Interactive Social Recommendation

Xin Wang
Tsinghua University
xin_wang@tsinghua.edu.cn

Steven C.H. Hoi
Singapore Management University
chhoi@smu.edu.sg

Chenghao Liu
Zhejiang University
twinsken@zju.edu.cn

Martin Ester
Simon Fraser University
ester@cs.sfu.ca

ABSTRACT

Social recommendation has been an active research topic over the last decade, based on the assumption that social information from friendship networks is beneficial for improving recommendation accuracy, especially when dealing with cold-start users who lack sufficient past behavior information for accurate recommendation. However, it is nontrivial to use such information, since some of a person's friends may share similar preferences in certain aspects, but others may be totally irrelevant for recommendations. Thus one challenge is to explore and exploit the extend to which a user trusts his/her friends when utilizing social information to improve recommendations. On the other hand, most existing social recommendation models are non-interactive in that their algorithmic strategies are based on batch learning methodology, which learns to train the model in an offline manner from a collection of training data which are accumulated from usersfi historical interactions with the recommender systems. In the real world, new users may leave the systems for the reason of being recommended with boring items before enough data is collected for training a good model, which results in an inefficient customer retention. To tackle these challenges, we propose a novel method for interactive social recommendation, which not only simultaneously explores user preferences and exploits the effectiveness of personalization in an interactive way, but also adaptively learns different weights for different friends. In addition, we also give analyses on the complexity and regret of the proposed model. Extensive experiments on three real-world datasets illustrate the improvement of our proposed method against the state-of-the-art algorithms.

1 INTRODUCTION

Recommender systems have become a hot research topic in academia and been widely adopted in industry as well. Moreover, the rising of social networks and rapid development of web services actuate the emergence of recommendation in social media. People not only rate movies or TV series on IMDB, but also interact with each other on Facebook and see the latest updates of their favorite idols on Twitter. This brings the idea of social recommendation which tries to utilize available information (e.g., ratings) from users' friends to infer their preferences. Lots of existing work [1–12] has proved that incorporating social information from social networks does help to improve the accuracy of conventional recommendation methods [16–18]. At the same time, as more and more web service providers begin incorporating social elements into their services, social recommendation has also become a well studied topic in which most of their algorithmic strategies are to learn the model off-line via batch learning without any interactions from users. The training data in batch learning is normally obtained through the accumulation of users' historical interactions with the recommender systems, which may run the risk of users in real world leaving the systems because of many boring items being recommended to them before enough data is collected for training a good off-line model, resulting in inefficient customer retention. Besides, although social information from friends has been proved to be very useful for the improvement of recommendation accuracy, some of these friends may share similar preferences with the target user while others may be totally irrelevant for recommendations because of domain differences. This poses two challenges to us: first, how can we provide good-quality recommendations as soon as possible even when the target user has little past behavior data in order to maximize user retention in social recommendation; second, how to dynamically learn different weights for different friends which can best serve the recommendation accuracy when receiving more and more feedback from users.

To handle the first challenge, multi-armed bandit (MAB) serves as a competent candidate for recommendation with user interactions given its capability of simultaneously exploiting existing information that matches user interest and exploring new information that can improve global user experience, which is known as the *exploitation-exploration* trade-off dilemma. Thus casting the mechanism of multi-armed bandit into social recommendation can help mitigate the dilemma of user retention. A significant amount of work has been done on stochastic multi-armed bandit algorithm to provide principled solutions to the exploitation-exploration dilemma [51, 53, 55, 56]. In addition to the vanilla stochastic linear bandit models, contextual bandit algorithms [23, 34, 43, 50] become promising solutions when side information like contextual content (e.g., texts, tags, etc.) about users and items is available in scenarios such as mobile recommender systems [36], news recommendation [45] and display advertising [39, 44]. In general, the multi-armed bandit based algorithms try to get a good understanding of user preferences and thus achieve a high-quality recommendation as soon as possible through collecting a small amount of interactive feedback (e.g., behaviors such as ratings, clicks and favorites etc.)

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'17, November 6–10, 2017, Singapore.
© 2017 ACM. ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3132880

from users. We will give a detailed description on how multi-armed bandit can be incorporated into social recommendation later.

As for the second challenge, it is also necessary to exploit and explore the extend to which the current user trusts her friends when utilizing social information to improve recommendations. Since our goal is to adaptively learn the weights of different friends as more and more user interactive feedback becomes available, we employ a modified multi-arm bandit schema to dynamically update these weights upon receiving new feedback from users after they interact with the systems (e.g., give feedback such as clicks or ratings).

On the other hand, all the contextual bandit models we mentioned utilize content data such as tags and texts to construct an explicit feature vector (for each user and item) which will be used to determine the expected reward of the bandit. In practice, it is not always the case that the content data used to extract user and item feature vectors can be easily obtained, which makes the contextual bandit algorithms ineligible for producing accurate recommendations. Inspired by Zhao et al.'s work [33] and Qin et al.'s work [32], we borrow the idea from matrix factorization [16] which factorizes the observed user feedback into latent feature vectors in order to address our social recommendation problem in the scenario where there is no content information to construct explicit feature vectors and only user feedback (e.g., ratings, clicks, bookmarks etc.) can be observed. We employ the factorized latent user and item feature vectors to represent content information, extend the classical matrix factorization and combine it with the contextual multi-armed bandit in social recommendation.

In summary, we make the following contributions.

- We propose a novel interactive social recommendation model (ISR) which differs from and is superior to previous work in the following aspects.
  (1) Previous work on social recommendation [1–12] does not consider interactive learning.
  (2) Given a user in social recommendation, some existing methods simply compute the weights (i.e., degree of trust) for his/her friends uniformly (i.e., give equal weight to every friend) [1, 2, 4], which is a suboptimal solution because of the domain differences. Some others obtain these weights by calculating the rating similarities between the given user and his/her friends [2, 7], which is in a static way as well.
  This being the case, our solution is novel in the sense of adaptively learning these weights.
- We give a rigorous regret analysis to show that part of our proposed interactive social recommendation model has a regret bound of $O(\sqrt{T})$.
- We conduct extensive experiments on three real-world datasets and demonstrate the improvement of our proposed ISR model against the state-of-the-art methods.

2 RELATED WORK

There has been no shortage of existing work on *social recommendation* whose appearance should be attributed to the advent of social networks. As the rich information on social network becomes available [13, 15], social recommendation which makes use of social information from social networks to enhance recommender systems has attracted lots of attention from researchers due to the encouraging improvement (particularly for cold-start users) obtained against its non-social counterpart. Indeed, we are inevitably much easier to be influenced by our friends than strangers to change habits, adopt novel technologies, accept new ideas etc. Therefore, the purpose of social recommendation [1–8, 10–12] is to utilize the information of social influence inferred from social networks to help boost the performance of traditional methods such as collaborative filtering in recommender systems. More specifically, Ma et al. [1] propose a probabilistic matrix factorization model which factorizes user-item rating matrix and user-user linkage matrix simultaneously. They later present another probabilistic matrix factorization model which aggregates a user's own rating and her friends' ratings to predict the target user's final rating on an item. In [4], Jamali and Ester introduce a novel probabilistic matrix factorization model based on the assumption that users' latent feature vectors are dependent on their social ties'. Last but not least, Wang et al. [12, 14] propose to distinguish different tie types in social recommendation through presenting a method which can simultaneously classify strong and weak ties in a social network with respect to optimal recommendation accuracy as well as learn the latent feature vectors for users and items.

There always exists a trade-off between utilizing the information available so far (exploitation) and acquiring new knowledge (exploration). This kind of problems has been widely studied extensively in many fields such as *Machine Learning*, *Theoretical Computer Science*, *Operations Research* etc. This mature, yet very active, research area is known as "multi-armed bandit" in literature [44, 46, 48, 49].

Being first introduced by Robbins [58], multi-armed bandit is able to provide us with a clean, simple theoretical formulation for analyzing the trade-off between exploration and exploitation, and thus has been widely utilized by researchers to solve the challenges in balancing the trade-off faced by exploitation-exploration problems. We refer readers to [30, 51, 54] for a more general treatment.

Depart from the conventional stochastic multi-armed bandit [51, 55, 56], contextual bandit algorithms [20–25, 28, 29, 31, 35, 36, 38, 42, 45, 47, 50, 52] have attracted lots of attention from researchers because they have achieved much more promising performance than their context-free counterparts. Contextual bandit settings normally assume that the expectation of the reward (also known as payoff) for an action of a user on an item is a linear function of the corresponding context, e.g., the dot product of user feature vector and item feature vector[50], which gives much flexibility for different choices of expected reward function. For instance, Chu et al. [38] and Li et al. [45] use ridge regression to calculate the expectation and confidence interval of the reward of an action. In particular, methodology for the unbiased evaluation of context bandit algorithm is introduced in [42]. Besides the linear reward, Filippi et al. propose a parametric bandit algorithm for non-linear rewards. Later, a general approach to encoding prior knowledge for accelerating contextual bandit learning is introduced in [35] through employing a coarse-to-fine feature hierarchy which dramatically reduces the amount of exploration required. Bouneffouf et al. [36] investigate exploitation and exploration dilemma in mobile context-aware recommender systems and present an approach to the adaptive balance of exploitation/exploration trade-off regarding the target user's situation. By utilizing a Gaussian process kernel and

taking context into consideration, Vanchinathan et al. [29] introduce a novel algorithm that can efficiently re-rank lists to reflect user preferences over the items displayed. Moreover, a contextual combinatorial bandit that plays a "super arm" at each round is proposed by Qin et al. [32] to dynamically identify diverse items which new users are very likely to be fond of. Tang et al. explore ensemble strategies of contextual bandit algorithms to obtain robust predicted click-through rate of web objects [31], and later they propose a parameter-free bandit strategy which uses online bootstrap to derive the distribution of predicting models [25]. Recently, a combination of linear bandit with cascade model is introduced in [21] to deal with the large-scale recommendation and the dynamical pattern of reward as well as the context drift in the course of time is taken into account to formulate a time varying multi-armed bandit by Zeng et al. [22].

Others have also explored another variant which is designed to model dependency in the bandit setting [20, 23, 24, 27, 28, 34, 37]. In particular, authors in [27, 34] conduct investigations about contextual bandit with the probabilistic dependencies of context and actions being taken into consideration. Gentile and Li et al. [24, 28] investigate adaptive clustering algorithms based on the learnt model parameters for contextual bandit under the assumption that content is recommended to different groups (clusters) of users such that users within each group (cluster) tend to share similar interest, followed by Zhou and Brunskill who propose a contextual bandit algorithm that explores the latent structure of users through learning the distribution of users over different (fixed number) latent classes to make personalized recommendations for new users [20].

There is also some work that incorporates matrix factorization into the bandit setting [26, 33], among which Kawale et al. [26] employ Thompson sampling to perform online recommendation and Zhao et al. [33] propose an interactive collaborative filtering method based on probabilistic matrix factorization. We remark that neither of these models takes social information into consideration.

Cesa-Bianchi et al. [34], Wu et al. [23] and Wang et al. [19] study the bandit setting where information from social networks is taken into account. Specifically, Cesa-Bianchi et al.'s model utilizes a graph Laplacian to regularize the model so that users and their friends have similar bandit parameters and Wu et al.'s model, on the other hand, assumes the reward in bandit is generated through an additive model, indicating that friends' feedback (reward) on their recommendations can be passed via the network to explain the target user's feedback (reward). Wang et al. examine the bandit setting from another view through combining it with matrix completion. However, all the proposed models in [19, 23, 34] assume the weights for different friends to be fixed, without learning these weights adaptively to best serve the recommendation accuracy. Besides, their focus is orthogonal to ours in this work as they are based on explicit features whose model formulations and experimental settings are different from those based on latent features.

3 MULTI-ARMED BANDIT METHODOLOGY IN RECOMMENDATION

In practice, we often face many situations where it is necessary to find a balance between exploiting our current knowledge and obtaining new knowledge through searching unknown space. Take recommender systems as an example, ultimately we would like to recommend "good" items to users with the best knowledge we have so far as well as explore users' other interests which we have no idea about through exposing some "random" items to them and observing their corresponding reactions to these random recommendations. As is discussed previously, multi-armed bandit is adequate as an appropriate solution for this exploitation-exploration dilemma. In this section, we will give a mathematical description of the general idea for multi-armed bandit (MAB) strategy in the context of recommender systems, as well as several existing multi-armed bandit models which are to be used as baselines for comparison with our proposed model in the experiments.

Formally, a K-armed bandit consists of K arms, representing K candidate items to be recommended to a user and pulling an arm means recommending an item to a user. In a general stochastic formation, for each user u, these K arms can also be treated as K probability distributions $[D_{u,1}, D_{u,2}, \cdots, D_{u,K}]$ with associated expected values (i.e., means) $[\mu_{u,1}, \mu_{u,2}, \cdots, \mu_{u,K}]$ and variances $[\sigma_{u,1}, \sigma_{u,2}, \cdots, \sigma_{u,K}]$ where the distribution $D_{u,i}$ is initially unknown. A bandit algorithm proceeds in discrete trials (rounds) $t = 1, 2, 3, \cdots$, and given a user u, it chooses one item i out of the K candidates(through pulling one of the K arms) and recommends it to the user u in each trial (round). After each recommendation, the algorithm receives a reward $r_{u,i}(t) \sim D_{u,i}(t)$ for picking item i as the recommendation for user u. The *total expected regret* is used to measure the performance of bandit algorithms. For a bandit algorithm running totally T trials (rounds), the *total expected regret* R_T is defined as follows:

$$R_T = \sum_{u \in U} \left[\mathbb{E}\Big[\sum_{t=1}^{T} \mu_* \Big] - \mathbb{E}\Big[\sum_{t=1}^{T} \mu_{u,i}(t) \Big] \right], \tag{1}$$

where U is the set of users for evaluation and $\mu^* = max_{j=1,2,\cdots,K} \mu_j$ is the expected reward from the best arm (i.e., best candidate item) in each round. Our objective is to find an optimal set of items, minimizing the total expected regret R_T, as the recommendation for each user, which equals to maximizing the cumulative expected reward during T rounds for every user:

$$I_u(T) = \bigcup_{t=1}^{T} \arg\max_i \mathbb{E}[r_{u,i}(t)] = \bigcup_{t=1}^{T} i_u(t) \ . \tag{2}$$

Most bandit strategies maintain empirical average rewards which will be updated in every round for each arm chosen. We denote $\hat{r}_{u,i}(t)$ as the empirical average reward of arm (i.e., item) i after t rounds for user u, and $p_{u,i}(t)$ as the probability of picking arm i for user u (i.e., recommending item i to user u) in round t.

ϵ-greedy. The ϵ-greedy algorithm is widely used because of its simplicity, and obvious generalizations for sequential decision problems. In each round $t = 1, 2, \cdots$ the algorithm selects the item with the highest empirical average reward from the K candidate items with probability $1 - \epsilon$, and selects a random item with probability ϵ. In other words, given initial empirical average rewards $\hat{r}_{u,1}(0), \hat{r}_{u,2}(0), \cdots, \hat{r}_{u,K}(0)$ for user u,

$$p_{u,i}(t+1) = \begin{cases} 1 - \epsilon + \epsilon/K, & \text{if } i = \arg\max_{j=1,\cdots,K} \hat{r}_{u,j}(t) \\ \epsilon/K, & \text{otherwise.} \end{cases} \tag{3}$$

Boltzmann Exploration (Softmax). Softmax methods are based on Luce's axiom of choice [57] and pick each item for recommendation with a probability that is proportional to its average reward. Therefore items with greater empirical average rewards should be

picked with higher probabilities. In the following we will describe Boltzmann Exploration [41], a Softmax method which selects an item using a Boltzmann distribution. Given the initial empirical average rewards of the K candidate items for user u (denoted as $\hat{r}_{u,1}(0), \hat{r}_{u,2}(0), \cdots, \hat{r}_{u,K}(0)$), the probability of picking item i as recommendation for user u in round $t+1$ is:

$$p_{u,i}(t+1) = \frac{e^{\hat{r}_{u,i}(t)/\tau}}{\sum_{j=1}^{K} e^{\hat{r}_{u,j}(t)/\tau}}, \tag{4}$$

where τ is a temperature parameter controlling the randomness of the choice. We would like to point out that Boltzmann Exploration acts like pure greedy when τ tends to 0, and selects items for recommendations uniformly at random as τ tends to infinity.

Upper Confidence Bounds (UCB). Lai and Robins are the first to introduce the technique of upper confidence bounds for the asymptotic analysis of regret in stochastic bandit models [55]. Later Auer employs the UCB based algorithm to show how confidence bounds can be applied to elegantly deal with the trade-off between exploitation and exploration in online learning [52]. Then the family of UCB algorithms are proposed in [51] as a simple and elegant implementation of the idea for optimism under uncertainty. In addition to the empirical average reward, UCB maintains the number of times that each item is picked for recommendation up to round t as well. Initially all the items are assumed to be chosen once and afterwards the algorithm greedily selects item i in round t as follows:

$$i(t) = \underset{j=1,\cdots,K}{\arg\max} \left(\hat{r}_{u,j}(t) + \sqrt{\frac{2\log t}{n_j(t)}} \right), \tag{5}$$

where $n_j(t)$ represents the number of times item j has been selected for recommendations so far. We note that $\hat{r}_{u,j}(t)$ is the empirical mean estimate of $r_{u,j}(t)$ in round t given previous observations in the past $t-1$ rounds and $\sqrt{\frac{2\log t}{n_j(t)}}$ is an upper confidence bound. This can be interpreted as a good trade-off between exploitation, i.e., $\hat{r}_{u,j}(t)$, and exploration, i.e., $\sqrt{\frac{2\log t}{n_j(t)}}$.

Linear UCB (LinUCB). Li et al. propose a linear model under the UCB framework (called LinUCB) through combining linear bandit and contextual bandit together to focus on the problem of personalized news article recommendation [45]. LinUCB assumes that the mean of $r_{u,i}(t)$ can be obtained through the dot product of an item-dependent coefficient with the concatenation of user u's and item i's feature vectors in round t, which is linear with respect to the item-dependent coefficient given that the user and item feature vectors are known to us.

However, explicit feature vector may not be always available in practice. Take movie recommendation as an example, most of the state-of-the-art methods are based on collaborative filtering where user and item latent feature vectors are learnt through low rank matrix factorization. Therefore, given the success of collaborative filtering in recommender systems, we formulate LinUCB through employing the latent feature vectors learnt by low rank matrix factorization instead of explicit feature vectors extracted directly from texts or labels in this paper, which is similar to algorithm 2 in [33].

As such, a common strategy widely adopted by many matrix factorization based collaborative filtering algorithms is to approximate the feedback (e.g., ratings, clicks etc.) through the inner product of user and item latent feature vectors ($\mathbf{p}_u$ and $\mathbf{q}_i$):

$$r_{u,i} = \mathbf{p}_u^\top \mathbf{q}_i \ . \tag{6}$$

To incorporate the low rank matrix factorization into LinUCB, we reformulate the bandit strategy for item selection in the same way as [33]:

$$\begin{aligned} i(t) &= \underset{j=1,\cdots,K}{\arg\max} \mathbb{E}\big[r_{u,j}(t)\big] \\ &= \underset{j=1,\cdots,K}{\arg\max} \mathbb{E}_{\mathbf{p}_u}\left[\mathbf{p}_u^\top | t\right] \mathbf{q}_j \\ &= \underset{j=1,\cdots,K}{\arg\max} \left(\hat{\mathbf{p}}_{u,t}^\top \mathbf{q}_j + c\sqrt{\mathbf{q}_j^\top \Sigma_{u,t}^{-1} \mathbf{q}_j} \right). \end{aligned} \tag{7}$$

And we treat the user feedback for an item as the reward of picking this item for recommendation.

We conclude this section by pointing out that all of these existing models handle users' preferences over items without considering the influences from their friends on social networks, nor do they adaptively learn the different weights for different friends to best serve the recommendation accuracy. This motivates us to develop a novel multi-armed bandit (MAB) model that is capable of taking not only user-item interactions but also social information from social networks into consideration and learning these weights dynamically so that a boost in terms of recommendation quality can be achieved.

4 INTERACTIVE SOCIAL RECOMMENDATION

In this section, we propose our interactive social recommendation model (ISR) which is capable of refining itself to best serve the customers after each interaction with a user.

Let $\mathcal{U}$ be the set of users for evaluation and $\mathcal{I}$ be the set of candidate items, given a user $u \in \mathcal{U}$, N_u denotes the set of her friends, i.e., her directly connected users, and $w_{u,f}$ is the weight for the edges (connections) between user u and her friend $f \in N_u$. Recall that the vanilla matrix factorization presented in (6) has been widely adopted by collaborative filtering in both academia and industry [17]. Thus given the great success of matrix factorization in recommendation during the past years, lots of social recommendation models [1, 2, 4, 5, 10] actually are extensions based on the vanilla matrix factorization, among which Ma et al. propose the STE (Recommendation with Social Trust Ensemble) model that uses a weighted aggregation of a user's own preferences and her friends' preferences to predict the target user's final feedback (e.g., rating) on an item:

$$r_{u,i} = \alpha \mathbf{p}_u^\top \mathbf{q}_i + (1-\alpha) \sum_{f \in N_u} w_{u,f} \mathbf{p}_f^\top \mathbf{q}_i \ , \tag{8}$$

where α is a pre-set parameter controlling the relative importance of the target user's own preferences and her friends' influences, which naturally simulates the real-world scenario in which people's final decisions depend on both own preferences and friends' influences. Although this idea is elegant and effective in reducing the inaccuracy of traditional matrix factorization, it has some limitations: 1) It is an offline method depending on batch learning and not applicable for real-world recommender systems which serve in an online and interactive manner. 2) It assumes a pre-calculated and fixed weight for each friend, which may not always hold as the degree of trust between users and their friends tends to change

Algorithm 1: Interactive Social Recommendation

Input: $c_1, c_2 \in \mathbb{R}_+$, $\alpha \in [0, 1]$, λ_p, λ_w
Graph $G(U, E)$, where $\mathcal{U}$ is the set of users, $\mathcal{E}$ is the set of edges indicating the connected linkage graph.
MAP solutions for item latent feature vectors:
$\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \cdots \mathbf{q}_{|I|}$

Initialization:
$\Sigma_{u,1} \leftarrow \lambda_p I, \mathbf{h}_{u,1} \leftarrow \mathbf{0}$
$\Delta_{u,1} \leftarrow \lambda_w I, \mathbf{z}_{u,1} \leftarrow \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  $\mathbf{p}_{u,t} \leftarrow \Sigma_{u,t}^{-1}\mathbf{h}_{u,t}$
  $\mathbf{w}_{u,t} \leftarrow \Delta_{u,t}^{-1}\mathbf{z}_{u,t}$
  where $\mathbf{w}_u = [w_{u,f_1}^{\dagger}, w_{u,f_2}^{\dagger}, w_{u,f_3}^{\dagger}, \cdots w_{u,f_{|N_u|}}^{\dagger}]^{\intercal}$,
  and $f_1, f_2, \cdots, f_3 \in N_u$.
  foreach $i \in \mathcal{I}$ **do**
    foreach $f \in N_u$ **do**
      $s_{f,i} = \mathbf{p}_f^{\intercal}\mathbf{q}_i$
    end
    $\mathbf{s}_{u,i} = [s_{f_1,i}, s_{f_2,i}, s_{f_3,i}, \cdots, s_{f_{|N_u|},i}]^{\intercal}$,
    where $f_1, f_2, \cdots, f_3 \in N_u$.
    $$g_{u,i}(t) \leftarrow \alpha\Big(\mathbf{p}_{u,t}^{\intercal}\mathbf{q}_i + c_1\sqrt{\mathbf{q}_i^{\intercal}\Sigma_{u,t}^{-1}\mathbf{q}_i}\Big) + (1-\alpha)\Big(\mathbf{w}_{u,t}^{\intercal}\mathbf{s}_{u,i} + c_2\sqrt{\mathbf{s}_{u,i}^{\intercal}\Delta_{u,t}^{-1}\mathbf{s}_{u,i}}\Big)$$
  end
  Choose the item $i = \arg\max g_{u,j}(t)$ where $j = 1, \cdots, K$, with ties broken arbitrarily.
  Receive a real-value reward $r_{u,i}(t)$.

  Update:
  $$\Sigma_{u,t+1} \leftarrow \Sigma_{u,t} + \mathbf{q}_i\mathbf{q}_i^{\intercal}$$
  $$\Delta_{u,t+1} \leftarrow \Delta_{u,t} + \mathbf{s}_{u,i}\mathbf{s}_{u,i}^{\intercal}$$
  $$\mathbf{h}_{u,t+1} \leftarrow \mathbf{h}_{u,t} + \frac{\big(r_{u,i}(t) - (1-\alpha)\mathbf{w}_{u,t}^{\intercal}\mathbf{s}_{u,i}\big)\mathbf{q}_i}{\alpha}$$
  $$\mathbf{z}_{u,t+1} \leftarrow \mathbf{z}_{u,t} + \frac{\big(r_{u,i}(t) - \alpha\mathbf{p}_{u,t}^{\intercal}\mathbf{q}_i\big)\mathbf{s}_{u,i}}{1-\alpha}$$
end
Output: $\mathbf{P} = \{\mathbf{p}_u : u \in \mathcal{U}\}$,$\mathbf{W} = \{\mathbf{w}_u : u \in \mathcal{U}\}$

when new user feedback is observed. Our proposed ISR model, on the other hand, is capable of addressing these limitations.

The ISR Model

A modified version of linUCB is proposed in [33] via replacing the dot product of contextual feature vectors and coefficients with probabilistic matrix factorization, where the reward of recommending an item i to a user u in round t is regarded as the feedback (such as ratings, clicks etc.) of user u on item i:

$$r_{u,i}(t) = \mathbf{p}_u^{\intercal}(t)\mathbf{q}_i \ . \tag{9}$$

The ISR model extends this formula (9) by incorporating the social part:

$$r_{u,i}(t) = \alpha\mathbf{p}_u^{\intercal}(t)\mathbf{q}_i + (1-\alpha)\sum_{f\in N_u} w_{u,f}^{\dagger}(t)\mathbf{p}_f^{\intercal}\mathbf{q}_i \ , \tag{10}$$

where same as in (8), α is the importance controlling parameter in range $[0, 1]$ and $w_{u,f}^{\dagger} = \frac{w_{u,f}}{\sum_{v\in N_u} w_{u,v}}$ is the normalized edge weight between u and f. Then the item that has the largest weighted sum of expected rewards from u and all her friends $f \in N_u$ is selected:

$$\begin{aligned} i(t) &= \arg\max_{j=1,\cdots,K} \mathbb{E}\Big[\alpha\hat{r}_{u,j}(t) + (1-\alpha)\sum_{f\in N_u}\hat{w}_{u,f}^{\dagger}r_{f,j}(t)\Big] \\ &= \arg\max_{j=1,\cdots,K}\Big(\alpha\hat{\mathbf{p}}_u^{\intercal}(t)\mathbf{q}_j + (1-\alpha)\sum_{f\in N_u}\hat{w}_{u,f}^{\dagger}(t)\mathbf{p}_f^{\intercal}\mathbf{q}_j\Big), \end{aligned} \tag{11}$$

where K is the number of candidate items for u in round t. For convenience, we construct a social weight coefficient vector for each user u (denoted as $\mathbf{w}_u$) that consists of all the edge weights for her friends : $\mathbf{w}_u = [w_{u,f_1}^{\dagger}, w_{u,f_2}^{\dagger}, w_{u,f_3}^{\dagger}, \cdots w_{u,f_{|N_u|}}^{\dagger}]^{\intercal}$, and by further denoting $\mathbf{s}_{u,i} = [s_{f_1,i}, s_{f_2,i}, s_{f_3,i}, \cdots, s_{f_{|N_u|},i}]^{\intercal}$ where $s_{f,i} = \mathbf{p}_f^{\intercal}\mathbf{q}_i$, we can rewrite (11) as follows:

$$i(t) = \arg\max_{j=1,\cdots,K}\Big(\alpha\hat{\mathbf{p}}_{u,t}^{\intercal}\mathbf{q}_j + (1-\alpha)\hat{\mathbf{w}}_{u,t}^{\intercal}\mathbf{s}_{u,j}\Big) \ . \tag{12}$$

In plain English, our ISR model aims to find an *optimal* set of items as recommendations for different users, such that the accumulated expected reward of the recommendations over all users will be maximized.

Given the fact that user preferences tend to change in the course of time while item characteristics normally remain static, it is natural for our ISR model to place more focus on the knowledge obtained in the last round for the target user rather than for the item, especially when a sufficient amount of feedback has been collected to infer the latent feature spaces for items. Therefore, we will assume that the item latent feature vectors have already been pre-learnt through the *maximum a posterior* (MAP) estimate under matrix factorization. We will talk more about this experimental setting later in Section 5.

If the item latent feature vectors remain fixed, then the reward in (10) becomes linear with respect to the user latent feature vectors with the social weight coefficient vectors treated as constants, and also linear with respect to the social weight coefficient vectors with the user latent feature vectors treated as constants. Our goal is to find the best user latent feature vectors and the optimal edge weights for their friends.

The uncertainty of the reward comes from two parts: self-reward ($\mathbf{p}_u^{\intercal}\mathbf{q}_j$) and social-reward ($\mathbf{w}_u^{\intercal}\mathbf{s}_{u,j}$), whose uncertainty derives from the estimation for user latent feature vector $\mathbf{p}_u$ and social weight coefficient vector $\mathbf{w}_u$ respectively. According to ridge regression, the uncertainty of estimation for $\mathbf{p}_u$ is:

$$||\mathbf{q}_i||_{\Sigma_{u,t}^{-1}} = \sqrt{\mathbf{q}_i^{\intercal}\Sigma_{u,t}^{-1}\mathbf{q}_i} \ , \tag{13}$$

where $\Sigma_{u,t}^{-1}$ is the inverse covariance matrix for u's self-reward in round t. And similarly, the uncertainty in the estimation of $\mathbf{w}_u$ can be formulated as follows:

$$||\mathbf{s}_{u,i}||_{\Delta_{u,t}^{-1}} = \sqrt{\mathbf{s}_i^{\intercal}\Delta_{u,t}^{-1}\mathbf{s}_i} \ , \tag{14}$$

where $\Delta_{u,t}^{-1}$ is the inverse covariance matrix for u's social-reward in round t. ISR chooses the item with the highest upper confidence bound in each round:

$$i(t) = \arg\max_{j=1,\cdots,K}\Bigg[\alpha\Big(\mathbf{p}_{u,t}^{\intercal}\mathbf{q}_j + c_1\sqrt{\mathbf{q}_j^{\intercal}\Sigma_{u,t}^{-1}\mathbf{q}_j}\Big) + (1-\alpha)\Big(\mathbf{w}_{u,t}^{\intercal}\mathbf{s}_{u,j} + c_2\sqrt{\mathbf{s}_{u,j}^{\intercal}\Delta_{u,t}^{-1}\mathbf{s}_{u,j}}\Big)\Bigg], \tag{15}$$

where c_1 and c_2 are two parameters used to determine the confidence. The details of our proposed ISR model are given in Algorithm 1.

Complexity. Exploitation-exploration is essentially all about the parameter space for exploration. Existing multi-armed bandit (MAB) based recommendation methods normally treat each item as an arm, which results in $|\mathcal{I}|$ (i.e., total number of candidate items) parameters for each user. LinUCB [45] reduces the number of parameters for each user to $O(d)$ (i.e., the sum of the length of user and item feature vectors) by a linear model, so does the modified LinUCB under matrix factorization introduced in [33] (whose number of parameters is exactly d, the length of item latent feature vector, for every user). As for our ISR model, given a user u, there is one more parameter w_{uf} for each friend $f \in N_u$ of user u, thus we will have $|N_u|$ (number of u's friends) parameters added to the social part of our ISR model. Therefore, ISR requires $d + |N_u|$ parameters for each user u.

Regret. We remark that the self-reward part of our proposed ISR model has a regret bound of $O(\sqrt{T})$ under certain assumptions. To maintain the continuity and readability, we leave the concrete regret analysis in the end of this paper. Readers are referred to appendix A for more details.

5 EMPIRICAL EVALUATION

We report the results of our experiments on three real-world public datasets and compare the performance of the proposed *Interactive Social Recommendation* (ISR) model with various baselines including bandit based interactive methods and non-bandit based offline methods in terms of different evaluation metrics.

5.1 Experimental Setup

Although an online experimental setting with real time user-system interactions is most appropriate for evaluations of different algorithms in this paper, it is typically impossible to have such an environment in academic research [45]. Therefore, we follow the unbiased offline evaluation strategy for bandit alorithms proposed in [42] under the assumption that the user-system interactions (ratings) recorded in our experimental datasets are not biased by the recommender systems and these records can be regarded as unbiased user feedback in our experimental setting.

	Flixster	*Douban*	*Epinions*
#users	76013	64642	10702
#items	48516	56005	39737
#ratings	7350235	9133529	482492
#ratings per user	96.70	141.29	45.08
#ratings per item	151.50	163.08	12.14
#social connections	1209962	1390960	219374

Table 1: Overview of datasets

Datasets. We use the following three real-world datasets, whose basic statistics are summarized in Table 1.

- *Flixster.* The Flixster dataset containing information of user-movie ratings and user-user friendships from Flixster, an American social movie site for discovering new movies (http://www.flixster.com/).
- *Douban.* This dataset is extracted from the Chinese Douban movie forum (http://movie.douban.com/), which contains user-user friendships and user-movie ratings.
- *Epinions.* This is the popular consumer review dataset, Epinions, which consists of user-user trust relationships and user-item ratings from Epinions (http://www.epinions.com/).

For all datasets, we split the data into two user-disjoint sets : training set and test set. The test set is constructed by randomly choosing 200 users who have at least 120 ratings and 20 social connections, leaving the remaining users and their ratings in the training set.

Methods for Comparisons. We compare ISR with several state-of-the-art approaches including three exploitation-exploration (i.e., MAB based) interactive methods (*ϵ-greedy, Softmax, LinUCB*), one non-interactive personalized social recommendation method (*STE*), one non-interactive personalized non-social recommendation method (*PMF*) and one non-interactive non-personalized non-social recommendation method (*Random*). Thus, the following seven recommendation methods, including six baselines, are tested.

- *ISR.* Our proposed ISR model, which is an interactive personalized social recommendation approach.
- *ϵ-greedy.* As is presented in (3), it is one of the most popular exploitation-exploration strategies in literature. In our problem setting, the expected reward of item i for user u at round t, $\hat{r}_{u,i}(t)$, is assumed to be estimated by the dot product of user latent feature vector at round t ($\mathbf{p}_{u,t}$) and item latent feature vector ($\mathbf{q}_j$). Thus the *ϵ-greedy* algorithm picks the item with the largest estimated reward based on the current knowledge with probability $1-\epsilon$ at round t:
$$i(t) = \arg\max_{j=1,\cdots,K} \hat{\mathbf{p}}_{u,t}^{\top} \mathbf{q}_j \,, \tag{16}$$
and randomly picks an item with probability ϵ.
- *Softmax.* Another well-studied exploitation-exploration strategy described in (4), which is fitted into our problem setting through substituting $\hat{r}_{u,i}(t)$ with $\hat{\mathbf{p}}_{u,t}^{\top}\mathbf{q}_j$ (i.e., $\hat{r}_{u,i}(t) = \hat{\mathbf{p}}_{u,t}^{\top}\mathbf{q}_j$), in a similar way to ϵ-greedy.
- *Linear UCB (LinUCB).* Algorithm 2 in [33] where c is a tuning parameter, see equation (7) in section 3.
- *STE.* This is a personalized social recommendation method proposed by Ma et al. [2] which aggregates a user's own rating and her friends' ratings to predict the target user's final rating on an item.
- *PMF.* The classic personalized non-social probabilistic matrix factorization model first introduced in [16].
- *Random.* Randomly recommend unrated items to each user.

As is pointed out in section 2 that the three models proposed in [19, 23, 34] are designed for explicit features rather than latent features, resulting in different model formulations and experimental settings from ours. This being the case, their work is orthogonal to ours and we are unable to compare ISR with these three models.

Evaluation Metrics. We evaluate different models in two aspects: 1) recommending one single item in each round and 2) recommending multiple items in each round. If we only recommend a single item in each round, one straightforward measure is to count the number of hit (i.e., recommendation in which the recommended item has a rating that is no smaller than 4) after T rounds and average it by the number of users. Thus based on this methodology, we adopt two metrics, cumulative *Precision@T* and cumulative *Recall@T*, for the evaluation in the scenario of single item recommendation per round.

	Flixster *Cumulative Precision*				**Douban** *Cumulative Precision*				**Epinions** *Cumulative Precision*			
Round T	20	40	80	120	20	40	80	120	20	40	80	120
ϵ-greedy	0.6065	0.5358	0.4346	0.3689	0.6362	0.5588	0.4496	0.3804	0.6943	0.5749	0.4337	0.3537
Softmax	0.6138	0.5427	0.4385	0.3719	0.6380	0.5616	0.4510	0.3814	0.6967	0.5776	0.4351	0.3547
LinUCB	0.7798	0.6393	0.4792	0.3897	0.8073	0.6582	0.4918	0.3989	0.6884	0.5763	0.4399	0.3616
ISR	**0.8442**	**0.6824**	**0.5059**	**0.4088**	**0.8790**	**0.7094**	**0.5221**	**0.4203**	**0.7790**	**0.6379**	**0.4786**	**0.3899**
Imprv	8.26%*	6.74%*	5.57%*	4.90%	8.88%*	7.78%*	6.16%*	5.37%*	11.81%*	10.44%*	8.80%*	7.83%*
	Cumulative Recall				*Cumulative Recall*				*Cumulative Recall*			
Round T	20	40	80	120	20	40	80	120	20	40	80	120
ϵ-greedy	0.0960	0.1698	0.2745	0.3464	0.0792	0.1283	0.1957	0.2338	0.1030	0.1629	0.2437	0.2811
Softmax	0.0975	0.1720	0.2772	0.3494	0.0797	0.1287	0.1960	0.2345	0.1037	0.1634	0.2443	0.2818
LinUCB	0.1229	0.2019	0.3022	0.3658	0.1007	0.1516	0.2138	0.2451	0.1021	0.1632	0.2472	0.2869
ISR	**0.1333**	**0.2161**	**0.3195**	**0.3845**	**0.1097**	**0.1629**	**0.2270**	**0.2582**	**0.1150**	**0.1811**	**0.2694**	**0.3094**
Imprv	8.46%*	7.03%*	5.73%*	5.11%	8.94%*	7.45%*	6.17%*	5.35%*	10.90%*	10.83%*	8.98%*	7.84%*

Table 2: Cumulative precision and recall on test users (bold font highlights the winner).

- Cumulative *Precision@T* (*Pre@T*).

$$Precision@T = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{T} \sum_{t=1}^{T} \theta_{hit} ,$$

  where $\theta_{hit} = 1$ if the rating of the target user u on the recommended item i in round t is equal to or higher than 4 and $\theta_{hit} = 0$ otherwise. $\mathcal{U}_{test}$ denotes those users in the test set.
- Cumulative *Recall@T* (*Rec@T*).

$$Recall@T = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \sum_{t=1}^{T} \frac{\theta_{hit}}{|\mathcal{R}_u|} ,$$

  where $\mathcal{R}_u$ is the set of items that have been rated no less than 4 by user u in the test set.

When recommending multiple items in each round, the relative rankings of these candidate items become fairly important for the evaluation. Normalized Discounted Cumulative Gain (*NDCG*) is such a top-n recommendation measure suitable for this purpose. Let $S(u)$ be the set of all items rated by user u in the test set and $C(u)$ be the set of candidate items to be ranked in the test set for user u. We denote $R(u)$ as the ranking of items in $C(u)$ in a descending order, then for any item i in $S(u)$, its position in $R(u)$ is noted as $rank_i^u$.

- *NDCG*. In the context of recommender systems, *NDCG* is defined as follows:

$$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{DCG_u}{IDCG_u} ,$$

  where *DCG* and *IDCG* (Ideal Discounted Cumulative Gain) are in turn defined as:

$$DCG_u = \sum_{i \in S(u)} \frac{1}{\log_2(rank_i^u + 1)} , \text{ and } IDCG_u = \sum_{i=1}^{|S(u)|} \frac{1}{\log_2(i+1)} .$$

Thus the *NDCG* value for exploitation-exploration (MAB based) interactive methods will take the summation over all T rounds and then average on the number of total rounds. In our experiments, we test *NDCG@n* (where $n = 3, 5$), indicating that $C(u)$ only contains items with top-n largest rating values from u.

5.2 Experimental Results

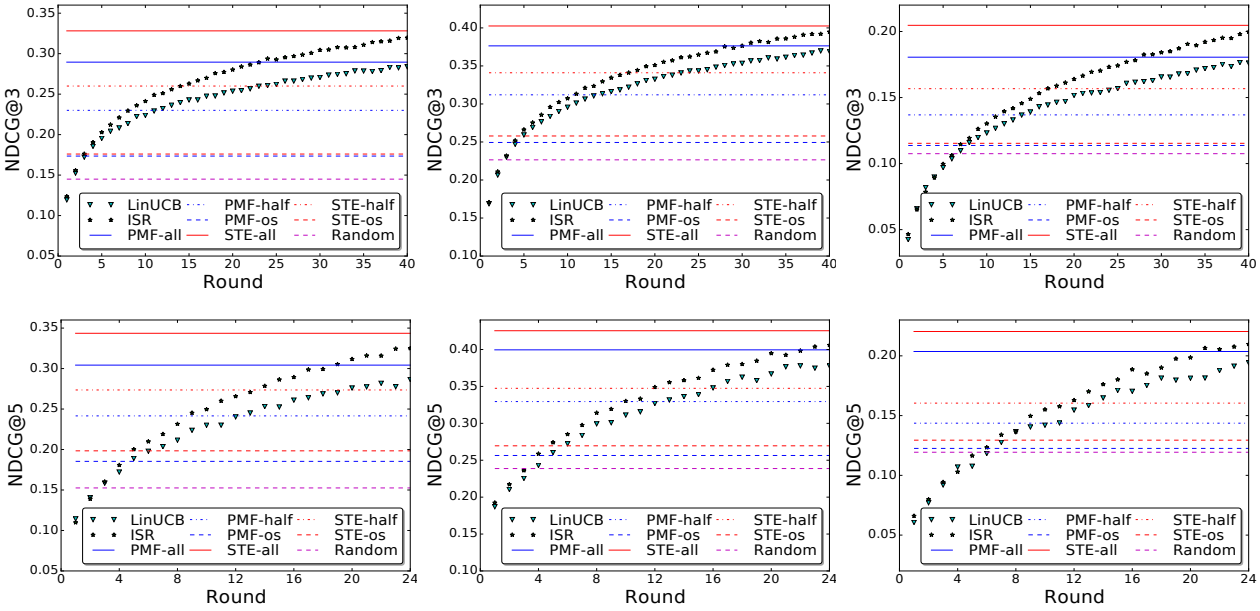
For exploitation-exploration (MAB based) algorithms including ϵ-greedy, Softmax, LinUCB and ISR, probabilistic matrix factorization is first used to train all the item latent feature vectors which will remain unchanged thereafter and be utilized to learn the user latent feature vectors (and the social weight coefficients for ISR) later. Furthermore, "*" in Table 2 and Table 3 indicates that the corresponding result is significant by Wilcoxon signed-rank test at $p < 0.05$.

Recommending a single item in each round. In this evaluation scenario, up to 120 rounds of interactions are studied for each exploitation-exploration algorithm, given that each user in the test sets has at least 120 ratings. We compare the performance of our proposed ISR model with other three exploitation-exploration methods: ϵ-greedy, Softmax and LinUCB, in term of cumulative precision and recall. Table 2 presents the performances of all four approaches on all three datasets for $T = 20$, 40, 80 and 120, with the last row showing the improvement of ISR over the best baseline. Clearly, the proposed ISR model outperforms all three exploitation-exploration baselines, with a trend towards a decreasing improvement as T becomes larger. Take cumulative precision as an example, as T increases from 20 to 120, the improvement of ISR over the best baseline decreases from 8.26% to 4.90% on Flixster, from 8.88% to 5.37% on Douban and from 11.81% to 7.83% on Epinions. One possible reason is that during the first several runs of the model when very little feedback is available, ISR model is capable of making much better recommendations than the baselines due to the benefit of taking social influences into consideration. On the other hand, these models will receive more and more feedback, which may increase their recommendation accuracy (especially for non-social exploitation-exploration baselines) as T increases, resulting in a less improvement for ISR against the baselines.

Recommending multiple items in each round. In the scenario of recommending m ($m > 1$) items per round, we study up to $T = \frac{120}{m}$ rounds of interactions when evaluating each algorithm. In our experiments, we test the performance of different algorithms by setting $m = 3$ and $m = 5$ and study up to $T = 40$ and $T = 24$ rounds of interactions. Moreover, each of the two non-MAB based baselines (i.e., PMF and STE) is designed to have three variants: *-os* (short for out of sample), *-half* and *-all*. For variant *-os*, we train the model on the training set and test its performance on the test set. Note that as the training set and test set are user-disjointed, users in the test set will never appear in the training set (i.e., out of sample), which may result in very poor performance for non-MAB based models. As for the other two variants, we randomly select η ratings to train the user latent feature vector for each user u in the test set. We set η to be the number of observable ratings during the first $\frac{T}{2}$ rounds in the test set for the *-half* variant and be the

	Flixster				Douban				Epinions			
	NDCG@3		*NDCG@5*		*NDCG@3*		*NDCG@5*		*NDCG@3*		*NDCG@5*	
Round T	20	40	12	24	20	40	12	24	20	40	12	24
ϵ-greedy	0.2398	0.2793	0.2267	0.2894	0.2646	0.3260	0.2932	0.3668	0.1503	0.1769	0.1559	0.1918
Softmax	0.2421	0.2859	0.2197	0.2739	0.2588	0.3202	0.2906	0.3617	0.1454	0.1733	0.1479	0.1888
LinUCB	0.2537	0.2838	0.2403	0.2865	0.3325	0.3692	0.3269	0.3780	0.1516	0.1762	0.1546	0.1943
ISR	**0.2802**	**0.3197**	**0.2657**	**0.3250**	**0.3510**	**0.3949**	**0.3490**	**0.4060**	**0.1640**	**0.1999**	**0.1629**	**0.2098**
Imprv	10.45%*	11.82%*	10.57%*	12.30%*	5.56%*	6.96%*	6.76%*	7.41%*	8.18%	13.00%*	4.49%	7.98%*

Table 3: NDCG@n for ϵ-greedy, Softmax, LinUCB and ISR on three datasets (bold font highlights the winner).

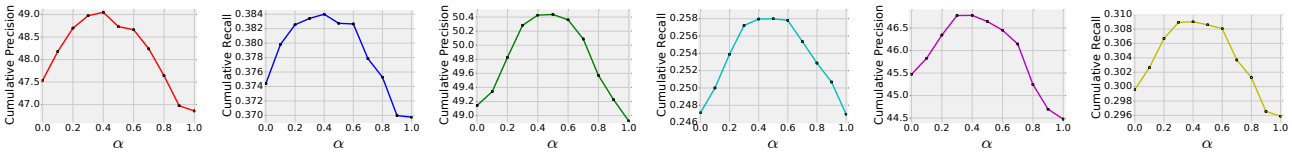


(a) Flixster (b) Douban (c) Epinions

Figure 1: NDCG@3 and NDCG@5 for Random, LinUCB, ISR, PMF and STE as well as their variants on three datasets

number of all available ratings in the test set for the *-all* variant. In other words, the - *all* variant is trained on all available observations in the test set, indicating the best solution we can obtain and the performance of *-half* should intuitively lie between *-all* and *-os*. In Figure 1, we can see seven straight horizontal lines (they are straight because these non-MAB based off-line models do batch trainings and have nothing to do with the rounds of interactions) in each of the six sub-figures, representing the Random baseline (the lowest one) as well as the three variants for each of PMF and STE: PMF-all, PMF-half, PMF-os and STE-all, STE-half, STE-os. It is easy to observe that PMF-half lies between PMF-all and PMF-os, and similarly STE-half lies between STE-all and STE-os, which verifies our assumptions above. On the other hand, LinUCB and ISR which can be regarded as the exploitation-exploration (MAB based) versions of PMF and STE to some extent, start with very poor performance, gradually get improved when receiving more and more feedback in rounds of interactions and closely approach PMF-all and STE-all respectively in round 120. For both *NDCG*@3 and *NDCG*@5 on all three datasets, the *-half* baselines outperform their MAB based algorithms (LinUCB and ISR) in early rounds before being surpassed by their exploitation-exploration counterparts soon after. This is reasonable since the *-half* variant can get access to a portion of the observations in the test set to learn the user preferences, but when more user feedback is available the MAB based algorithm gets improved through dynamically adapting to user feedback and finally reaches a comparable performance with the *-all* variant. Besides, our proposed ISR outperforms LinUCB which does not utilize social information, through the benefit of taking social influences from friends into account and adaptively learning weights for these friends. In addition to LinUCB, we also compare ISR with other exploitation-exploration baselines including ϵ-greedy and Softmax, whose results are list in Table 3. With no surprise, we observe that ISR beats both of them in all cases .

Impact of controlling parameter α. As a controlling parameter, α balances the target user's own preferences and the tastes of her friends. It controls the extent to which ISR should trust the target user' own interests and how much the model should emphasis on the tastes of her friends. In two extreme cases, ISR will only consider the target user's own preferences without any social influences when α is set to 1 and merely take the preferences of the target user's friends into account when α is set to 0. With α being set to other real values between 1 and 0, ISR will take both the target user's and her friends' interests into consideration when making recommendations. Figure 2 shows the impact of α on both cumulative precision and recall for all three datasets. We observe that the optimal α equals to 0.4 on Flixster and Epinions, and equals to 0.5 on Douban, which confirms the efficacy of fusing favors of the target user and her friends together in improving the recommendation accuracy. Moreover, each of the plots in Figure 2 looks analogous to a parabolic shape for both cumulative precision and recall on all datasets, indicating that α with either a larger or smaller value than the optimal one may cause a decline in the



(a) Flixster (b) Douban (c) Epinions

Figure 2: Impact of different α values in ISR on cumulative precision and recall for three datasets in round 120

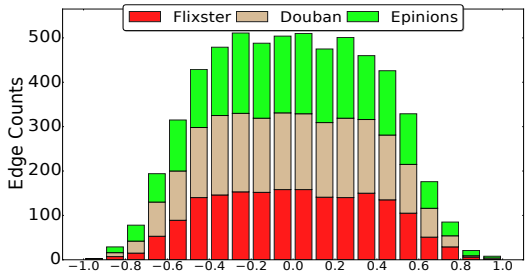


Figure 3: Relative edge weight changes after round 120

performance of the algorithm. In other words, it is necessary to find a good balance between the tastes of the target users and their friends — leaning too much against either of them may result in suboptimal recommendations.

Learning the edge weights. Last but not least, we also present some statistics on the learned edge weights by ISR. As discussed in section 4, we adopt the normalized edge weights so that the initial edge weights depend on the number of friends for each user (i.e., initial weights are equally set to $\frac{1}{|N_u|}$ for all edges of user u). Thus we show the relative changes in edge weights with respect to their initial values after 120 rounds of ISR in Figure 3, where positive bin values on X axis indicate relative increases and negative ones indicate relative decreases. We observe that weights of 1829 edges in Flixster, 2117 edges in Douban and 2075 edges in Epinions are updated during the 120 rounds where most of them have a relative change between −80% and 80% of their initial values, demonstrating the necessity of learning the edge weights.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel interactive social recommendation model (ISR), which can not only dynamically adapt itself based on user feedback but also adaptively learn different weights for different friends in social networks. We employ the similar idea of multi-armed bandit (MAB) strategy for the interactive learning procedure and analyze the regret bound of our proposed ISR model. We evaluate the performance of the proposed ISR model and compare with various baselines including MAB based algorithms and non-MAB based ones in terms of cumulative precision, cumulative recall and *NDCG@n* on three real-world datasets, demonstrating the advantages of ISR against these state-of-the-art approaches.

Despite the promising results obtained, some open issues remain unsolved in this paper. First of all, some users might get new friends during the interactions, which will lead to the problem of incremental social information. Second, there always exist popular users who have lots of friends, making the exploration space considerably huge. It will be quite interesting and challenging to investigate these two problems and we leave them for future work.

ACKNOWLEDGMENTS

This research is supported by China Postdoctoral Science Foundation No. BX201700136 , National Program on Key Basic Research Project No. 2015CB352300, National Natural Science Foundation of China Major Project No. U1611461 and the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

REFERENCES

[1] H. Ma *et al.,* Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.
[2] H. Ma *et al.,* Learning to recommend with social trust ensemble. In *SIGIR*, 2009.
[3] M. Jamali and M. Ester Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *KDD*, pages 397–406, 2009.
[4] M. Jamali and M. Ester A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.
[5] H. Ma *et al.,* Recommender systems with social regularization. In *WSDM*, 2011.
[6] S. Yang *et al.,* Like like alike: joint friendship and interest propagation in social networks. In *WWW*, pages 537–546, 2011.
[7] H. Ma *et al.,* Learning to recommend with explicit and implicit social relations. *TIST*, 3(2):29, ACM, 2011.
[8] M. Ye *et al.,* Exploring social influence for recommendation: a generative model approach. In *SIGIR*, pages 671–680, 2012.
[9] S. Purushotham *et al.,* Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems. In *ICML*, 2012.
[10] B. Yang *et al.,* Social collaborative filtering by trust. In *IJCAI*, 2013.
[11] T. Zhao *et al.,* Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pages 261–270, 2014.
[12] X. Wang *et al.,* Social Recommendation with Strong and Weak Ties. In *CIKM*, pages 5–14, 2016.
[13] J. Li *et al.,* Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *IJCAI*, 2017
[14] X. Wang *et al.,* Learning personalized preference of strong and weak ties for social recommendation. In *WWW*, pages 1601–1610, 2017.
[15] K. Cheng *et al.,* Unsupervised Feature Selection in Signed Social Networks. In *KDD*, pages 777—786, 2017
[16] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.
[17] Y. Koren *et al.,* Matrix factorization techniques for recommender systems. *Computer*, 8(42):30–37, 2009.
[18] X. Wang *et al.,* Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization. In *AAAI*, 2016.
[19] H. Wang *et al.,* Factorization Bandits for Interactive Recommendation. In *AAAI*, 2017.
[20] L. Zhou and E. Brunskill. Latent Contextual Bandits and their Application to Personalized Recommendations for New Users. In *IJCAI*, 2016.
[21] S. Zong *et al.,* Cascading Bandits for Large-Scale Recommendation Problems. In *UAI*, 2016.
[22] C. Zeng *et al.,* Online Context-Aware Recommendation with Time Varying Multi-Arm Bandit. In *KDD*, 2016.
[23] Q. Wu *et al.,* Contextual Bandits in a Collaborative Environment. In *SIGIR*, 2016.
[24] S. Li *et al.,* Collaborative Filtering Bandits. In *SIGIR*, 2016.
[25] L. Tang *et al.,* Personalized recommendation via parameter-free contextual bandits. In *SIGIR*, pages 323–332, 2015.
[26] J. Kawale *et al.,* Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *NIPS*, pages 1297–1305, 2015.
[27] A. Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 1(15):2533–2568, 2014.
[28] C. Gentile *et al.,* Online Clustering of Bandits. In *ICML*, pages 757–765, 2014.
[29] H. Vanchinathan *et al.,* Explore-exploit in top-n recommender systems via gaussian processes. In *RecSys*, pages 225–232, 2014.
[30] V. Kuleshov and D. Precup. Algorithms for multi-armed bandit problems. In *arXiv:1402.6028*, 2014.
[31] L. Tang *et al.,* Ensemble contextual bandits for personalized recommendation. In *RecSys*, pages 73–80, 2014.

[32] L. Qin *et al.,* Contextual Combinatorial Bandit and its Application on Diversified Online Recommendation. In *SDM*, pages 461–469, 2014.
[33] X. Zhao *et al.,* Interactive collaborative filtering. In *CIKM*, 2013.
[34] N. Cesa-Bianchi *et al.,* A gang of bandits. In *NIPS*, pages 737–745, 2013.
[35] Y. Yue *et al.,* Hierarchical exploration for accelerating contextual bandits. In *ICML*, 2012.
[36] D. Bouneffouf *et al.,* A contextual-bandit algorithm for mobile context-aware recommender system. In *ICONIP*, pages 324–331, 2012.
[37] K. Amin *et al.,* Graphical Models for Bandit Problems. In *UAI*, 2011.
[38] W. Chu *et al.,* Contextual Bandits with Linear Payoff Functions. In *AISTATS*, 2011.
[39] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *NIPS*, pages 2249–2257, 2011.
[40] Y. Abbasi-Yadkori *et al.,* Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320, 2011.
[41] R. Sutton and A. Barto. Reinforcement learning: An introduction. Cambridge Univ Press, 2011.
[42] L. Li *et al.,* Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *WSDM*, pages 297–306, 2011.
[43] S. Filippi *et al.,* Parametric bandits: The generalized linear case. In *NIPS*, 2010.
[44] W. Li *et al.,* Exploitation and exploration in a performance based contextual advertising system. In *KDD*, pages 27–36, 2010.
[45] L. Li *et al.,* A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.
[46] D. Agarwal *et al.,* Explore/exploit schemes for web content optimization. In *ICDM*, pages 1–10, 2009.
[47] J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NIPS*, pages 817–824, 2008.
[48] A. Mahajan and D. Teneketzis. Multi-armed bandit problems. *Foundations and Applications of Sensor Management*, 121–151, 2008.
[49] J. Langford *et al.,* Exploration scavenging. In *ICML*, pages 528–535, 2008.
[50] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, Nov(3):397–422, 2002.
[51] P. Auer *et al.,* Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2-3(47):235–256, 2002.
[52] P. Auer. Using upper confidence bounds for online learning. In *41st Annual Symposium on Foundations of Computer Science*, pages 270–279, 2000.
[53] P. Auer *et al.,* Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *36th Annual Symposium on Foundations of Computer Science*, 1995.
[54] D. Berry and B. Fristedt. Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). Springer, 1985.
[55] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 1(6):4–22, Elsevier, 1985.
[56] J. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 148–177, JSTOR, 1979.
[57] R. Luce. Individual choice behavior, a theoretical analysis. *Bull. Amer. Math. Soc*, 66(1960):259-260, 1960.
[58] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 5(58):527–535, 1952.

A REGRET OF SELF-REWARD FOR ISR

Recall that for UCB based algorithm, take (5) and (7) for instance, the choice of item in each round is:

$$i(t) = \arg\max_{j=1,\cdots,K} \left(\hat{r}_j(t) + \hat{c}_j(t) \right), \tag{17}$$

where for each item $j = 1, \cdots, K$, the true mean reward $r_j(t)$ in round t lies in a confidence interval:

$$C_j(t) : \left[\hat{r}_j(t) - \hat{c}_j(t) \quad , \quad \hat{r}_j(t) + \hat{c}_j(t) \right]. \tag{18}$$

To be brief, the estimation of $r_j(t)$ is supposed to be as optimistic as possible and then the item with the best optimistic estimate will be chosen.

As such, we formulate the regret in the vanilla stochastic multi-arm bandit setting as a simpler version of that indicated in (1):

$$R_T = \sum_{t=1}^{T} \left(\mu_* - r_i(t) \right), \tag{19}$$

where u_* denotes the expected reward of the best item. Then [51] shows that after running the UCB based algorithms, with high probability:

$$\begin{aligned} R_T &= \sum_{t=1}^{T} \left(\mu_* - r_i(t) \right) \le \sum_{t=1}^{T} \left(\hat{r}_i(t) + \hat{c}_i(t) - r_i(t) \right) \\ &\le \sum_{t=1}^{T} \left(\hat{r}_i(t) + \hat{c}_i(t) - \left(\hat{r}_i(t) - \hat{c}_i(t) \right) \right) = 2 \sum_{t=1}^{T} \hat{c}_i(t) \,. \end{aligned} \tag{20}$$

Confidence Intervals. It is easy to show that through concatenating all feature vectors into a single "larger" one, the self-reward part of ISR can be treated as a special case of general linear stochastic bandit [40], which in each round chooses the item such that:

$$i(t) = \arg\max_{j=1,\cdots,K} \left(\hat{\mathbf{p}}_t^\intercal \mathbf{q}_{t,j} + c \sqrt{\mathbf{q}_{t,j}^\intercal \Sigma_t^{-1} \mathbf{q}_{t,j}} \right). \tag{21}$$

And the ellipsoid confidence interval for $\mathbf{p}$ is:

$$C_t = \{ \mathbf{p} \mid ||\mathbf{p} - \hat{\mathbf{p}}||_{\Sigma_t^{-1}} \le c \}, \tag{22}$$

where $||x||_\Sigma = \sqrt{\mathbf{x}^\intercal \Sigma \mathbf{x}}$. Given that Σ_t is a symmetric positive definite matrix and:

$$||\mathbf{p} - \hat{\mathbf{p}}_t||_{\Sigma_t^{-1}} = \sqrt{(\mathbf{p} - \hat{\mathbf{p}}_t)^\intercal \Sigma_t^{-1} (\mathbf{p} - \hat{\mathbf{p}}_t)}, \tag{23}$$

if we set Σ_t to be identity matrix, resulting in a norm-2 regularization on $\mathbf{p} - \hat{\mathbf{p}}_t$, then $\hat{\mathbf{p}}_t$ can be estimated through the standard ridge regression:

$$\hat{\mathbf{p}}_t = \arg\min_{\mathbf{p}} \sum_{t'=1}^{t-1} \left(\hat{r}_i(t') - \mathbf{p}^\intercal \mathbf{q}_{t',i} \right) + \lambda ||\mathbf{p}||^2 \,. \tag{24}$$

The corresponding regret is then measured as follows:

$$R_T = \sum_{t=1}^{T} \left(\mathbf{p}_t^\intercal \mathbf{q}_{t,j^*} - \mathbf{p}_t^\intercal \mathbf{q}_{t,j} \right), \tag{25}$$

where $j^* = \arg\max_{j=1,\cdots,K} \mathbf{p}_t^\intercal \mathbf{q}_{t,j}$.

As a common setting, we follow the assumption that everything is Gaussian, e.g., the distribution D described in Section 3 follows a Gaussian distribution with μ and σ as mean and variance respectively. Thus from the solution of ridge regression, we have:

$$\Sigma_t = \lambda_p I + \sum_{t'=1}^{t} \mathbf{q}_{t',i} \mathbf{q}_{t',i}^\intercal \,, \tag{26}$$

making C_t in (22) a valid ellipsoid confidence set containing the true $\mathbf{p}$ with a very high probability controlled by c. Abbasi-Yadkori et al. [40] give a general condition on the use of valid confidence ellipsoid, which says if the linearity of true model and the independence of the rewards with R-sub-Gaussian (with $R \ge 0$) hold, and $\mathbf{p}$ as well as $\mathbf{q}$ are bounded by some constants, i.e., $||\mathbf{p}|| \le S$ and $||\mathbf{q}|| \le L$, then for any $0 \le \delta \le 1$ and all $t \ge 0$, with probability at least $1 - \delta$, the true optimal value $\mathbf{p}_*$ lies in the following ellipsoid confidence set C_t:

$$\mathbf{p} \in \mathbb{R}^d : ||\mathbf{p} - \hat{\mathbf{p}}_t||_{\Sigma_t^{-1}} \le R \sqrt{d \log \left(\frac{1 + tL^2/\lambda}{\delta} \right)} + \lambda^{\frac{1}{2}} S \,. \tag{27}$$

We refer readers to Theorem 2 in [40] for more details.

Therefore, applying (27) with R-sub-Gaussian tails on the noise, $\mathbf{p}$ and $\mathbf{q}$ upper bounded by S and L, C_t in (22) will be at most:

$$O\left(R \sqrt{d |\mathcal{I}| \log \frac{t}{\delta}} + \lambda^{\frac{1}{2}} S \right), \tag{28}$$

where d is the latent feature dimension and $|\mathcal{I}|$ is the number of candidate items.

Regret Bound. Under the assumption that $\lambda \ge \max_{\mathbf{q}} ||\mathbf{q}||^2$ and based on the proof of Theorem 3 in [40], we can further write (20) as follows:

$$R_T \le 2 \sum_{t=1}^{T} c_i(t) = 2 \sum_{t=1}^{T} c_t ||\mathbf{q}_{t,i}||_{\Sigma_t^{-1}} \le 2 \sqrt{\sum_{t=1}^{T} c_t^2 ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} \tag{29}$$

$$\le 2 \sqrt{c_T^2 \sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} = 2 c_T \sqrt{\sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}}} \,, \tag{30}$$

where (29) is obtained by applying Cauchy-Schwarz inequality [1] and (30) is obtained based on the fact that c_t is monotonically increasing. Again, Abbasi-Yadkori et al. [40] prove that if $\lambda \ge \max_{\mathbf{q}} ||\mathbf{q}||^2$ holds, then:

$$\sum_{t=1}^{T} ||\mathbf{q}_{t,i}||^2_{\Sigma_t^{-1}} \le 2 \log \det(\Sigma_T) \le O\left(d |\mathcal{I}| \log T \right). \tag{31}$$

Last, by putting (29) and (31) together, we have:

$$R_T \le O\left(dRS |\mathcal{I}| \lambda^{\frac{1}{2}} \log \left(\frac{T}{\delta} \right) \sqrt{T} \right), \tag{32}$$

and if we further ignore the logarithmic factors and regards the latent feature dimension parameter d as a constant, then the regret of the self-reward part of ISR is at most $O(\sqrt{T})$.

[1] https://en.wikipedia.org/wiki/Cauchy-Schwarz_inequality