to another node j is calculated as follows:

$$p_+(j|i) = (\mathbf{A}_+)_{ij}/d_i \quad (1)$$

where $d_i = \sum_j (\mathbf{A}_+)_{ij}$ is the degree of i . The *transition matrix* $\mathbf{T}_+ \in \mathbb{R}^{n \times n}$ of a random walk is given by

$$\mathbf{T}_+ = \mathbf{D}_+^{-1} \mathbf{A}_+ \quad (2)$$

where $(\mathbf{D}_+)_{ii} = d_i$ is the degree diagonal matrix, with $\mathbf{D}_+ \in \mathbb{R}^{n \times n}$. A vector $\mathbf{p}_+^{(t)} \in \mathbb{R}^n$ represents the visiting distribution over all n nodes at a certain time t . With these settings, if the walk continues at the next time $t+1$, the distribution vector will be updated as follows:

$$\mathbf{p}_+^{(t+1)} = \mathbf{p}_+^{(t)} \times \mathbf{T}_+ \quad (3)$$

In the case of isolated users, random walks are not performed, as these users have not expressed their social preferences.

Multiple Random Walks. Instead of performing a single walk, we run multiple random walks from a source node in graph \mathcal{G}_+ to better infer the implicit trust relationships. The main reason that we can achieve better inference is that *multiple random walks start from the source user sou to seek more alternatives for the implicit (indirect) relationship to the target user tar* . Given the graph \mathcal{G}_+ , we define s as the total length of a single walk for which we recursively update

the distribution vector $\mathbf{p}^{(s)}$. For a target node tar we consider all its in-linked edges, denoted by $(\mathbf{A}_+)_{*tar}$, that is the tar -th column vector of \mathbf{A}_+ . With these settings, the returned value for a random walk terminated at time s is:

$$(\overline{\mathbf{A}}_+)_{sou,tar}|s = \mathbf{p}^{(s)}(\mathbf{A}_+)_{*tar} \quad (4)$$

Theoretically, we can perform random walks with infinite lengths from the source node. Aggregating the multiple random walks from the source node we have:

$$(\overline{\mathbf{A}}_+)_{sou,tar} = \sum_{t=1}^{\infty} \omega_+(t) \mathbf{p}_+^{(0)} \mathbf{T}_+^t (\mathbf{A}_+)_{*tar} \quad (5)$$

where $\mathbf{p}_+^{(0)}$ is the starting distribution of a walk on \mathcal{G}_+ and $\omega_+(t)$ expresses the probability that a random walk will terminate at a certain time t :

$$\omega_+(t) = p_+(s = t|\xi) = \xi_t \prod_{i=1}^{t-1} (1 - \xi_i) \quad (6)$$

Therefore, the adjacency matrix $\overline{\mathbf{A}}_+$ with the inferred trust relationships is calculated as follows:

$$\overline{\mathbf{A}}_+ = \sum_{t=1}^{\infty} \omega_+(t) \mathbf{T}_+^t \mathbf{A}_+ \quad (7)$$

In our implementation, we avoid long (infinite) walks on the graph, following the idea of the ‘‘six degrees of separation’’, that is most nodes can be reached with a six step walk length [13]. This means that if a walk has reached more than six steps, then the walk is terminated. In practice, we observed that random walks do not reach more than 4 steps in our experiments with $\xi_t = 0.85$, equal to the dampening factor of PageRank [14]. Next, the inferred trust relationships are stored in matrix $\overline{\mathbf{A}}_+$.

When performing multiple random walks on graph \mathcal{G}_+ , the distrust relationships in graph \mathcal{G}_- are ignored. Consequently, an inferred trust relationship between a source user sou and a target user tar in $\overline{\mathbf{A}}_+$, might have a distrust relationship between sou and tar in graph \mathcal{G}_- . Since users do not accept the recommendations of distrust users [2, 7], we recompute matrix $\overline{\mathbf{A}}_+$ by setting $\overline{\mathbf{A}}_+ \leftarrow 0$, if $(\overline{\mathbf{A}}_+)_{ij} > 0 \wedge (\mathbf{A}_-)_{ij} > 0, \forall i, j = 1 \dots n$. Finally, the filtered trust relationships and their positive weights are stored into the initial adjacency matrix with the trust relationships, by setting $\mathbf{A}_+ \leftarrow \overline{\mathbf{A}}_+$.

4.2 Regularization Method

To generate the recommendations in our regularization method, we have to compute the low rank approximation $\hat{\mathbf{R}}$ based on the inferred trust relationships in \mathbf{A}_+ and the distrust relationships in \mathbf{A}_- . We first capture the user-based similarities using the rating matrix \mathbf{R} . If users i and j have interacted with at

least a common item q , then users i and j are connected based on their preferences [22]. The preferences' connections/similarities are stored in a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, whose ij -th entries are calculated as follows:

$$\mathbf{S}_{ij} = \begin{cases} \frac{\sum_{q=1}^m \mathbf{R}_{iq} \mathbf{R}_{jq}}{\sqrt{\sum_{q=1}^m \mathbf{R}_{iq}^2} \sqrt{\sum_{q=1}^m \mathbf{R}_{jq}^2}}, & \text{if users } i \text{ and } j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

with $i, j = 1, \dots, n$.

Next, we form the neighbourhoods \mathcal{N}_+^i and \mathcal{N}_-^i based on the adjacency matrices \mathbf{A}_+ and \mathbf{A}_- , respectively, where \mathbf{A}_+ contains the explicit as well as the inferred implicit trust relationships. Given the latent vector \mathbf{U}_i of user i and the latent vector \mathbf{U}_j of her friend j , with $j \in \mathcal{N}_+^i$, their distance $\|\mathbf{U}_i - \mathbf{U}_j\|_2^2$ should be as close as possible, weighted by the trust degree $(\mathbf{A}_+)_{ij}$ and the preference similarity \mathbf{S}_{ij} . The reason that we consider the similarity between trust users i and j is that trust friends do not necessarily have similar preferences [4]. The higher the similarity between two friends the most likely would be that they have similar preferences. Thus, we weigh the influence of trust friends i and j as follows:

$$\sum_{j \in \mathcal{N}_+^i} (\mathbf{A}_+)_{ij} \mathbf{S}_{ij} \|\mathbf{U}_i - \mathbf{U}_j\|_2^2 \quad (9)$$

Accordingly, for a distrusted user $k \in \mathcal{N}_-^i$, we have to penalize their distance $\|\mathbf{U}_i - \mathbf{U}_k\|_2^2$, weighted by the distrust degree $(\mathbf{A}_-)_{ik}$ and their preference dissimilarity $1 - \mathbf{S}_{ik}$ as follows:

$$- \sum_{k \in \mathcal{N}_-^i} (\mathbf{A}_-)_{ik} (1 - \mathbf{S}_{ik}) \|\mathbf{U}_i - \mathbf{U}_k\|_2^2 \quad (10)$$

By aggregating Eqs. (9) and (10), we have the following regularization term with respect to a latent vector \mathbf{U}_i , to measure the weighted influence of trust and distrust relationships:

$$\Psi(\mathbf{U}_i) = \sum_{j \in \mathcal{N}_+^i} (\mathbf{A}_+)_{ij} \mathbf{S}_{ij} \|\mathbf{U}_i - \mathbf{U}_j\|_2^2 - \sum_{k \in \mathcal{N}_-^i} (\mathbf{A}_-)_{ik} (1 - \mathbf{S}_{ik}) \|\mathbf{U}_i - \mathbf{U}_k\|_2^2 \quad (11)$$

To compute the latent matrices \mathbf{U} and \mathbf{V} we formulate the following objective function \mathcal{L} as a minimization problem:

$$\min_{\mathbf{U}, \mathbf{V}} \mathcal{L}(\mathbf{U}, \mathbf{V}) = \|\mathbf{R} - \mathbf{UV}^\top\|_F^2 + \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) + \alpha \sum_{i=1}^n \Psi(\mathbf{U}_i) \quad (12)$$

where the second and third terms are the regularization term to avoid model overfitting and the social regularizer of Eq. (11), respectively. Parameters λ and α control the influences of the respective regularization terms.

To optimize Eq. (12) we use a gradient-based strategy. As the term $\Psi(\mathbf{U}_i)$ in Eq. (11) may become negative⁵, making the objective function \mathcal{L} non-convex, we define an auxiliary matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$. During the optimization of the objective function \mathcal{L} for the iteration *iter* and $i = 1 \dots n$ we set $\mathbf{H}_i^{iter} = 1$ if $\Psi(\mathbf{U}_i) > 0$, and 0 otherwise. Thus, at the iteration *iter* we have the following objective function:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \mathcal{L}(\mathbf{U}, \mathbf{V}) &= \|\mathbf{R} - \mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \\ &+ \alpha \sum_{i=1}^n \mathbf{H}_i^{iter} \left[\sum_{j \in \mathcal{N}_+^i} (\mathbf{A}_+)_{ij} \mathbf{S}_{ij} \|\mathbf{U}_i - \mathbf{U}_j\|_2^2 - \sum_{k \in \mathcal{N}_-^i} (\mathbf{A}_-)_{ik} (1 - \mathbf{S}_{ik}) \|\mathbf{U}_i - \mathbf{U}_k\|_2^2 \right] \end{aligned} \quad (13)$$

As \mathcal{L} becomes convex with the auxiliary matrix \mathbf{H}^{iter} , we follow a gradient-based optimization strategy to calculate matrices/variables \mathbf{U} and \mathbf{V} . Given a learning parameter η we have the following update rules:

$$\begin{aligned} \mathbf{U}_i^{iter+1} &\leftarrow \mathbf{U}_i^{iter} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{U}_i^{iter}}, \quad i = 1 \dots n \\ \mathbf{V}_q^{iter+1} &\leftarrow \mathbf{V}_q^{iter} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{V}_q^{iter}}, \quad q = 1 \dots m \end{aligned} \quad (14)$$

where the respective gradients at iteration *iter* are calculated as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{U}_i^{iter}} &= 2 \sum_{q=1}^m \mathbb{I}_{iq}^R (\mathbf{U}_i^\top \mathbf{V}_q - \mathbf{R}_{iq}) \mathbf{V}_q + 2\lambda \mathbf{U}_i \\ &+ 2\alpha \mathbf{H}_i^{iter} \sum_{j \in \mathcal{N}_+^i} (\mathbf{A}_+)_{ij} \mathbf{S}_{ij} (\mathbf{U}_i - \mathbf{U}_j) \\ &- 2\alpha \mathbf{H}_i^{iter} \sum_{k \in \mathcal{N}_-^i} (\mathbf{A}_-)_{ik} (1 - \mathbf{S}_{ik}) (\mathbf{U}_i - \mathbf{U}_k) \end{aligned} \quad (15)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}_q^{iter}} = 2 \sum_{i=1}^n \mathbb{I}_{iq}^R (\mathbf{U}_i^\top \mathbf{V}_q - \mathbf{R}_{iq}) \mathbf{U}_i + 2\lambda \mathbf{V}_q \quad (16)$$

where $\mathbb{I}_{iq}^R \in \{0, 1\}^{n \times m}$ is an indicator matrix, with $\mathbb{I}_{iq}^R = 1$ if $\mathbf{R}_{iq} > 0$ and 0 otherwise. Based on the gradients in Eqs. (15) and (16) we can solve the minimization problem of Eq. (13) using the update rules in Eq. (14). Having computed matrices \mathbf{U} and \mathbf{V} , we reconstruct the initial rating matrix \mathbf{R} by computing $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$ to generate the final recommendations.

⁵ If $\sum_{k \in \mathcal{N}_-^i} (\mathbf{A}_-)_{ik} (1 - \mathbf{S}_{ik}) \|\mathbf{U}_i - \mathbf{U}_k\|_2^2 > \sum_{j \in \mathcal{N}_+^i} (\mathbf{A}_+)_{ij} \mathbf{S}_{ij} \|\mathbf{U}_i - \mathbf{U}_j\|_2^2$.

5 Experimental Evaluation

5.1 Experimental Setup

In our experiments, we use a real-world dataset from Epinions⁶ [3]. For comparison reasons, we conduct our experiments on a down-sampled dataset, at the same scale as in [2, 15]. The down-sampled dataset contains $n=119,867$ users, $m=676,436$ product-items and 12,328,927 ratings, including 452,123 trust and 92,417 distrust social relationships. The reason for selecting the dataset is that it is among the most challenging datasets in the relevant literature, as it contains many users and items with high sparsity⁷ in users' preferences, as well as it includes users' trust and distrust relationships.

To evaluate the performance of the proposed method, we randomly select a percentage of ratings (i, q) as a test set \mathcal{T} , while the remaining ratings are used to train our model. Following relevant studies [2, 11], we evaluate the performance of our model in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are formally defined as follows

$$\text{MAE} = \frac{\sum_{(i,q) \in \mathcal{T}} |\mathbf{R}_{iq} - \hat{\mathbf{R}}_{iq}|}{|\mathcal{T}|}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,q) \in \mathcal{T}} (\mathbf{R}_{iq} - \hat{\mathbf{R}}_{iq})^2}{|\mathcal{T}|}}$$

where $\hat{\mathbf{R}}$ is the prediction of our model e.g., the low rank approximation matrix, and \mathbf{R} is the rating matrix with the ratings of the test set \mathcal{T} . The difference between the two evaluation metrics is that RMSE emphasizes more in larger prediction errors than MAE. We repeated our experiments five times and we averaged our results over the five runs.

5.2 Compared Methods

In our experiments, we use the following baseline methods:

- **NMF** [9]: a baseline non-negative matrix factorization method, which does not consider neither trust or distrust relationships.
- **MF-distrust** [11]: a matrix factorization strategy that incorporates the distrust information, trying to maximize the user latent features of users who are connected with a explicit distrust social relationship. This strategy does not use trust relationships.
- **TrustWalker** [5]: a random walk model that exploits explicit trust relationships in a neighborhood-based collaborative filtering strategy. This model ignores users' distrust relationships.

⁶ <http://www.trustlet.org/epinions.html>

⁷ It approximately includes 0.02% of all entries in the rating matrix \mathbf{R} .

- **TrustSVD** [4]: a model that extends SVD++ [8] to learn both the user preferences and the social influence of her friends. This method does not exploit distrust relationships as well.

We also compare the proposed approach with the following competitive strategies that exploit both trust and distrust relationships:

- **MF-TD** [2]: a method that performs matrix factorization such that the latent features of foes who are distrusted by a certain user have a guaranteed minimum dissimilarity gap from the worst dissimilarity of friends who are trusted by this user.
- **JNMF-SG** [15]: a method that co-factorizes user-item and user-cluster associations, by partitioning users into clusters with a spectral clustering approach based on users’ trust and distrust explicit relationships.
- **RecSSN** [19]: a recommendation method in social signed networks, that considers trust and distrust explicit relationships when generating recommendations. RecSSN captures both local and global information from the signed graph and then exploits both types of information in a matrix factorization scheme.
- **RF**: a variant of the proposed method, where we avoid the inference step of Section 4.1 and use only the trust and distrust explicit (direct) relationships. This variant is used to show the importance of the social inference step in our regularization method, when implicit trust relationships are missing.
- **MRW-RF**: the proposed method that infers social relationships in our regularization method to exploit explicit and implicit relationships.

The parameters of the examined methods have been determined via cross-validation and in our experiments we report the best results. The parameter analysis of the proposed method is further studied in Section 5.5.

5.3 Comparison with State-of-the-Art

In Figure 1 we evaluate the examined models in terms of MAE and RMSE. To show the negative effect of sparsity we train the models with different percentages of ratings. In this set of experiments we use all social explicit relationships. We observe that reducing the training set sizes degrades the recommendation accuracy, indicated by larger errors of MAE and RMSE in all models. Compared to baseline NMF, the trust-based models TrustWalker and TrustSVD are less affected by the presence of sparsity in the reduced training sets, as they exploit users’ trust explicit relationships. However, both TrustWalker and TrustSVD ignore users’ distrust relationships, which explains their limited performance. Also, MF-distrust can reduce the sparsity problem using the distrust relationships. The main reason that MF-distrust performs lower than TrustWalker and TrustSVD is that there are less users’ distrust relationships than trust ones in the Epinions dataset (Section 5.1). This complies with several studies reporting that users tend to establish less distrust relationships than trust ones [19, 20].

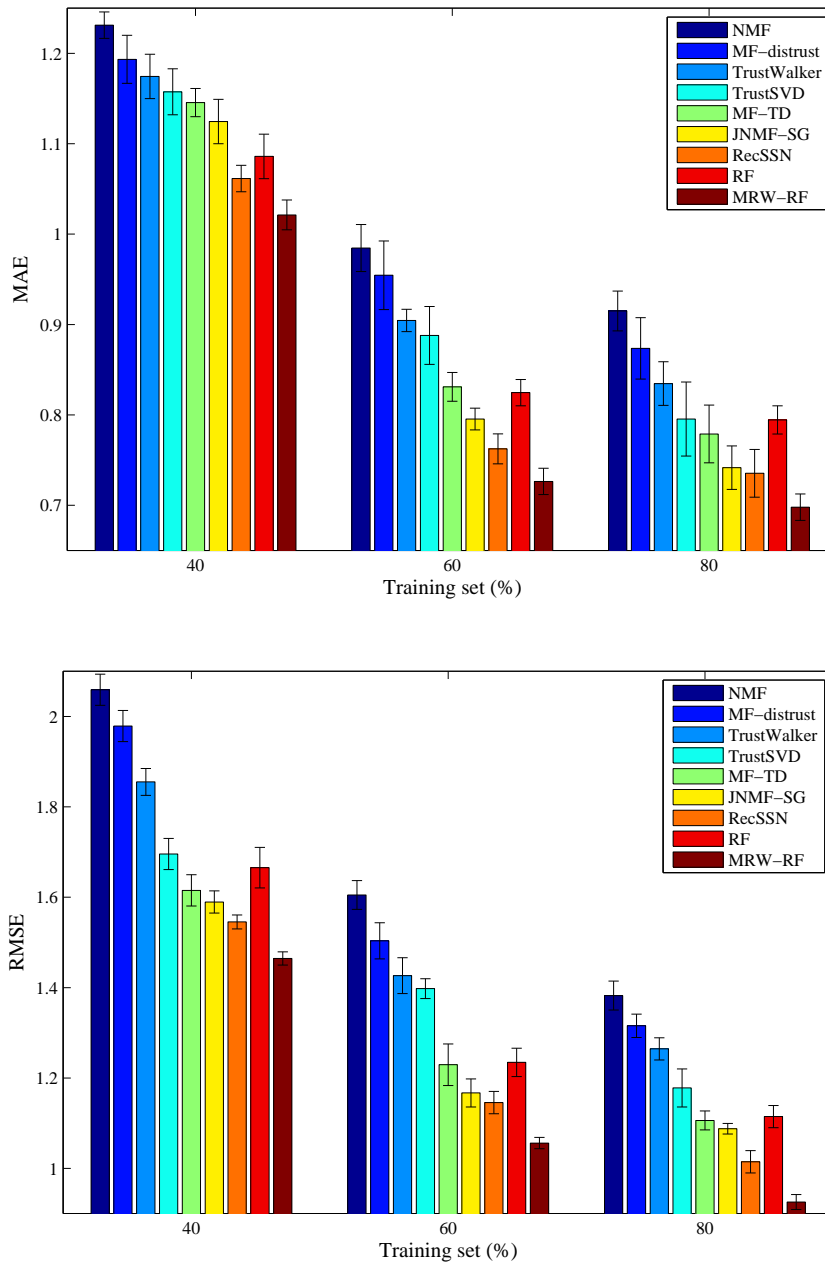


Fig. 1: Effect on MAE and RMSE when varying the percentage of ratings in the training set, using all explicit relationships.

Clearly, all the recommendation strategies MF-TD, JNMF-SG, RecSSN, RF and MRW-RF that exploit both trust and distrust relationships outperform the baseline methods NMF, MF-distrust, TrustWalker and TrustSVD. The results in Figure 1 reveal that it is important to exploit both trust and distrust relationships when generating recommendations. An interesting observation is that our RF variant performs significantly lower than the proposed MRW-RF approach in all settings. This demonstrates the importance of social inference of implicit relationships in our approach, as RF produces recommendations only with explicit trust and distrust relationships in the regularization method of Section 4.2. Instead, the proposed MRW-RF approach efficiently infers the implicit trust relationships and significantly enhances the explicit ones by a factor of 1.72 on average in all five runs. As shown in Figure 1 MRW-RF beats all its competitors, MF-TD, JNMF-SG and RecSSN, because the competitors use explicit trust and distrust relationships and ignore the implicit trust relationships that users might have. Using the paired t-test, we found that MRW-RF is superior over the competitors in all runs, for $p < 0.01$. Moreover, as friends’ trust and foes’ distrust degrees do not necessarily match with users’ preferences, our regularization method also weighs the influence of the implicit and explicit relationships, which explains the high performance of MRW-RF in all settings. To verify this, we re-ran MRW-RF without considering the terms \mathbf{S}_{ij} and $1 - \mathbf{S}_{ik}$ in the social regularization term $\Psi(\mathbf{U}_i)$ of Eq. (11). In our runs, we observed that the prediction error metrics of MAE and RMSE were significantly increased by 11.29% and 15.62%, respectively, indicating the importance of the weighting strategy in the social regularization term to capture the correlations of users’ preferences with trust and distrust degrees.

5.4 Impact of Social Relationships

Figure 2 shows the impact on MAE and RMSE when varying the percentage of explicit social relationships in the training set. All models are trained with 50% of the ratings. In this set of experiments the explicit relationships, both trust and distrust ones, are randomly removed from the training set to vary the percentage of explicit relationships in 40, 60 and 80%. When downsizing the relationships, we ensure that the same percentage of trust and distrust relationships is removed. The baseline method NMF is used for reference, as its performance is not influenced when varying the percentages of relationships. Figure 2 demonstrates that all the social-based models are negatively affected when using less relationships. Since state-of-the-art models use only explicit relationships, they do not perform well, having high prediction errors in terms of MAE and RMSE. When varying the percentages of explicit relationships, the proposed MRW-RF approach preserves the recommendation accuracy relatively high, as MRW-RF correctly infers the implicit relationships with the inference step of Section 4.1. In the case of 40, 60 and 80% relationships, our inference step significantly enhances the reduced relationships, by 3.28, 2.64 and 2.11, respectively. Consequently, MRW-RF is less affected in the presence of sparsity in social relationships. While state-of-the-art methods have limited performance

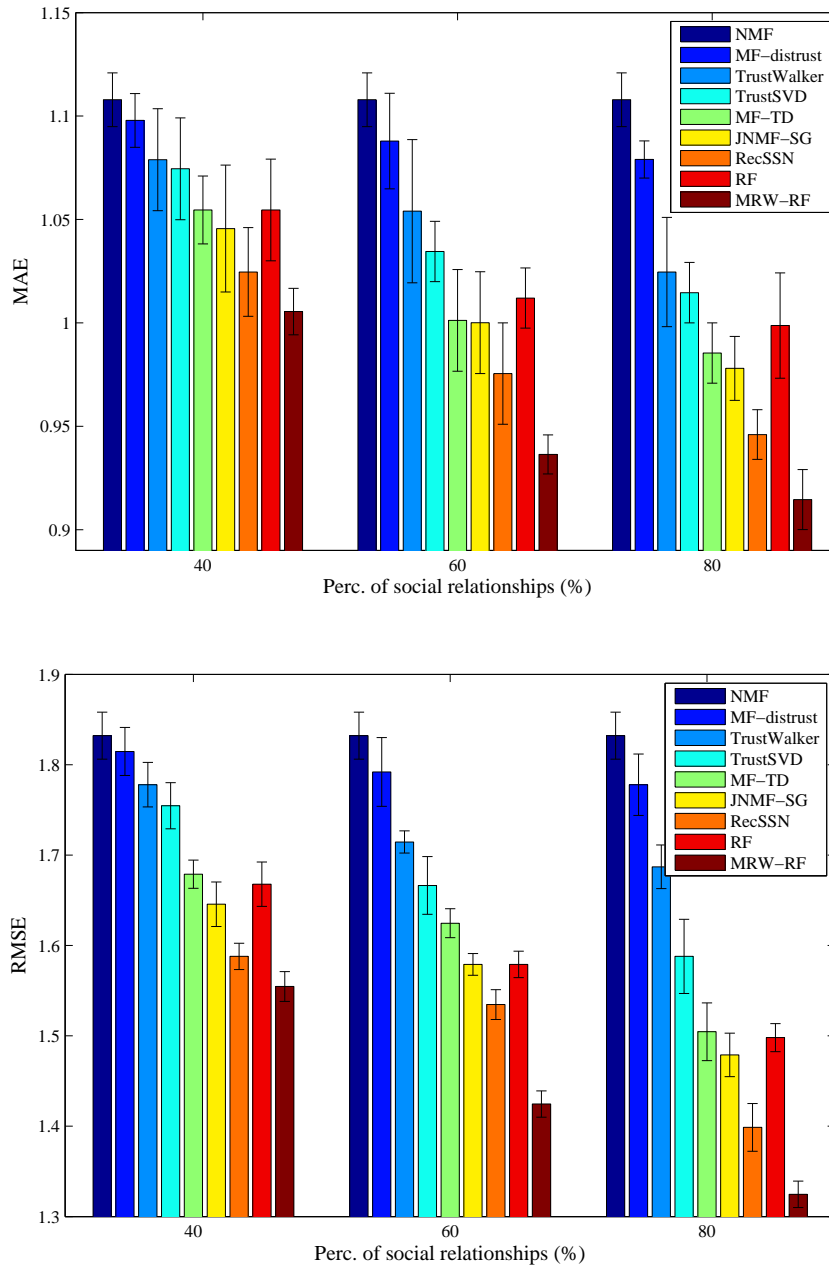


Fig. 2: Effect on MAE and RMSE when varying the percentage of explicit relationships and using 50% of the ratings as training set.

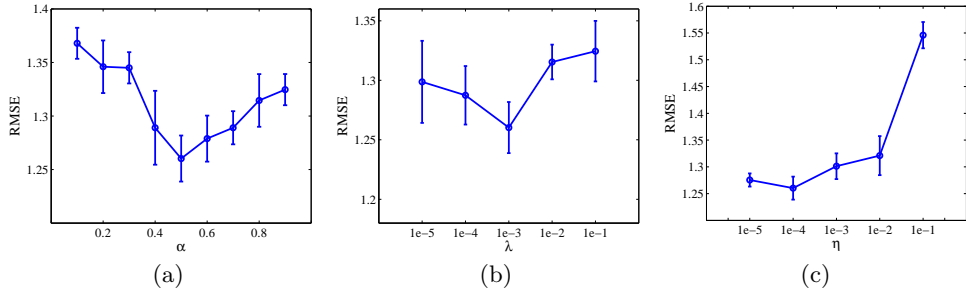


Fig. 3: Effect on RMSE when varying (a) the social regularization parameter α , (b) the regularization parameter λ and (c) the learning rate η .

as there are only a few relationships on which to train their models, MRW-RF boosts the recommendation accuracy by efficiently capturing users' similarity preferences in both the implicit and explicit relationships.

5.5 Parameter Analysis

In Figure 3, we evaluate the influence of parameters α , λ and η . In this set of experiments we train our model using 50% of the ratings as training set with all social explicit relationships. Figure 3(a) presents the effect on RMSE with changes of the α parameter in the objective function of Eq. (13). Parameter α is varied from 0.1 to 0.9 by a step of 0.1. We observe that when α is in the range of 0.4-0.6 the lowest error RMSE is achieved, thus we fix α to 0.5. Out of this range, the RMSE metric is significantly increased, where higher values of α make the selections of the social friends/foes dominate each user's personalized selections. On the other hand, lower values of α make the social selections have less influence on the objective function of Eq. (13), thus not handling well the data sparsity problem in users' preferences. In Figure 3(b) we study the effect of the regularization parameter λ in Eq. (13). We observe that there is a drop on RMSE when $\lambda = 1e - 03$, as higher/lower λ values result in model overfitting/underfitting. In Figure 3(c) we vary the learning rate η of the update rules of Eq. (14). Clearly, a more conservative learning strategy of $\eta = 1e - 4$ is required, as for larger η values RMSE is significantly increased.

6 Conclusions

In this paper we presented MRW-RF, an efficient recommendation strategy to exploit users' trust and distrust relationships and solve the sparsity in both the users' preferences and social relationships. The two key factors of the proposed approach are (i) the correct inference of the missing-implicit trust relationships while considering users' distrust relationships during the inference step, and (ii) the capture of the influence of friends and trust degrees on users' preferences in

our regularization method. In the proposed inference step, MRW-RF efficiently computes the trust degrees of implicit relationships and significantly enhances the explicit ones. In doing so, the proposed approach handles the sparsity in users' relationships, which plays a crucial role on boosting the recommendation accuracy when a part of the social relationships is not available. Then, in our regularization method we weigh the influence of the inferred/implicit and explicit social relationships by taking into account the users' preferences of friends and foes. Hence, our proposed MRW-RF method achieves high recommendation accuracy by exploiting the selection of the inferred and explicit friends, as well as the selection of explicit foes. Our experiments demonstrate the superiority of MRW-RF over several baselines when varying the sparsity in users' preferences and in social relationships. By enhancing users' social relationships in the inference step and efficiently incorporating the trust and distrust degrees into our regularization method, our approach significantly outperforms competitors in all settings. As future work we plan to investigate the performance of social inference of evolving trust and distrust relationships, to capture users' preference dynamics, a challenging task for recommender systems [16–18].

Acknowledgments. Dimitrios Rafailidis was supported by the COMPLEXYS and INFORTECH Research Institutes of University of Mons.

References

1. Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A., Radha, H.: Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In: Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria. pp. 51–58 (2015)
2. Forsati, R., Mahdavi, M., Shamsfard, M., Sarwat, M.: Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans. Inf. Syst.* 32(4), 17:1–17:38 (2014)
3. Guha, R.V., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th ACM International Conference on World Wide Web, New York, NY, USA. pp. 403–412 (2004)
4. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, Texas, USA. pp. 123–129 (2015)
5. Jamali, M., Ester, M.: *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France. pp. 397–406 (2009)
6. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 2010 ACM Conference on Recommender Systems, Barcelona, Spain. pp. 135–142 (2010)
7. Jang, M., Faloutsos, C., Kim, S., Kang, U., Ha, J.: PIN-TRUST: fast trust propagation exploiting positive, implicit, and negative information. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA. pp. 629–638 (2016)

8. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA. pp. 426–434 (2008)
9. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, Denver, CO, USA. pp. 556–562 (2000)
10. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA. pp. 203–210 (2009)
11. Ma, H., Lyu, M.R., King, I.: Learning to recommend with trust and distrust relationships. In: Proceedings of the 2009 ACM Conference on Recommender Systems, New York, NY, USA. pp. 189–196 (2009)
12. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA. pp. 931–940 (2008)
13. Milgram, S.: The small world problem. *Psychology Today* 2 (1967)
14. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. In: Technical report, Stanford Digital Libraries SIDL-WP-1999-0120 (1999)
15. Rafailidis, D.: Modeling trust and distrust information in recommender systems via joint matrix factorization with signed graphs. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy. pp. 1060–1065 (2016)
16. Rafailidis, D., Kefalas, P., Manolopoulos, Y.: Preference dynamics with multimodal user-item interactions in social media recommendation. *Expert Syst. Appl.* 74, 11–18 (2017)
17. Rafailidis, D., Nanopoulos, A.: Modeling the dynamics of user preferences in coupled tensor factorization. In: Proceedings of the 8th ACM Conference on Recommender Systems, Foster City, Silicon Valley, CA, USA. pp. 321–324 (2014)
18. Rafailidis, D., Nanopoulos, A.: Modeling users preference dynamics and side information in recommender systems. *IEEE Trans. Systems, Man, and Cybernetics: Systems* 46(6), 782–792 (2016)
19. Tang, J., Aggarwal, C.C., Liu, H.: Recommendations in signed social networks. In: Proceedings of the 25th ACM International Conference on World Wide Web, Montreal, Canada. pp. 31–40 (2016)
20. Tang, J., Chang, Y., Aggarwal, C., Liu, H.: A survey of signed network mining in social media. *ACM Comput. Surv.* 49(3), 42:1–42:37 (2016)
21. Tang, J., Hu, X., Chang, Y., Liu, H.: Predictability of distrust with interaction data. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China. pp. 181–190 (2014)
22. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. *Social Netw. Analys. Mining* 3(4), 1113–1133 (2013)
23. Victor, P., Cornelis, C., Cock, M.D., Teredesai, A.: Trust- and distrust-based recommendations for controversial reviews. *IEEE Intelligent Systems* 26(1), 48–55 (2011)