Improving Personal Product Recommendation via Friendships’ Expansion

Chunxia Yin, Tao Chu

The First Aircraft Institute, Aviation Industry Corporation of China, Xi’an, China.
Email: dingdong1104@163.com

Received July 2013

ABSTRACT

The trust as a social relationship captures similarity of tastes or interests in perspective. However, the existent trust information is usually very sparse, which may suppress the accuracy of our personal product recommendation algorithm via a listening and trust preference network. Based on this thinking, we experiment the typical trust inference methods to find out the most excellent friend-recommending index which is used to expand the current trust network. Experimental results demonstrate the expanded friendships via superposed random walk can indeed improve the accuracy of our personal product recommendation.

Keywords: Personal Product Recommendation; Trust Inference; Listening and Trust Preference Network

1. Introduction

Nowadays some web sites, such as last.fm and delicious, have taken the lead in guiding users to build trust relationships, that is, a user on such a website can select some other users with similar tastes as his/her trusted friends. The effectiveness of a recommendation algorithm rests with the effective use of personal information from net users. Personal trust information obviously contains reliable personal friend relations and therefore has the potential to help provide better recommendations. Based on this thinking, we proposed the LTPN algorithm [1], namely the artist recommendation algorithm via the listening and trust preference network. However, the provided user-trusted friend relations by web users are only a small fraction of the potential pairings between users so that the existent trust data is very sparse.

The trust as a social relationship captures similarity of tastes or interests in perspective. Obviously, it may be difficult to know tastes or interests of a person from his sparse friendships. Instinctively, if the friendships can be accurately inferred, the sparse problem may be relieved so that the expanded friendships are expected to improve the accuracy of our LTPN algorithm. That is, the sparsity of trust data may severely influence the behavior of our LTPN algorithm. In a word, if trust is used to support decision making, it is important to have an accurate estimate of trust when it is not directly available, as well as a measure of confidence in that estimate [2].

Many biological, technological, social systems can be well described by complex networks with nodes representing individuals or items and links denoting the relations or interactions between nodes [3-6]. As part of the surge of research on complex networks, a considerable amount of attention has been devoted to the computational analysis of social networks [7]. An important scientific issue relevant to social network analysis is personal trust inference, which aims at estimating the likelihood of existence of trust with respect to a particular topic between two individuals. A number of algorithms for computing trust from social networks rely on the knowledge of the network topology, which are the mainstream class of trust inference algorithms. They provide us an entry of inferring friendships for online web users.

In this paper, we investigate the use of trust inference for improving personal product recommendation. To this end, we review and discuss typical trust inference methods to find out the most effective friendship-expanding method, and then explore whether the expanded user-trusted friend relations can improve the accuracy of the LTPN algorithm. The numerical results indicate that the inferred friendships via superposed random walk can improve our existent trust-aware product recommendation.

2. Method

In this paper, we still discuss the artist as the special product. We assume that a music artist recommendation system consists of a user-set $U = \{u_1, u_2, \ldots, u_n\}$ and an artist-set $A = \{a_1, a_2, \ldots, a_m\}$. Each user has listened to
some artists and selected some other users with similar musical tastes as his/her trusted friends. The user-listened artist and user-trusted friend relations can uniformly be described by a listening and trust relation network (LTRN for short), $G (U, A, E1, E2)$, where $E1$ is the set of user-trusted friend relations and $E2$ is the set of user-listened artist relations. Note that multiple links and self-connections are not allowed in the network. Clearly, $G (U, A, E1, E2)$ can be seen as a combination of the trust network $G (U, E1)$ and bipartite network $G (U, A, E2)$.

**Figure 1** gives a simple example of the listening and trust relation network. Each black solid line represents a listening count for a [user, artist] pair. For example, $u_1$ has listened to $a_1$ 80 times. The red dashed curves show the mutual (undirected) trust relations between users. The listening and trust relation network (a) can be seen as a combination of the trust network (b) and the bipartite network (c).

To improve our LTPN algorithm, we need solve two problems. First, the personal trust between two individuals without a direct connection needs to be inferred. We will solve this problem by applying an excellent personal trust inference method to the existent trust network. After these inferred trust relations with maximal evaluated values are added to LTRN, the expanded listening and trust relation network (ELTRN for short) will be built. Second, ELTRN is transformed into the listening and trust preference network. Obviously, ELTRN is a special LTRN. For convenience, the process of transforming the listening and trust relation network into the listening and trust preference network is called as the LTRN-LTPN transformation. We use the LTRN-LTPN transformation method in Ref. [1] to solve the second problem.

Based on the above analysis, we put forward the improved LTPN algorithm via friendships’ expansion (FELTPN for short).

FELTPN is described as the following four steps:

1. Expand the user-trusted friend relations by making use of an excellent trust inference method.

2. Build the listening and trust preference network by applying the LTRN-LTPN transformation to ELTRN.

3. Apply the LTPN algorithm to the listening and trust preference network for the personal artist recommendation.

Next, our LTRN-LTPN transformation and LTPN algorithm are shortly introduced. Readers are encouraged to see Ref. [1] for their details.

The user-listened artist relations can be expressed by a listening count matrix, $C_{n \times m}$, where $C_{if}$ is the listening count of $u_i$ to $a_f$. The user-trusted friend relations can be expressed by an adjacent matrix $T_{n \times n}$. If there exists a mutual trust between $u_i$ and $u_j$, $T_{ij} = T_{ji} = 1$, otherwise $T_{ij} = T_{ji} = 0$.

The LTRN-LTPN transformation is as follows:

1. Transform the user-listened artist relations into the impartial listening preferences, $C'_{n \times m}$, in which

   $C'_{if} = \frac{C_{if}}{\sum_g C_{ig}}$.  \hspace{1cm} \text{(1)}$

   (2) Take advantage of the cosine similarity between the listening preferences of users to build trust preferences, $\hat{T}_{n \times n}$, in which

   $\hat{T}_{ij} = \begin{cases} 
   \frac{C'_{i} \cdot C'_{j}}{\|C'_{i}\|_2 \cdot \|C'_{j}\|_2}, & \text{if} \quad T_{ij} = 1 \\
   0, & \text{otherwise}
   \end{cases}$. \hspace{1cm} \text{(2)}$

**Figure 1.** (Color online) A simple listening and trust relation network consisting of four users and eight artists.
where $C'_i$ is the $i$th row vector of matrix $C'_{n \times m}$.

Now, $C'_{n \times m}$ and $T_{n \times n}$ express together the listening and trust preference network evolving from the corresponding relation network.

Suppose that $u_i$ is a target user, $a_l$ is an artist to which he/she has never listened, and $\{u_l\}$ is a set consisting of all his/her trusted friends having listened to $a_l$.

The basic idea of our LTPN algorithm is as follows:

1. Find all these two-step trust and listening preference paths from $u_i$ to $u_l$ to $a_l$, each of which obviously contains the trust preference and listening one in turn. Each such path is called $\text{path}(u_i,u_l,a_l)$.

2. Acquire the single-path level of preference of $u_i$ for $a_l$ from $\text{path}(u_i,u_l,a_l)$, $P_{ijl}$, by doing the multiplication on listening and trust preferences of the path, that is,

\[ P_{ijl} = T_{il}C'_{jl} \]

3. Obtain the level of listening preference of $u_i$ for $a_l$, $P_{il}$, by adding all the above single-path levels of preference of $u_i$ for $a_l$ together, that is,

\[ P_{il} = \sum_j P_{ijl} = \sum_j T_{il}C'_{jl} \]

Figure 2 explains the general idea of FELTPN from the angle of the target user. Now suppose $u_3$ is a target user.

To avoid interference, plot (a) refines Figure 1(a) from the angle of $u_3$ and emphasizes these artists not selected by $u_3$ with green bold solid lines. Plot (b), in which the blue bold dashed curve represents the inferred friendship for $u_3$, is the ELTRN only for $u_3$ after a certain trust inference algorithm is applied to the trust network in Figure 1(b).

Figure 2. (Color online) Illustration of FELTPN.
user-listening artist links of \( u_3 \) represent the levels of listening preference of \( u_3 \) for the artists, presents the listening and trust preference network only for \( u_3 \) after the LTRN_LTPN transformation is applied to the ELTRN in plot (b). Eventually, the predicted scores for \( u_3 \) are shown in plot (d) after our LTPN algorithm is applied to the listening and trust preference network in plot (c).

3. Experiment Data and Evaluation Metrics

We use a benchmark data set, lastfm, to evaluate the effectiveness of the personal trust inference method and artist recommendation algorithm. The lastfm data, released in the framework of HetRec 2011 (http://ir.ii.uam.es/hetrec2011), consists of 1892 users and 17632 artists from Last.fm online music system (http://www.last.fm). It records user-trusted friend and user-listened artist relations. For the purpose of effectively testing our algorithm, we remove those users having listened to less than ten artists. As a result, the listening and trust relation network, originating from a random sampling of 990 users, contains 49191 user-listened artist relations and 3878 mutual trust relations. Therefore, the average number of friend relations per user is 7.8 and the average number of user-listened artist relations per user is 49.69.

Figure 3, where \( N_d \) is the number of users with \( d \) friends, exhibits the degree distribution of trust network from the lastfm data. Obviously, it satisfies heavy-tailed distribution, which means most users own sparse friendships.

The personal trust inference method need calculate the likelihood of existence of all non-observed trust relations of the target user, then sorts them in a descending order according to their likelihoods and finally recommend the target user these top-Len friends. To determine how accurately a personal trust inference method is able to mine friendships, the observed user-trusted friend relations \( E_1 \) need to be randomly divided into two parts: The training set Figure 3.

\( E_1^{Tr} \) contains 90% of user-trusted friend relations from each user, and the remaining 10% constitutes the probe set \( E_1^{Pr} \). \( E_1^{Tr} \) is treated as known information, while no information in \( E_1^{Pr} \) is used for prediction.

Similarly, the personal artist recommendation algorithm need calculate the listened likelihood of all non-listened artists of the target user in the future, then sorts them in a descending order according to their likelihoods and finally recommend the target user these top-Len artists. To determine the effectiveness of our artist recommendation algorithm, user-listened artist relations \( E_2 \) also need to be randomly divided into two parts: The training set \( E_2^{Tr} \) contains 90% of user-listened artist relations from each user, and the remaining 10% constitutes the probe set \( E_2^{Pr} \). \( E_2^{Tr} \) and (observed and expanded) mutual trust relations are treated as known information, while no information in \( E_2^{Pr} \) is used for prediction.

We use three standard metrics, area under the receiver operating characteristic curve (AUC) [8], Precision [9] and Recall [9], to quantify the accuracy of a personal trust inference method. Note that the AUC evaluates the whole ranking list of all candidate friends while both Precision and Recall only focus on the top-Len candidate friends of the list.

We use precision, recall, inter-diversity [10], intra-diversity [11] and popularity [10] to evaluate the quality of an artist recommendation algorithm. The first two are different accuracy metrics. The inter-diversity (\( D^\) ) measures the personalization of recommendations for different users with different habits and tastes. The intra-diversity (\( D^\) ) surveys the diversity of topics in a recommendation list. The last one measures the capability of recommending dark (less popular) artists.

Before introducing concrete definitions of the above metrics, we must interpret some symbols to be used in these definitions.

Denote by \( \Omega \), the set containing all \( n-1 \) possible trust relations of \( u_i \). Then the set of all evaluated trust relations of \( u_i \) is \( \Omega - E_1^{Tr} \) where \( E_1^{Tr} \) denotes the training set of trust relations of \( u_i \). Accordingly, \( E_1^{Pr}_j \) is the probe set of trust relations of \( u_i \). Here, trust relations of \( u_i \) can be interpreted as his/her trusted friends.

As a matter of convenience, the (friend or artist) recommendation list for \( u_i \) is consistently expressed as \( L_i \) and the (friend or artists) probe set for \( u_i \) is consistently expressed as \( P_i \).

\[ | \cdot | \] denotes the number of elements in a set or list. The number of users having listened to \( a_j \) is expressed as \( d_j \). \( B_{u_{\text{sym}}} \) is an adjacency matrix. If \( u_i \) has listened to \( a_j \), \( B_{u_{\text{sym}}}(i,j) = 1 \), otherwise \( B_{u_{\text{sym}}}(i,j) = 0 \).

In the present case, the AUC statistic can be inter-
Considering the probability that a randomly chosen trust relation in \( E_t^{Pi} \) is given a higher trust score than a randomly chosen nonexistent relation. Given the whole ranking list of all evaluated trust relations of \( u_i \), among \( w \) independent comparisons, if there are \( w' \) times the relation in \( E_t^{Pi} \) having a higher score and \( w'' \) times being of the same score, the AUC value for \( u_i \) is computed as follows:

\[
AUC_i = (w' + 0.5w'')/w. \tag{5}
\]

The AUC for the whole system is the average over all users, that is,

\[
AUC = \frac{\sum AUC_i}{n}. \tag{6}
\]

The degree to which the AUC value exceeds 0.5 indicates how much better predictions are than pure chance.

Precision is defined as the ratio of the number of relevant items to the total number of recommended items. Given the top-\( n \) items of a ranking list for \( u_i \), the precision for \( u_i \) obviously equals \( |L_i \cap P_i|/|P_i| \). Therefore the precision of the whole system is

\[
Precision = \frac{1}{n \times \text{Len}} \sum |L_i \cap P_i|/|P_i|. \tag{7}
\]

Clearly, a higher value of precision means a better performance.

Recall is defined as the ratio of the number of relevant items to the total number of items in the probe set. Given the top-\( n \) items of a ranking list for \( u_i \), the recall for \( u_i \) obviously equals \( |L_i \cap P_i|/|P| \). Therefore the recall of the whole system is

\[
Recall = \frac{1}{n} \sum |L_i \cap P|/|P|. \tag{8}
\]

A higher value of recall means a better performance.

A basic \( D^a \) measures the diversity of artists between two recommendation lists. \( D^a \) between two recommendation lists can be quantified by the Hamming distance between the two lists. Therefore the inter-diversity of the whole system is

\[
D^a = \frac{2}{n(n-1)} \sum_{i\neq j} \left( 1 - \frac{|L_i \cap L_j|}{\text{Len}} \right). \tag{9}
\]

Generally speaking, the greater value of \( D^a \) means the more personalized recommendations for different users.

A basic \( D^a \) measures the diversity of artists within a recommendation list. Therefore, the intra-diversity of the whole system is

\[
D^a = \frac{1}{n} \sum |1 - \frac{2}{|L_i|(|L_i|-1)} \sum S_{fgi}|, \tag{10}
\]

where \( S_{fg} \) denotes the cosine similarity between \( a_f \) and \( a_g \), namely,

\[
S_{fg} = \frac{1}{\sqrt{k_fk_g}} \sum B_fB_g. \tag{11}
\]

Generally speaking, the higher value of \( D^a \) means the more diversified topics within recommendation lists.

The popularity of a recommendation list is defined as the average degree of artists in the list. Therefore the popularity of the whole system is

\[
Popularity = \frac{1}{n} \sum \frac{1}{|L_i|} \sum_{s_{fgi}} d_f. \tag{12}
\]

The smaller \( Popularity \), corresponding to the stronger capability of mining dark artists, is preferred since those lower-degree artists are hard to be found by users themselves.

### 4. The Inference of User-Trusted Friend Relations

In this section, we seek the most excellent (accurate and fast) trust inference method among the most influential and representative sixteen trust inference indices. These indices can be classified into three categories: local, quasi-local and global in accordance with the scope of topological information used by the inference measure. Clearly, a local measure only requires the information about node degree and the nearest neighborhood, and a global measure asks for the global knowledge of network topology. Note that the quasi-local measure [12] considers more topological information than the local but does not require global topological information. Therefore, the global is the most time-consuming, the quasi-local takes second place, and the local needs the minimum time.

We compare the accuracies of sixteen indices, which are CN [13,14], Jacard [15], Salton [16], LHN [17], HPI [18], HDI [10], Sørensen [19], PA [20], AA [21], RA [22], LP [22], LRW [23], SRW [23], Katz [24], MFI [25] and RWR [26], about predicting friends in Table 1.

<table>
<thead>
<tr>
<th>friend inference indices</th>
<th>AUC</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>0.8353</td>
<td>0.014</td>
<td>0.6797</td>
</tr>
<tr>
<td>Jacard</td>
<td>0.8292</td>
<td>0.0126</td>
<td>0.67</td>
</tr>
<tr>
<td>Salton</td>
<td>0.8277</td>
<td>0.0115</td>
<td>0.6646</td>
</tr>
<tr>
<td>LHN</td>
<td>0.8096</td>
<td>0.0014</td>
<td>0.6271</td>
</tr>
<tr>
<td>HPI</td>
<td>0.819</td>
<td>0.0028</td>
<td>0.6372</td>
</tr>
<tr>
<td>HDI</td>
<td>0.8288</td>
<td>0.0126</td>
<td>0.668</td>
</tr>
<tr>
<td>Sørensen</td>
<td>0.8292</td>
<td>0.0126</td>
<td>0.67</td>
</tr>
<tr>
<td>PA</td>
<td>0.7919</td>
<td>0.0029</td>
<td>0.6289</td>
</tr>
<tr>
<td>AA</td>
<td>0.8375</td>
<td>0.0147</td>
<td>0.6848</td>
</tr>
<tr>
<td>RA</td>
<td>0.8362</td>
<td>0.0137</td>
<td>0.6794</td>
</tr>
<tr>
<td>LP (( \alpha = 0.001 ))</td>
<td>0.6662</td>
<td>0.0021</td>
<td>0.6314</td>
</tr>
<tr>
<td>LRW (( t = 3 ))</td>
<td>0.9635</td>
<td>0.0226</td>
<td>0.7723</td>
</tr>
<tr>
<td>SRW (( t = 3 ))</td>
<td>0.9875</td>
<td>0.0776</td>
<td>0.9565</td>
</tr>
<tr>
<td>Katz (( \alpha = 0.001 ))</td>
<td>0.6662</td>
<td>0.0021</td>
<td>0.6314</td>
</tr>
<tr>
<td>MFI</td>
<td>0.429</td>
<td>0.0019</td>
<td>0.6305</td>
</tr>
<tr>
<td>RWR (( c = 0.9 ))</td>
<td>0.9702</td>
<td>0.049</td>
<td>0.8318</td>
</tr>
</tbody>
</table>
Jacard, Salton, LHN, HPI, HDI, Sørensen, PA, AA and RA are local indices. LP, LRW and SRW are quasi-local indices. Katz, MFI and RWR are global indices.

The results reported here are averaged over 30 independent runs, each of which corresponds to a random division of training and testing sets of user-trusted friend relations. The entries corresponding to the highest accuracies are emphasized in bold. The entries of LP, LRW, SRW, Katz, and RWR are the accuracies corresponding to their optimal parameters. The numbers inside the brackets denote the optimal values of their parameters.

We can see that the highest accuracies are all acquired by SRW with \( t = 3 \). Clearly, this optimal walking step is very small, which unfolds two points: SRW with the optimal step owns the low computational complexity allowing high scalability and it abandons the superfluous information that makes no contribution or negative contribution to the prediction accuracy. Therefore, SRW can be expected to improve the artist recommendation of LTPN algorithm by expanding users’ friendships.

5. The Personal Artist Recommendation via Friendships’ Expansion

These results in Figure 4 report the comparisons between FELTPN and three other typical recommendation algorithms, LTPN, the collaborative filtering algorithm (CF) and the standard network-based inference (NBI) [27], for different moderate list lengths.

The basic CF recommends artists for the target user \( u_t \) by the following three steps:

1) Compute the affinity \( Q_{ij} \) between \( u_t \) and every other user \( u_i \) as follows:

\[
Q_{ij} = \frac{1}{\sqrt{d(u_i)d(u_j)}} \sum_j B_{ij} B_{ij},
\]

where \( d(u_i) \) denotes the number of artists \( u_i \) has listened to.

2) Compute the level of listening preference \( P_{tl} \) of \( u_t \) for his/her unheard artist \( a_l \) as follows:

\[
P_{tl} = \frac{\sum_{j \in \Omega} Q_{ij} B_{ij}}{\sum_{j \in \Omega} Q_{ij}}.
\]

3) Sort these artists, which \( u_t \) has never heard before, in decreasing order according to their predicted scores, and finally recommend the top \( \text{Len} \) artists to \( u_t \).

The results are averaged over 30 independent runs, each of which corresponds to a random division of train-
ing and testing sets of user-listened artist relations. Here, ELTRN are produced by the following friendship-expanding method: only when the number of friends of a user is less than 10, his friendships are expanded via SRW and the number of expanded friends is set to 1. We can see from Figure 4 that the expanded friendships via SRW can improve both precision and recall of our LTPN algorithm while both outstanding diversity and excellent popularity are maintained.

6. Conclusion and Discussion

In this paper, we exploit SRW to expand the personal friendships and then make use of our LTPN algorithm to recommend personal artists. The experimental results show that the expanded friendships via SRW can improve the recommendation accuracy of our LTPN algorithm. However, we can also see that the improvement is not large. External information, such as tag information and rating information, may provide more useful information and insights for personal recommendation. Furthermore, the sparsity of existent data and the huge size of real systems are two most main difficulties for the studies of personal recommendation. The former may result in large difficulties in building statistical models. The latter requires highly efficient algorithms because any highly accurate algorithm will become meaningless if the consuming time is unacceptable. Therefore, designing a better algorithm in both accuracy and speed is our future challenge, especially for sparse and huge networks.

7. Acknowledgements

The authors thank the anonymous referees for helpful suggestions and comments and acknowledge HetRec 2011 for providing us the lastfm data.

REFERENCES


[19] T. Sørensen, “A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vege-
tation on Danish Commons,” *Biological Skr*, Vol. 5, No. 4, 1948, pp. 1-34.


