Trust Prediction via Aggregating Heterogeneous Social Networks

Jin Huang
University of Texas at Arlington
Arlington, TX 76013
jin.huang@mavs.uta.edu

Feiping Nie
University of Texas at Arlington
Arlington, TX 76013
feipingnie@gmail.com

Heng Huang
University of Texas at Arlington
Arlington, TX 76013
heng@uta.edu

Yi-Cheng Tu
University of South Florida
Tampa, FL 33620
ytu@cse.usf.edu

ABSTRACT

Along with the increasing popularity of social web sites, users rely more on the trustworthiness information for many online activities among users. However, such social network data often suffers from severe data sparsity and are not able to provide users with enough information. Therefore, trust prediction has emerged as an important topic in social network research. Traditional approaches explore the topology of trust graph. Previous research in sociology and our life experience suggest that people who are in the same social circle often exhibit similar behavior and tastes. Such ancillary information, is often accessible and therefore could potentially help the trust prediction. In this paper, we address the link prediction problem by aggregating heterogeneous social networks and propose a novel joint manifold factorization (JMF) method. Our new joint learning model explores the user group level similarity between correlated graphs and simultaneously learns the individual graph structure, therefore the shared structures and patterns from multiple social networks can be utilized to enhance the prediction tasks. As a result, we not only improve the trust prediction in the target graph, but also facilitate other information retrieval tasks in the auxiliary graphs. To optimize the objective function, we break down the proposed objective function into several manageable subproblems, then further establish the theoretical convergence with the aid of auxiliary function. Extensive experiments were conducted on real world data sets and all empirical results demonstrated the effectiveness of our method.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms,Experimentation

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

Keywords

Trust Prediction, Social Network, Transfer Learning, Nonnegative Matrix Factorization

1. INTRODUCTION

The ever-increasing popularity of social web sites such as Facebook and LinkedIn has generated complicated social networks and corresponding data sets with enormous sizes. Among the various forms of online activities, adding (accepting) other users as friends is a primary one. With the increasing risk of exposing private profile to malicious users, the question of whom to trust has become an important challenge to individual users. Many online social communities allow users to tag other users to facilitate the trustworthiness evaluation. Trust tags in a social network can be represented as a trust graph $G = \langle V, E \rangle$, where V represents the collection of nodes (users) and an edge between node i and j denotes a trust vote from user i to user j. Due to the lack of diligence and privacy concern on users' part, there are often a large number of missing values in the trust graph, making the trust link prediction a difficult task. In the literature, there are a few trust prediction papers using trust propagation [8, 11]. The assumption for these methods is that users tend to trust each other given a trustable common friend. However, since only a very small portion of entries in trust graph are explicitly tagged, the prospect of these approaches seems gloomy.

It has been discovered in [15], people who are in the same social circle often share similar behavior and tastes. In [5], Crandall et al. give the following two main reasons. One is that people generally adopt behavior exhibited by those they interact with. Such process is called social influence. The other more distinct reason is people incline to form relationships with others who are already similar to them. Prior research works on inferring individual user's interests and attributes from his or her social neighbors [1, 13, 16]. These papers show the possibility of improving the users' attributes prediction from the trust graph. In this paper, we will explore the trust graph structure with the users' behavior profile instead, use the ancillary information to help the trust prediction. We propose a joint manifold factorization (JMF) model to predict the trust and distrust in social network by aggregating heterogeneous social networks from both target trust domain and auxiliary information domain. When we say two graphs are heterogenous, it implies they are from different domains and have no apparent structural similarity and their entries generally have different scale. Our approach is

to alleviate the sparsity problem in trust graph by taking advantage of the supplementary knowledge about user behavior and discovering the implicit group-level similarity, which are jointly determined by the user-user trust graph and user behavior auxiliary graph. This helps us find the optimal like-minded user groups across both domains. Moreover, we construct the individual affinity graphs to explore the individual geometric structures of the feature manifold to improve the prediction of the missing elements.

The remainder of this paper is organized as follows. In Section 2, we describe the notations used in this paper and formulate the objective function. We will derive our optimization method and provide the algorithm in Section 3. In Section 4, we will give the outline of the convergence proof in our new algorithm. We empirically validate the effectiveness of our method for trust prediction in Section 5 and conclude the paper in Section 6.

2. JOINT MANIFOLD FACTORIZATION

In this section, we will introduce our JMF objective function that aggregates the heterogeneous social networks. Prior to this, we first reiterate our motivation and then give an example to demonstrate this.

As mentioned in the introduction, trusted users in a social network often display similar behavior and taste. Meanwhile, social network users become friends due to the similar background and interest. Therefore, the trust graph and behavior graph should share some structure similarity in spite of the apparent difference, if the coincidence of the similar ratings contributes to the trust votes. As a result, the trust prediction accuracy can be improved with the aid of behavior graph information and vice versa. In summary, we transfer the knowledge from different domains to circumvent the sparsity constraint and help predict the entries in both matrices, this answers the most important questions of transfer learning: what to transfer and how to transfer. Fig. 1 is a demonstration of our motivation.

2.1 Notations

We use boldface uppercase letters, such as $\mathbf{X}$ to denote matrices, $\mathbf{X}_{i.}$, $\mathbf{X}_{.j}$, $\mathbf{X}(i,j)$ (or $\mathbf{X}_{ij}$ when needed) to denote the ith row, jth column and the entry located at (i,j) of $\mathbf{X}$, respectively. In our setting, for simplicity, we only discuss two matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ case, then it is natural to extend the objective function to multiple matrices case. For the ease of discussion, we further assume $\mathbf{G}_1 \in \mathbb{R}^{n\times m_1}$, $\mathbf{G}_2 \in \mathbb{R}^{n\times m_2}$ are the trust graph and rating graph (a special category of behavior graph) respectively, where n is the number of identical users in both domains, m_1 is the number of users who receive trust votes, m_2 is the number of different items. $\Omega_1 \subset \mathbf{G}_1$ and $\Omega_2 \subset \mathbf{G}_2$ are entries known in corresponding graphs.

2.2 Objective Function Formulation

Inspired by the preceding discussions, we target at the joint matrix factorization to find out the shared group structure between two graphs.

$$\min_{\mathbf{U},\mathbf{V}_1,\mathbf{V}_2,c} \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \tag{1}$$

Here $\mathbf{U} \in \mathbb{R}^{n\times l}$, $\mathbf{V}_1 \in \mathbb{R}^{m_1\times l}$, $\mathbf{V}_2 \in \mathbb{R}^{m_2\times l}$ where l is the number of group parameter determined by user. $c > 0$ is a scalar adjusting the scale inconsistency between graphs since the two graphs are from different domains. Here $\mathbf{U}$ is jointly determined by the trust graph and rating graph, therefore it provides the shared group structure for both graphs. Since rows represent users in both graphs, we could group users based on $\mathbf{U}$ and then conduct the trust and rating prediction with $\mathbf{V}_1$,$\mathbf{V}_2$ respectively. It can be observed that $\mathbf{U}$ carries the knowledge of both trust graph and rating graph, such framework becomes especially useful since both graphs usually have data sparsity issues for real data sets.

While the above model takes into account of the common row group structure in terms of both matrices, it fails to consider the individual column information for both matrices. To overcome this drawback, we include the Laplacian regularity term [9] in the above formula. To be specific,

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}_1,\mathbf{V}_2,c} \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \\ &+\lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) + \lambda Tr(\mathbf{V}_2^T\mathbf{L}_2\mathbf{V}_2) \\ &s.t. \quad \mathbf{V}_1\mathbf{V}_1^T = \mathbf{I}, \mathbf{V}_2\mathbf{V}_2^T = \mathbf{I}, \quad \mathbf{U} \geq 0, \mathbf{V}_1 \geq 0, \mathbf{V}_2 \geq 0\end{aligned} \tag{2}$$

Here $\lambda > 0$ is a scalar parameter to be tuned, $\mathbf{L}_1$ and $\mathbf{L}_2$ are the Laplacian graph based on the columns of $\mathbf{G}_1$ and $\mathbf{G}_2$ respectively, Tr is the trace operation which yields the sum of diagonal elements of the matrix. The detailed construction for $\mathbf{L}_1$ and $\mathbf{L}_2$ would be given in the next section. We impose the orthogonal constraints on $\mathbf{V}_1$ and $\mathbf{V}_2$ to ensure the uniqueness of the solution. Suppose $\mathbf{U}^*$, $\mathbf{V}_1^*$ and $\mathbf{V}_2^*$ are the solutions to Eq. (2), then for any given nonzero constant $c_1 > 1$, $c_1\mathbf{U}^*$ and $\mathbf{V}_1^*/c_1$ would give same value in the first term and lower value for the third term, this is true no matter $\mathbf{U}^*$ and $\mathbf{V}_1^*$ are local or global optimum solutions, the same argument applies to $\mathbf{V}_2$. In other words, the optimal solution to Eq. (2) is not unique without the constraint.

3. OPTIMIZATION

In the following, we will give solution to Eq. (2). As we see, minimizing Eq. (2) is with respect to $\mathbf{U}$, $\mathbf{V}_1$, $\mathbf{V}_2$ and c, and we can not give a closed-form solution. We will present an alternating scheme to optimize the objective, this procedure repeats until convergence.

3.1 Initialization

In this paper, for any missing entry $\mathbf{G}(i,j)$, we use mean of the available entries in the corresponding row and column, for a user-item rating matrix, such initialization combines the available information for both the individual user rating habit and other users' ratings on a particular item. For a user-user trust matrix, such initialization consider both user i and user j's social circle influence.

After the missing values initial imputation, we construct the Laplacian graphs for both social networks. We define the edge weight matrix $\mathbf{W}$ as follows:

$$w_{ij} = \begin{cases} 1 : \mathbf{X}_{i.} \in N_k(\mathbf{X}_{.j}) \ or \ \mathbf{X}_{j.} \in N_k(\mathbf{X}_{.i}) \\ 0 : otherwise \end{cases}$$

where $N_k(\mathbf{X}_{i.})$ denotes the set of k nearest neighbors of $\mathbf{X}_{i.}$. It is easy to see $\mathbf{W}$ is symmetric. Let graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix whose entries are column sums of $\mathbf{W}$, $\mathbf{D}_{ii} = \sum_j w_{ij}$.

After that, we construct $\mathbf{V}_1$ and $\mathbf{V}_2$ based on k-means on columns for $\mathbf{G}_1$ and $\mathbf{G}_2$ respectively. For i-th row of $\mathbf{V}_1$, if this row belongs to j-th cluster, then $\mathbf{V}_1(i,j) = 1$, all other elements in i-th row are 0. $\mathbf{V}_2$ is initialized in the same manner.

Now we come to the optimization of our objective function, we iteratively solve $\mathbf{U}$, $\mathbf{V}_1$, $\mathbf{V}_2$ and c in an alternating manner. In other words, we will optimize the objective with respect to one variable while fixing the other variables. Such process repeats until convergence.

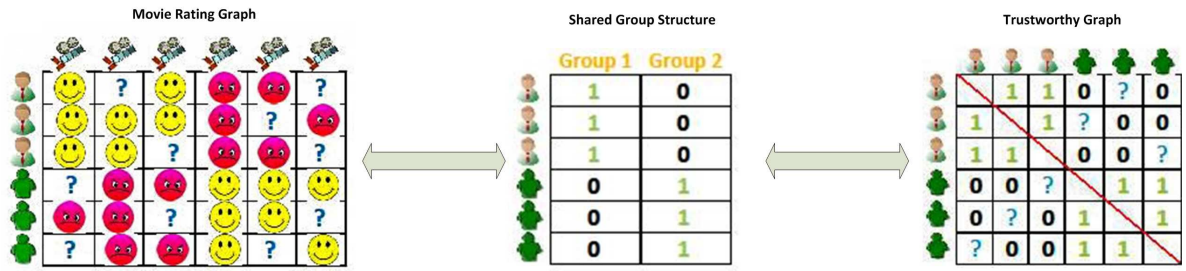

Figure 1: A demonstration for our motivation and learning process. The shared group structure matrix is jointly determined by the rating graph and trust graph. The rating matrix contains 2 groups of users' reviews about movies, where a smile face represents a satisfactory review and an angry face represents an unsatisfactory message. The trust matrix contains users' trust evaluation towards other users, where 1 represents trust and 0 represent distrust. The question mark represents missing value in both graphs. The 1s in cluster information matrix indicate users are in the corresponding group while 0s represent users are not in that group.

3.2 Computation of $\mathbf{U}$

Optimizing Eq. (2) with respect to $\mathbf{U}$ is equivalent to optimizing

$$\begin{aligned} J_1 &= \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \\ s.t. \quad & \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}, \quad \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}, \mathbf{V}_1 \geq 0, \mathbf{V}_2 \geq 0 \end{aligned} \tag{3}$$

Setting $\frac{\partial J_1}{\partial \mathbf{U}} = 0$ leads to the following updating formula

$$\mathbf{U} = \frac{\mathbf{G}_1\mathbf{V}_1 + c\mathbf{G}_2\mathbf{V}_2}{2} \tag{4}$$

3.3 Computation of $\mathbf{V}_1$

Optimizing Eq. (2) with respect to $\mathbf{V}_1$ is equivalent to optimizing

$$\begin{aligned} J_2 &= \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) \\ s.t. \quad & \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}, \mathbf{V}_1 \geq 0 \end{aligned} \tag{5}$$

For the constraint $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}$, we can not get a closed-form solution of $\mathbf{V}_1$. Therefore we will present an iterative multiplicative updating algorithm. We introduce the Lagrangian multiplier $\boldsymbol{\alpha} \in \mathbb{R}^{l\times l}$, the corresponding Lagrangian function is

$$L(\mathbf{V}_1) = \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) - Tr(\boldsymbol{\alpha}(\mathbf{V}_1^T\mathbf{V}_1 - \mathbf{I})) \tag{6}$$

Setting $\frac{\partial L(\mathbf{V}_1)}{\partial \mathbf{V}_1} = 0$ and use the orthogonal constrain $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}$, we obtain

$$\begin{aligned} & -\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{L}_1\mathbf{V}_1 - \mathbf{V}_1\boldsymbol{\alpha} = 0 \\ & \Rightarrow \boldsymbol{\alpha} = -\mathbf{V}_1^T\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1 \end{aligned} \tag{7}$$

Using the Karush-Kuhn-Tucker condition [3] $\boldsymbol{\alpha} \cdot \mathbf{V}_1 = 0$, where $\cdot$ is the element-wise product operator and thereafter, we get

$$(-\mathbf{V}_1^T\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) \cdot \mathbf{V}_1 = 0 \tag{8}$$

Introduce $\mathbf{L}_1 = \mathbf{L}_1^+ - \mathbf{L}_1^-$, $\mathbf{V}_1 = \mathbf{V}_1^+ - \mathbf{V}_1^-$ and $\mathbf{U} = \mathbf{U}^+ - \mathbf{U}^-$ where $U_{ij}^+ = (|U_{ij}| + U_{ij})/2$ and $U_{ij}^- = (|U_{ij}| - U_{ij})/2$ [7] and $\mathbf{L}_1$,$\mathbf{V}_1$ defined in a similar fashion, we obtain

$$(\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1 + \mathbf{V}_1\boldsymbol{\alpha}^- - \mathbf{G}_1^T\mathbf{U}^+ - \lambda\mathbf{L}_1^-\mathbf{V}_1 - \mathbf{V}_1\boldsymbol{\alpha}^+) \cdot \mathbf{V}_1 = 0 \tag{9}$$

Eq. (9) leads to the following updating formula

$$\mathbf{V}_1(i,j) = \mathbf{V}_1(i,j)\sqrt{\frac{\left[\mathbf{G}_1^T\mathbf{U}^+ + \lambda\mathbf{L}_1^-\mathbf{V}_1 + \mathbf{V}_1\boldsymbol{\alpha}^+\right]_{ij}}{\left[\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1 + \mathbf{V}_2\boldsymbol{\alpha}^-\right]_{ij}}} \tag{10}$$

3.4 Computation of $\mathbf{V}_2$

Optimizing Eq. (2) with respect to $\mathbf{V}_2$ is equivalent to optimizing

$$\begin{aligned} J_3 &= \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 + \lambda Tr(\mathbf{V}_2^T\mathbf{L}_2\mathbf{V}_2) \\ s.t. \quad & \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}, \mathbf{V}_2 \geq 0 \end{aligned} \tag{11}$$

The optimization with the above equation is almost identical to the previous subsection, so we only give formula without details.

$$\mathbf{V}_2(i,j) = \mathbf{V}_2(i,j)\sqrt{\frac{\left[c\mathbf{G}_2^T\mathbf{U}^+ + \lambda\mathbf{L}_2^-\mathbf{V}_2 + \mathbf{V}_2\boldsymbol{\beta}^+\right]_{ij}}{\left[c\mathbf{G}_2^T\mathbf{U}^- + \lambda\mathbf{L}_2^+\mathbf{V}_2 + \mathbf{V}_2\boldsymbol{\beta}^-\right]_{ij}}} \tag{12}$$

3.5 Computation of c

Optimizing Eq. (2) with respect to c is equivalent to optimizing

$$J_4 = \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \tag{13}$$

The above task is equivalent to

$$\min_c Tr(c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T)(c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T)^T$$

This can be written as

$$\min_c Ac^2 - 2Bc + D,$$

where $A = Tr(\mathbf{G}_2\mathbf{G}_2^T)$, $B = Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{G}_2^T)$, $D = Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{V}_2\mathbf{U}^T)$. It is a quadratic function in c, the solution is then

$$c = \frac{Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{G}_2^T)}{Tr(\mathbf{G}_2\mathbf{G}_2^T)}. \tag{14}$$

4. CONVERGENCE ANALYSIS

In this section, we prove the convergence of our algorithm. Since the objective function is naturally lower bounded, we just need to prove the objective function is monotonically decreasing at each iteration. The decreases in objective function for updating $\mathbf{U}$ and c are obvious, so we focus on the proof for the $\mathbf{V}_1$ and $\mathbf{V}_2$. We use classic auxiliary function approach used in [12].

DEFINITION 1. *[12] $Z(h,h')$ is an auxiliary function for $F(h)$ if the conditions*

$$Z(h,h') \geq F(h), Z(h,h) = F(h)$$

are satisfied.

LEMMA 1. *[12] If Z is an auxiliary function for F, then F is non-increasing under the update*

$$h^{(t+1)} = \arg\min_h Z(h, h^{(t)})$$

LEMMA 2. *[7] For any nonnegative matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{k\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times k}$, $\mathbf{S}' \in \mathbb{R}^{n\times k}$, and $\mathbf{A}$, $\mathbf{B}$ are symmetric, then the following inequality holds*

$$\sum_{i=1}^{n}\sum_{p=1}^{k} \frac{(\mathbf{AS}'\mathbf{B})_{ip}\mathbf{S}_{ip}^2}{\mathbf{S}'_{ip}} \geq Tr(\mathbf{S}^T\mathbf{ASB})$$

THEOREM 3. *Let*

$$J(\mathbf{V}_1) = Tr(\lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1 - 2\mathbf{G}_1^T\mathbf{U}\mathbf{V}_1^T + \boldsymbol{\alpha}\mathbf{V}_1^T\mathbf{V}_1) \quad (15)$$

Then the following function

$$\begin{aligned}
&Z(\mathbf{V}_1, \mathbf{V}'_1)\\
&= \lambda\sum_{ij}\frac{(\mathbf{L}_1^+\mathbf{V}'_1)_{ij}\mathbf{V}_{1,ij}^2}{\mathbf{V}'_{1,ij}} - \lambda\sum_{ijk}(\mathbf{L}_1^-)_{jk}\mathbf{V}'_{1,ji}\mathbf{V}'_{1,ki}(1+\log\frac{\mathbf{V}_{1,ji}\mathbf{V}_{1,ki}}{\mathbf{V}'_{1,ji}\mathbf{V}'_{1,ki}})\\
&-2\sum_{ij}\mathbf{G}_1^T\mathbf{U}^+\mathbf{V}'_{1,ij}(1+\log\frac{\mathbf{V}_{1,ij}}{\mathbf{V}'_{1,ij}}) + 2\sum_{ij}\mathbf{G}_1^T\mathbf{U}^-\frac{\mathbf{V}_{1,ij}^2+\mathbf{V}'^2_{1,ij}}{2\mathbf{V}'_{1,ij}}\\
&+\sum_{ij}\boldsymbol{\alpha}^+\mathbf{V}_{1,ij}^2 - \sum_{ijk}\boldsymbol{\alpha}^-\mathbf{V}'_{1,ij}\mathbf{V}'_{1,ik}(1+\log\frac{\mathbf{V}_{1,ij}\mathbf{V}_{1,ik}}{\mathbf{V}'_{1,ij}\mathbf{V}'_{1,ik}})
\end{aligned}$$

is an auxiliary function for $J(\mathbf{V}_1)$. Furthermore, it is a convex function in $\mathbf{V}_1$ and its global minimum is

$$\mathbf{V}_1(i,j) = \mathbf{V}_1(i,j)\sqrt{\frac{\left[\mathbf{G}_1^T\mathbf{U}^+ + \lambda\mathbf{L}_1^-\mathbf{V}_1 + \mathbf{V}_1\boldsymbol{\alpha}^+\right]_{ij}}{\left[\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1 + \mathbf{V}_2\boldsymbol{\alpha}^-\right]_{ij}}} \quad (16)$$

Proof *We have to omit the proof due to page limit. Interested reader may contact the first author for details.* □

THEOREM 4. *Updating $\mathbf{V}_1$ using Eq. (10) will monotonically decrease the value of the objective in Eq. (2).*

Proof *By Lemma 1 and Theorem 3, we can get that $J(\mathbf{V}_1^0) = Z(\mathbf{V}_1^0, \mathbf{V}_1^0) \geq Z(\mathbf{V}_1^1, \mathbf{V}_1^0) \geq J(\mathbf{V}_1^1) \geq \ldots$so $J(\mathbf{V}_1)$ is monotonically decreasing.* □

The monotonical decrease of the objective in Eq. (2) via updating $\mathbf{V}_2$ using Eq. (12) can be proved in a similar way. Therefore it is obvious our algorithm will converge.

5. EXPERIMENTS

In this paper, we want to compare the prediction performance with other methods on both trust graph and rating graph from a real data set.

5.1 Description of Data Set

This data set was collected by Paolo Massa [14] in a 5-week crawl from Epinions.com. It consists of two parts, one is the ratings part, the other is the trust votes part. The Epinions data set consists of 49,290 users, 139,738 items, 664,824 reviews from users to items, 487,181 trust statement between users. Users express their web of trust, i.e, reviewers whose reviews and ratings they have consistently found to be valuable and offensive [14]. Therefore it is reasonable to assume most individual users tend to cast trust votes towards other users if the users have similar rating patterns towards those items. As a result, the rating matrix and trust matrix could have similar row structure given common users, in other words, the assumption of our method holds on this data set.

5.2 Competitive Methods and Procedure

The competitive methods include average filling (AF), k-nearest neighbors (KNN), SimRank [10], SVD [2] and matrix completion via trace norm (MC) [4]. AF uses the average of their available row and column entries to impute the missing entries. KNN imputes the missing values based on nodes similarity using Jaccard's coefficient while SimRank imputes based on path ensembles. SVD finds a low-rank matrix that approximates the target matrix and MC seeks a low-rank matrix with trace norm regularity, where trace norm of a matrix is the convex relaxation of its rank. Note that we stack the trust graph and rating graph using the common users as rows for MC method, that serves the benchmark method for transfer learning.

For KNN, we search k in the list $\{1, 2, \ldots, 9\}$, to impute the missing value using the node with the highest Jaccard similarity score. For SimRank method, we set the parameters using the default value suggested by the author. For SVD, we choose the rank from the list ($\frac{R}{10}, \frac{2R}{10}$..., R), where $R = \min(n, m)$, the minimum of the number of rows and columns. For MC, regularity coefficient ς is tuned from the list $\{10^{-2}, 10^{-1}, 1, 10\}$.

We design the experiments as follows: we select top 2,000 users with the highest degrees (cast and receive most votes), then we select items with more than 68 ratings from the above selected users. The resulting trust graph $\mathbf{G}_1$ of size $2,000 \times 2,000$ has 149,146 trust votes (represented by 1), which consists of 3.73 % of all possible votes, those distrust or unknown votes are represented by 0. The rating graph $\mathbf{G}_2$ of size $2,000 \times 96$ has 10,225 ratings (from 1 to 5), which consists of 5.33 % of all possible ratings, those missing ratings are represented by 0. Among those available ratings, the number of ratings 1,2 and 3 are roughly equal, 4 is twice as many as 1 and 5 is about 4 times as many as 1, such skew distribution might be due to users' reluctance to give low ratings for unsatisfactory items.

5.3 Evaluation Metric

The evaluation of rating graph is relatively easy, it is conventional to use Mean Average Error (MAE) and Root Mean Square Error (RMSE).

$$\begin{aligned}
\text{MAE} &= \sum_{r_{ij}\in T_E} |r_{ij} - \hat{r}_{ij}| / |T_E|\\
RMSE &= \sqrt{\sum_{r_{ij}\in T_E} (r_{ij} - \hat{r}_{ij})^2 / |T_E|}
\end{aligned} \quad (17)$$

where r_{ij} and $\hat{r}_{ij}$ are the true and predicted ratings respectively, $|T_E|$ is the number of test ratings.

The evaluation of trust graph is more complicated. Since the binary trust votes have a very skewed distribution, precision and recall are more suitable than receiver operating characteristic (ROC) [6]. However, since most methods described in this paper do not restrict the output to the discrete binary domain, in order to use the metric measure precision and recall, we have to convert the continuous predictions to the binary values using threshold values. For real data set, it is difficult (if not impossible) to choose an appropriate threshold, therefore it is inappropriate to compare all methods solely based on the optimal recall and precision values. As a result, we evaluate the performance of all methods based on Area Under Curve (AUC) for recall-precision curves and the optimal F1 values from all methods, where F1 is defined in Eq. (18).

$$recall = \frac{TP}{TP+FN}, \quad precision = \frac{TP}{TP+FP},$$
$$F1 = \frac{2\times recall \times precision}{recall + precision}, \quad (18)$$

Table 1: Evaluation on Trust and Rating Graphs

(a) Recall-Precision Evaluations

Link	Methods	AUC	F1
Trust	AF	0.207	**0.223**
	KNN	0.183	0.218
	SimRank	0.185	0.218
	SVD	0.123	0.160
	MC	0.075	0.122
	JMF	**0.215**	0.221
Distrust	AF	0.914	0.977
	KNN	0.916	0.977
	SimRank	0.916	0.977
	SVD	0.583	0.971
	MC	0.972	0.981
	JMF	**0.992**	**0.991**

(b) Rating Graph Evaluation Results

Prediction Measure	Methods	Result
MAE	AF	0.864
	KNN	0.839
	SimRank	0.832
	SVD	0.924
	MC	0.828
	JMF	**0.772**
RMSE	AF	1.062
	KNN	1.045
	SimRank	1.034
	SVD	1.263
	MC	1.024
	JMF	**0.963**

where TP, FN and FP are numbers of true positives, false negatives and false positives, respectively.

We randomly hide half of the available entries, conduct the predictions via all methods and evaluate via 2-fold cross validation. Such procedure is repeated 10 times and we report the average results in Table 1. JMF has better performance than other methods except F1 score for trust links, where AF shows some slight advantage. Note that it is impractical and time consuming to tune threshold value for real application, therefore our method still shows better performance in trust link prediction than AF method considering the significant AUC advantage. We can conclude our method has the best performance in trust prediction in all the methods we listed in terms of trust links and distrust links. Table 1 also lists all methods' optimal value in terms of MAE and RMSE for the rating graph, again JMF has the best MAE and RMSE results. We can therefore conclude that transfer learning does provide the bridge for the trust graph and rating graph to share the valuable information with each other. This helps alleviate the common data sparsity issue in social network data. On the other hand, as we have shown, naive transfer learning MC method does not work very well here, MC method fails to extract the common row structure with matrices stacked.

6. CONCLUSION

In this paper, we developed the joint manifold factorization (JMF) method to perform trust prediction with the ancillary rating matrix. We transfer the common group structure knowledge between two related matrices and simultaneously explore the individual matrix geometric structure. With publicly available data sets, our method shows its advantage over classical trust prediction methods for both the trust matrix and rating matrix.

7. ACKNOWLEDGEMENT

The authors would like to thank the authors of Epinions for graciously making data available for this study. We also appreciate the code from other authors.

8. REFERENCES

[1] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2677–2682, 2007.

[2] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–54, 1998.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[4] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[5] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of 14th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 160–168, 2008.

[6] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 233–240, 2006.

[7] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorization. *IEEE Transactions on PAMI*, 32(1):45–55, 2010.

[8] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of Thirteenth International World Wide Web Conference*, pages 403–412, 2004.

[9] X. He and P. Niyogi. Locality preserving projections. In *Proceedings of Seventeenth Annual Conference on Neural Information Processing Systems*, pages 153–160, 2003.

[10] G. Jeh and J. Widom. A measure of structural-context similarity. In *Proceedings of 8th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 538–543, 2002.

[11] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of Twelfth International World Wide Web Conference*, pages 640–651, 2003.

[12] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of Fourteenth Annual Conference on Neural Information Processing Systems*, pages 556–562, 2000.

[13] P. Massa and P. Avesani. Trust-aware recommender systems. In *ACM RecSys*, pages 17–24, 2007.

[14] P. Massa and P. Avesani. *Trust metrics in recommender systems. Computing with Social Trust*. Springler, 2009.

[15] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27, 2001.

[16] Z. Wen and C. Lin. On the quality of inferring interests from social neighbors. In *Proceedings of 16th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 373–382, 2010.