ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-601

Social Recommendation with Interpersonal Influence

Junming Huang[1] and **Xue-Qi Cheng**[1] and **Jiafeng Guo**[1] and **Hua-Wei Shen**[1] and **Kun Yang**[2]

Abstract. Social recommendation, that an individual recommends an item to another, has gained popularity and success in web applications such as online sharing and shopping services. It is largely different from a traditional recommendation where an automatic system recommends an item to a user. In a social recommendation, the interpersonal influence plays a critical role but is usually ignored in traditional recommendation systems, which recommend items based on user-item utility. In this paper, we propose an approach to model the utility of a social recommendation through combining three factors, i.e. receiver interests, item qualities and interpersonal influences. In our approach, values of all factors can be learned from user behaviors. Experiments are conducted to compare our approach with three conventional methods in social recommendation prediction.Empirical results show the effectiveness of our approach, where an increase by 26% in prediction accuracy can be observed.

1 INTRODUCTION

Social recommendations are common activities in daily life, where a person (*sender*) recommends an *item* to another person (*receiver*). For example, a professor recommends a reference book to his student. Recently, social recommendations have also gained great success on online sharing and shopping services, such as the online sharing website Douban.com allowing users to recommend interesting books, movies and music to their friends and being favored by millions. As Sinha [26] points out, social recommendations with interpersonal influence are more appropriate than traditional recommendations. Social recommendation is becoming a popular online recommendation scheme beyond the traditional recommendation, where an automatic system recommends items to users.

As we can see, a social recommendation can be represented as a triple (*sender*, *receiver*, *item*), while the traditional recommendation focuses on the pair of (*receiver*, *item*). The involvement of sender and receiver brings *interpersonal influence* into a social recommendation, which makes it largely different from a traditional recommendation. Interpersonal influence actually plays a critical role. For example, a girl might not pick a dress recommended by her mother, but would say yes if her boyfriend likes it. Therefore, how to model the utility of social recommendations, the recommendations with interpersonal influence, becomes a natural question. With this problem addressed, we will not only be able to understand social recommendation activity better, but also investigate new applications in the future, e.g. to automatically recommend a product to a potential costumer by reminding him that his friend has already bought it.

However, existing approaches cannot be directly applied to address this problem as they only capture partial factors of a social recommendation. Standard recommendation approaches [16, 1, 21, 2, 17] focus on modeling utility of the pair (receiver,item), ignoring the factor of the sender. Trust-based recommendation approaches [5, 14, 7] also focus on the utility of the pair (receiver,item) by incorporating interpersonal trust. Those trust are obtained from either manual assignment[5, 14], which limits the scalability, or the correlations between (receiver,item) pairs[7], which do not actually reflect the interpersonal factor. Besides, innovation diffusion approaches [12, 6, 8, 9] focus on modeling the adoption propagation through the pairs of (sender,receiver), without taking items into account.

Inspired by the studies over traditional recommendations and innovation diffusion, we first introduce *social utility* of a social recommendation, which measures the usefulness of the item, recommended by the sender, to the receiver. Here we consider that the social utility is mainly affected by three kinds of factors, i.e. receiver interest, item quality and interpersonal influence. The first two factors describe the attributes of the receiver and the item, while the third describes the relation between the sender and the receiver. In this way, we define a social utility function to map those three factors to a real number. Specifically, we represent the three factors with vectors. Each vector contains same number of elements, and each element represents an aspect like topic or genre. The social utility is given by the norm of Hadamard product of the interest of receiver, the quality of item and the influence between sender and receiver.

Empirically we find that the tendency for a receiver to accept a social recommendation increases with its social utility. Therefore, we propose an approach to model social utility in the social recommendation data set, where each recommendation is labeled either "accepted" or "refused". We estimate factors by maximizing social utilities of accepted recommendations and minimizing those of refused ones. With all factors estimated, we can then predict social utility of an incoming social recommendation automatically.

Some might confuse the interpersonal influence with user similarity. However, they come with totally different mechanism. User similarity is content-dependent. Two people with similar tastes would buy same books on their own choice independently. In contrary, influence is content-free. The influence phenomenon relies on social relation rather than item content. Given that a person can affect another's action, they are not necessarily similar in interests, e.g. a fan is willing to buy whatever advertised by his idol.

We conduct experiments to demonstrate the effectiveness of our approach over social recommendation prediction. Three conventional methods are considered as baselines, i.e., standard and trust-based recommendation systems, and an innovation diffusion model. Empirical results show that our approach can obtain improved performance in social recommendation prediction, by increasing 26% in prediction accuracy compared with the baselines.

The rest of this paper is organized as follows. Related research work is introduced in section 2. A qualitative analysis on data is in section 3. Our model is formulated in section 4 and empirical results are reported in section 5. Conclusion comes in section 6.

2 BACKGROUND

2.1 Recommendation systems

Targeting at recommending to users items they might appreciate, a traditional recommendation system is required to solve the core problem formulated as follows. Consider the set U of all users and the set A of all items, a utility function $f : U \times A \to R$ must be built to measure how useful an item is to a user, where R is a simply ordered set, e.g. real or integer values in a given range. Then the recommending

[1] Institute of Computing Technology, the Chinese Academy of Sciences, email: {huangjunming,guojiafeng,shenhuawei}@software.ict.ac.cn, cxq@ict.ac.cn
[2] School of Computer Science & Electronic Engineering, University of Essex, email: kunyang@essex.ac.uk

task for a certain user $u \in U$ is to pick an item $a \in A$ to maximize $f(u, a)$. In most actual scenarios, values of $f(u, a)$ are accessible only on a subspace of $U \times A$, e.g., seldom a user rates all restaurants in a city. Thus a recommendation system has to extrapolate $f(u, a)$ values based on the known subspace.

As an early representative method, a ***content-based*** method represents items with content vectors and calculates similarity between a new coming item a' and all items a user u preferred to estimate $f(u, a')$ [1, 21]. The method requires items in a form of explicit content that can be parsed automatically, and thus limits the system from scenarios where items are difficult to parse, like music and movies.

Differently, ***collaborative filtering*** techniques are developed to calculate similarity with rating history rather than understanding of content. Such a system can be either *user-based*[2, 17] or *item-based*[23, 3]. Take the user-based systems for example. A user-based system recommends to a user u items preferred by similar users. User similarity can be calculated by a variety of approaches, including vector cosine between two users' history vector whose elements represent previous ratings on commonly viewed items [2, 17], and widely-used Pearson's correlation coefficient [19, 25] as follows,

$$sim(u, v) = \frac{\sum_{a \in A(u,v)} (r_{u,a} - \bar{r}_u)(r_{v,a} - \bar{r}_v)}{\sqrt{\sum_{a \in A(u,v)} (r_{u,a} - \bar{r}_u)^2 \sum_{a \in A(u,v)} (r_{v,a} - \bar{r}_v)^2}} \tag{1}$$

where $A(u, v)$ indicates the set of items that both users u and v have rated, $r_{u,a}$ is the rating u made on item a which can be valued binary (acceptance/refusal), integer or real , and $\bar{r}_u$ is the average rating u made on all items. The larger similarity two users have, the stronger tendency they have to act like each other. The utility $f(u, a')$ is estimated as the predicted rating $\hat{r}(u, a')$ with Resnick's formula [19] as follows

$$\hat{r}(u, a') = \bar{r}(u) + \frac{\sum_v sim(u, v)(r(v, a) - \bar{r}_v)}{|\sum_v sim(u, v)|} \tag{2}$$

More directly related to the problem issued in this paper are several recent research works on collaborative filtering techniques to incorporate *user trust* to replace or combine with user similarity to serve as weights in rating aggregation.

Golbeck [5] uses explicit interpersonal trust in an online social network to replace user similarity. Jamali and Ester [10] run a random walk algorithm on an explicitly expressed trust network to find proper items for each customer. Since trust values are manually input and thus suffer from sparsity problem, Ma et al [14] use matrix factorization to estimate missing trust values with user node degrees, while Massa and Avesani [15] estimates with trust propagation. However, all of above approaches heavily rely on manually assigned trust values, which severely limits their scalability.

O'Donovan and Smyth [18] calculate trust for each user by the average accuracy of using one's rating to predict another's. As we can see, the trust value actually reflects one's global authority on items rather than interpersonal influence.

He and Chu [7] calculate three factors to predict customer rating, including user preference, item's general acceptance and influence from social friends. However, their model still focuses on estimating the utility of a pair (receiver, item) rather than a social recommendation triple (sender, receiver, item). Furthermore, their influence is defined as a probability that two friends rate like each other. Thus it might be difficult for the model to distinguish the two phenomenons: (1) two users act similarly because of similar interest, and (2) a user has strong faith on his friend so he copies opinions from that friend. Note that the latter case shows right the power of interpersonal influence which will be addressed in our work.

2.2 Innovation diffusion models

Richardson and Domingos mine the mechanism in viral marketing [4, 20] that customers can differently influence their social contacts to buy same products. The customers with high social influence are preferred when a merchant delivers trial products to start a viral marking. Typical innovation diffusion models like *Linear Threshold Model* [6] and *Independent Cascade Model* [8, 9] are developed to model the dynamics that how information spread through social networks. In the spread of an innovation, an individual node is considered in either of two states, *active* or *inactive*, corresponding to adopting that innovation or not.

The *Linear Threshold Model* [6] assumes that each node v defines a set of normalized weights $b(u, v)$ on all of its in-edges (u, v) to indicate neighbor importance. It becomes active if the sum of weights from active neighbors is beyond a certain threshold θ_v, and keeps still otherwise. The *Independent Cascade Model* [8, 9] defines a probabilistic process in which each node u, after it is activated, has one chance to activate its inactive neighbor v with a probability $p_{u,v}$, which is defined on the edge (u, v) and independent from history.

Lescovec [13] measures actual recommendation diffusion on a person-to-person network. Pointing out that products of different categories are differently preferred, however, he does not go further to analyze attributes of single products and single users.

Innovation diffusion models are proven to reach accurate performance in simulating and predicting the propagation of innovation through social networks. However, most innovation diffusion models focus only on external motivations but ignore internal motivations, i.e. the attributes of users and items (e.g. item quality). Thus they cannot easily explain the cases that social recommendations on different items end with different results even through the same sender and receiver. Therefore, those models are rarely directly applied for recommendation systems.

3 PRELIMINARY STUDIES

3.1 Data collection

Before we dive into the technique details, we first conduct some preliminary studies on the social recommendation data set. We collect data from the online sharing website Douban.com, where a user can declare friendship with or become a fan of another user, collect interesting movies, books and music albums, or recommend them to his friends and fans. We crawled relationships and activities of all unique users in the hottest 15 discussion groups. If a user v is a fan of another user u, we build a directed edge (u, v) between them to indicate u could recommend items to v. If u and v are friends, two edges with opposite directions are built. When a user recommends a book, movie or music album to his friends or fans, we consider it as a social recommendation to them. We label the social recommendation "accepted" if the receiver collects that item later, or "refused" otherwise. We assume that all social recommendations are independent from each other, thus when a user gets multiple recommendations of an item before he collects it, we simplify the case as that the receiver refuses all earlier recommendations and accepts the latest one. Some statistics of the crawled data are described in table 1.

Table 1. Data description

Value	Description
35,211	number of users
231,600	number of directed edges
7,987	number of items
1,174,627	number of social recommendations
62,451	number of accepted social recommendations

3.2 Factor analysis

We qualitatively analyze the factors that might determine whether a social recommendation will be accepted. As is sufficiently discussed in earlier works, receiver interests and item qualities, which are usually combined as the utility of a pair (*receiver*, *item*), are believed to affect the recommendation result. However, it is still unclear how the interpersonal influence affects. Therefore, we design an experiment to gain some intuitive perception. For each social recommendation triple (sender u, receiver v, item a) in data, we calculate the traditional utility of pair (v, a) with average cosine similarity [22] between a and $A(v)$, the set of items that v has collected before.

$$Utility(v, a) = \frac{1}{|A(v)|} \sum_{a' \in A(v)} \frac{|C_a \cap C_{a'}|}{\sqrt{|C_a| \cdot |C_{a'}|}} \tag{3}$$

where C_a is the set of users that have collected item a. The above utility comes from the average similarity of item a to all items a' that user v collects. We also calculate the recommendation successful ratio from user u to user v to roughly estimate the interpersonal influence of (u, v),

$$Influence(u, v) = \frac{|Acc(u, v)|}{|Rec(u, v)|} \tag{4}$$

where $Rec(u, v)$ is the set of social recommendations that u has sent to v, and $Acc(u, v)$ its subset of accepted ones.

We analyze the relationship between recommendation results and the two factors, i.e. utility and influence. To avoid linear correlation between the recommendation results and the influence, we conduct an open test. We randomly select 90% of social recommendations as the training set to calculate utility and influence, and the rest are held out as the testing set. For each social recommendation in testing set, we plot it as a point w.r.t. its utility and influence in figure 1. Accepted recommendations are denoted as big blue circles while refused ones small red. Besides, we show the average values of utility (0.22) and influence (0.17) as dash lines to divide the figure into four sections, and report the acceptance ratio in each corner. As we see from the result, the acceptance ratio increases with influence, i.e. from 0.0985 to 0.2077 and from 0.1105 to 0.4000, indicating that larger influence encourages the receiver to accept a social recommendation. Especially, the acceptance ratio in the zone of high utility and low influence is relatively small, implying that a social recommendation is less likely to be accepted with low influence, even if it brings high utility. Therefore, we might conclude that with utility only we cannot sufficiently explain whether a social recommendation will be accepted. We need to leverage interpersonal influence to complete the mechanism.

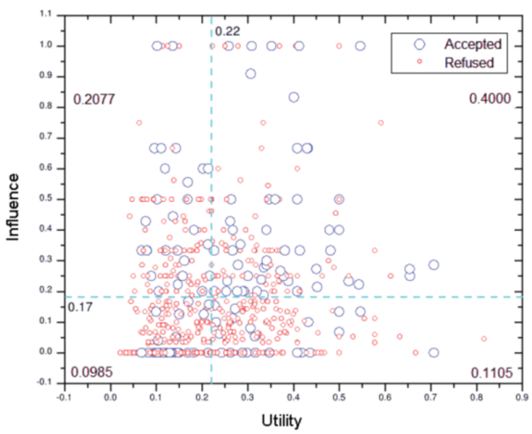

Figure 1. Utility and influence

4 OUR APPROACH

In this section, we propose an approach to model the utility of a social recommendation, named *social utility*, to represent the extent how a receiver will be willing to accept a social recommendation. In this way, we build our model based on the social recommendation data set, where each recommendation is labeled "accepted" or "refused", and predict the social utility for any incoming social recommendation.

First we introduce the social utility of a social recommendation. As we can see, a social recommendation is an activity where a person (*sender*) recommends a certain *item* to another person (*receiver*), which can be denoted by a triple (sender, receiver, item). Similar as that in the traditional recommendation scenario, we can also define a utility for the social recommendation, named as *social utility*. Social utility measures the usefulness of the item, *recommended by the sender*, to the receiver. Formally, consider the set U of all users and the set A of all items that might be recommended, the task to obtain the social utility then becomes to define the mapping function as $f : U \times U \times A \rightarrow R$, where R is a simply ordered set, such as real numbers with a certain range. Note that in traditional recommendation, the utility function is defined as $f : U \times A \rightarrow R$.

To build up the social utility function, let us consider a typical social recommendation scenario first. As a big fan of drama and fantasy movies, Alice heard that a drama and thriller movie *Twilight* had gained a box office record recently. Since she was not quite sure whether it was worth watching, Alice hesitated in making a decision to watch it until her friend Brown came to recommend it to her. Given that Brown had recommended several wonderful drama movies before, Alice was quite confident of his taste on movies of that genre and finally accepted his recommendation.

Reviewing this scenario, we find that three major factors contribute to the social utility of a social recommendation, which lead to Alice's final acceptance. First, Alice had *interests* in drama and fantasy movies. Second, the box office record showed the movie had high *quality* in drama and thriller, suggesting it worth watching for the fans of those genres. Third, Alice believed Brown's taste on dramas, whose *interpersonal influence* brought more confidence of the movie. Finally, the overlapped part (drama) of the three factors resulted in a high social utility which encouraged her final acceptance.

As we can see from the above analysis, the social utility function should be defined with three factors, i.e. receiver interest, item quality and interpersonal influence. The first two factors describe the attributes of the receiver and the item respectively, while the third describes the relation between the sender and the receiver. Moreover, each factor could be measured in different aspects, e.g. book topics or movie genres such as drama, fantasy and thriller.

Formally, we use u, v and a to denote the sender, receiver and item in a social recommendation respectively. We represent the *interest vector* of receiver v as I_v, the *quality vector* of item a as Q_a, and the *influence vector* from u to v as T_{uv}. Each vector contains D elements to indicate different aspects. All elements can be valued positive real. A greater value indicates higher interest, better quality or stronger influence in corresponding aspect. Take the above case for example: $D = 3$, $I_{Alice} = (1, 1, 0)$, $Q_{Twilight} = (1, 0, 1)$, $T_{Bob,Alice} = (1, 0, 0)$, where the aspects are drama, fantasy and thriller. In this way, the social utility function for the social recommendation (u, v, a) could be defined as follows

$$f(u, v, a) = \|I_v * T_{uv} * Q_a\| = \sqrt{\sum_{d=1}^{D} (I_{v,d} T_{uv,d} Q_{a,d})^2} \tag{5}$$

where $*$ indicates the Hadamard product. The above definition captures the intuition that a social utility increases with the aggregation over all aspects, and in each aspect increases with the joint value of the three vectors.

In order to constrain the result in a certain range of real numbers, we simply constrain that any element must be valued real in the range $[0, 1]$. We further constrain $\|I_v\| = 1$ for every receiver v since its values indicate how v's interests distributed over aspects. Therefore, we ensure that social utility will be valued in the range $[0, 1]$. The social utility will reach 1 if the influence and quality vectors maximize in all aspects corresponding to non-zero elements in interest vector, and 0 if minimize. It captures the intuition that a receiver will be most willing to accept a social recommendation if the most trustworthy friend recommends the best item in his favorite genres, and vice versa.

4.1 Learning

To predict the social utility of any incoming social recommendation, we need to first estimate values of all interest, quality and influence vectors based on training data. Unfortunately, those vectors cannot be simply explicitly extracted since all observations are results of coupling variables. Instead, we consider those vectors as hidden variables and use a machine learning method to estimate their values. For better scalability, no prior value are required.

Intuitively, a properly estimated model should simultaneously maximize the social utilities of accepted recommendations and minimize those of refused ones. Thus we estimate interest, quality and influence vectors by maximizing the product of $f(u, v, a)$ of accepted recommendations and $1 - f(u, v, a)$ of refused ones.

$$\{\hat{I}, \hat{Q}, \hat{T}\} = \arg\max_{\{I,Q,T\}} \prod_{(u,v,a)} f(u, v, a)^{\delta(u,v,a)} (1 - f(u, v, a))^{1-\delta(u,v,a)} \quad (6)$$

where the indicator function $\delta(u, v, a) = 1$ if user v accepts user u's recommendation of item a , and 0 otherwise. The product is over all recommendation triples (u, v, a) in the training data. To avoid floating-point overflow, we define the inference objective function as the logarithm of the above expression.

$$L = \sum_{(u,v,a)} (\delta(u, v, a) \ln f(u, v, a) + (1 - \delta(u, v, a)) \ln(1 - f(u, v, a))) \quad (7)$$

Values of all interest, quality and influence vectors are estimated by maximizing L with respect to them. We run a gradient ascent algorithm to gradually achieve maximum L. In each iteration, we calculate partial differentials of L with respect to each element in hidden variables as follows and update its value with a plus of that differential. The process ends when stable maximum L is reached.

$$\frac{\partial}{\partial I_{v,d}} L = \sum_{(u,v,a)} \frac{\delta(u, v, a) I_{v,d} T^2_{uv,d} Q^2_{a,d}}{p^2(u, v, a)} - \frac{(1 - \delta(u, v, a)) I_{v,d} T^2_{uv,d} Q^2_{a,d}}{f(u, v, a)(1 - f(u, v, a))} \quad (8)$$

$$\frac{\partial}{\partial T_{uv,d}} L = \sum_{(u,v,a)} \frac{\delta(u, v, a) I^2_{v,d} T_{uv,d} Q^2_{a,d}}{p^2(u, v, a)} - \frac{(1 - \delta(u, v, a)) I^2_{v,d} T_{uv,d} Q^2_{a,d}}{f(u, v, a)(1 - f(u, v, a))} \quad (9)$$

$$\frac{\partial}{\partial Q_{a,d}} L = \sum_{(u,v,a)} \frac{\delta(u, v, a) I^2_{v,d} T^2_{uv,d} Q_{a,d}}{p^2(u, v, a)} - \frac{(1 - \delta(u, v, a)) I^2_{v,d} T^2_{uv,d} Q_{a,d}}{f(u, v, a)(1 - f(u, v, a))} \quad (10)$$

The computational cost in the learning process increases with data size since every social recommendation must be counted in each iteration. We may reduce the cost by using cache or investigating an incremental algorithm in the future.

4.2 Prediction

For an incoming recommendation triple (u, v, a), our model calculate its social utility with equation (5). A social recommendation will be expected to be accepted if its social utility is larger than a certain threshold μ_p , and refused if smaller.

$$\hat{\delta}(u, v, a) = \begin{cases} 1, & f(u, v, a) \geq \mu_p, \\ 0, & f(u, v, a) < \mu_p. \end{cases} \quad (11)$$

The value of μ_p should be carefully selected in order to best distinguish accepted and refused recommendations. In our model, the value is selected when the model reaches the highest F1 measure in a cross validation experiment. Besides, in later experiments, we might also vary μ_p in the range $[0, 1]$ to make predictions tighter or looser, so as to access different precision-recall samples to draw a P-R curve.

5 EMPIRICAL RESULTS

We conduct experiments to compare performance of our model and conventional methods with the data described in section 3.1. For each method, 90% social recommendations are randomly selected as training set, and remaining are held out as testing set. Empirical results show that our model could better predict social recommendations.

5.1 Learning process

As discussed in section 4.1, we implement an iteratively gradient ascent algorithm to estimate hidden variables of interest, quality and influence vectors. We report L on training and testing set for every 10 iterations. As is shown in figure 2, L in the testing set increases consistently until convergence after around 2,000 iterations, indicating that our learning method has effectively and efficiently estimated hidden vectors.

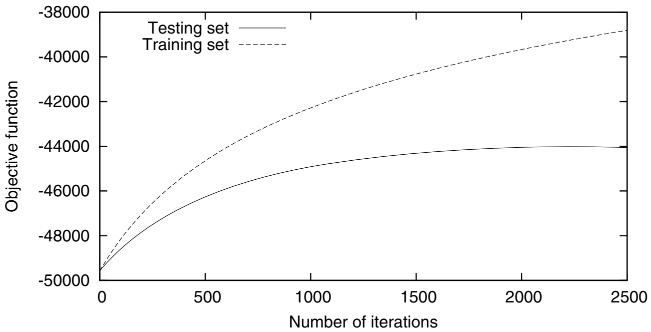

Figure 2. Objective function in learning process

We also evaluate the model accuracy with F1 measure on the testing set to check learning algorithm, which is the harmonic mean of *precision*, the ratio of true positives among all "predicted successful" recommendations, and *recall*, the ratio of true positives among all "actual successful" recommendations. When predicting, we simply take $\mu_p = \frac{1}{2}$. As is shown in figure 3, our learning algorithm succeeds in increasing F1 measure before reaching convergence after 700 iterations.

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (12)$$

To select a proper number of aspects, D, is another basic task in our learning process. Too few aspects limit the model to well distinguish interests and qualities in different aspects, while too many aspects could lead to overfitting. We train several models with different values of D and evaluate them by calculating their best F1 measure among all possible values of μ_p respectively. As is shown in figure 4, a model allowing multiple aspects could better fit data than one with single aspect, which could be considered as "do not distinguish

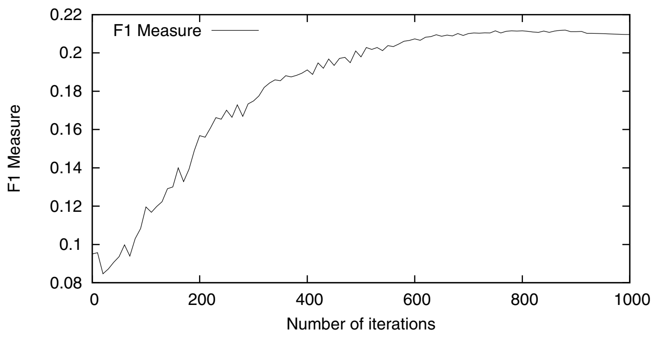

Figure 3. F1 measure in learning process

aspects". The single peak of F1 measure indicates the best value of *D*, which we use in following experiments.

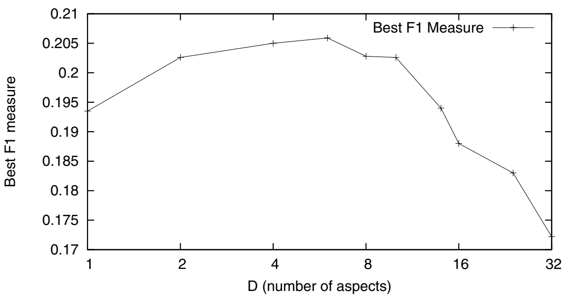

Figure 4. Selecting the number of aspects

5.2 Baselines

Three typical baselines are chosen as described in section 2: standard collaborative filtering method (represented as "Resnick" in legends), trust-based collaborative filtering method and independent cascade model. Firstly we prepare real ratings for the two collaborative filtering methods: an "accepted" social recommendation is considered with rating 1.0, and a "refused" one -1.0. For standard collaborative filtering method, we calculate the utility of $(receiver, item)$ with Resnick formula equation (2), where similar users are selected according to Pearson's correlation equation (1). For trust-based collaborative filtering, we calculate the utility of $(receiver, item)$ with CItem strategy provided in [18]. For Independent Cascade model, we estimate the acceptance probability of a social recommendation with $p_{sender,receiver}$, the successful ratio among all social recommendations through the edge $(sender, receiver)$ in the training set. Real predictions of those estimated models are finally turned into binary predictions by being compared with a certain threshold respectively, similar to equ (11).

5.3 Evaluation

We evaluate each model with the accuracy that it predicts whether a social recommendation in testing set will be accepted. The accuracy is usually measured with mean absolute error(MAE) or mean squared error(MSE) in recommendation systems. Since all recommendation labels in our data are binary instead of real ratings, those two metrics are both degenerated to zero-one loss, the ratio of incorrectly predicted recommendations. We also report precision, recall and F1 measure, which is usually used in information retrieval work. We make the predictions looser or tighter by varying thresholds of each model, which are used to turn a real prediction (utility or probability) into a binary choice, in order to get a best F1 measure and multiple precision-recall samples. The precision-recall curve shows how flexible a model is, which could be required to focus on high precision or high recall on request in various scenarios, for example, a product recommendation system should perform high precision if customers are impatient of advertisement, or high recall if customers do not want to miss any coupon.

Table 2. Prediction results

Method	F1 measure	Average 0-1 loss
Resnick	0.113	0.723
Trust-based	0.142	0.457
Independent Cascade	0.175	0.215
Influence-based	**0.221**	**0.125**

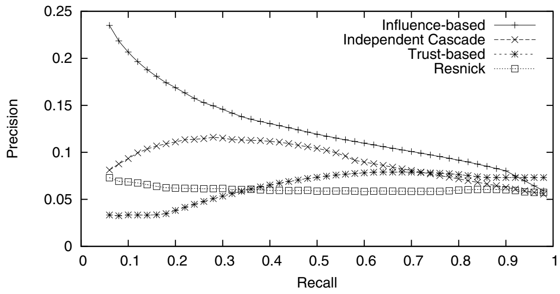

Figure 5. Precision-recall curve

As is shown in table 2, our approach, represented as "influence-based", has performed significantly leading accuracy upon baselines. The $0-1$ loss of our model drops by 41% from the closest baseline, and $F1$ measure increases by 26%. The precision-recall curve of our model in figure 5 is beyond baselines almost everywhere, indicating better accuracy on any requests, no matter tighter or looser.

Furthermore, to evaluate the distinguish ability of each model, we analyze the real prediction of each testing recommendation, i.e., predicted social utility in our model, predicted probability in Independent Cascade model, and predicted utility in two collaborative filtering systems. A T-test is performed to report the statistics significance between those real predictions of actually accepted and refused recommendations.

Table 3. Recommendation prediction statistics

Method	Average and variance of predictions		T-test
	accepted recommendations	refused recommendations	
Resnick	0.177(0.061)	0.456(0.525)	18.533
Trust-based	-0.156(0.536)	-0.525(0.609)	8.324
Independent Cascade	0.159(0.080)	0.047(0.032)	39.950
Influence-based	0.157(0.073)	0.016(0.008)	**72.641**

As is shown in table 3, the average social utility of accepted recommendations predicted by our model is clearly distinguishable from that of refused ones. The highest T statistics supports that the our model can best distinguish accepted social recommendations from refused ones.

Those results lend solid support that social recommendations could be better explained and predicted by taking into account all three kinds of factors: receiver interests, item qualities and interpersonal influences.

5.4 Case analysis

What is interesting in the data is that a user might receive several recommendations of the same item from multiple friends. Taking a case for example, the 62# user (let's call him Adam) has been recommended the same movie *Slumdog Millionaire* by 4191# user (let's call him Bob) and 271# user (let's call her Carol) in sequence. Adam took no action in a month after Bob recommended it, but collected the movie right after Carol made the same recommendation. That case is difficult to explain with collaborative filtering methods, which predict that Adam would always accept if the movie brings him utility large enough, or never accept it otherwise. Our model could explain that the interpersonal influence from Carol to Adam (0.52) was stronger than that from Bob (0.46), which made her recommendation greater social utility (0.82) than Bob's (0.65).

Table 4. A sample user Adam's reactions

Item recommended	*Slumdog Millionaire*	*Slumdog Millionaire*	*Mad Detective*
Recommended by	Bob	Carol	Carol
Adam's reaction	ignore	accept	ignore
Resnick utility	0.83	0.83	-0.27
Trust-based utility	-0.52	-0.52	-0.49
Independent Cascade probability	0.22	0.66	0.66
Influence-based social utility	0.65	0.82	0.77

Though the above case could also be explained with innovation diffusion models, they cannot perfectly explain the reason that Carol recommended another movie *Mad Detective* to Adam a month later but Adam refused it, since her two recommendations are undistinguishable with the same sender and receiver and thus should have same acceptance probability. In contrast, our model could tell the difference by item qualities and thus correctly predict the latter recommendation with a lower social utility (0.77) because the quality of the latter movie (0.45) is lower than the former one (0.51).

6 CONCLUSIONS

This paper proposes an approach to model the social utility of a social recommendation triple (*sender*, *receiver*, *item*). We identify that the social utility is mainly affected by three factors, i.e. receiver interest, item quality and interpersonal influence. By learning on binary labeled data, we are able to predict the social utility of any incoming social recommendation. Empirical results show our model improves recommendation prediction with an increase by 26% in prediction accuracy upon conventional approaches, i.e., standard and trust-based collaborative filtering techniques, and independent cascade model.

As the future work, we will further analyze the mechanism among those three major factors and try to turn our model into a practical online system to check its prediction ability in real world.

ACKNOWLEDGEMENTS

The work of this paper was funded by the National Natural Science Foundation of China under grants No 60873245, 60933005 and 60903139, the K. C. Wong Education Foundation of Hong Kong, and the Research Fund of ICT-CAS No 20097290.

REFERENCES

[1] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[3] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[4] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM.

[5] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *iTrust*, pages 93–104, 2006.

[6] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6):1420–1443, 1978.

[7] J. He and W. W. Chu. A social network-based recommender system. Technical Report 090014, Computer Science Department, UCLA, 2009.

[8] E. M. J. Goldenberg, B. Libai. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[9] E. M. J. Goldenberg, B. Libai. Using complex systems analysis to advance marketing theory development. *Academy of Marketing Science Review*, 2001.

[10] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 397–406, NY, USA, 2009. ACM.

[11] J.J. Jung. Visualizing Recommendation Flow on Social Network. *Journal of Universal Computer Science*, 11(11):1780–1791, 2005.

[12] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.

[13] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5, 2007.

[14] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 931–940, New York, NY, USA, 2008. ACM.

[15] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In *CoopIS/DOA/ODBASE (1)*, pages 492–508, 2004.

[16] G. Member-Adomavicius and A. Member-Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[17] A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 395–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[18] J. O'Donovan and B. Smyth. Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, New York, NY, USA, 2005. ACM.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM.

[20] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, New York, NY, USA, 2002. ACM.

[21] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[22] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[24] A. Seth and J. Zhang. A Social Network Based Approach to Personalized Recommendation of Participatory Media Content. In *ICWSM*, 2008.

[25] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[26] 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.