World Wide Web
https://doi.org/10.1007/s11280-018-0618-6

Evidence-driven dubious decision making in online shopping

Qiao Tian[1] · Jianxin Li[2] · Lu Chen[3] · Ke Deng[5] · Rong-hua Li[4] · Mark Reynolds[2] · Chengfei Liu[3]

Received: 7 December 2017 / Revised: 11 June 2018 / Accepted: 22 June 2018

Abstract Nowadays, Online shopping has been tremendous lifestyle choices due to the lower management cost for product/service providers and the cheaper prices for buyers/customers. Meanwhile, it raises a big challenge for both buyers and sellers to identify the right product items from the numerous choices and the right customers from a large number of different buyers. This motivates the study of recommendation system which computes recommendation scores for product items and filters out those with low scores. Recently, a promising direction involves the consideration of the social network influence in recommendation system. While significant performance improvement has been observed, it is still unclear to which extension the social network influence can help differentiate product items

This article belongs to the Topical Collection: *Special Issue on Social Computing and Big Data Applications*
Guest Editors: Xiaoming Fu, Hong Huang, Gareth Tyson, Lu Zheng, and Gang Wang

✉ Jianxin Li
jianxin.li@uwa.edu.au

Qiao Tian
tianqiao@hrbeu.edu.cn

Lu Chen
luchen@swin.edu.au

Ke Deng
ke.deng@rmit.edu.au

Rong-hua Li
rhli@szu.edu.cn

Mark Reynolds
mark.reynolds@uwa.edu.au

Chengfei Liu
cliu@swin.edu.au

1 College of Computer Science and Technology, Harbin Engineering University, Harbin, China

in terms of recommendation scores. This is an interesting problem in particular in the situation that the recommended product items have the highly similar (or identical) scores. As the first effort to this problem, this paper probes the boundary of social network influence to recommendation outputs by solving an optimization problem called *evidence-driven dubious decision making*. Two solutions have been proposed and the evaluation on two real world datasets has verified the effectiveness of the proposed solutions.

Keywords Collaborating Filtering · Social Influence · Recommendation

1 Introduction

With the fast development of online e-commerce nowadays, online shopping has been dominating the daily life of most peoples. Meanwhile, it raises a big challenge for both buyers and sellers to identify the right products from the numerous choices (e.g., books, movies or computers) and the right customers from a large number of different buyers. This motivates the study of recommendation system which narrows down the number of products for a particular buyer according to the buyer's preference (e.g., [9, 10, 13, 20]). One of the most popular recommendation techniques is collaborative filtering (CF) which, for a buyer, computes recommendation scores of product items by exploiting the purchasing history of many buyers.

Due to the commercial importance, the recommendation system has attracted significant attentions and the state-of-the-art is now beyond purchasing history. It has been recognized that a significant source of information to improve recommendation is the influence between users of social networks. The motivation is that peoples often share in social networks the user experience of purchased products. Recently, a great effort have been put to develop advanced collaborative filtering technique with the consideration of social network influence from different perspectives and significant improvements have been reported (e.g., [4, 7, 11, 13, 17, 18]).

However, the existing studies ignore a fundamental question, that is, to which extension the social network influence can help differentiate the recommended product items. Answering this question is critical in the situation that the recommended product items have similar (or identical) scores. Without a proper answer, a recommendation system has no evidence to evaluate the optimality of recommendations, for example, whether or not the recommended product items may have more difference in terms of recommendation scores by exploring influence of social networks.

[2] Department of Computer Science and Software Engineering, University of Western Australia, Perth, Australia

[3] Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, Australia

[4] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

[5] School of Science, RMIT University, Melbourne, Australia

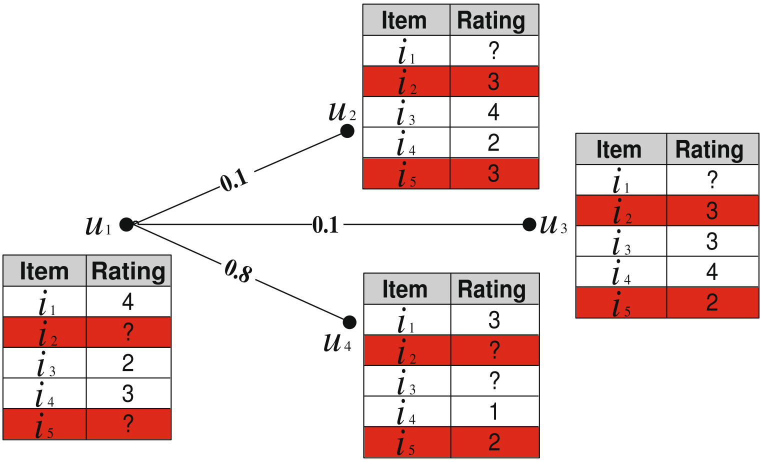


Figure 1 Example of small social graph

Figure 1 shows an example. Each user has rated the purchased product items and the influences between users in a social network are labeled on the edges between them. Assume u_1 is a query user who wants to buy i_2 or i_5. The recommendation score can be computed using collaborative filtering together with social influence. To do that, we first get CF scores $P^{cf}_{u_1,i_2} = 3$ and $P^{cf}_{u_1,i_5} = 2.56$. Then we consider the social network influence of u_1's neighbors, i.e., $P^{si}_{u,i} = \sum b_{uv} * p^{cf}_{v,i}$ where v is u's neighbor and b_{uv} is the weight of edge (u, v). So, the social network influence of u_1's neighbors on i_2 is $P^{si}_{u_1,i_2} = 0.1 * 3 + 0.1 * 3 + 0.8 * 2 = 2.2$. As such, the overall recommendation score with consideration of social network influence is $f(i, u) = P^{si}_{u,i} * P^{cf}_{u,i}$. We have $f(i_2, u_1) = 6.6$, $f(i_5, u_1) = 5.38$. But they are very similar. The question is whether it is possible to further differentiate the recommendation scores for i_2 and i_5 by exploring more social network influence information.

Motivated by the observation, this study aims to probe the boundary social network influence to enhance recommendations by solving an optimization problem called *evidence-driven dubious decision making*. Consider a social network $\mathcal{G}$ where each node is a user associated with purchasing history and each user-to-user link represents the interaction between them regarding purchase behaviors. Given a query user u, a set of query product items $\mathbb{I}$ and an integer k, the *evidence-driven dubious decision making* problem finds the maximum difference between product items in $\mathbb{I}$ in terms of recommendation scores when considering the social network influence from no more than k users. The contributions of this study are three-fold:

- To the best of our knowledge, this is the first study with aim to probe the boundary of social network influence in recommendation. It provides a benchmark to measure the effectiveness of recommendation system which explores social network influence.
- To achieve the aim, this study defines an optimization problem called *evidence-driven dubious decision making* which has been proved Np-hard.

– This study has developed two solutions to solve the *evidence-driven dubious decision making* problem and tests on real world datasets have verified the effectiveness of the proposed solutions.

The rest of this work is organized as follows. We first present the preliminaries in Section 3 and discuss the related work in Section 2. Section 4 formally defines the proposed problem. After that, we develop the solutions in Section 5. Finally, we present and discuss the experimental results in Section 6 and conclude the work in Section 7.

2 Related work

The recommendation system has been well studied in the past two decades and various methods have been developed (see surveys [8, 12, 14] for details). This section focuses on the recommendation techniques where the social network influence has been considered.

A line of research has used the social influence of users in social networks to improve the quality of recommendation. In [11], authors use the result of collaborative filtering prediction and social contagion outcome to compute the recommendation result. The paper focuses on the process of decision-making and only considers the user's neighborhood. In [13], authors propose a model-based recommendation method which also considers the social influence. The paper utilizes the *matrix factorization* and *learning to rank* techniques. Also, authors incorporate the social influence into the model. The paper mainly targets at the latent preference of users and updates to avoid the information overload problem in the social network sites. It first finds the valuable friends and then recommends them to the query user. The paper only finds the valuable friends and does not consider the effect of all the friends.

In [16], authors indicate that the collaborative filtering is suffering the issues of data sparsity and cold start. To address such issues, the collaborative filtering recommendation is improved by means of elaborately integrating the conventional rating data and the social trust network among the same users. In [15], cold start problem in collaborative filtering is studied based on a framework of tightly coupled collaborative filtering approach and deep learning neural network. A specific deep neural network is used to extract the content features of the items. The state of the art collaborative filtering model is modified to take the content features into prediction of ratings for cold start items.

In a recommendation scenario, the system must consider many different factors which may influence a user concurrently. Authors of [7] takes three factors into account, *receiver interests*, *item qualities* and *interpersonal influence*, to model the utility of a social recommendation. The paper uses machine learning method to derive the interpersonal influence based on the recommendation items across all users. However, it is difficult for a query user to know why he gets these recommended items and who in the social network are involved. In [17], authors discuss the differences between social influence and social correlation. Unlike [7], this paper proposes the probabilistic generative model to incorporate various information, including social influence, user behavior and item content. It selects the items with the highest probability to recommend to query user.

The above mentioned studies aim to optimize recommendation systems by considering the influence between users in social networks. However, it is still unclear to what extent the social network influence can help differentiate the recommended product items, in particular, in the situation that the recommended product items have similar (or identical) scores. To answer this fundamental question, this paper solves *evidence-driven dubious decision* *making* problem. So, this study is related but very different from all existing studies. Being a critical supplement, this study fills the gap of this important research field.

3 Preliminaries

3.1 Collaborative filtering (CF)

In a typical scenario, there is a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and a set of items $\mathcal{I} = \{i_1, i_2, \ldots i_n\}$. Each user u_i has a list of rated items $\mathcal{I}_{u_i}$. These data form a user-to-item rating matrix. The ratings can either be explicit or implicit. In this paper, we use a 1-5 scale.

CF is a prevalent research problem in recommendation system, and many works have been done in this field. It helps people make choices based on the preference history of other peoples. Generally, a CF algorithm assumes people have similar interests if they have similar profiles. The profile of a person can be represented as a rating matrix that this person rated on items [19]. Each element in the rating matrix is the rating value that users gave to the item, such as a book, a restaurant or a movie. The missing ratings indicate that user has not yet rated the item. The missing ratings can be predicted by the following steps: *Similarity Computation* and *Recommendation Computation* in [14].

Similarity Computation: We use $Pearson\ Correlation$ to measure the similarity between two users [3]. The similarity $w_{u,v}$ between users u and v is

$$w_{u,v} = \frac{\sum\limits_{i \in I} \left(r_{u,i} - \overline{r}_u\right)\left(r_{v,i} - \overline{r}_v\right)}{\sqrt{\sum\limits_{i \in \mathcal{I}} \left(r_{u,i} - \overline{r}_u\right)^2} \sqrt{\sum\limits_{i \in \mathcal{I}} \left(r_{v,i} - \overline{r}_v\right)^2}} \tag{1}$$

$w_{u,v}$ represents the similarity between users u and v. $r_{u,i}$ is the rating of user u on the item i and $\overline{r}_u$ is the average rating of the rated items of user u.

Recommendation Computation: This is a critical step for collaborative filtering [14]. We take a weighted average of all the ratings to predict the rating of a user u on an item i.

$$P_{u,i}^{cf} = \overline{r}'_u + \frac{\sum\limits_{v \in \mathcal{U}} \left(r_{v,i} - \overline{r}'_v\right) w_{u,v}}{\sum\limits_{v \in \mathcal{U}} \left|w_{u,v}\right|} \tag{2}$$

$\overline{r}'_u$ and $\overline{r}'_v$ are the average ratings for users u and v on all other rated items.

3.2 Top-*l* shortest path computation

The social relevance between users in a social network can be properly measured using the top-l shortest path distances. The breadth first search (BFS) and $Dijkstra$ algorithm can be applied to find the shortest paths between nodes in a graph. However, it is computationally expensive to compute top-l shortest paths online. To address this challenge, authors in [2] propose a framework to efficiently find the top-l shortest paths. The framework is based on 2-hop cover and an index with robust pruning scheme proposed in [1]. In this study, we apply this method to find the top-l shortest paths for evaluating the influence between users.

4 Problem statement

Definition 1 (Information Network) An information network is defined as an undirected graph $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{I})$ where $\mathcal{U}$ is the set of nodes to represent users, $\mathcal{E}$ is the set of edges which denotes interactions between users, $\mathcal{I}$ is the set of items to represent product items such as books, movies or computers.

Definition 2 (Rating Matrix) Given m users $\mathcal{U} = \{u_1, \ldots, u_m\}$ and n items $\mathcal{I} = \{i_1, \ldots, i_n\}$, we have a rating matrix $\mathcal{R} \in \mathbb{R}^{m \times n}$ where $\mathcal{R} = \left\{\mathcal{R}_{ij}, \text{if } u_i \text{ rated on } i_j\right\}$.

$\mathcal{R}_{ij}$ is the rating that user u_i rated on the item i_j. e.g., $\mathcal{R}_{22}$=3 denotes the rating of u_2 on i_2 is 3. The value in rating matrix may be the scores given actually by users to the items or, if missing, it is estimated by using collaborative filtering. The rating can either be explicit, such as a 1-5 scale in Netflix, or implicit such as purchases or clicks. In this paper, we use 1-5 scales.

Definition 3 (Social influence) Given a query user $u_q \in \mathcal{U}$ and a set of query items $\mathbb{I}$, the top-l shortest path distances are used to measure the influence of another user in $\mathcal{U}$ to u_q. Let $\mathcal{SP}$ be the set of paths between u_q and a user $v \in \mathcal{U}$, $|sp|$ be the length of a path $sp \in \mathcal{SP}$. The social influence of v to u_q on item $i \in \mathbb{I}$ is:

$$P^{si}_{u_q,i}(v) = \sum_{sp \in \mathcal{SP} \wedge |\mathcal{SP}| \leq l} \frac{1}{|sp|} \times pre(v, i). \tag{3}$$

l is an application parameter which is indicated by user. That is, only the top-l shortest paths are considered in the social influence. Given users v_1 and v_2, if v_1 has shorter top-l shortest paths than v_2, then v_1 has higher influence to u_q than v_2.

$pre(v, i)$ denotes the prestige of user v on item i which is defined as:

$$pre(v, i) = pre(v) \times pre(i). \tag{4}$$

$pre(v, i)$ consists of two factors, user's rating activeness $pre(v)$ and the actual rating ratio of i, $pre(i)$. $pre(v)$ = (number of $v's$ rated items) / (number of all users' rated items);

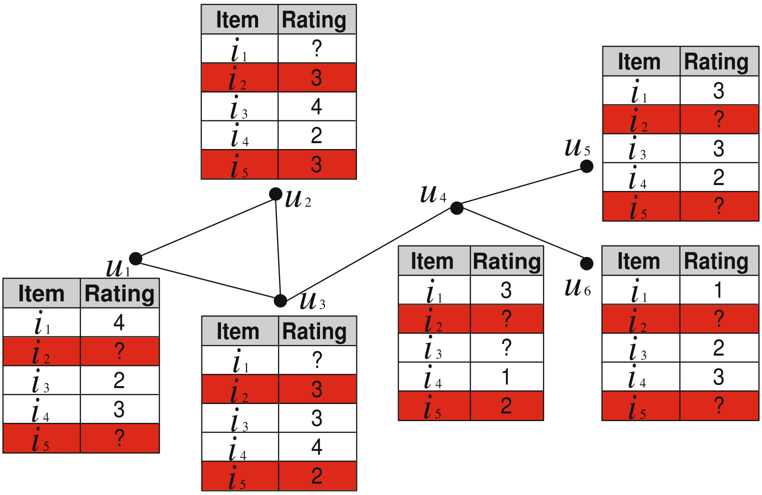


Figure 2 A small social graph with online ratings

$pre(i)$ = (number of users rated on i) / (number of users rated on all the items). In general, the rating follows the Poisson distribution [6]. As shown in (5), λ is the average of the number of all users' rated items or the average of the number of users rated on all items. t is the number of items rated by current user or the number of users rated on the current item.

$$P(t) = \frac{\lambda^t}{t!} e^{-\lambda}, t = 0, 1, \ldots \tag{5}$$

The Poisson distribution is considered to obtain $pre(v)$ and $pre(i)$. Also, we normalize the value of $pre(v, i)$ to range [0, 1].

Figure 2 shows a social graph with ratings. We compute $pre(v, i)$ using (4) and (5). Taking $pre(u_2, i_2)$ as an example. $\lambda_u = (3+4+4+3+3+3)/6 = 3.33$, u_2 actually rated on 4 items, so $t = 4$. Then we get $P(4) = 0.183$, and $pre(u_2) = 0.183/(0.220*4+0.183*2) = 0.147$ as defined above. Similary, we can get $pre(i_2) = 0.147/(0.147 + 0.195 + 0.195 + 0.156 + 0.104) = 0.184$, and normalize the value of $pre(v, i)$ to range [0, 1], we get $pre(u_2, i_2) = 0.270$. Poisson Distributions for different t are shown in Table 1.

Given a set of users $\mathbb{S} \subset \mathcal{U}$, the overall social influence to u_q on item i is obtained by integrating their social influences. If a user v in $\mathbb{S}$ rates on i, then its actual rating is used; otherwise, its influence to u_q on i is ignored by setting to be 1.

$$P^{si}_{u_q,i}(\mathbb{S}) = \sum_{v \in \mathbb{S}_i} P^{si}_{u_q,i}(v) \times P^{cf}_{v,i} + \sum_{v \in \mathbb{S} \setminus \mathbb{S}_i} 1. \tag{6}$$

Definition 4 (Social Recommendation Score) For an item $i \in \mathbb{I}$, social recommendation score $f(i|u_q, \mathbb{S})$ is the value that u_q rates i by concurrently considering the ratings and the social influence of other users. Formally,

$$f(i|u_q, \mathbb{S}) = P^{cf}_{u_q,i} \times P^{si}_{u_q,i}(\mathbb{S}). \tag{7}$$

$P^{cf}_{u_q,i}$ is the estimated rating of u_q to i considering the ratings of other users only as defined by (2); $P^{si}_{u_q,i}(\mathbb{S})$ is the social influence of other users to u_q on item i as defined in 6.

Table 1 Poisson distributions for examples in Figure 2

t	$P(t)$	$pre(v)$
0	0.036	0.029
1	0.119	0.096
2	0.198	0.159
3	0.220	0.177
4	0.183	0.147
5	0.122	0.098
t	$P(t)$	$pre(i)$
0	0.018	0.023
1	0.073	0.092
2	0.147	0.184
3	0.195	0.245
4	0.195	0.245
5	0.156	0.196
6	0.104	0.130

Definition 5 (Evidence-driven Dubious Decision Making Problem) Given a query user u_q, a set of query items $\mathbb{I}$, and a parameter k on the information network $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{I})$, the Evidence-driven Dubious Decision Making problem finds a subset of users $\mathbb{S}$ from $\mathcal{U}'$ that can maximally differentiate the social recommendation scores of the query user u_q on the set of query items, formally

$$\Delta = \arg \max_{\mathbb{S} \subseteq \mathcal{U}', |\mathbb{S}| \leq k} \sum_{i,j \in \mathbb{I}, i \neq j} \left| f(i|u_q, \mathbb{S}) - f(j|u_q, \mathbb{S}) \right| \tag{8}$$

where $\mathcal{U}'$ is a subset of $\mathcal{U}$ and each user in $\mathcal{U}'$ has actual rating on at least one query item.

Property 1 The evidence-driven dubious decision making problem is NP-hard.

Proof We reduce our dubious decision making problem to a decision problem. If the decision problem can be solved in polynomial time, then the dubious decision making problem can be solved in P; otherwise, the dubious decision making problem is NP-hard. Given an instance S, we first check whether $|S| <= k$. If it is, we calculate the differentiation score using $\sum |f(i|u_q, S) - f(j|u_q, S)|$ according to (8). After all instances are calculated, we can solve the dubious decision making problem by selecting the instance with the maximum score. However, the number of instances generated is $O(k^n)$. So the decision problem cannot be solved in polynomial time. So, the property is proved. □

The notations used in this paper are presented in Table 2.

5 Our proposed solutions

This section introduces two solutions to solve the evidence-driven dubious decision making problem.

5.1 *h*-hop based exact solution

The exact solution of the evidence-driven dubious decision making problem is designed by taking three steps as follows.

Table 2 Notations used in this work

Notations	Descriptions
$\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{I})$	Information Network, the set of users, edges and items
$\mathbb{I}$	The set of the query items
$\mathbb{S}$	The set of the selected users
$\mathcal{R}$	The set of ratings
u_q	The query user
f	The social recommendation score
$P^{cf}_{u_q,i}$	The predictive rating for u_q on item i from the set $\mathbb{I}$
$P^{si}_{u_q,i}(\mathbb{S})$	The social influence of u_q on item i by the set $\mathbb{S}$

- First, we pre-compute the recommendation ratings for the missing data in the rating matrix using the collaborative filtering method [14] as introduced in Section 3.1.
- Then, the users who actually rate at least one of the query items are identified. Note that "actually rate" means a user has actual rating scores, other than the predicted rating scores. For each such user, the shortest paths to the query user is computed and the one within h hops is retained. Note h is used to control the search space. After that, the top-l shortest paths from each retained user to the query user are computed using the method in [2] as introduced in Section 3.2.
- Finally, the k best users which maximize the objective function in (8) are reported. This is done by checking all possible combinations of k users among all retained users.

The example in Figure 2 shows the procedure of the h-hop based exact solution where u_1 is the query user, and i_2 and i_5 are the query items. The values of $pre(v, i)$ have been presented in Table 1. According to collaborative filtering, $P^{cf}_{u_1,i_2} = 3.00$ and $P^{cf}_{u_1,i_5} = 2.56$. When h=1 and $k = 2$, it requires to check all possible combinations of three candidate sets, i.e., any two of the three users $\{u_1, u_2, u_3\}$. If l is specified as 1, then $P^{si}_{u_1,i_2}(u_3) = 1/1 * 0.270 = 0.270$ using (3) where $pre(u_3, i_2) = 0.270$. We further integrate social network influence of all users in $\mathbb{S}$ using (6). Based to (7) and (8), the differentiated score is $\Delta = 0.252$ for $\{u_2, u_3\}$. Similarly, we get the differentiated scores of $\{u_2, u_4\}$ and $\{u_3, u_4\}$ as 1.55 and 2.48 respectively. So, $\mathbb{S} = \{u_3, u_4\}$ is the solution of the evidence-driven dubious decision making problem. The time complexity of the proposed exact solution is $(\frac{|\mathcal{U}|}{k})(|\mathcal{U}|+|\mathcal{E}|)lk(\frac{|\mathbb{I}|}{2})$ where $(\frac{|\mathcal{U}|}{k})$ is the time cost to calculate the possible candidate sets; $(|\mathcal{U}|+|\mathcal{E}|)l$ is the time cost to compute the top-l shortest paths for each vertex pair; k the the upper bound of $|\mathbb{S}|$, and $(\frac{|\mathbb{I}|}{2})$ is the possible size of calculating the differentiation scores for a given set $\mathbb{I}$ of product items.

Algorithm 1 Pruning-based Advanced Solution

Require:
  $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{I})$;
  a query user u_q with query item set $\mathbb{I}$ and parameters h, l, and k;
Ensure:
  Result Heap H={(item, score)};
  a set $\mathbb{S}$ of k selected users;

1: $\mathbb{S} \leftarrow$ Select k nearest users in $\mathcal{G}$ by using BFS method;
2: Initialize H and Δ_0 based on $\mathbb{S}$ using (8);
3: Update $(f_i^{upper}, f_i^{lower})$ for each item $i \in \mathbb{I}$, initialize the current hop number h_0, and flag = false;
4: **while** !flag **do**
5:     **for** each pair of items $i, j \in \mathbb{I}$ and $i \neq j$ **do**
6:         **if** $\Delta_0 < \sum_{i,j\in\mathbb{I},i\neq j} \left| f(i|u_q, \mathbb{S}) - f(j|u_q, \mathbb{S}) \right|$ **then**
7:             flag = false;
8:             Process the next user using BFS method;
9:             Update Δ_0, $\mathbb{S}$, H, h_0, and the inverted node lists;
10:             Break;
11:         **else**
12:             flag = true;
13: **return** H and $\mathbb{S}$;

5.2 Pruning-based advanced solution

The *h-hop based Exact Solution* has three major issues. The first issue is that the suitable size of h is hard to specify. If we remove the constraint of h, the size of generated k-sized possible combination set is too large to be practical in finding the best k users. The second issue is that it is unnecessary to compute all users' top-l shortest paths. This is because many candidates are not in the result set. The third issue is that it is unnecessary to generate the huge number of k-sized combinations. The reason is that some combinations are irrelevant to the solution.

This section provides an advanced solution which can properly address the three issues. The idea is to explore the nearest neighbors of the query user which can be obtained by using the breadth first search (BFS) algorithm. If there are some neighbors with the actual rating score about one query item, then we put them into the result candidate set. Meanwhile, we calculate the maximum differentiated score using (8). Repeating the process until we find the first k users in the neighborhood of the query user. After that, we check an early stop condition before we go ahead in exploring other neighbors. A good early stop condition should be able to safely stop the breadth first search algorithm as early as possible and output the results immediately once the stop condition is met.

In this work, we use the inverted node list as index to maintain the item-to-user ratings. Given an item, we can find a list of users with actual rating scores and predicted scores, and sort them by scores. As such, the head and tail of the list provide the upper bound value and lower bound value of collaborative filtering scores for users regarding the corresponding item. Similarly, we have the upper bound value and lower bound value of $pre(.)$.

As we discussed before, the social influence can be measured by the top-l shortest paths from a user to the query user. From the strategy of BFS algorithm, it is easy to derive the upper bound value and lower bound value of the top-l shortest path distances for a user that has not been seen yet. Assume the current depth is h_0. We can have that the upper bound value of the top-l shortest path distances is $\frac{l}{h_0+1}$, and the lower bound value is $\frac{1}{h_0+1}$.

By integrating the above three components (i.e., upper and lower bound of collaborative filtering scores, $pre(.)$, and the top-l shortest path distances), we can get the global upper bound value f_i^{upper} and lower bound value f_i^{lower} that a user can make contribution for an item i. The detailed procedure is provided in Algorithm 1. The bounds lead to an early stop condition.

Property 2 (Early Stop Condition) Assume the current candidate set is $\mathbb{S}$, the current visited hop number is h_0, and the current differentiated score is Δ_0. The BFS can be stopped safely if $\Delta_0 \geq \sum_{i,j \in \mathbb{I}, i \neq j} \left| f(i|u_q, \mathbb{S}') - f(j|u_q, \mathbb{S}') \right|$ where $\mathbb{S}'$ is a k-sized subset of users in $\mathbb{S} \cup \{u_x\}$. Here, u_x is a virtual node and its contribution to an item i is bounded by f_i^{upper} and f_i^{lower}.

In Figure 3, assume u_1 is the query user, i_2 and i_5 are query items, $k = 1$ and $l = 1$. At the 1^{st} iteration, the pruning-based solution exploits the direct neighbors $\{u_2, u_3\}$. u_3 is the best choice with its differentiated score Δ_0=0.587. The visited u_2 and u_3 are removed from the inverted node lists. Now, the current hop number h_0 is 1. Using Property 2, we check if the early stop condition holds at this moment, i.e., computing the two extreme conditions $f_{i_2}^{upper}$, $f_{i_5}^{lower}$ and $f_{i_5}^{upper}$, $f_{i_2}^{lower}$. Thus, we have $\Delta_{i_2^U - i_5^L} = 0.434 < \Delta_0$ and $\Delta_{i_5^U - i_2^L} = 0.753 > \Delta_0$. So the algorithm cannot terminate. At the next iteration, u_4 is visited with its differentiated score 1.89, which is larger than that of u_3. So the best choice is u_4 now. The

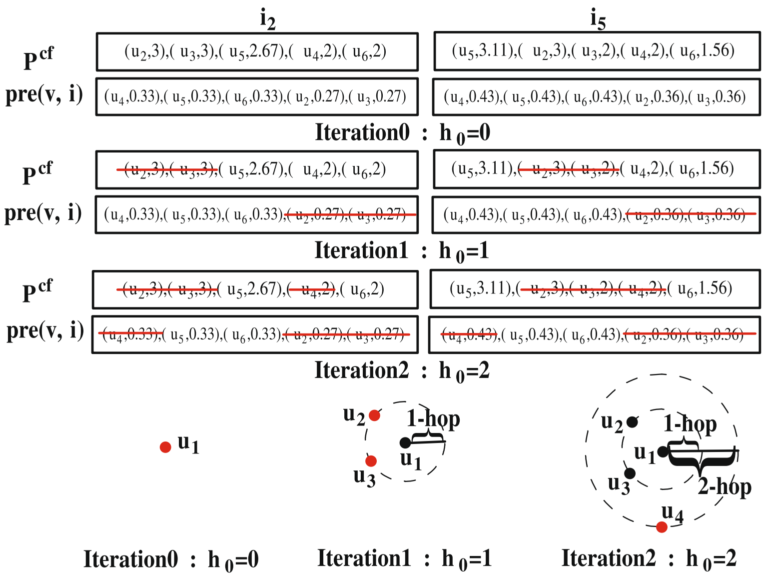


Figure 3 Early stop condition

current $\Delta_0 = 1.89$ and $h_0 = 2$. Based on Property 2, the algorithm can be safely stopped without further computation because $\Delta_{i_2^U - i_5^L} = 0.288 < \Delta_0$ and $\Delta_{i_5^U - i_2^L} = 0.501 < \Delta_0$; otherwise, the above process will be repeated.

6 Experimental study

To verify the performance of our algorithms, we conduct extensive experiments with the baseline algorithm BF and the pruning-based algorithm PruneAlg. These algorithms are implemented in Java and the experimental evaluation was conducted on a machine with a quad-core Intel i7-4870HQ, 3.7GHz processor, 16GB of memory, and macOS Sierra installed.

6.1 Experiment settings

The accuracy of our proposed methods are evaluated using two real world datasets from:

- Epinions[1] is a platform for customers to put their reviews and to help people determine buying decisions.
- Filmtrust[2] [5] is an online film website for users to rate movies and establishes their trust relationships each other. The statistic information is shown in Table 3.

[1] http://www.trustlet.org/downloaded_epinions.html

[2] https://www.librec.net/datasets.html#filmtrust

Table 3 Statistics of the datasets

Dataset	Size	Users	Items	Ratings	Avg. ratings per user
Epinions	8.4M	49288	139738	664824	16
Filmtrust	387KB	874	1309	35497	24

Dataset	Avg. number of connected vertices	Avg. number of rated times per item
Epinions	14216	4
Filmtrust	426	17

We conduct experiments using different settings of $|\mathbb{I}|$, $|\mathbb{S}|$ and l, i.e., the query item size, the selected user set size, and the number of shortest paths. The query users and query items are randomly selected. The configurations are shown in Table 4.

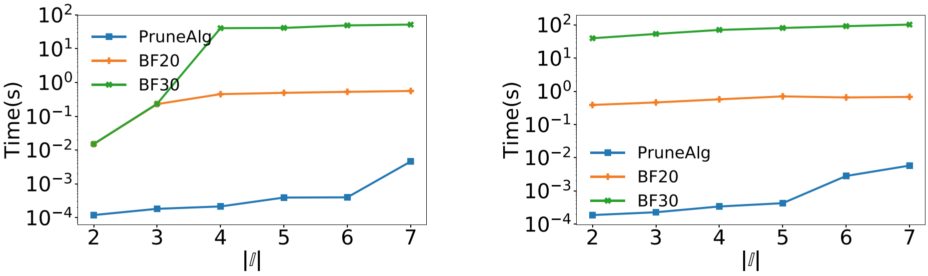
6.2 Evaluation of efficiency

We notice that the algorithm BF runs extremely slow in both real datasets since it has to evaluate overwhelming number of combinations. As a result, we report the efficiency of BF when it selected nearby users only, that is, users closest to the query user. BF20 means BF searches the optimal result for a query user from its 20-closest users. BF30 means BF searches the optimal result for a query user from its 30-closest users. On the other hand PruneAlg can perform well with the complete search space, that is, all connected users of the query user. Therefore, we report the efficiency of PruneAlg when it searches the optimum result from the complete search space.

Varying Query Item Size Figure 4a and b compare the query efficiency under different query item sizes. In both datasets, the time costs increase as we enlarge the number of query items for all algorithms, however, PruneAlg runs several orders of magnitude faster than BF20 and BF30. PruneAlg is slightly more sensitive to the changes of the query item size. For Epinions dataset, when the query item size is small, the time costs of BF20 and BF30 are almost identical for the following reasons. First, when query item size is small, the number of users rated the query item is small, which makes the number of users they search over tend to be small. Second, Epinions dataset has a characteristic that users who are close tend to rate similar items. As a result, the users to be sought are more likely close to the query users. As the query item size increases to 4, the time cost difference between BF20 and BF30 becomes clear. Filmtrust dataset does not have similar phenomenon as Epinions dataset. This is because with Filmtrust dataset each user rates much more items and each item is rated by much more users on average. That means even when the query item size is small, the total number of users rating the query item in Filmtrust dataset is much higher than that in Epinions dataset. Because of that, the optimal results for Filmtrust dataset usually cannot be found from the users close to the query user.

Table 4 Parameter settings

Parameter	Range	Default value
$\lvert\mathbb{I}\rvert$	2, 3 ,4, 5, 6, 7	5
$\lvert\mathbb{S}\rvert$	4, 6, 8, 10, 12, 14, 16	10
l	2, 3, 4, 5, 6, 7	5



(a) Epinions (b) Filmtrust

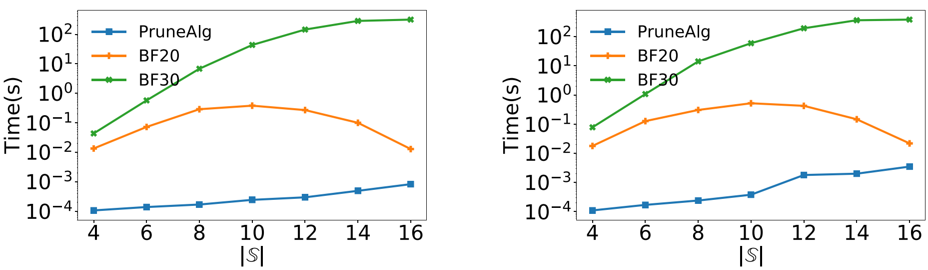
Figure 4 Time cost when varying $|\mathbb{I}|$

Varying Selected User Size Figure 5a and b show the query efficiency when varying selected user size, i.e., the number of selected users in $\mathbb{S}$. The time costs of BF20 and BF30 are exponential to the number of selected users before the number reaches 10 in Epinions dataset and 15 in Elimtrust dataset. After the number reaches 10 in Epinions dataset and 15 in Elimtrust dataset, time costs of BF20 and BF30 remains the same in large. This is because the maximum number of combinations evaluated by BF20 C_{20}^{10} and by BF30 evaluates is C_{30}^{15} in theory. Compared to BF20 and BF30, PruneAlg is much less sensitive to the number of selected user and it scales well when the number increases for both datasets.

Varying *l* Figure 6a and b show the efficiency as l changes. For both BF20 and BF30, the time costs are very insensitive to l. This is because they only search selected users over users close to the query user. However, as l increases, users that are distant to the query user may become as competitive as the users nearby the query user, whereas neither BF20 nor BF30 considers the users far away from the query user. This is why the time cost of PruneAlg is more sensitive to l than BF20 and BF30. The PruneAlg is about 5 orders of magnitude faster than BF20 and BF30 for both datasets.

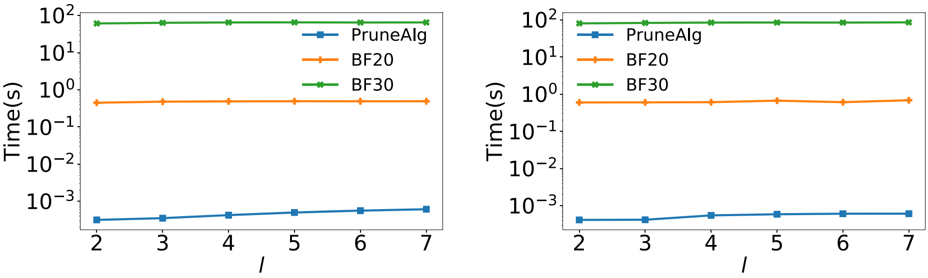
6.3 Evaluation of effectiveness

In theory, the selected users in BF20 and BF30 are the local optimal solution only, whereas, the selected users derived by PruneAlg are the global optimum solution. Therefore, in this



(a) Epinions (b) Filmtrust

Figure 5 Time cost when varying $|\mathbb{S}|$



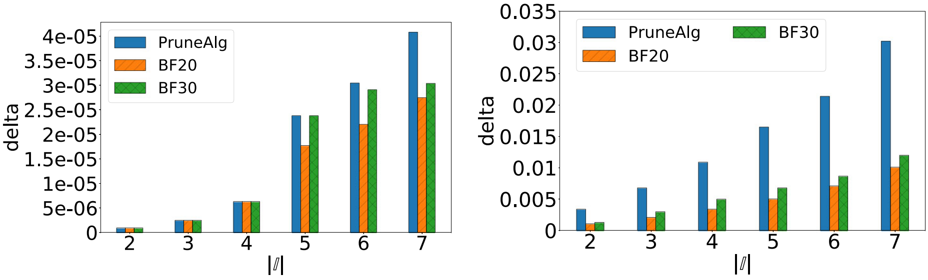
(a) Epinions (b) Filmtrust

Figure 6 Time cost when varying l

section, we only demonstrate the effectiveness by showing how the Δ score changes as configuration varies.

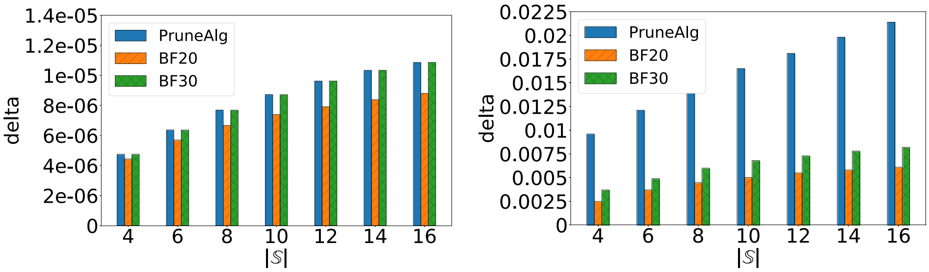
Varying Query Item Size Figure 7a and b report Δ scores when increasing the query item size. For both datasets and both algorithms, Δ scores increase as the size of query item increases. For Filmtrust dataset, the effectiveness of PruneAlg is better than that of BF20 and BF30. The superiority becomes more obvious as the size of query item increases and its effectiveness is almost three times better than that of BF20 and BF30 when $\mathbb{I}$ becomes 7. For Epinions dataset, the effectivenesses of BF20 and BF30 are very close to that of PruneAlg. This is because users rating the same items tend to be close with each other whereas Filmtrust does not have such characteristic. However, the effectiveness of PruneAlg is clearly superior than that of BF20 and BF30 when the size of query items is greater than 5. Its effectiveness is over 20% better than BF20 and BF30 when $\mathbb{I}$ is 7 on Epinions dataset.

Varying Selected User Size Figure 8a and b show the Δ scores as the size of selected users changes. For both datasets, the Δ scores increase when selecting more users for all algorithms. Interestingly, for Epinions, the effectiveness of BF30 and PruneAlg are identical. There are two reasons. First, we pre-filtered the zero-scored results in the plots, otherwise, on average, the effectiveness of BF20 will be very low. Second, for Epinions dataset, the users close to each other tend to rate similar items than other users. In such a situation, given a query user, when query item size is small, it can find the optimum results by searching its



(a) Epinions (b) Filmtrust

Figure 7 Differentiated score when varying $|\mathbb{I}|$



(a) Epinions (b) Filmtrust

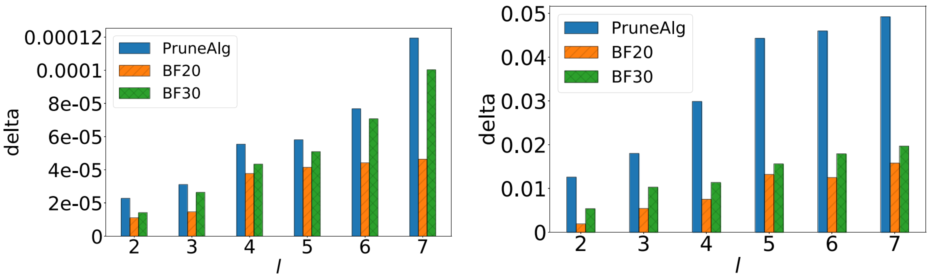
Figure 8 Differentiated score when varying $|\mathbb{S}|$

nearby users. For Filmtrust dataset, there is no such property. The effectiveness of PruneAlg is significantly better than that of BF20 and BF30. The superiority becomes increasingly obvious when selecting more users. Its effectiveness is over three times better than that of the other two algorithms when selecting 16 users.

Varying l Figure 9 demonstrates Δ scores for different l values. In both datasets, Δ score increases as l grows. For Epinions dataset, the effectiveness of PruneAlg is superior over BF30 slightly in all settings while the effectiveness of PruneAlg and BF30 are much better than that of BF20 if l is greater than 5. For Filmtrust dataset, the effectiveness of PruneAlg is much better than that of BF20 and BF30. And such superiority becomes more clear when l increases. PruneAlg performs much better than the other two methods due to (1) the social relationships in Epinions dataset are more loose than Filmtrust data on average; and (2) the nearby users in Epinions dataset tend to rate similar items while in Filmtrust dataset users rating the same item tend to distribute uniformly.

6.4 Evaluation of precision

Processing Datasets In both datasets, we select a number of users and, for each query user, the actual rating on some query items are deleted. By solving the evidence-driven dubious decision making problem, these deleted ratings are estimated as recommendation scores where the difference between these recommendation scores are maximized. We randomly select 100 users in Epinions and Filmtrust dataset respectively.



(a) Epinions (b) Filmtrust

Figure 9 Differentiated score when varying l

Table 5 Result of precision evaluation

Dataset	Perfect match ratio	Avg.number of inversions
Epinions	73.5%	2.33
Filmtrust	81.32%	1.62

Evaluation Metrics We use two metrics to evaluate the experiment results where the sequence order of the actual ratings are used as the ground truth.

The first metric is *perfect match ratio* defined as follows:

$$Perfect\ Match\ Ratio = \frac{the\ number\ of\ exact\ match\ results}{the\ total\ number\ of\ selected\ users} \tag{9}$$

The *number of exact match results* means the number of the users for whom the sequence order of recommendation score is exactly same as the actual rating sequence order. The higher ratio indicates the better performance can be achieved if social network influence is considered.

The second metric is the *average number of inversions*. For a user, let item $i_1, i_2, \ldots, i_n$ be the actual rating sequence order where suppose $i_h < i_j$. In the recommendation score sequence, if $f_h > f_j$. The item pair (i_h, i_j) is called an inversion. For example, if the actual rating sequence order is i_1, i_2, i_3 and the recommendation score sequence order is i_2, i_1, i_3, the number of inversions is 1. The average number of inversions is the sum of inversions of all the users divided by the total number of queries. A smaller value indicates the better performance can be achieved if social network influence is considered.

Evaluation Result The tests results based on the above two metrics, i.e., *perfect match ratio* and *average number of inversions*, are reported in Table 5. We can see that the social network influence in Epinions dataset is likely to contribute more in recommendation than that in Filmtrust dataset.

7 Conclusion

This work sheds light to understand the impact of social network influence in the field recommendation. We have proposed and addressed a significant problem, *evidence-driven dubious decision making*. It probes the boundary that social network influence can contribute to recommendation. With the flourish of social networks and the recommendation system in e-commerce, this boundary is essential since it is the benchmark to evaluate the method which exploring social network influence in recommendation. In this study, we point out this problem is NP-hard and two solutions have been developed. Their performance have been verified on two real world datasets.

Acknowledgements This work was partially supported by the ARC Discovery Projects under Grant No. DP160102114 and DP160102412.

References

1. Akiba, T., Iwata, Y., Yoshida, Y.: Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In: Proceedings of the ACM SIGMOD international conference on management of data, SIGMOD, 2013, New york, June 22-27, 2013, pp. 349–360 (2013)
2. Akiba, T., Hayashi, T., Nori, N., Iwata, Y., Yoshida, Y.: Efficient top-k shortest-path distance queries on large networks by pruned landmark labeling. In: Proceedings of the Twenty-Ninth AAAI, conference on artificial intelligence, January 25-30, 2015, Austin, pp. 2–8 (2015)
3. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI '98 Proceedings of the Fourteenth conference on uncertainty in artificial intelligence, University of Wisconsin Business School, Madison, July 24-26, 1998, pp. 43–52 (1998)
4. Chen, C.C., Shih, S.-Y., Lee, M.: Who should you follow? Combining learning to rank with social influence for informative friend recommendation. Decis. Support. Syst. **90**, 33–45 (2016)
5. Guo, G., Zhang, J., Yorke-Smith, N.: A novel bayesian similarity measure for recommender systems. In: Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI), pp. 2619–2625 (2013)
6. Hsu, C.-N., Chung, H.-H., Huang, H.-S.: The hybrid poisson aspect model for personalized shopping recommendation. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, pp. 545–548 (2003)
7. Huang, J., Cheng, X., Guo, J., Shen, H., Yang, K.: Social recommendation with interpersonal influence. In: ECAI 2010 - 19th European conference on artificial intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings, pp. 601–606 (2010)
8. Koren, Y., Bell, R.: Advances in collaborative filtering. Recommender systems handbook, pp. 78–118 (2015)
9. Lee, J., Lee, D., Lee, Y.-C., Hwang, W.-S., Kim, S.-W.: Improving the accuracy of top-n recommendation using a preference model. Inf. Sci. **348**, 290–304 (2016)
10. Li, X., Xu, G., Chen, E., Li, L.: Learning user preferences across multiple aspects for merchant recommendation. In: 2015 IEEE international conference on data mining, ICDM 2015, Atlantic city, November 14-17, 2015, pp. 865–870 (2015)
11. Shang, S., Hui, P., Kulkarni, S.R., Cuff, P.W.: Wisdom of the crowd. In: Corporating social influence in recommendation models. arXiv:1208.0782 (2012)
12. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix A survey of the state of the art and future challenges. ACM Comput. Surv. **47**(1), 3 (2014)
13. Shih, S.-Y., Lee, M., Chen, C.C.: An effective friend recommendation method using learning to rank and social influence. In: 19th Pacific Asia conference on information systems, PACIS 2015, Singapore, July 5-9, 2015, pp. 242 (2015)
14. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv Artificial Intellegence **2009**, 421425:1–421425:19 (2009)
15. Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 29–39 (2017)
16. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017)
17. Ye, M., Liu, X., Lee, W.-C.: Exploring social influence for recommendation: a generative model approach. In: The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, August 12-16, 2012, pp. 671–680 (2012)
18. Zhang, J., Liu, B., Tang, J., Chen, T., Li, J.: Social influence locality for modeling retweeting behaviors. In: IJCAI Proceedings of the 23rd international joint conference on artificial intelligence, Beijing, August 3-9, 2013, pp. 2761–2767 (2013)
19. Zhao, Z.-D., Shang, M.: User-based collaborative-filtering recommendation algorithms on hadoop. In: Third international conference on knowledge discovery and data mining, WKDD, 2010, Phuket, Thailand, 9-10 January 2010, pp. 478–481 (2010)
20. Zhao, Z., Hanqing, L., Cai, D., He, X., Zhuang, Y.: User preference learning for online social recommendation. IEEE Trans. Knowl. Data Eng. **28**(9), 2522–2534 (2016)