A Revisit to Social Network-Based Recommender Systems*

Hui Li, Dingming Wu, Nikos Mamoulis
Department of Computer Science
The University of Hong Kong, Hong Kong
{hli2,dmwu,nikos}@cs.hku.hk

ABSTRACT

With the rapid expansion of online social networks, social network-based recommendation has become a meaningful and effective way of suggesting new items or activities to users. In this paper, we propose two methods to improve the performance of the state-of-art social network-based recommender system (SNRS), which is based on a probabilistic model. Our first method classifies the correlations between pairs of users' ratings. The other is making the system robust to sparse data, i.e., few immediate friends having few common ratings with the target user. Our experimental study demonstrates that our techniques significantly improve the accuracy of SNRS.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; J.4 [**Computer Applications**]: Social and Behavioral Sciences

Keywords

Recommender System, Social Network, Social Influence

1. INTRODUCTION

Recommender systems have become an important and independent research area since the mid-1990s [1], achieving great success in e-commerce. Many famous online vendors such as Amazon and Netflix leverage recommender systems to advertise products to customers. These systems also serve as effective tools to help users in finding useful information and overcoming the problem of information overload.

Recently, many online social networks (OSNs), such as Facebook and LinkedIn, have emerged and attracted a vast number of users. Social networks contain enormous information which is still growing rapidly as more and more users join them. Besides, it is a common phenomenon that we

---

usually turn to our family, friends and acquaintances for suggestions when we want to make a decision or a purchase. Also, we tend to affiliate and establish relationships with people who share the same interests with us. Since social influence may affect users' acceptance of recommendations, researchers have proposed to mine valuable contextual information (e.g., time, location, mood and relationship) from OSNs and take social influence into consideration in the recommendation process, to supplement the traditional data (i.e., user ratings). These social-based recommendation approaches are briefly introduced in Section 2.2.

In this paper, we improve the state-of-art social network-based recommender system (SNRS) [3]. Specifically, we (i) classify the correlations between pairs of users' ratings to improve the recommendation accuracy and (ii) make SNRS robust against sparse data. Our experiments on real data confirm the effectiveness of our improvements over the original SNRS and a collaborative filtering approach.

2. BACKGROUND

2.1 Social Network Based Recommendation

The state-of-the-art SNRS [3] utilizes the information in social networks to improve the performance of traditional recommender systems. The idea behind SNRS is to use a simplified homogeneous social network as a Bayesian network. The probability distribution of the target user U's rating on the target item I is calculated as:

$$Pr(R_{UI} = k) = \frac{1}{Z} Pr(R_{UJ} = k | a_J = a_I) \times Pr(R_I = k)$$
$$\times \prod_{V \in N(U)} \frac{1}{Z_V} H_{UV}(R_{UI} - R_{VI}), \quad (1)$$

where a_I and a_J are the attribute values for item I and item J in a m dimensional attribute space A. For example, $a_I = \{Italian, Low\}$ are the attribute values for restaurant I in a restaurant rating system with $A = \{Cuisine, Price\}$. $Pr(R_{UJ} = k | a_J = a_I)$ is U's preference for items similar to I, i.e., the conditional probability that the target user U will give a rating k to a item J with the same attribute values a_I as item I. $Pr(R_I = k)$ is the general acceptance of the target item I, i.e., the prior probability that the target item I receives a rating value k. H_{UV} is a histogram recording the differences in ratings between users U and V. For example $H_{UV}(-1)$ is the number of items J such that $R_{UJ} = R_{VJ} - 1$. $N(U)$ denotes the immediate friends of U; influences of all users in $N(U)$ to U can be modeled as a product of the normalized histograms of individual friend

pairs: $\prod_{V \in N(U)} \frac{1}{Z_V} H_{UV}(R_{UI} - R_{VI})$, where Z_V is the normalizing constant for the histogram of each immediate friend pair. Z serves as the normalizing constant to guarantee $\sum_k (Pr(R_{UI} = k) = 1)$. The estimated value of R_{UI} is the expectation of the distribution $Pr(R_{UI} = k)$.

The drawback of Equation 1 is that it cannot be used if the target user has no immediate friends or the immediate friends have no ratings on target item. For this reason, SNRS uses an iterative classification method [10] to utilize friend-of-friends inference, based on the fact that estimating the class of an entity often depends on the class estimations of its neighbors. It predicts the user U's rating on each target item on a limited user set $N'(U) = U \cup N(U)$. Each iteration processes all users of $N'(U)$ at a random permutation. A user is skipped if lacking immediate friends' information. The process stops after M iterations or when no more updates happen in the current iteration.

Table 1: Ratings

Item	Price	R_{U_A}	R_{U_B}
I_1	High	5	4
I_2	High	4	3
I_3	Low	3	3
I_4	High	4	4
I_5	Low	4	5
I_6	Low	?	3
I_7	Low	?	5

Table 2: Conversion

Item	ΔR_{U_A}	ΔR_{U_B}
I_1	1	0
I_2	0	-1
I_3	-1	-1
I_4	0	0
I_5	0	1
I_6	?	-1
I_7	?	1

EXAMPLE 1. *Figure 1 and Table 1 show an example social network and the items rated by users U_A and U_B. Suppose that we are targeting user U_A's rating on item I_6. The limited user set $N'(U_A)$ contains users U_A, U_B, and U_C, since users U_B and U_C are U_A's immediate friends in the social network. Consider a random order (U_A, U_C, U_B) of $N'(U_A)$ in the first iteration. User U_A is firstly considered. Assume that only U_B has a rating on I_6. According to Table 1, the histogram of rating differences between U_A and U_B are $H_{U_A U_B}(0) = 2$, $H_{U_A U_B}(1) = 2$ and $H_{U_A U_B}(-1) = 1$. Therefore, $\prod_{V \in N(U_A)} = \frac{1}{Z_{U_B}} H_{U_A U_B}(R_{U_A I_6} - R_{U_B I_6}) = 0, 0.2, 0.4, 0.4, 0$ when $R_{U_A I_6} = 1, 2, 3, 4, 5$, $R_{U_B I_6} = 3$ and $Z_{U_B} = \sum H_{U_A U_B} = 5$. Hence, there are three possible ratings 2, 3 and 4 for $R_{U_A I_6}$. Then, user preference can be calculated using:*

$$Pr(R_{U_A J} = k | a_J = \{Low\}) = Pr(R_{U_A} = k) \times Pr(a_J = \{Low\} | R_{U_A} = k)$$

$$= \frac{|I(R_{U_A} = k)| + 1}{|I(U_A)| + n} \times \frac{|I(a_J = \{Low\}, R_{U_A} = k)| + 1}{|I(R_{U_A} = k)| + m},$$

where $n = 5$ and $m = 1$ represent the possible ratings and dimensionality of A, respectively. $I(Q)$ is the set of items which meet the conditions of Q. To be precise, $Pr(R_{U_A J} = k | a_J = \{Low\}) = 0.2, 0.4, 0.4$, when $k = 2, 3, 4$, after normalization. When user attributes are not available, item acceptance can be calculated using:

$$Pr(R_{I_6} = k) = \frac{|I(R_{I_6} = k)| + 1}{|I(I_6)| + m}$$

Thus, $Pr(R_{I_6} = k) = 0.25, 0.5, 0.25$ when $k = 2, 3, 4$ after normalization. Finally, the probability distribution that user U_A will give item I_6 a rating of k, after the first iteration can be obtained using Equation 1, i.e., $Pr(R_{U_A I_6} = k) = 0.0769, 0.6154, 0.3077$ when $k = 2, 3, 4$. The predicted rating is the expectation of the probability distribution which is 3.2308.
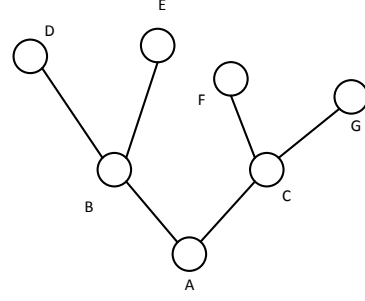

Figure 1: Example Social Network

2.2 Other Related Work

With the popularity of social networks, traditional recommender systems take advantage of additional information, such as user friendships and social influence, to improve their effectiveness. Social network-based collaborative filtering approaches [2, 6, 7] give higher weight to friends of users rather than considering a set of anonymous similar users. Matrix factorization based social recommender systems [8, 9] also leverage the information from social networks and take the influence from friends into consideration. Besides, there are some other approaches that consider social influence [3, 5, 11].

3. OUR PROPOSAL

In this paper we improve the SNRS described in Section 2.1 in two ways. First, we classify the correlations between pairs of users ratings to improve the accuracy of the system. Second, we make the system robust to sparse data, i.e., when there are few immediate friends with few common ratings with regard to the target user.

3.1 Classifying User Correlations

The correlations between the target user and his/her immediate friends learned from their common rated items are based on the assumption that each pair of friends behave consistently on reviewing the items. However, disagreement exists among friends in practice. We propose to use a Bipolar distribution to classify correlations between friends into two categories: LIKE and DISLIKE, such that conflict ratings (the target user likes item I, while one immediate friend dislikes it, or vice versa) can be ignored when doing recommendation. In addition, each user may have a personal rating preference. For instance, given a 5 scale rating system, a strict user may give ratings ranging from 1 to 3, while a lenient user usually offers ratings ranging from 3 to 5. If these two users are immediate friends, such different ratings may have a negative effect on each other's recommendation. Thus, we propose to use the differences between rating deviations from each user's average rating values when computing the correlations, instead of the original rating differences.

Specifically, the existing rating of each user U on each item I is firstly converted into the deviation ΔR_{UI} from the average rating value $\overline{R_U}$ for U and then the rated items for each user U are classified into S_U^{LIKE} and $S_U^{DISLIKE}$ categories. Note that ΔR_{UI} is rounded to the nearest integer. Formally,

$$S_U^{LIKE} = \{I_i | \Delta R_{UI_i} = R_{UI_i} - \overline{R_U} \geq 0\}$$
$$S_U^{DISLIKE} = \{I_i | \Delta R_{UI_i} = R_{UI_i} - \overline{R_U} < 0\} \tag{2}$$

The correlations between the target user U and all the immediate friends V are calculated as

$$\begin{cases} \prod_{V \in N(U)} \frac{1}{Z_V} H_{UV}^{LIKE}(\Delta R_{UI} - \Delta R_{VI}) & , \Delta R_{UI} \geq 0 \\ \prod_{V \in N(U)} \frac{1}{Z_V} H_{UV}^{DISLIKE}(\Delta R_{UI} - \Delta R_{VI}), \Delta R_{UI} < 0 \end{cases} \quad (3)$$

EXAMPLE 2. *Consider the social network and the ratings in Figure 1 and Table 1. Users U_A's and U_B's average rating values are 4 and 3.9, respectively. The converted ratings are shown in Table 2. The items rated by U_A and U_B are classified into two categories which are displayed in Table 3.*

Table 3: Item Categories

User	S^{LIKE}	$S^{DISLIKE}$
U_A	I_1, I_2, I_4, I_5	I_3
U_B	I_1, I_4, I_5, I_7	I_2, I_3, I_6

When estimating user U_A's rating on item I_6, we obtain $S_{U_A U_B}^{DISLIKE} = S_{U_A}^{DISLIKE} \cap S_{U_B}^{DISLIKE} = \{I_3\}$, since $\Delta R_{U_B I_6} = -1 < 0$ falls in the DISLIKE category. Then we have the histogram of converted rating differences between U_A and U_B are $H_{U_A U_B}^{DISLIKE}(0) = 1$. Therefore, $\prod_{V \in N(U_A)} \frac{1}{Z_V} H_{U_A V}^{DISLIKE} (\Delta R_{U_A I_6} - \Delta R_{V I_6}) = \frac{1}{Z_{U_B}} H_{U_A U_B}^{DISLIKE}(\Delta R_{U_A I_6} - \Delta R_{U_B I_6}) = 1$ when $\Delta R_{U_A I_6} = -1$, $\Delta R_{U_B I_6} = -1$ and $Z_{U_B} = 1$. At last, the probability distribution of U_A's rating on I_6 can be estimated using Equation 1.

Similarly, when estimating U_A's rating on I_7, we obtain $S_{U_A U_B}^{LIKE} = S_{U_A}^{LIKE} \cap S_{U_B}^{LIKE} = \{I_1, I_4, I_5\}$, since $\Delta R_{U_B I_7} = 1 > 0$ falls in the LIKE category. Then we have the histogram of converted rating differences between U_A and U_B are $H_{U_A U_B}^{LIKE}(-1) = 1, H_{U_A U_B}^{LIKE}(0) = 1, H_{U_A U_B}^{LIKE}(1) = 1$. Therefore, $\prod_{V \in N(U_A)} \frac{1}{Z_V} H_{U_A V}^{LIKE}(\Delta R_{U_A I_7} - \Delta R_{V I_7}) = \frac{1}{Z_{U_B}} H^{LIKE} (\Delta R_{U_A I_7} - \Delta R_{U_B I_7}) = 0.33, 0.33, 0.33$ when $\Delta R_{U_B I_7} = 0, 1, 2$, $\Delta R_{U_B I_7} = 1$ and $Z_{U_B} = 3$. At last, the probability distribution of U_A's rating on I_7 can be estimated using Equation 1.

3.2 Temporal Influence Links

In a social network, some users may have few immediate friends, and friends may have few common ratings; then, the user set $N'(U)$ (including the target user and the immediate friends) considered in each estimation iteration may be too small and the social influence may not be well considered in the recommendation. To address this data sparsity problem, we define temporal influence links in the social network. In each iteration, $N'(U)$ is extended to also include all users that can reach the target user via temporal influence links.

DEFINITION 1. *A temporal influence link $E(V, U)$ is a directed edge from user V to user U if and only if (i) U and V are immediate friends, (ii) they both rated at least one common item I, and (iii) V rated I (at timestamp t) before U did (at timestamp $t + \Delta t$).*

DEFINITION 2. *User V_0 can reach user V_n via temporal influence links if there exists a sequence of users $V_1, V_2, \cdots, V_{n-1}$, such that temporal influence links $E(V_i, V_{i+1}), i = 0, 1, \cdots, n-1$ exist.*

EXAMPLE 3. *Figure 2 shows exemplary temporal influence links in a social network. Edge $E(U_D, U_B)$ is an temporal influence link, since (i) users U_D and U_B are immediate friends, (ii) they both rated item I_4, and (iii) user U_D rated*

I_4 *(at timestamp t_2) before U_B did (at timestamp t_4). User U_H can reach user U_A via temporal influence links, since temporal influence links $E(U_H, U_F), E(U_F, U_C), E(U_C, U_A)$ exist. Suppose we are estimating user U_A's rating on an item. By considering the temporal influence links, the user set $N'(U_A)$ includes users $U_A, U_B, U_C, U_D, U_F, U_H$ in one iteration. Recall that in the SNRS system, only $U_A, U_B,$ and U_C are considered in one iteration.*
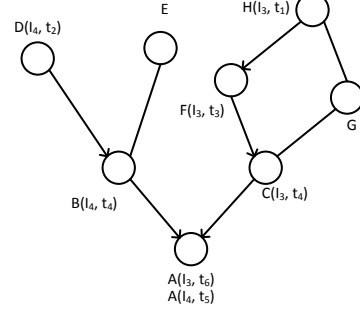

Figure 2: Example Temporal Influence Links

4. EXPERIMENTS

4.1 Dataset

We crawled a dataset from a real social network-based recommender system called Dianping[1], which is a leading local restaurant search and review platform in China. The dataset contains a social network of 11,352 users and users' ratings on 10,657 restaurants, where a 5-scale rating system is used. The statistics of the dataset are displayed in Table 4. Figure 3 illustrates the rating differences d between friends on restaurants in this dataset. For example, 248 for rating difference of -4 means that there are 248 pairs of rating records where a user gave a rating which is 4 less than the previous rating given by one friend on the same restaurant. Note that the counts for $\pm |d|$ are not equal since we consider the chronological order. From Figure 3 we can see that indeed disagreements may exist among friends, as we describe in Section 3.1. Each user in the dataset has rated at least one restaurant. Ratings are ordered in ascending order of the time when they are created. The $K\%$ most recent ratings from each users are held out for testing and the $(100\text{-}K)\%$ older ratings are used for training.

Table 4: General statistics of Dianping dataset

Statistics	Dianping
Users	11,352
Social Relations	759,289
Items	10,657
Ratings	501,472
Sparsity	99.59%

4.2 Performance Metrics

Most of the recommender systems adopt the coverage and accuracy metrics to evaluate performance [1]. Coverage measures the percentage of items for which a recommender system can make predictions. We use Mean Absolute Error (MAE), which is defined as the average absolute deviation of the predicted rating to the real rating, to measure the accuracy of our approach because it is widely used in the evaluation of rating-based recommendation [1,4]. We apply
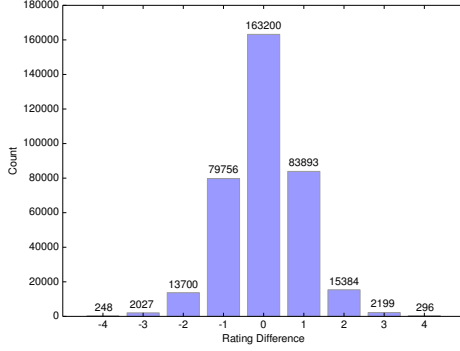
---

[1]http://www.dianping.com

Figure 3: Rating Difference Between Friends

a leave-one-out technique, which means hiding a real rating and trying to predict it in our test. The definition of MAE is as follows:

$$MAE = \frac{\sum_{u \epsilon U} \sum_{i \epsilon I} |R_{ui} - R'_{ui}|}{N} \qquad (4)$$

where R_{ui} is the real rating of target user u on target restaurant i, R'_{ui} is the predicted rating, and N denotes the number of tested ratings.

4.3 Competitors

We compare our proposed recommendation approaches to two methods:

- **CF**: standard user-based collaborative filtering algorithm using Pearson correlation to measure user similarity.
- **SNRS**: the social network-based recommender system [3] before applying our enhancements.

4.4 Results

The experimental results on Dianping dataset are displayed in Table 5. Approach SNRS* uses the classified correlations (Section 3.1) and Approach SNRS** incorporates both enhancements proposed in Section 3.

Table 5: Results on Dianping dataset

Training Data	Metrics	CF	SNRS	SNRS*	SNRS**
90%	MAE	0.8062	0.6941	0.5859	0.5910
	Coverage	0.7153	0.7033	0.7033	0.7416
80%	MAE	0.7900	0.6532	0.5584	0.5642
	Coverage	0.6204	0.7076	0.5917	0.6252

From the results, we can see that:

(1) SNRS* and SNRS** outperform CF (more than 26%) and SNRS (more than 13%) in terms of MAE, since classifying friends' correlations based on their rating histories and incorporating more social influences by the temporal influence links can help improving the accuracy of recommendation. SNRS also has lower MAE than CF, showing the effectiveness of social network-based recommendation. SNRS** is slightly worse than SNRS* on MAE showing that the expanding of the user set may have side effect, since some less similar users are taken into account in recommendation.

(2) As for coverage, SNRS has a stable performance of around 0.7. CF also has a good performance since it can always give prediction if the target item has at least one rater and the Pearson correlation similarity between this rater and the target user is calculable. The filtering of ratings where friends disagree is a possible reason why SNRS* and SNRS** do not have a good coverage compared to SNRS.

Also, the breaking of temporal influence links may result in the decrease of coverages of SNRS* and SNRS**, since the percentage of training data decreases. Still, SNRS** performs no worse than CF in terms of coverage. Compared to SNRS*, SNRS** has a better coverage, indicating that temporal influence links can incorporate more information that increases the estimation probability.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we extended a social network-based recommender system (SNRS) [3], which utilizes the information in social networks to improve the performance of traditional recommender systems. Two methods were proposed to improve performance. One is classifying the correlations between pairs of users ratings to improve the accuracy of the system. The other is making the system robust to sparse data, i.e., when there are few immediate friends with few common ratings with regard to the target user. Experiments on a real dataset demonstrate that our approach has a better performance compared to a collaborative filtering method and the original SNRS. In the future, we plan to further study the impact of social influence in recommendation and investigate ways to improve the coverage of our method when the data is sparse.

6. 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 TKDE*, 17(6):734–749, 2005.

[2] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *Australasian Conference on Artificial Intelligence*, 2010.

[3] J. He and W. W. Chu. A social network-based recommender system (snrs). In *Data Mining for Social Network Data*, pages 47–74. 2010.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM TOIS*, 22(1):5–53, 2004.

[5] M. Jamali and M. Ester. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *KDD*, 2009.

[6] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[7] F. Liu and H. J. Lee. Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37(7):4772–4778, 2010.

[8] X. Liu and K. Aberer. Soco: a social network aided context-aware recommender system. In *WWW*, 2013.

[9] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

[10] J. Neville and D. Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, 2000.

[11] Y. Pan, F. Cong, K. Chen, and Y. Yu. Diffusion-aware personalized social update recommendation. In *RecSys*, 2013.