"Like Attracts Like!"– A Social Recommendation Framework Through Label Propagation

Dingyan Wang[†], Irwin King[†‡] and Kwong Sak Leung[†]
[†] The Chinese University of Hong Kong
Shatin, N.T. Hong Kong
{dywang, king, ksleung}@cse.cuhk.edu.hk

[‡] AT&T Labs - Research
San Francisco, CA, USA
irwin@research.att.com

ABSTRACT

Recently label propagation recommendation receives much attention from both industrial and academic fields due to its low requirement of labeled training data and effective prediction. Previous methods propagate preferences on a user or item similarity graph for making recommendation. However, they still suffer some major problems, including data sparsity, lack of trustworthiness, cold-start problem. By observation, the currently booming social network has some characteristics to remedy these problems. (1) Most of the user connections in either social network or real life can inflect information about users' interest similarity by "Like Attracts Like", which can improve propagation graph construction. (2) Social connections can inflect trustworthiness information for user similarity, where connections are not built randomly but based on their trust. (3)Social network can provide user connection data as the supplementation of sparse ratings, which can also solve the cold-start problem when one new user has no rating history but social network. In order to improve the recommendation accuracy, we propose a social label propagation recommendation framework. In addition, we also construct the traditional user similarity graph for combination with social network to solve the noise and multi-interest problem in social network. Finally, we implement Green's function semi-supervised learning algorithm for label propagation recommendation on the real world recommendation data. The empirical results demonstrate the effectiveness of our proposed social recommendation framework.

Categories and Subject Descriptors

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

General Terms

Algorithm, Experimentation

Keywords

Recommender Systems, Social Network, Label Propagation, Collaborative Filtering

1. INTRODUCTION

In modern days, people are usually overwhelmed with various of choices on the web which provides a huge number of information. Recommendation, as the technology to suggest personalized items to meet special needs and tastes of different persons [1, 9], has been widely applied into many e-commerce web sites, like Amazon, IMDb, Epinions, etc. The feedback shows that correct recommendations not only improve consumer satisfaction but also increase the profit of e-commercial systems. On the other hand, recommendation has been studied widely in academic. Typically, recommendation in collaborative filtering (CF) is a hot topic, which can automatically predict the preferences of users to items only based on the history rating information. It is based on a simple idea that users with similar interests will have the similar preference to items. More formally, recommendation can be regarded as a prediction task (illustrated by a toy example in Table 1 and 2): given a partially observed user-item rating matrix $R_0 \in \mathbb{R}^{M \times N}$, whose rows represent M users, columns represent N items, non-zero elements represent observed ratings and zero elements represent those unknown ratings, the goal is to predict unknown ratings to complete the matrix, with each element r_{jk} $(1 \leq j \leq M, 1 \leq k \leq N)$ in the range of rating $1, ..., R$ $(R > 1, R \in \mathbb{Z})$.

Among many recommendation methods, recently label propagation recommendation becomes a popular method [3]. Originally, label propagation is one graph-based semi-supervised learning algorithm, formulated as that a node's label propagates to neighboring nodes based on their proximity. The label propagation recommendation takes a novel view by treating recommendation as the process of label (rating) information propagation from labeled data (i.e., items with ratings) to unlabeled data (i.e., items without ratings). Since label propagation only requires a small number of training labeled data for prediction, its application in recommendation is attractive in the real world by reducing effect from

data sparsity. What's more, the empirical results also show the effectiveness of label propagation recommendation compared to other traditional methods.

However, the previous traditional label propagation recommendation suffers from some weaknesses: (1) **Data sparsity and low accuracy.** Rating data sparsity can cause low accuracy of constructing graph for label propagation. The previous constructing graph methods usually calculate item or user similarity by *cosine similarity* and *Pearson Correlation Coefficient* (PCC). However, these two methods can suffer from data sparsity since they both assume that two users have rated at least some items in common. Hence, users with fewer ratings tend to have low accuracy in similarity calculation. (2) **No trustworthiness information.** Only with the rating information, traditional recommendation methods have no trustworthiness information about the recommendation from similar users. Therefore, we do not know whether the recommendation should be trusted or not. (3) **Cold-start problem.** Previous recommender systems in CF usually cannot make any relevant recommendation at the beginning to a new user since the new user has no rating history before.

On the other hand, a new form of social communication between peoples, that is, social network sites are booming in the last decade. For example, the famous online social network site Facebook currently is utilized by over 300 million active users and about 70% of Facebook users are outside the United States. Social network has many types of user links, including the professional online social networks like Facebook and Twitter, and friendship or trust between users in some recommender system like Epinions (shown in Fig.1), and so on. In fact, some characteristics of social network can overcome or reduce the above problems in the current recommender systems.

- **Data sufficiency**. The popularity of social network sites provides sufficient user data to reduce the data sparsity problem in the current CF recommender systems. In additional, it is also helpful to solve the cold-start problem when a new user in recommender systems has information in social networks.

- **Similar Preferences with Trustworthiness**. Most of the time, the user connections in the recommender systems are based on their similar preferences since "Like attracts like". At the same time, the recommendation from his or her connected friends should have higher trustworthiness than strangers'. This scenario is similar to the case in reality that most people are willing to believe recommendations from friends with high trustworthiness.

In order to overcome the problems above and improve recommendation accuracy, we propose a social label propagation recommendation framework utilizing social network information. Firstly we propose a model to calculate the user similarity in social network. We utilize the distances between users in social network to calculate the similarity based on the assumption that direct social connections like friends can more likely to indicate user interest similarity [10] [12]. Secondly, we combine the social graph with the user graph constructed from the rating information for the final label propagation. Finally we implement *Green's function* [3] semi-supervised learning algorithm to demonstrate the

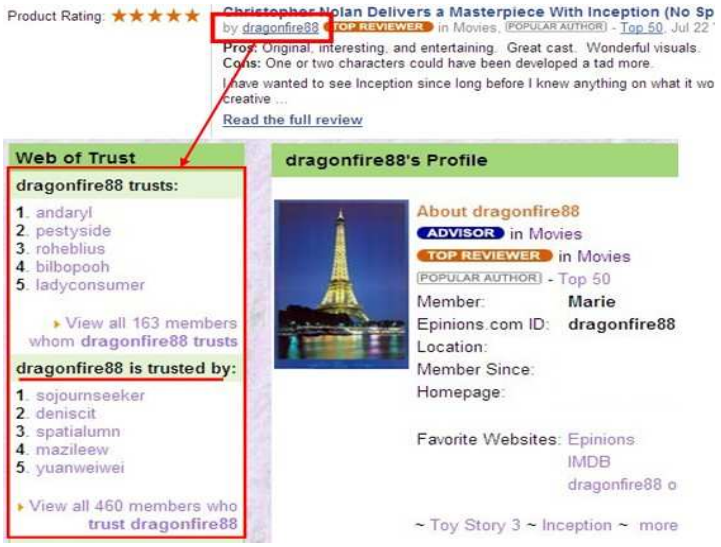

Figure 1: A Recommender System with Social Network Information

effectiveness of our framework. We conduct a series of experiments on the real world dataset from Epinions to evaluate the performance of our model. Comparing to previous traditional recommendation models, the experimental results demonstrate the outperformance of our social recommendation framework.

The contributions of our work can be mainly divided into three aspects:

- First, we improve the recommendation accuracy by combining social network and constructed graph from rating information to represent a more concise user-interest similarity graph.

- Second, we reduce the bad effective by data sparsity by utilizing social network, which can provide user connections related to interest as the supplementation as parse ratings.

- Third, to ensure the effectiveness, we compare with some previous recommendation methods on the real world data Epinions with social network inforamtion.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of label propagation recommendation approaches and some other recommendation related work. Section 3 presents the Green's function learning framework. Our social recommendation framework is presented in Section 4. Section 5 gives the experimental analysis, followed by the conclusion and discussion in Section 6.

2. RELATED WORK

Broadly speaking, current technologies of recommender systems fall into either of the two strategies: *content-based* and *collaborative filtering* (CF) [6, 9]. In content-based recommendation one tries to recommend items similar to those a given user preferred in the past. Usually, content-based recommendation approach requires external information such as user profiles, explicit item descriptions, etc., to

Table 1: User-Item Rating Matrix R_0

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	2	3	**0**	5	**0**	1	**0**
u_2	1	**0**	**0**	5	**0**	**0**	2
u_3	**0**	2	4	4	5	3	**0**
u_4	3	2	4	5	**0**	**0**	**0**
u_5	2	**0**	1	3	**0**	5	4

Table 2: Predicted User-Item Rating Matrix $\widehat{R}$

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	2	3	**2**	5	**3**	1	**3**
u_2	1	**4**	**3**	5	**3**	**4**	2
u_3	**3**	2	4	4	5	3	**3**
u_4	3	2	4	5	**4**	**3**	**3**
u_5	2	**3**	1	3	**4**	5	4

analyze item similarity or user preference. In contrast, CF recommendation is based on the core assumption that similar users on similar items express similar interest, and it usually relies on the rating information. Among CF methods, CF memory-based methods are widely employed due to its low complexity and high effectiveness. CF methods are mainly divided into two categories: *memory-based* and *model-based* methods. The most studied examples of memory-based collaborative filtering include user-based [7, 19] and item-based approaches [2, 15].

Recently, label propagation based memory-based CF methods are proposed [3, 20, 8]. Label propagation needs only a small number of labeled training data, and therefore, this scenario suits the data sparsity in real world recommender systems well. In addition, CF recommendation is inherently similar to label propagation, that is, to predict a given user's preference by propagating preference information through the pairwise similarity between users or items. Hence, label propagation is applied to CF recommendation, which is to utilize graph-based semi-supervised learning algorithm to predict unknown ratings. Chris Ding et al. [3] proposed a label propagation learning framework using Green's function and applied it to the item-based recommendation. In Ding's work, the recommendation algorithm is simple since the Green's function is proved to be the optimal solution to the iterative label propagation procedure. This method employs cosine similarity on rating information to construct an item-graph for label propagation. However, it still suffers from the data sparsity in the process of constructing item-graph.

One way to solve data sparsity and improve recommendation accuracy, social network information is considered to apply into this field. As to the social-network based recommendation, there is one paper [8] based on the Random Walk algorithm to utilize social connection and other social annotations to improve recommendation. However, this method does not utilize the rating information and is not applicable to constructing a Random Walk graph in real dataset. A more specific recommendation application with label propagation was proposed for document recommendation [20]. This method combines multiple graphs from document citation, author and venue information to construct a final similarity document-graph. However, citation is one-sided relation, which can not totally reflect the common similarity between two items. Most recently, Ma [13] proposed an idea based on social regularized matrix factorization to make recommendation based on social network information. It has a good performance with matrix factorization method but it suffers from one problem that when one new rating or user enters, it has to redo the algorithm again not preferred as an online algorithm. In order to have high accuracy, we can combine their strength and overcome their weakness. Inspired by [20], we propose the social network combined by a user graph derived from rating information to get a user graph for label propagation. It can be applicable to all recommender systems if we can have user social network information, and more practical without much cost by using only rating information.

3. LABEL PROPAGATION LEARNING

3.1 Green's Function Approach

Originally for the Laplace operator in Eq. (1),

$$\mathcal{L}f(r) = \nabla^2 f(x,y,z) = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})f(x,y,z), \quad (1)$$

Green's function plays an essential role in solving partial differential equations by transforming them into integral equations [3]. Since the Laplace operator is involved in many physical phenomenon, especially the diffusion, which is a process of particle random walk driven by a heat gradient, Green's function becomes the foundation of solving many physics problems. The physical explanation is that Green's function represents the propagation of influence of point sources. Considering the similarity between label propagation and the diffusion process, Green's function is applied into semi-supervised learning using label propagation. However, label propagation emphasizes the global and coherent nature of influence propagation, comparing to the diffusion with the local and random nature. Therefore, in order to better take advantage of Green's function in machine learning, a modified Green's function learning algorithm is proposed in [3]. Details of the Green's function learning framework are as follows:

Definition 3.1 (**Combinatorial Laplacian**). *Given a graph G with edge weights W, the combinatorial Laplacian is defined as $L = D - W$, where D is the diagonal matrix with sums of each row of W: $D = diag(We), e = (1...1)^T$.*

Definition 3.2 (**Green's Function**). *Green's function is defined as a generic graph as the inverse of the combinatorial Laplacian $L = D - W$ with zero eigen-mode discarded when it is constructed using eigenvectors of L.*

Green's function can be constructed by computing the eigenvectors of L:

$$Lv_i = \lambda_i v_i, \quad v_p^T v_q = \delta_{pq}, \quad (2)$$

where $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{l+n}$ are the eigenvalues of the corresponding eigenvectors $v_1, v_2, ..., v_{l+n}$. Assuming the graph G is connected, the first eigenvector v_1 of the zero eigenvalue λ_1 is a constant vector with multiplicity one, $v_1 = e/(l+n)^{1/2}$. According to the theory proof in [3], getting rid of zero-mode, that is, the zero eigenvalue λ_1, can make the

label propagation process consistency and global. Green's function is then the positive definite part of L:

$$G = L_+^{-1} = \frac{1}{(D-W)_+} = \sum_{i=2}^{l+n} \frac{v_i v_i^T}{\lambda_i}, \quad (3)$$

where G represents the Green function and $(D-W)_+$ indicates the zero-mode is discarded. According to the Eq. (3), it is easy to obtain that G is a square matrix.

Random walks on a graph can well illustrate the label propagation on a graph [4, 5]. Given a graph with nonnegative edge weight W, random walk can run on the graph, with the transition probability $t_{ij} = p(i \to j) = w_{ij}/D_{ii}$, or $T = D^{-1}W$. It is shown in [14] that $R_{ij} = G_{ii} + G_{jj} - 2G_{ij}$ can be a distance metric between two nodes from the random walk point of view, which is the critical role od the Green's function.

Utilized as a graph-based learning method, Green's function can be effective in semi-supervised learning with the label propagation. Assuming we have labeled data $\{x_i\}_{i=1}^{l}$ with the labels $\{y_i\}_{i=1}^{l}$ and unlabeled data $\{x_i\}_{i=l+1}^{l+n}$, the algorithm to predict the label of unlabeled data for the two-class case is showed as follows:

$$y_j = \text{sign} \sum_{i=1}^{l} G_{ji} y_i, \quad l < j < l+n. \quad (4)$$

The algorithm for the multi-class case can be written as follows:

$$y_{jk} = \begin{cases} 1, & k = \arg\max_k \sum_{i=1}^{l} G_{ji} y_{ik} \\ 0, & otherwise \end{cases}, \quad 1 \le j \le l+n, \quad (5)$$

where there are K classes and the label $Y = \{y_1, ..., y_k\}$, $Y_{ik} = 1$ if the label of x_i is the class k and otherwise $Y_{ik} = 0$.

Algorithm 1 The Algorithm for Green's Function Label Propagation

1: **Inputs**: A similarity weighted graph and labeled data
2: To calculate Green's function G:

$$G = L_+^{-1} = \frac{1}{(D-W)_+} = \sum_{i=2}^{l+n} \frac{v_i v_i^T}{\lambda_i}.$$

3: To predict unknown rating matrix $\widehat{R}^T$ according to

$$y_{jk} = \begin{cases} 1, & k = \arg\max_k \sum_{i=1}^{l} G_{ji} y_{ik} \\ 0, & otherwise \end{cases}, \quad 1 \le j \le l+n.$$

4: **Outputs**: The predicted labels for unlabeled data.

4. A SOCIAL RECOMMENDATION FRAMEWORK THROUGH LABEL PROPAGATION

4.1 Social Network

Though there are many methods to formulate a social network in a graph, we utilize the undirected graph to describe a social network since the empirical results show that label propagation on the undirected graph can have a better performance. What's more, making friends is based on mutual agreement in our real world. Therefore, it is reasonable to define the social network as an undirected graph. The definition is as follows (illustrated by Fig.2):

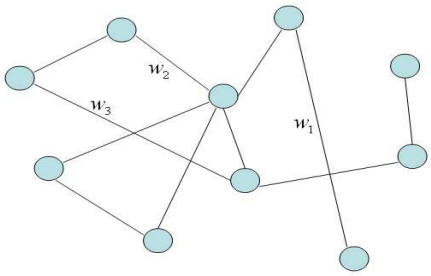

Figure 2: Social Network

Definition 4.1 (**Social Network**). *We define a social network as an undirected weighted graph $G(V, E, W)$, where V represents the set of users, E represents links among users and W is the weight to measure the similarity between two users' interest.*

In the graph of social network, the weight of one edge measures the similarity of the two connected nodes, where the similarity in our application represents the similarity of two users interest or preference. We assume that the friendship in the social network of the recommender systems is based on the common or similar interest, which is fact is proved to be right in some extent. At the same time, we believe there are chances that the friend of one user's friend may have similar interest with this user but the similarity will decrease with the distance between the two users.

Definition 4.2 (**Distance**). *Distance d_{ij} between two users i, j on a social network is defined as the shortest length of the path from i to j. The distance of two directly connected nodes(users) is 1.*

In order to constrain the value of W in the range of $[0, 1]$ and dropping gently with the distance increasing, we adopt the following common-used metric to calculate the similarity:

$$W_{ij} = \beta exp(-d_{ij}^2/\alpha^2), \quad (6)$$

$$\beta = \frac{min(soc_i, soc_j)}{max(soc)}, \quad (7)$$

where α is a control factor, the β is a weight parameter to measure the trust value of this similarity, where soc_i, soc_j are the numbers of user i, j's social connections respectively. The maximal distance d_{ij} is set to 6 based on the famous six-degree theory and the higher value of W_{ij} represents the higher similarity. In the real world, we observe that the user who is an expert will attract many people to follow him. These fans are not the real friends but just one type of appreciation, and then the unbalanced trust between the expert and the followers are lower than that of two persons who are equally trusted. This is also one way to reduce the trust noise. That is the reason why we have a β in the similarity. According to [17], the user interest similarity

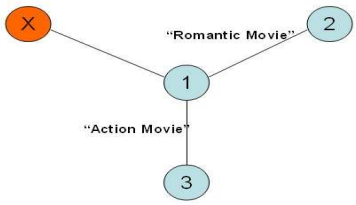

Figure 3: Noise & Multi-interest Problem in Social Network

is largest for the direct connections and hence, the $W_{i,j}$ is largest when $d_{ij} = 1$. In order to reduce the complexity, we only consider the distance between users to be 3 as the maximum. Our empirical results in the later section show that this maximal distance is sufficient to calculate similarity between most each two users.

4.2 User Graph

However, the social network may have the noisy nodes and suffers from the multi-interest problem. Illustrated in Fig. 3, the red node may be a noise user that is added wrongly without any common interest with user 1 and its friends 2 and 3. The noise node should be deleted from the graph in order to have a correct label propagation. The other problem is the multi-interest problem. For example, user 1 prefers two kinds of movies: action movie and romantic movie. Therefore he or she has two kinds of friends with two kinds of different interests, with friend 2 who is a girl loving romantic movies and friend 3 who is a boy loving action movies. If we utilize Eq.6 to calculate the similarity of user 2 and 3, their similarity may be high with the distance 2. In fact, they have quite different preferences and the the multi-interest problem can lead label propagation on social network into error.

In order to solve the problem, we combine the user graph derived from history rating information, which can provide some information of the interest between users. Assuming there are M users $U = \{u_1, u_2, ..., u_m\}$ and N items $I = \{i_1, i_2, ..., i_n\}$ in a recommender system. Then a $M \times N$ rating matrix called the user-item matrix R_0 will be obtained, with each element r_{jk} as the rating by a user u_j to an item i_k $(1 \leq j \leq M, 1 \leq k \leq N)$. Usually the rating r is an integer belonging to the set $\{1, ..., R\}$. If user u_j has not rated item i_k yet, the rating r_{jk} is set as 0.

The item graph is essential in the item-based recommendation using label propagation. The definition of item graph is as follows:

Definition 4.3 (**User Graph**). *A user graph is an undirected graph $G_u = (V, E^u)$ with a weight w^u in each edge e^u and each node $v \in V$ as an item, where $w^u_{jk} = w^u_{kj}$, $0 \leq w^u_{jk} < 1$ when $j \neq k$ and $w^u_{jk} = 1$ when $j = k$.*

In the user graph, the weight between two nodes represents the similarity between them. In this paper, the similarity between two users is symmetrical, which indicates that if item i is similar to user$j$ then user j is also similar to user i with the same similarity value.

In order to construct an item graph, we have to construct an $M \times M$ similarity matrix W^u. Many similarity computation approaches have been proposed in the memory-based recommendation and some of them have been the typical ones widely used, including *cosine similarity* [2, 18], *Pearson Correlated Coefficient* (PCC) [16, 11], etc. Among them, *cosine similarity* is widely used in user-based recommendation.

- **Cosine Similarity**

  In *cosine similarity*, each item is treated as a vector in the space of users, e.g., user u_k is denoted by the k-th row of R_0 as $\mathbf{u}_k = \langle r_{1k}, r_{2k}, ..., r_{Nk} \rangle$. The *cosine similarity* between two users j and k is given by:

$$sim(j,k) = (\mathbf{u}_j, \mathbf{u}_k) = \frac{\mathbf{u}_j \cdot \mathbf{u}_k}{\|\mathbf{u}_j\|_2 \|\mathbf{u}_k\|_2}, \tag{8}$$

  where '$\cdot$' indicates the vector dot-product operation and $\| \ \|_2$ denotes the L-2 norm distance.

  From Eq. (8), we can find that *cosine similarity* measures the angle between two vectors and therefore it is symmetrical. The resulted similarity ranges from -1 representing exactly opposite, to 1 representing exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity. However, in our paper the *cosine similarity* of two users will in the range $[0, 1]$, since the rating for each user cannot be negative. The higher *cosine similarity* value is, the more similar the two users are.

- **Pearson Correlation Coefficient (PCC)**

  In practice, different users have different rating styles. For example, one user who is a girl preferring love stories usually rates romantic movies higher than horror movies. Therefore, PCC is proposed to consider different rating styles of users. PCC between two items is based on the common users rating the two items. The definition of PCC is in the Eq. (9):

$$sim(j,k) = \frac{\sum_{i \in \mathbf{i}_j \cap \mathbf{i}_k} (r_{j,i} - \overline{r_j})(r_{k,i} - \overline{r_k})}{\sqrt{\sum_{i \in \mathbf{i}_j \cap \mathbf{i}_k} (r_{j,i} - \overline{r_j})^2} \sqrt{\sum_{i \in \mathbf{i}_j \cap \mathbf{i}_k} (r_{k,i} - \overline{r_k})^2}}, \tag{9}$$

  where $sim(j,k)$ represents the similarity between user j and k, and i belongs to the subset of items that are rated commonly. $r_{j,i}$ is the rating user j gave to item i, and $\overline{r_j}$ denotes the mean of all ratings for user j. Resembling *cosine similarity*, the value of PCC similarity ranges from 0 to 1 with higher value indicating higher similarity.

4.3 Social-User Graph

In order to construct a more concise similarity user-graph for label propagation recommendation, we combine the social network and the user-graph derived from rating information to wipe out the noise nodes and solve the multi-interest problem. The method we employ is to linear sum up the two similarities to obtain a new user similarity W^{su}_{ij}:

$$W^{su}_{i,j} = \mu W_{i,j} + (1 - \mu) W^u_{i,j}, \tag{10}$$

where μ is a control parameter we have to train or tune in the experiment.

When we get the new similarity, we should do some preprocess to get a sparse graph which can improve the performance of label propagation according to previous empirical

results. Given a threshold β, if $W_{i,j}^{su}$ is no less than β, we keep the edge or add an edge between user i and j. Otherwise, we delete the edge. We can get a new graph called ***Social-User Graph*** as the label propagation recommendation framework. This social-user graph is applied to harmonic function or Green's function, and the observed ratings are regarded as labeled data and unknown ratings are unlabeled data. The labels are the integer ratings of $1, .., R$. Finally, we apply the harmonic function and Green's function semi-supervised multi-class learning algorithms to predict the unknown ratings.

5. EXPERIMENTAL ANALYSIS

In this section, we conduct several experiments to compare the recommendation quality of our recommendation approach with social label propagation by comparing with other CF label propagation recommendation methods. Besides, we also compare our approach with other state-of-the-art recommendation methods which are not based on label propagation. The rating background in our paper is set to be discrete-valued. Our experimental analysis is expected to address the following questions:

1. What is the performance of our label propagation social recommendation framework comparing with previous label propagation recommendation algorithms purely utilizing history rating with *cosine similarity* or PCC?

2. How does our approach compare to other current recommendation methods?

3. How does the parameter μ in constructing social-user graph affect the performance of our approach?

5.1 Dataset

For our proposed recommendation framework, we have to utilize the social network information between users in the recommender systems. As for the social network information, there have been many famous popular online social network sites, like Facebook[1], to influence millions of people nowadays and hence, there are many opportunities to easily mine the social network information. However, currently few recommender systems do not cooperate with the online social network sites to improve the recommendation quality. Despite all that, there are already some online recommender systems successfully establishing social network between users without online social networks sites, such as Epinions[2]. In our paper, we choose the two online systems as the data source for our experiments on recommendation with social network information.

5.1.1 Epinions Dataset

Epinions.com is a well known knowledge sharing site and review site, which was established in 1999. In Epinions.com, users can assign reviews for the products with integer ratings from 1 to 5. These ratings will influence future customers when they are about to decide whether a product is worth buying or a movie is worth watching. Every member of Epinions maintains a "trust" list which can be regarded as a social network of trust relationships between users. Epinions is thus an ideal source for experiments on social trust recommendation.

[1]http://www.facebook.com/
[2]http://www.epinions.com/

Table 3: Statistics of Epinions Dataset

Epinions Dataset	Statistics
#User	975
#Item	1732
#Rating	30,547
#Training(90%)	27,445
#Test(10%)	3,102
#Training(80%)	24,017
#Test(20%)	6,530
Density	1.81%

Table 4: Social Network Statistics of Epinions Dataset

Epinions Dataset	Statistics
Avg. Links per User	74.41
Max. Links per User	445
Min. Links per User	6

The dataset used in our experiments was collected by crawling the Epinions.com site on Jan 2009. It consists of 11,880 users who have rated a total of 226,101 different items. The total number of ratings is 588,552. The density of the user-item rating matrix is less than 0.022%. However, in our experiment, we conduct it on small subset, containing 975 users and 1732 items, with a total 30547 ratings and 1.81% density. We can observe that the user-item rating matrix of Epinions is very sparse, since the densities for the most famous collaborative filtering dataset Movielens (943 users, 1,682 movies and 100,000 ratings) are 6.3%. Moreover, an important factor that we choose the Epinions dataset is that user social trust network information is not included in the classical Movielens. In our label propagation recommendation algorithms, we split the data by about 80%/20% and 90%/10% into training data and test data. The statistics of the Epinions dataset is summarized in Table 3. As to the user social trust network, the total number of issued trust statements is $71,580$. The statistics of this data source is summarized in Table 4.

5.2 Metrics

We use three most widely used metrics to measure the prediction quality of recommendation approaches in our experiments.

- **Mean Absolute Error (MAE)**. MAE is defined as:

$$MAE = \frac{1}{n}\sum_{j,k}|r_{jk} - \widehat{r_{jk}}|, \tag{11}$$

  where n is the number of tested ratings, r_{jk} is the rating that user j gave to item k and $\widehat{r_{jk}}$ denotes the predicted rating that user j gave to item k.

- **Mean Zero-one Error (MZOE)**. MZOE is defined as:

$$MZOE = \frac{1}{n}\sum_{j,k} 1_{r_{jk} \neq \widehat{r_{jk}}}, \tag{12}$$

where n is also the number of tested ratings. MZOE calculates the fraction of incorrect predictions.

- **Rooted Mean Squared Error (RMSE)**. RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{j,k} (r_{jk} - \widehat{r_{jk}})^2}{n}}, \qquad (13)$$

where n is also the number of tested ratings.

5.3 Experiment Setting

5.3.1 Model Social Connections

In order to construct the social-user graph, firstly we have to obtain the social network information by modeling the social connections between users. In the Epinions dataset, user connections are called trust. If two users have the direct trust relationship, we model their distance in social connections as 1. Then to calculate the interest similarity based on social connections, we utilize the formula $W_{ij} = \beta * exp(-d_{ij}^2/\alpha^2)$ where d_{ij} is the distance between two users and α is a parameter to control the speed of descending. In our assumption, we regard that the direct links can more reflect the high chances of common interest and with the distances increasing, the similarity will go down. According to the Six Degrees of Separation, we set the similarity as 0 when the distance is over 6. In our experiment, we set the similarity as about 0.4 when the distance is 3. In fact, at hen same time, we tune the parameter α to make our modeling have a good result for our approach. In our experiments, we find that the maximal distance does not need to be set as 6 since the difference is sufficiently small when the distance is larger than 3. Considering this phenomenon as well as the complexity problem, we set the maximal distance as 3.

5.3.2 Constructing Social-User Graph

After modeling the social network, we calculate the user graph with rating history of training data. In order to construct the user-graph, we utilize the most two common-used methods: *cosine similarity* and PCC. Since the data is sparse, we may also have a sparse user-graph with few similarity values. We can combine the previous modeled social network and the user-graph to obtain the social-user graph with $W_{i,j}^{su} = \mu W_{i,j} + (1-\mu) W_{i,j}^{u}$, where $W_{i,j}^{u}$ is the similarity of the user-graph. The control parameter is one objective to find out in our experiment. We tune the value of μ to measure its impact to our framework. When we get the value of $W_{i,j}^{su}$, we do some processing to make the graph more suitable for label propagation, according to the empirical result that sparse graph is more helpful to label propagation. We set a threshold β to filter the connections with low similarity. Usually, the β is lower than or equal to the average of the similarity values. The similarity values lower than β will be set to be 0.

5.4 Impact of Control Parameter μ

In this subsection, we measure the impact of control parameter μ in our recommendation framework. Since both the social network and the user-graph have advantage and disadvantage, we try to find out a good tradeoff between them. We implement our framework with Green's function in the Epinions dataset. The results are shown in Fig. 4 and Fig. 5.

The main advantage of our model is that we combine the social network information with rating history analysis, which helps to construct a more accurate user graph for Green's function recommendation. There is a weight parameter μ to balance the social network information and rating history. In fact, it is to obtain a balance between traditional similarity (*cosine similarity* or PCC) from ratings and social similarity from social connections. When $\mu = 0$, the user similarity is only the classical *cosine similarity*, and when $\mu = 1$, the user similarity is only similarity from social connections. In other cases that μ is between $(0, 1)$, we obtain the user similarity combining both. From both figures, we can see that combining social connection information can improve the recommendation accuracy. However, the pure social connection information has less effect than the rating history on the recommendation. The reason is that social connections have many noises and do not completely reflect the user preferences. Rating information plays a more important role in recommendation, which is directly related to ratings.

Fig. 4 shows the impacts of μ on MAE, MZOE and RMSE on the Epinions dataset, which has 90% training data and 10% test data. We can observe from this figure that the value of μ affects the performances of our model significantly, which demonstrates that adding social connection information on user graph construction can improve the Green's function recommendation greatly. As shown in all the three charts in Fig. 4, when μ increases from 0, the three prediction errors decrease first, that is, the prediction accuracy increases first. But when μ passes 0.3, the prediction accuracy begins to decrease dramatically with further increase of μ. The optimal value of weight parameter is near 0.3 to get the best prediction accuracy when the dataset is divided into 90% training data and 10% test data. However, in FIg. 5 which illustrates the recommendation accuracy when the dataset is divided into 80% training data and 20% test data, the optimal parameter has a different value. When μ is around 0.6, the model can obtain the best performance in the current training data. This difference may be caused by the different training data. Since in both experiments, the part of similarity from social connections are the same, but the similarity from training data is different. The first training data are more than the second one, which is helpful for the calculation of cosine similarity. This shows that when we have sufficient rating information, the social connection information works less in the recommendation. However, when the rating data are not sufficient, social information can play a more important role. The result supports what we propose in our paper, that is, social information can solve the cold-start problem when a new user comes into a recommender system.

5.5 Performance Comparison

In order to measure the recommendation quality of our approach, we compare it with different recommendation algorithms: user-based recommendation with Green's function with *cosine similarity*(UCOS) and PCC(UPCC), previous Green's function recommendation with *cosine similarity*(GCOS) and PCC(GPCC). As for our approach, we have two different ones (GSUCOS and GSUPCC) based on different user-graph constructing methods. The training data

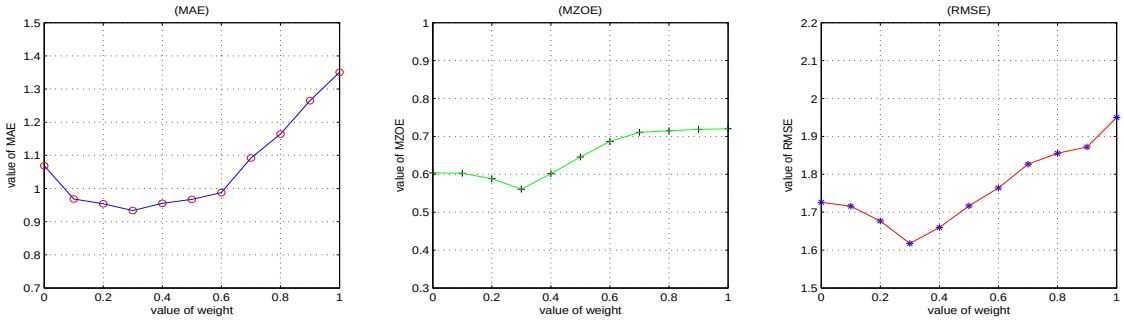

Figure 4: Impact of Control Parameter μ in Epinions Data with 90% Training Data

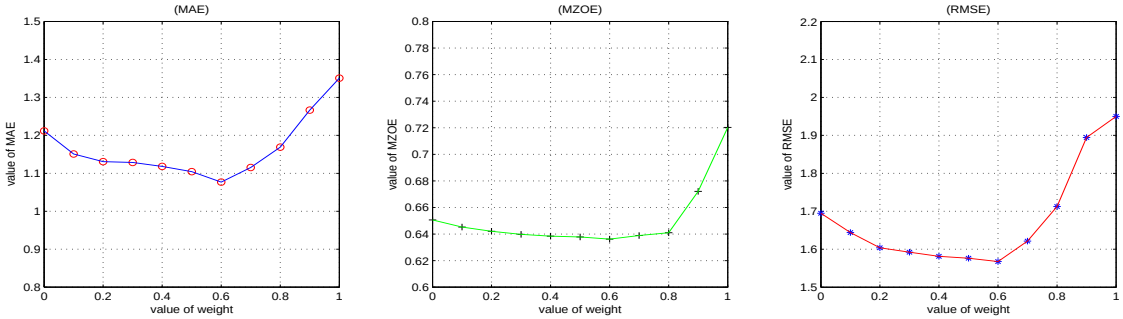

Figure 5: Impact of Control Parameter μ in Epinions Data with 80% Training Data

in this part of experiments is 80% out of the whole data, and the results of our model are both the best one with the optimal parameter μ.

The main goal of recommendation model in CF is to provide accurate recommendation for users when only rating information is given. The prediction accuracy of memory-based methods is empirically lower than that of model-based methods. Though Green's function model is an memory-based recommendation model in a novel way, it improves the prediction accuracy significantly. The essential point of its good performance is the user graph construction. However, the way of building user graph in the previous model suffers from data sparsity problem. In our paper, we propose a novel model combining social connection information for Green's function recommendation. In order to evaluate the efficiency and superiority of our model, we compare our model to the previous model with *cosine similarity*, PCC as well as the other classical user-based methods. Besides, we set the parameter $\mu = 0.6$ in our model which achieves the best accuracy in this training dataset.

Table 5 shows the results of performance of different recommendation methods. We observe that our new approach has the lowest MAE, MZOE and RMSE among these errors. As for MAE, MZOE and RMSE, the lower the value is, the better the recommendation performs. Because these three metrics measure the prediction from different aspects, if the results in these three metrics have the coincident trends then they can demonstrate the performance more powerfully. The results demonstrate efficiency and superiority of our model over user-based methods. The comparison also shows that the Green's function recommendation with PCC has a better performance compared to the model with *cosine similarity*, which agrees with the idea that PCC is a better measurement than cosine in recommendation. Compared to MAE of the previous model GCOS, our model GSUCOS gains an increase by 3.2%. Compared to MAE of GPCC, though GSUCOS performs worse than it, GSUPCC has a higher MAE by 6.7%. These results demonstrate that our model can improve the accuracy of user-based recommendation effectively.

6. CONCLUSION

Predicting unknown ratings in collaborative filtering can be viewed as a process of label propagation, that is, the influence propagation from observed ratings to unknown ratings. Green's function can be applied to recommendation as a method of label propagation. Before the Green's function works, an accurate graph should be constructed first, which is based on the user similarity computation. Previous work used the classical *cosine similarity* which suffers from the data sparsity problem and being lack of trustworthiness value. In this paper, we propose a new similarity computation approach combining social connections and classical similarity calculation to improve performance of Green's function in recommendation. The currently popular social network can provide sufficient user connection data, which can be utilized to model user similarity. Besides, social connections can inflect user trust-worthiness. By combining both similarities to construct a user-social graph, the framework is able to improve the recommendation accuracy. Finally, we also conduct some experiments with real data to demonstrate that our approach outperforms the previous methods.

7. ACKNOWLEDGEMENT

Table 5: Performance Comparisons in Epinions Data with Green's Function

Metrics \ Methods	UCOS	UPCC	GCOS	GPCC	GSUCOS	GSUPCC
MAE	1.224	1.1332	1.1123	0.9667	**1.0768**	**0.9023**
MZOE	0.6621	0.6592	0.6410	0.6028	**0.6362**	**0.5728**
RMSE	1.7113	1.6983	1.6948	1.5471	**1.6176**	**1.4938**

The work in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413210).

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[2] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):177, 2004.

[3] C. Ding, H. Simon, R. Jin, and T. Li. A learning framework using Green's function and kernel regularization with application to recommender system. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 269. ACM, 2007.

[4] P. Doyle and J. Snell. Random walks and electric networks, volume 22 of Carus Mathematical Monographs. *Mathematical Association of America, Washington, DC*, 52, 1984.

[5] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic processes and their applications*, 2(4):311–336, 1974.

[6] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[7] R. Jin, J. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344. ACM, 2004.

[8] I. Konstas, V. Stathopoulos, and J. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM, 2009.

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

[10] D. Lee and P. Brusilovsky. Social networks and interest similarity: the case of CiteULike. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 151–156. ACM, 2010.

[11] H. Ma, I. King, and M. Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 46. ACM, 2007.

[12] H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[13] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[14] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Machine Learning: ECML 2004*, pages 371–383, 2004.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, page 295. ACM, 2001.

[16] U. Shardanand and P. Maes. Social information filtering: algorithms for automating Şword of mouthŤ. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[17] R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *Proceedings of the Delos-NSF workshop on personalization and recommender systems in digital libraries*, 2001.

[18] F. Wang, S. Ma, L. Yang, and T. Li. Recommendation on item graphs. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 1119–1123, 2006.

[19] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM New York, NY, USA, 2005.

[20] D. Zhou, S. Zhu, K. Yu, X. Song, B. Tseng, H. Zha, and C. Giles. Learning multiple graphs for document recommendations. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 141–150. ACM Press, 2008.