User Participation in Collaborative Filtering-Based Recommendation Systems: A Game Theoretic Approach

Lei Xu, Chunxiao Jiang, *Senior Member, IEEE*, Yan Chen, *Senior Member, IEEE*, Yong Ren, *Senior Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Collaborative filtering is widely used in recommendation systems. A user can get high-quality recommendations only when both the user himself/herself and other users actively participate, i.e., provide sufficient ratings. However, due to the rating cost, rational users tend to provide as few ratings as possible. Therefore, there exists a tradeoff between the rating cost and the recommendation quality. In this paper, we model the interactions among users as a game in satisfaction form and study the corresponding equilibrium, namely satisfaction equilibrium (SE). Considering that accumulated ratings are used for generating recommendations, we design a behavior rule which allows users to achieve an SE via iteratively rating items. We theoretically analyze under what conditions an SE can be learned via the behavior rule. Experimental results on Jester and MovieLens data sets confirm the analysis and demonstrate that, if all users have moderate expectations for recommendation quality and satisfied users are willing to provide more ratings, then all users can get satisfying recommendations without providing many ratings. The SE analysis of the proposed game in this paper is helpful for designing mechanisms to encourage user participation.**

***Index Terms*—Behavior rule, collaborative filtering (CF), game theory, satisfaction equilibrium (SE).**

Manuscript received September 5, 2017; revised December 11, 2017; accepted January 16, 2018. Date of publication February 13, 2018; date of current version February 22, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61571300, Grant 61371079, Grant 61672137, and Grant 61602090, in part by the Young Elite Scientists Sponsorship Program By CAST under Grant 2016QNRC001, and in part by the 111 Project under Grant B17008. The work of Y. Chen was supported by the Thousand Youth Talents Program of China. This paper was recommended by Associate Editor Y. S. Y. S. Ong. *(Corresponding author: Chunxiao Jiang.)*

L. Xu is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: xul04ster@gmail.com).

C. Jiang is with the Tsinghua Space Center, Tsinghua University, Beijing 100084, China (e-mail: chx.jiang@gmail.com).

Y. Chen is with the Center for Future Media and School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: eecyan@uestc.edu.cn).

Y. Ren is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: reny@tsinghua.edu.cn).

K. J. R. Liu is with the Department of Electrical and Computer Engineering, University of Maryland at College Park, College Park, MD 20742 USA (e-mail: kjrliu@umd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2800731

I. Introduction

A. Collaborative Filtering-Based Recommendation

RECOMMENDATION system has been successfully applied in a variety of applications [1]. The predominant approach to building recommendation systems is collaborative filtering (CF) [2], where the key idea is to utilize the *ratings* collected from users to identify users with similar interests and to predict which items the users may be interested in. Conventionally, ratings are organized into a user-item matrix $\mathbf{R} = [r_{ij}]_{N\times M}$ with the rating r_{ij} indicating user i's preference for item j. The task of the recommendation server (RS) is to predict the missing values in the matrix.

Users' rating data are the fundamental resources of CF-based recommendation systems, which means user participation is of vital importance for the success of recommendation. Generally, a user assigns ratings to items after he[1] has obtained experience of the items. In practice, the number of total items available for recommendation is much larger than the number of items that a user has experienced, thus the rating matrix is sparse. To make things worse, due to the cost incurred by rating items, such as time consumption and privacy disclosure [3], users will not rate every item that they have experienced. The insufficiency of rating data inevitably impairs the recommendation quality [4].

B. Encourage User Participation

To deal with the aforementioned problem, researchers have proposed various approaches, such as exploring the content information [5] and user relationships [6]. Apart from improving the recommendation algorithms [7]–[9], one can circumvent the problem by designing incentive mechanisms to encourage user participation. Though mechanisms proposed particularly for recommendation systems are rare, incentive mechanisms have been extensively studied in similar contexts such as crowdsourcing [10] and cooperation in wireless communications [11], [12].

In recommendation systems, the RS can offer various incentives to users so as to compensate their rating cost. In addition to monetary rewards and other forms of external incentives, the recommendation quality can be considered as an intrinsic incentive for users to rate items. In this paper, we investigate

[1]For ease of description, in this paper we sometimes use *he* to refer to the user.

the influence of the recommendations themselves on users' rating behaviors. Specifically, we are interested in the following questions: whether users, motivated by recommendation quality solely, can contribute sufficient rating data so that the RS can generate *satisfying* recommendations for all users? How should users behave so that the cost of rating and the quality of recommendations can be balanced?

C. Game-Theoretic Approach

Intuitively, a user may get better recommendations if he reveals more information about his preferences to the RS by rating more items, while in the meantime, the user has to pay more cost. When deciding whether to rate an item or not, a user needs to make a tradeoff between the cost of rating and the quality of recommendation. Moreover, as the name CF suggests, whether a user can get good recommendations depends not only on the ratings provided by the user himself, but also on the ratings provided by others. Therefore, interactions of individuals' rating behaviors should be considered when one makes decisions on rating. Furthermore, users are usually rational, in the sense that a user wishes to obtain good recommendations without rating many items. In such a case, it is natural to employ game theory [13] to model the interactions among users in a CF system.

In our previous work [14], we have built a game-theoretic model to study users' rating behaviors in a CF-based recommendation system. Application of game theory has been seen in a few studies of user behavior in a context where individuals' behaviors affect each other [15]–[17]. Particularly, Halkidi and Koutsopoulos [15] employed game theory to model the interactions among users in a recommendation system. They developed a mathematical framework to address the tradeoff between privacy preservation and high-quality recommendation. Different from their study, we model the interactions among users as a satisfactory game with incomplete information: each user only has the knowledge of his own ratings and recommendations, while others' ratings cannot be observed. Meanwhile, the CF algorithm adopted by the RS is also unknown to users. Inspired by Perlaza *et al.*'s work [18], we apply the notion of *satisfaction equilibrium* (SE), which was originally introduced by Ross and Chaib-draa [19], to analyze the game with incomplete information. A game is said to be in SE when all players simultaneously satisfy their individual constraints. In the context of CF, a user's expectation for recommendation quality is seen as his constraint.

Based on the game model proposed in [14], in this paper, we carefully study how to find the SE of the game. As mentioned above, the proposed game is a game with incomplete information. Hence, different from the equilibrium concepts in the context of complete information games, the SE arises as a result of a learning process, rather than the result of rational thinking on players' beliefs and observations [19]. Based on the characteristics of recommendation systems, we design a learning algorithm which allows users to achieve an SE. Convergence of the proposed learning algorithm is analyzed theoretically. And we conduct a series of experiments on the Jester data set and the MovieLens data set to verify the feasibility of the learning algorithm. We think that the derived convergence conditions can provide some implications to the design of external incentives.

The rest of this paper is organized as follows. Section II presents the experimental proof for the basic assumption based on which we build the game model. Section III briefly describes the system model while Section IV presents in details the game formulation. In Section V, we introduce the proposed learning algorithm to achieve the SE. The convergence analysis is conducted in Section VI. Finally, the simulation results are shown in Section VII and the conclusions are drawn in Section VIII.

II. Preliminary Analysis

A fundamental assumption of this paper is that given the items that the users have experienced and the recommendation algorithm adopted by the RS, the quality of recommendations increases as users provide more ratings. This assumption is quite general. Yet in order to make this paper more rigorous, we have conducted some simple experiments to verify the assumption. Experiments were performed on a set of ratings chosen from the Jester data set [20]. Given the original rating matrix, we randomly set some nonzero elements to "0" (denoting missing values). Let σ_R denote the ratio of remaining nonzero elements to original nonzero elements. By this way, we can observe how the recommendation quality changes with the number of ratings. Detailed information about the rating data will be presented in Section VII.

Experiment results are stored in a matrix $\mathbf{Q} = [q_{ij}]$, where each column represents a user, each row represents a particular value of σ_R, and q_{ij} denotes the corresponding recommendation quality. Fig. 1 shows the matrix $\mathbf{Q}$ obtained by applying a user-based CF algorithm, and the recommendation quality is measured by the difference between the predicated ratings and the user's true preferences. Details of the recommendation algorithm and the evaluation metric of recommendation quality will be presented in Section III. As we can see, as more ratings are available (σ_R increases), the recommendation quality improves.

To better illustrate the change of recommendation quality, for each value of σ_R we compute the average of the recommendation quality perceived by all users. Fig. 2 shows the results obtained under different settings of recommendation algorithms and evaluation metrics. It is clear that for any given recommendation algorithm, the recommendation quality improves with σ_R. Suppose that each user has an expectation for the recommendation quality, then from Fig. 2 we can learn that, if users do not have high expectations, a relatively small number of ratings (e.g., $\sigma_R = 0.5$) will be enough to generate satisfying recommendations. With these preliminary results, we can proceed to formal study of the user participation problem.

III. System Model

Consider a CF system where a set of users $\mathcal{N} = \{1, 2, \dots, N\}$ interact with an RS. The RS maintains information about a set of items $S = \{s_1, s_2, \dots, s_M\}$. Each user experiences a set of items and assigns ratings to some of them.

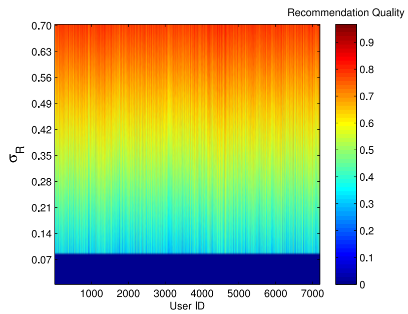
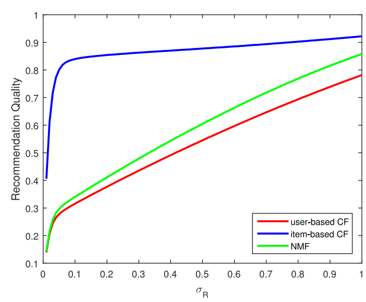
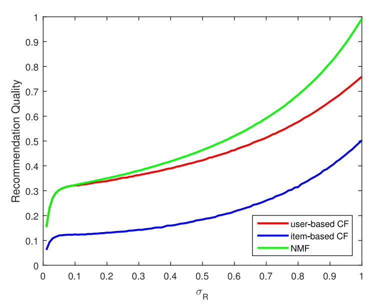


Fig. 1. Illustration of the experiment result $\mathbf{Q} = [q_{ij}]$. Each row of $\mathbf{Q}$ corresponds to a given value of σ_R ($\sigma_R = (70/100) \cdot (k/100), k = 1, 2, \ldots, 100$). Each column of $\mathbf{Q}$ corresponds to a user. The element q_{ij} represents the corresponding recommendation quality. Different values of q_{ij} are indicated by different colors: blue represents low quality, red represents high quality.



(a)



(b)

Fig. 2. Recommendation quality changes with the number of ratings. Each curve corresponds to one of the following three recommendation algorithms: user-based CF [21], item-based CF [22], and non-negative matrix factorization [23]. (a) Recommendation quality is evaluated by the difference between the predicated ratings and the user's true preferences [see (4)]. (b) Recommendation quality is evaluated by the overlap between the recommended items and the items that the user is mostly interested in [see (3)].

Let S_i and $\tilde{S}_i$ denote the set of items that user i has experienced and rated, respectively, then we have $\tilde{S}_i \subseteq S_i \subseteq S$. From the perspective of the RS, a rating vector $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iM})$ is provided by user i when a set $\tilde{S}_i$ is chosen. We define $r_{ij} \in (0, r_{\max}]$ if $s_j \in \tilde{S}_i$, $r_{ij} = 0$ if $s_j \notin \tilde{S}_i$ $(j = 1, \ldots, M)$. Usually, a high value of r_{ij} implies user i has a strong preference for item s_j.

As mentioned in Section I, the reason that the user will not rate all the items in S_i is the time consumption and the privacy loss incurred by rating. In order to protect privacy, the user can provide fake ratings to the RS [24], [25], so that the true preferences of the user will not be disclosed. However, considering that the recommendation quality will be hurt by fake ratings, falsifying ratings can be nontrivial and time-consuming. Also, in practical recommendation systems, the profile of user interest is often represented by some kind of distribution over different types of items [26], which means the values of ratings have little influence on user profile. What matters more is whether the user has rated an item. Hence in this paper, we assume that as long as the user decides to rate an item, the user will provide a rating that coincides with his true preference.

The ratings provided by all users form a rating matrix $\mathbf{R} = [r_{ij}]_{N\times M}$. The RS applies some recommendation algorithm to $\mathbf{R}$ to predict users' preferences for those unrated items. A recommendation vector $\hat{\mathbf{r}}_i = (\hat{r}_{i1}, \ldots, \hat{r}_{iM})$ is computed for each user i, where $\hat{r}_{ij}$ is defined as follows:

$$\hat{r}_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} \neq 0 \\ f_{ij}(\mathbf{R}), & \text{if } r_{ij} = 0 \end{cases} \tag{1}$$

with $f_{ij}(\mathbf{R})$ being the predicted rating determined by both the recommendation algorithm and the whole ratings. For example, if user-oriented neighborhood-based CF [21] is applied, then $f_{ij}(\mathbf{R})$ can be defined as

$$f_{ij}(\mathbf{R}) = \frac{\sum\limits_{k\in\text{Neighbor}(i)} r_{kj} F_{\text{sim}}(i,k)}{\sum\limits_{k\in\text{Neighbor}(i)} F_{\text{sim}}(i,k)} \tag{2}$$

where $F_{\text{sim}}(i,k)$ represents the similarity between user i and user k, Neighbor(i) represents the set of users who are most similar to user i. The similarity $F_{\text{sim}}(i,k)$ can be measured by Pearson correlation or vector cosine similarity [2].

After computing the recommendation vector, generally the RS will select several items with high $f_{ij}(\mathbf{R})$ and recommend them to the user. Then the user can evaluate whether the recommended items match his interest. Let $\mathbf{p}_i = (p_{i1}, \ldots, p_{iM})$ denote user i's interest, where p_{ij} represents user i's true preference for item s_j $(j = 1, \ldots, M)$. We assume $0 \leq p_{ij} \leq r_{\max}$ and define $p_{ij} = r_{ij}$ for $s_j \in \tilde{S}_i$. Let $\hat{S}_i$ denote the set of K items recommended by the RS. Let $\breve{S}_i$ denote the set of K items that correspond to the K highest p_{ij} in the set $S\backslash S_i$. That is, $\breve{S}_i$ denote the set of items that user i has not experienced yet but is interested in. Then the quality of the recommendation result $\hat{S}_i$, denoted as QoR$(\hat{S}_i)$, can be defined as

$$\text{QoR}\left(\hat{S}_i\right) = \frac{\left|\hat{S}_i \cap \breve{S}_i\right|}{K} \tag{3}$$

where $|A|$ denotes the cardinality of the set A.

In the study of recommendation systems, the recommendation quality is often evaluated by mean absolute error or root mean squared error (RMSE) [2]. Based on the definition of RMSE, we assume that the RS returns the whole vector $\hat{\mathbf{r}}_i$ to the user, and the quality of $\hat{\mathbf{r}}_i$ is evaluated by a user-specific function $g_i : \mathbb{R}^M \to \mathbb{R}$ which is defined as

$$g_i\left(\hat{\mathbf{r}}_i\right) = 1 - \frac{\sqrt{\sum\limits_{j=1}^{M}\left(\hat{r}_{ij} - p_{ij}\right)^2}}{r_{\max}\sqrt{M}}. \tag{4}$$

A large $g_i(\hat{\mathbf{r}}_i)$ implies high similarity between $\mathbf{r}_i$ and $\mathbf{p}_i$, namely high recommendation quality. In the subsequent analysis, we mainly use (4) as the metric of recommendation quality.

From (1), (3), and (4), we can see that the recommendation quality obtained by one user is affected by other users' ratings. In other words, users in a CF system interact with each other via providing ratings to the RS. In the following section, we will use satisfactory game to formulate the interaction among users.

IV. Satisfactory Game Formulation

A. Players and Actions

We consider all the users in $\mathcal{N}$ as players and the set $\tilde{S}_i$ as user i's action, i.e., $a_i = \tilde{S}_i$. Let $\mathcal{A}_i$ denote the action space of user i. All users share the same action space, i.e., for any $i \in \mathcal{N}$, there is $\mathcal{A}_i = (A^{(1)}, \ldots, A^{(K)})$, where $K = 2^{|S|} - 1$, $A^{(k)} \subseteq S$ $(k = 1, \ldots, K)$ and $A^{(k)} \neq \varnothing$. When choosing an action, each user follows his own probability distribution over the action space. We use $\boldsymbol{\pi}_i = (\pi_i^{(1)}, \ldots, \pi_i^{(K)})$ to denote the distribution, where $\pi_i^{(k)} \triangleq \Pr(a_i = A^{(k)})$ represents the probability that user i chooses the action $A^{(k)}$.

Given an action profile $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}$ $(\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N)$, the rating matrix $\mathbf{R}$ obtained by the RS is determined. Considering that the recommendation $\hat{\mathbf{r}}_i$ is fully determined by $\mathbf{R}$ when the recommendation algorithm is specified, we introduce a mapping $h_i : \mathcal{A} \to \mathbb{R}$ to show the influence of users' actions on recommendation quality

$$g_i\big(\hat{\mathbf{r}}_i\big) = h_i(\mathbf{a}) = h_i(a_i, \mathbf{a}_{-i}) \tag{5}$$

where $\mathbf{a}_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N) \in \mathcal{A}_{-i}$, $\mathcal{A}_{-i} = \mathcal{A}_1 \times \cdots \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \cdots \times \mathcal{A}_N$.

Intuitively, either the user i himself or other users rate more items, the rating matrix will become less sparse, and user i can get better recommendations. We introduce the notion of *rating completeness* to measure the relative amount of ratings provided by the user. Given the set S_i, user i's rating completeness σ_i is defined as

$$\sigma_i = \frac{|a_i|}{|S_i|}. \tag{6}$$

Notice that $a_i \subseteq S_i$ and $a_i \neq \varnothing$, hence $0 < \sigma_i \leq 1$. A large σ_i means user i actively participates in the rating activity. We use σ_{-i} to denote the average of other users' rating completeness

$$\sigma_{-i} = \frac{1}{N-1} \sum_{j \in \mathcal{N},\ j \neq i} \sigma_j. \tag{7}$$

By introducing σ_i and σ_{-i}, we can rewrite $h_i(a_i, \mathbf{a}_{-i})$ as

$$h_i(a_i, \mathbf{a}_{-i}) = h(\sigma_i, \sigma_{-i}; \mathbf{p}_i) \tag{8}$$

where the function $h(\cdot; \mathbf{p}_i)$ with parameter $\mathbf{p}_i$ takes σ_i and σ_{-i} as input.

As mentioned in Section I, rating items incurs some cost. The more items the user rates, the higher cost he has to pay. Let $c_i(a_i)$ denote the cost paid by user i when he chooses the action a_i, then for any $a'_i \in \mathcal{A}_i$, $a''_i \in \mathcal{A}_i$, if $a'_i \subset a''_i$, there is $c_i(a'_i) < c_i(a''_i)$.

B. Satisfaction Form

Due to the rating cost, usually the user will not rate all the items he has experienced. As we have discussed in Section II, given the recommendation algorithm, the evaluation metric of recommendation quality, and the items that users have experienced, the recommendation quality perceived by every user increases with the number of ratings provided by users. This means that when every user has rated all the items he has experienced, i.e., each user i chooses the action $a_i^* \triangleq S_i$, every user can receive the best recommendation that he can get. In such a case, the rating completeness of every user is 1. If we use $\Gamma_i^{\max}$ to denote the best recommendation quality then there is

$$\Gamma_i^{\max} = h(1, 1; \mathbf{p}_i). \tag{9}$$

In most cases, the rating completeness of a user is less than 1, hence the best result $\Gamma_i^{\max}$ can hardly be realized. Suppose that each user i has a relatively low expectation Γ_i $(\Gamma_i < \Gamma_i^{\max})$ for the recommendation quality. Given an action profile $\mathbf{a}$, as long as $h_i(\mathbf{a}) \geq \Gamma_i$, user i will be *satisfied*.

From (5) we know that, given the actions of other users, certain actions should be chosen by user i so that user i can get satisfying recommendations. We use $f_i(\mathbf{a}_{-i})$ to denote the set of such actions

$$f_i(\mathbf{a}_{-i}) = \{a_i \in \mathcal{A}_i : h_i(a_i, \mathbf{a}_{-i}) \geq \Gamma_i\}. \tag{10}$$

For any $\mathbf{a}_{-i} \in \mathcal{A}_{-i}$, the mapping $f_i : \mathcal{A}_{-i} \to 2^{\mathcal{A}_i}$ determines the actions available for user i to satisfy his expectation. It should be noted that, for some $\mathbf{a}_{-i}$, $f_i(\mathbf{a}_{-i})$ may be empty. For example, suppose that each user in $\mathcal{N}$, expect user i, rates only one item. Then even if user i rates all the items he has experienced, the ratings are not enough to reflect the real similarities between users. Consequently, user i cannot get satisfying recommendations.

Based on the above discussions, we can describe the proposed game by the following triplet:

$$\hat{G}_{\mathrm{CF}} = \big(\mathcal{N}, \{\mathcal{A}_i\}_{i \in \mathcal{N}}, \{f_i\}_{i \in \mathcal{N}}\big). \tag{11}$$

This formulation of game is called *satisfaction form*, which was first introduced by Perlaza *et al.* [18] to model the problem of quality-of-service provisioning in decentralized self-configuring networks.

C. Satisfaction Equilibrium

An important outcome of a game in satisfaction form is the one where all players are satisfied. This outcome is referred to as SE [18]

Definition 1 (Satisfaction Equilibrium): An action profile $\mathbf{a}^+$ is an equilibrium for the game $\hat{G}_{\mathrm{CF}} = (\mathcal{N}, \{\mathcal{A}_i\}_{i \in \mathcal{N}}, \{f_i\}_{i \in \mathcal{N}})$, if $\forall i \in \mathcal{N}$, there is $a_i^+ \in f_i(\mathbf{a}_{-i}^+)$.

We have assumed that for all $i \in \mathcal{N}$, there is $\Gamma_i < \Gamma_i^{\max}$, hence the action profile $\mathbf{a}^* \triangleq (S_1, S_2, \ldots, S_N)$ is an SE of the proposed game. However, $\mathbf{a}^*$ requires every user to pay the highest cost $c_i(S_i)$, which may exceed the necessary cost for achieving user's expectation. It is more practical to find a lower-cost SE $\mathbf{a}^+ = (a_1^+, \ldots, a_N^+)$ which satisfies:

1) $\forall i \in \mathcal{N}$, there is $a_i^+ \in f_i(\mathbf{a}_{-i}^+)$ and $c_i(a_i^+) \leq c_i(S_i)$;

2) there is at least one user who does not have to provide his complete ratings, that is, $\exists i \in \mathcal{N}$, $c_i(a_i^+) < c_i(S_i)$.

V. Learning Satisfaction Equilibrium

The game described above is a game with incomplete information, since each user has no knowledge of other users' actions. Different from general equilibrium concepts of games with complete information, the SE is obtained as the result of a learning process, rather than the result of rational thinking on players' beliefs and observations [19]. In this section, we study the behavior rule that allows users to learn a SE. The equilibrium learning is essentially an iterative process of information exchange between users and the RS. For the RS, the iterative process provides a way to acquire a certain amount of information to build a profile for a user [27], [28]. During the learning process, each user chooses his actions as follows.

Initially, user i chooses an action $a_i(0)$ based on the probability distribution $\boldsymbol{\pi}_i(0) = (\pi_i^{(1)}(0), \ldots, \pi_i^{(K)}(0))$, where for any $k \in \{1, \ldots, K\}$, $\pi_i^{(k)}(0)$ is defined as follows:

$$\pi_i^{(k)}(0) = \begin{cases} \beta_i(0)/\alpha^{c_i(A^{(k)})}, & \text{if } A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where parameter $\alpha > 1$ shows how much users care about the cost. A large α means it is more likely that the user will rate a small number of new items (i.e., unrated items) in one iteration. On the other hand, if α is small, users may provide sufficient ratings in a few iterations, which means an SE can be quickly achieved. The normalization factor $\beta_i(0)$ is defined as

$$\beta_i(0) = \frac{1}{\sum\limits_{k\,:\,A^{(k)} \subseteq S_i} \alpha^{-c_i(A^{(k)})}}. \tag{13}$$

After every user has chosen his action, the RS computes the recommendations based on the initial rating matrix $\mathbf{R}(0)$ and returns $\hat{\mathbf{r}}_i(0)$ to user i.

At the beginning of iteration n ($n = 1, 2, \ldots$), user i evaluates $\hat{\mathbf{r}}_i(n-1)$ to see whether it is satisfactory. We use a binary variable $v_i(n-1)$ to indicate the evaluation result

$$v_i(n-1) = \begin{cases} 1, & \text{if } g_i\big(\hat{\mathbf{r}}_i(n-1)\big) \geq \Gamma_i \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

According to $v_i(n-1)$, user i updates the probability distribution $\boldsymbol{\pi}_i(n) = (\pi_i^{(1)}(n), \ldots, \pi_i^{(K)}(n))$ and then chooses an action $a_i(n)$. Notice that the RS utilizes all the historical ratings of a user to compute recommendations. Even if the user does not rate any item in this iteration, the RS can still compute recommendations for him based on the ratings that the user has provided in previous iterations. Therefore, we use $a_i(n)$ to denote all the items that user i has rated by the end of iteration n, and naturally we have $a_i(n) \supseteq a_i(n-1)$.

If $v_i(n-1) = 0$, then user i may: 1) choose more items to rate, if he believes it is because he did not provide enough ratings that the recommendation result is unsatisfactory and 2) keep previous action, i.e., rate no more items, if he blames the unsatisfying result on other users. For any $k \in \{1, \ldots, K\}$, $\pi_i^{(k)}(n) \triangleq \Pr(a_i(n) = A^{(k)})$ is computed as follows:

$$\pi_i^{(k)}(n) = \begin{cases} \sigma_i(n-1), & \text{if } A^{(k)} = a_i(n-1) \\ \beta_i(n)/\alpha^{c_i(A^{(k)})}, & \text{if } a_i(n-1) \subset A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $\sigma_i(n-1)$ is the rating completeness of user i

$$\sigma_i(n-1) = \frac{|a_i(n-1)|}{|S_i|}. \tag{16}$$

A large $\sigma_i(n-1)$ means user i has already rated many items in S_i, thus the user possibly rates no more items even if he is not satisfied with current recommendation. The normalization factor $\beta_i(n)$ is defined as follows:

$$\beta_i(n) = \frac{1 - \sigma_i(n-1)}{\sum\limits_{k: a_i(n-1) \subset A^{(k)} \subseteq S_i} \alpha^{-c_i(A^{(k)})}}. \tag{17}$$

If $v_i(n-1) = 1$, then it is very likely that user i no longer rates the rest items in S_i. For any $k \in \{1, \ldots, K\}$, $\pi_i^{(k)}(n)$ is now defined as follows:

$$\pi_i^{(k)}(n) = \begin{cases} \mu, & \text{if } A^{(k)} = a_i(n-1) \\ \beta_i(n)/\alpha^{c_i(A^{(k)})}, & \text{if } a_i(n-1) \subset A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where the parameter μ denotes to what extent a satisfied user would keep the previous action, and usually there is $0.5 < \mu \leq 1$. The normalization factor $\beta_i(n)$ is defined as follows:

$$\beta_i(n) = \frac{1 - \mu}{\sum\limits_{k: a_i(n-1) \subset A^{(k)} \subseteq S_i} \alpha^{-c_i(A^{(k)})}}. \tag{19}$$

After every user has chosen his action, the RS computes the recommendations based on the rating matrix $\mathbf{R}(n)$ and returns $\hat{\mathbf{r}}_i(n)$ to user i. The learning process goes to the next iteration. If after a finite number of iterations, say n_s, all users have been satisfied, then the process stops. We say the behavior rule converges to an SE $\mathbf{a}^+ = (a_1(n_s), \ldots, a_N(n_s))$. A summary of the learning process is shown in Algorithm 1.

VI. Convergence of the Learning Algorithm

In this section, we study the convergence of learning algorithm proposed in the previous section. First, we introduce the basic assumption for the convergence analysis and the notion of *user state*. Then we present a simple analysis of the convergence. After that, we make some simplifications of the learning algorithm and present a quantitative analysis of the convergence.

A. Basic Assumption

The learning algorithm proposed above implies the following assumption we make about the relationship between the rating completeness and the recommendation quality.

Assumption 1: $\forall i \in \mathcal{N}$, the following two conditions hold for all $\sigma_i \in (0, 1]$ and $\sigma_{-i} \in (0, 1]$.

1) $([\partial h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)]/[\partial \sigma_i]) > 0$.
2) $([\partial h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)]/[\partial \sigma_{-i}]) \geq 0$.

This assumption indicates that the recommendation quality perceived by one user can be improved by either the user

Algorithm 1 Learning the SE of the Game $\hat{G}_{\text{CF}} = \left(\mathcal{N}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{f_i\}_{i\in\mathcal{N}}\right)$

$n = 0$;

$\forall k \in \{1, \cdots, K\}$,
$$\pi_i^{(k)}(0) = \begin{cases} \beta_i(0)/\alpha^{c_i\left(A^{(k)}\right)}, & \textit{if } A^{(k)} \subseteq S_i \\ 0, & \textit{otherwise} \end{cases},$$
where $\beta_i(0) = \frac{1}{\sum_{k:\ A^{(k)}\subseteq S_i} \alpha^{-c_i\left(A^{(k)}\right)}}$.

$a_i(0) \sim \boldsymbol{\pi}_i(0)$;

for all $n > 0$ **do**

update $\boldsymbol{\pi}_i(n)$: $\forall k \in \{1, \cdots, K\}$,
$$\pi_i^{(k)}(n) = \begin{cases} \gamma_i(n), & \textit{if } A^{(k)} = a_i(n-1) \\ \beta_i(n)/\alpha^{c_i\left(A^{(k)}\right)}, & \textit{if } a_i(n-1) \subset A^{(k)} \subseteq S_i \\ 0, & \textit{otherwise} \end{cases},$$
where
$$\gamma_i(n) = \begin{cases} \sigma_i(n-1), & \textit{if } v_i(n-1) = 0 \\ \mu, & \textit{if } v_i(n-1) = 1 \end{cases},$$
$$\beta_i(n) = \frac{1-\gamma_i(n)}{\sum_{k:\ a_i(n-1)\subset A^{(k)}\subseteq S_i} \alpha^{-c_i\left(A^{(k)}\right)}}.$$

$a_i(n) \sim \boldsymbol{\pi}_i(n)$;

end for

himself or other users. During the learning process, unsatisfied users continually provide more ratings. For an unsatisfied user i, even if σ_{-i} no longer increases, so long as σ_i increases with iterations, the recommendation quality gradually improves, and the user may get a satisfying result after a few iterations. If the assumption does not hold, then it is impossible for the learning algorithm to achieve an SE if someone is unsatisfied with the initial recommendation. And in such a case, rating more items only increases the user's cost, and the user will lose his motivation to participate. Notice that the function $h(\cdot, \cdot; \mathbf{p}_i)$ is actually determined by the recommendation algorithm and the evaluation metric of recommendation quality. Hence we assume that given the evaluation metric of recommendation quality, the recommendation algorithm adopted by the RS should satisfy the above assumption. We have demonstrated the rationality of the assumption via simulation results on real data set (see Section VII-C).

B. User State

Based on Assumption 1, given a user's expectation Γ_i, the relationship between σ_i and σ_{-i} can be depicted by a curved section in the $\sigma_i - \sigma_{-i}$ plane. As shown in Fig. 3, only when both σ_i and σ_{-i} exceed the corresponding thresholds, it is possible that user i will be satisfied. The two thresholds $\sigma_{i,\min}$ and $\sigma_{-i,\min}$ are determined by the following two equations, respectively:

$$h\left(\sigma_{i,\min}, 1; \mathbf{p}_i\right) = \Gamma_i \tag{20}$$
$$h\left(1, \sigma_{-i,\min}; \mathbf{p}_i\right) = \Gamma_i. \tag{21}$$

According to Assumption 1, if $\sigma_i < \sigma_{i,\min}$, then for any $\sigma_{-i} \in (0, 1]$, there is

$$h(\sigma_i, \sigma_{-i}; \mathbf{p}_i) < h\left(\sigma_{i,\min}, \sigma_{-i}; \mathbf{p}_i\right) \leq h\left(\sigma_{i,\min}, 1; \mathbf{p}_i\right). \tag{22}$$

Similarly, if $\sigma_{-i} < \sigma_{-i,\min}$, then for any $\sigma_i \in (0, 1]$, there is

$$h(\sigma_i, \sigma_{-i}; \mathbf{p}_i) < h\left(\sigma_i, \sigma_{-i,\min}; \mathbf{p}_i\right) \leq h\left(1, \sigma_{-i,\min}; \mathbf{p}_i\right). \tag{23}$$

Therefore, given Γ_i, $\sigma_{i,\min}$ represents the minimum requirement for user i and $\sigma_{-i,\min}$ represents the minimum requirement for other users.

During the learning process, each user's rating completeness increases with iterations. We define $\sigma_i(n-1)$ and $\sigma_{-i}(n-1)$ as follows:

$$\sigma_i(n-1) = \frac{|a_i(n-1)|}{|S_i|} \tag{24}$$
$$\sigma_{-i}(n-1) = \frac{1}{N-1} \sum_{j\in\mathcal{N},\ j\neq i} \frac{\left|a_j(n-1)\right|}{\left|S_j\right|}. \tag{25}$$

For any $n \geq 1$, there is $\sigma_i(n) \geq \sigma_i(n-1)$ and $\sigma_{-i}(n) \geq \sigma_{-i}(n-1)$. We assume that there exists some n_0 ($n_0 \geq 1$) that $\sigma_i(n_0) \geq \sigma_{i,\min}$ holds for all i. From iteration $n_0 + 1$, each user i is in one of the following three states.

1) *Satisfied:* As depicted by the green area in Fig. 3, user i has already got satisfying recommendations, namely $h(\sigma_i(n-1), \sigma_{-i}(n-1); \mathbf{p}_i) \geq \Gamma_i$. Once the user is satisfied, he will always in the *Satisfied* state. The reason is that with the increase of iterations, σ_i and σ_{-i} either increase or remain the same, and according to Assumption 1, $h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)$ will not decrease.
2) *Proximity to Satisfied:* As depicted by the cyan area in Fig. 3, user i has not been satisfied, namely $h(\sigma_i(n-1), \sigma_{-i}(n-1); \mathbf{p}_i) < \Gamma_i$, and user i has not rated all the items in S_i, namely $\sigma_i(n-1) < 1$, while other users have rated enough items, namely $\sigma_{-i}(n-1) \geq \sigma_{-i,\min}$. In this case, even if other users no longer rate more items, user i is able to enter the *Satisfied* state by rating more items.
3) *Far From Satisfied:* As depicted by the yellow area in Fig. 3, user i has not been satisfied, and the amount of ratings provided by other users has not achieved the minimum requirement of user i, namely $\sigma_{-i}(n-1) < \sigma_{-i,\min}$. In this case, if other users provide enough ratings in subsequent iterations, user i can enter the *Proximity to satisfied* state. Otherwise, the user will stuck in this state and never be satisfied.

We use Z_S, Z_P, and Z_F to denote the three states, respectively, and we use $z_i(n)$ to denote user i's state at the beginning of iteration n ($n \geq n_0$), then $z_i(n) \in \{Z_S, Z_P, Z_F\}$.

C. Simple Analysis of the Convergence

At the beginning of iteration n ($n \geq n_0$), users can be grouped into two sets: 1) the set of satisfied users $\mathcal{N}_S(n) \triangleq \{i | i \in \mathcal{N},\ z_i(n) = Z_S\}$ and 2) the set of unsatisfied users $\mathcal{N}_{US}(n) \triangleq \{i | i \in \mathcal{N},\ z_i(n) = Z_P \vee z_i(n) = Z_F\}$. As the learning process continues, the number of unsatisfied users decreases. For any user $i \in \mathcal{N}_{US}(n)$:

1) if $z_i(n) = Z_P$, then according to the definition of the state Z_P, the user will eventually become satisfied;
2) if $z_i(n) = Z_F$, there is $\sigma_{-i}(n-1) < \sigma_{-i,\min}$.

When users continue to provide more ratings after they are satisfied, namely $\mu < 1$, both $\sigma_{-i}(n-1)$ and $\sigma_i(n-1)$ keep increasing with n, hence at some iteration n' ($n' > n$), there will be $z_i(n') = Z_P$. However, when users in $\mathcal{N}_S(n)$ make no contributions to the increase of $\sigma_{-i}(n-1)$, namely $\mu = 1$, it is possible that the user will stay in the state Z_F permanently. Consider the case that the satisfied users have provided few

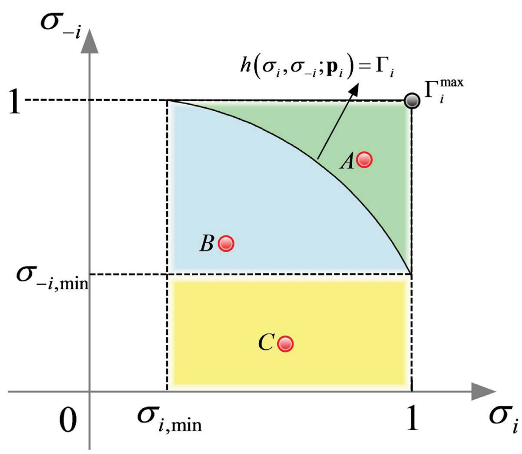


Fig. 3. Illustration of user state: *satisfied* (cyan area), *proximity to satisfied* (blue area), and *far from satisfied* (yellow area).

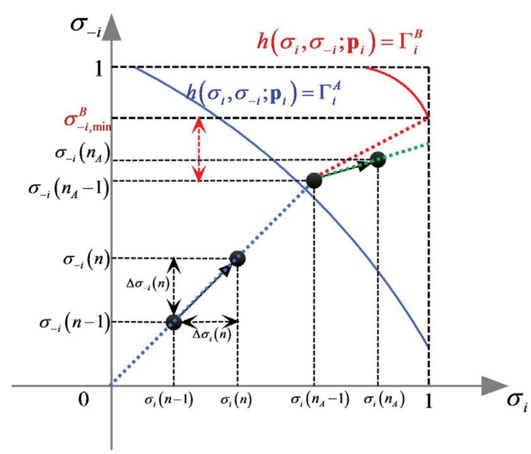


Fig. 4. Illustration of how user state changes with the rating completeness.

ratings that σ_{-i} cannot reach $\sigma_{-i,\min}$ even when all the unsatisfied users provide their complete ratings. In such a case, user i will never be satisfied.

To sum up, given $\mu = 1$, if the following inequality holds for some $i \in \mathcal{N}$ and some $n \in \{1, 2, \ldots\}$, then the learning algorithm cannot converge:

$$\frac{1}{N-1}\left[\sum_{j\in\mathcal{N}_S(n)}\sigma_j(n-1) + \sum_{j\in\mathcal{N}_{US}(n),\ j\neq i} 1\right] < \sigma_{-i,\min}. \quad (26)$$

Notice that $\sigma_{-i,\min}$ is determined by user i's expectation Γ_i. According to (26), we can conclude that if a *small* portion of users have relatively *high* expectations for the recommendation quality, then the proposed learning algorithm cannot converge to an SE. Next we will present an elaborate analysis of this conclusion.

D. Quantitative Analysis of the Convergence

From above discussion we can see that, to judge the convergence of the learning algorithm, the key is to analyze how each user's rating completeness changes over time. Let $\Delta\sigma_i(n)$ denote the increment of user i's rating completeness in iteration n, namely $\Delta\sigma_i(n) = \sigma_i(n) - \sigma_i(n-1)$. According to Algorithm 1, the value of $\Delta\sigma_i(n)$ is random. Hence it is difficult to quantitatively analyze the transition of user state. In this section, we simplify the learning process described in Algorithm 1 and discuss the convergence of the simplified version.

1) Simplified Learning Algorithm: The simplified learning process can be described as follows. Initially, each user i randomly chooses one item from S_i, thus for any $i \in \mathcal{N}$, $\sigma_i(0) = (1/|S_i|)$. At the beginning of iteration n ($n > 0$), each user i judges whether the recommendation quality is satisfactory. If the user is satisfied, then he does not change the action, namely $a_i(n) = a_i(n-1)$; if the user is unsatisfied, then he randomly chooses one item from $S_i \backslash a_i(n-1)$.

Based on above simplification, we can easily determine the value of $\Delta\sigma_i(n)$: if $i \in \mathcal{N}_S(n)$, then $\Delta\sigma_i(n) = 0$; if $i \in \mathcal{N}_{US}(n)$, then $\Delta\sigma_i(n) = (1/|S_i|)$.

2) Two Types of Users: In addition to simplifying the learning process, we also make some assumptions about users.

Assumption 2: Users in $\mathcal{N}$ can be divided into two groups $\mathcal{N}_A$ and $\mathcal{N}_B$.

1) For all $i \in \mathcal{N}_A$, $\Gamma_i = \eta_A \Gamma_i^{\max}$, where $0 < \eta_A < 1$.
2) For all $i \in \mathcal{N}_B$, $\Gamma_i = \eta_B \Gamma_i^{\max}$, where $\eta_A < \eta_B < 1$.
3) For all $i \in \mathcal{N}$, $|S_i| = M_0$, where M_0 is constant and $1 \le M_0 < |S|$.

3) Quantify the Change of Rating Completeness: With above simplifications, we can now quantitatively analyze when the transition of user state happens and explain why users in $\mathcal{N}_B$ may stay unsatisfied. Consider a user $i \in \mathcal{N}_B$. As shown in Fig. 4, with the increase of iterations, the user "moves" upward and/or rightward in the square $[0, 1]^2$. To judge whether the user can enter the *Satisfied* area which is determined by η_B, we need to analyze how the user moves from $(\sigma_i(n-1), \sigma_{-i}(n-1))$ to $(\sigma_i(n), \sigma_{-i}(n))$ in each iteration n.

According to the simplified algorithm, at the beginning of iteration 1, user i is at the point $([1/M_0], [1/M_0])$. Then, the user moves along the line $\sigma_{-i} = \sigma_i$ (the blue dotted line in Fig. 4) until users in $\mathcal{N}_A$ are satisfied. This is because before anyone is satisfied, for all $i \in \mathcal{N}$, both $\Delta\sigma_i(n)$ and $\Delta\sigma_{-i}(n)$ equal to $(1/M_0)$. Users in $\mathcal{N}_A$ have relatively low expectations, thus they become satisfied earlier than users in $\mathcal{N}_B$. There exists some $n_A \in \{1, 2, \ldots, M_0 - 1\}$ that at the beginning of iteration n_A, all the users in $\mathcal{N}_A$ are satisfied, while all the users in $\mathcal{N}_B$ are unsatisfied.

At the beginning of iteration n_A, user i is at the point $([n_A/M_0], [n_A/M_0])$. As we have discussed in Section VI-C, if user i is in the state Z_P, then he can be satisfied by rating more items. Here we focus on the other case, namely $z_i(n_A) = Z_P$. During iteration n_A, each user in $\mathcal{N}_B$ rates one more item, while users in $\mathcal{N}_A$ no longer rate more items, hence user i moves along a line whose slope k_B is less than 1

$$\begin{aligned} k_B &= \frac{\Delta\sigma_{-i}(n_A)}{\Delta\sigma_i(n_A)} \\ &= \frac{\frac{1}{N-1}\sum_{j\in\mathcal{N}_B, j\neq i}\left[\sigma_j(n_A) - \sigma_j(n_A-1)\right]}{\frac{1}{M_0}} \\ &= \frac{|\mathcal{N}_B| - 1}{N-1}. \end{aligned} \quad (27)$$

In subsequent iterations, user i moves along the same direction until one of the following two situations happens: 1) user i becomes satisfied and 2) user i has not been satisfied but has rated all the items in S_i, namely $\sigma_i = 1$. If the second situation happens, user i can never be satisfied. This is because that users in $\mathcal{N}_B$ are assumed to have same expectations, which implies all the other users in $\mathcal{N}_B$ also have provided their complete ratings. As a result, no user can make contributions to the increase of σ_{-i}. As depicted by the green dotted line in Fig. 4, the second situation will happen if k_B is smaller than some threshold $k_{\min}$ (see the red dotted line in Fig. 4)

$$k_{\min} = \frac{\sigma_{-i,\min} - \sigma_{-i}(n_A - 1)}{1 - \sigma_i(n_A - 1)} = \frac{\sigma_{-i,\min} - \frac{n_A}{M_0}}{1 - \frac{n_A}{M_0}}. \quad (28)$$

Plugging (27) and (28) into $k_B < k_{\min}$ we can get

$$\sigma_{-i,\min} > \frac{|\mathcal{N}_B| - 1}{N - 1} + \frac{N - |\mathcal{N}_B|}{N - 1} \cdot \frac{n_A}{M_0}. \quad (29)$$

The right part of above inequality is exactly the formula for calculating σ_{-i} when users in $\mathcal{N}_B$ provide their complete ratings and users in $\mathcal{N}_A$ provide the necessary amount of ratings to make themselves satisfied. Similar with the conclusion we have drawn in Section VI-C, the inequality implies that if a user has very high expectation which requires too much effort of other users, then the user cannot get satisfying recommendations. If the relationship between σ_i, σ_{-i}, η_A, and η_B can be explicitly expressed, then we can rewrite $\sigma_{-i,\min}$ in a specific form, and the influence of users' expectations on the convergence of the learning algorithm can be shown more clearly. Please see the Appendix for more details.

VII. Simulation

To verify the feasibility of the proposed SE learning algorithm, we have conducted a series of simulations by using real rating data. In this section, we first describe the preparation of data and experiment setup, then we present a comparison of the learning results which are obtained under different settings of users' expectations. After that, we provide some experimental proofs for the assumptions we have made and for the theoretical analysis presented in Section VI. In addition, to demonstrate the satisfactory game analysis can help to design incentive mechanisms for user participation, we conduct experiments to investigate how monetary rewards affect the result of equilibrium learning.

A. Data Set and Parameter Setting

Two data sets, namely Jester [20] and MovieLens,[2] are chosen for simulation. These two data sets are commonly used in the study of CF. Details of the data sets and corresponding parameter settings are given below.

1) Jester: The Jester data set contains about 4.1 million ratings of 100 jokes from 73 421 users. Considering that the "ground truth" of a user's preference for each item is required for the evaluation of recommendation quality, we only keep 720 000 ratings from the 7200 users who have rated all the 100 jokes. Ratings are real values ranging from −10.00 to +10.00 (the value "99" corresponds to "unrated"). As described in Section III, we have defined $0 \leq r_{ij} \leq r_{\max}$, so we adjust the ratings to the range [10.00, 30.00] and use 0 to represent "not rated." Finally, we get a user-item matrix $\mathbf{R} = [r_{ij}]_{7200\times 100}$ which contains no zero elements.

Parameters of the SE learning algorithm are set as follows.

1) $\mathbf{p}_i$*:* Each row of $\mathbf{R}$ is treated as the corresponding user's interest vector.
2) S_i*:* For each user i, we set $|S_i| = 70$ and randomly set 30% of the user's ratings to 0. The resulting rating matrix is denoted by $\mathbf{R}'$.
3) $g_i(\hat{\mathbf{r}}_i)$*:* The quality of the recommendation $\hat{\mathbf{r}}_i$ is evaluated according to (4).
4) $c_i(a_i)$*:* The rating cost is defined as the number of rated items, that is, $c_i(a_i) = |a_i|$.
5) α*:* This parameter affects the convergence speed. Considering the shape of the function $f(x) = 1/\alpha^x$ on the interval $[0, |S_i|]$, we set $\alpha = 1.2$.
6) μ*:* We set $\mu = 0.9$ and $\mu = 1$ to simulate the situation that satisfied users continue to provide ratings and the situation that satisfied users no longer rate more items, respectively.

We have implemented a user-based CF algorithm with MATLAB. Unknown ratings are predicted according to (2) where $|\text{Neighbor}(i)|$ is set to 36. The quality of recommendations is evaluated according to (4). Based on $\mathbf{R}$ and $\mathbf{R}'$, we calculate the best result $\Gamma_i^{\max}$, then we set $\Gamma_i = \eta_i \Gamma_i^{\max}$, where $0 < \eta_i < 1$. Settings of $\{\eta_i\}_{i=1}^N$ will be described later.

2) MovieLens: The MovieLens data set contains about 1 million ratings from 6040 users on 3900 movies. To conduct simulations, we set $|S_i| = 70$ and discard users who rated less than 70 movies. The resulting data set consists of ratings from 3631 users on 3675 movies. Let $\mathbf{R}' = [r_{ij}]_{3631\times 3675}$ denote the rating matrix, where $r_{ij} \in \{0, 1, \ldots, 5\}$ and $r_{ij} = 0$ means unrated. This rating matrix is quite sparse: the proportion of nonzero elements is only 6.78%. To determine the interest vector $\mathbf{p}_i$ of each user, we first apply the CF algorithm to predict the unknown ratings in $\mathbf{R}'$. Let $\mathbf{R}$ denote the matrix which consists of original ratings and predicted ratings. Then each row of $\mathbf{R}$ is treated as the corresponding user's interest vector. Other parameters of the SE learning algorithm are set in the same way as the Jester data.

B. Simulation Results of SE Learning

To verify the convergence of Algorithm 1, we test multiple groups of $\{\eta_i\}_{i=1}^N$. For a given μ, we run simulations with the following four settings: 1) $\eta_i = 0.5$ for all $i \in \mathcal{N}$; 2) $\eta_i = 0.85$ for all $i \in \mathcal{N}$; 3) $\eta_i = 0.85$ for 1% of the users, $\eta_i = 0.5$ for the rest; and 4) $\eta_i = 0.85$ for 20% of the users, $\eta_i = 0.5$ for the rest. To reduce the influence of randomness, we run the algorithm five times for each setting. In each run, the iterative process stops when all users are satisfied or the number of iterations reaches 10 000. After each run, we record the number of iterations n_{stop}, the number of satisfied users N_S, and the average rating completeness $\bar{\sigma}_i \triangleq (1/N)\sum_{i=1}^N |a_i(n_{\text{stop}})|/|S_i|$.

[2]http://grouplens.org/datasets/movielens/1m/

TABLE I
SIMULATION RESULTS OF SE LEARNING ON JESTER DATA SET

		$\mu = 0.9$					$\mu = 1$				
	runID	1	2	3	4	5	1	2	3	4	5
$\eta_i = 0.5$	n_{stop}	24	28	20	22	26	158	84	130	257	151
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.531	0.553	0.507	0.519	0.543	0.280	0.281	0.282	0.282	0.282
$\eta_i = 0.85$	n_{stop}	1203	1307	1306	1206	1203	8893	9560	9813	8988	9560
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.913	0.936	0.931	0.928	0.914	0.781	0.782	0.781	0.782	0.782
1%: $\eta_i = 0.85$ 99%: $\eta_i = 0.5$	n_{stop}	406	404	504	504	403	10000	10000	10000	10000	10000
	N_S	7200	7200	7200	7200	7200	7130	7130	7129	7130	7131
	$\bar{\sigma}_i$	0.906	0.888	0.895	0.896	0.881	0.325	0.326	0.326	0.324	0.324
20%: $\eta_i = 0.85$ 80%: $\eta_i = 0.5$	n_{stop}	804	905	802	902	903	10000	10000	10000	10000	10000
	N_S	7200	7200	7200	7200	7200	6901	6899	6885	6891	6896
	$\bar{\sigma}_i$	0.911	0.920	0.897	0.903	0.910	0.441	0.441	0.441	0.442	0.440

TABLE II
SIMULATION RESULTS OF SE LEARNING ON MOVIELENS DATA SET

		$\mu = 0.9$					$\mu = 1$				
	runID	1	2	3	4	5	1	2	3	4	5
$\eta_i = 0.5$	n_{stop}	27	29	48	28	73	57	28	39	53	37
	N_S	3631	3631	3631	3631	3631	3631	3631	3631	3631	3631
	$\bar{\sigma}_i$	0.814	0.824	0.884	0.821	0.921	0.224	0.224	0.223	0.223	0.226
$\eta_i = 0.85$	n_{stop}	528	679	533	494	560	734	610	640	780	571
	N_S	3631	3631	3631	3631	3631	3631	3631	3631	3631	3631
	$\bar{\sigma}_i$	0.990	0.994	0.990	0.989	0.991	0.745	0.744	0.743	0.745	0.745
1%: $\eta_i = 0.85$ 99%: $\eta_i = 0.5$	n_{stop}	118	78	223	68	157	10000	10000	10000	10000	10000
	N_S	3631	3631	3631	3631	3631	3630	3629	3629	3628	3628
	$\bar{\sigma}_i$	0.952	0.928	0.975	0.919	0.964	0.234	0.231	0.229	0.231	0.229
20%: $\eta_i = 0.85$ 80%: $\eta_i = 0.5$	n_{stop}	482	458	302	462	678	10000	10000	10000	10000	10000
	N_S	3631	3631	3631	3631	3631	3624	3619	3625	3622	3624
	$\bar{\sigma}_i$	0.989	0.989	0.981	0.989	0.993	0.350	0.349	0.350	0.349	0.350

Simulation results are shown in Tables I and II, from which we can make the following observations:

When users have similar expectations for the recommendation quality, even if the expectation is high ($\eta_i = 0.85$) and user becomes inactive after he is satisfied ($\mu = 1$), an SE can be reached. For a given μ, as users' expectations become higher, the convergence time becomes longer, and $\bar{\sigma}_i$ becomes higher, which means users need to rate more items. Given the setting of η_i, by comparing the results of different μ we can see that, when satisfied users no longer rate more items, the convergence time becomes longer, while the average rating completeness decreases. For example, as shown in Table I, given $\eta_i = 0.5$ for all $i \in \mathcal{N}$, when $\mu = 0.9$, an SE can be reached in 30 iterations, and averagely a user needs to rate 50%~60% of the items that he has experienced; when $\mu = 1$, usually more than 100 iterations are required to reach an SE, while the user only needs to rate less than 30% of the items. During the learning process, due to the randomness of users' actions, different users become satisfied at different time. If $\mu = 0.9$, satisfied users continue to make contributions to the improvement of recommendation quality, hence those unsatisfied users can be satisfied in a short time. While if $\mu = 1$, unsatisfied users can only rely on themselves to improve the recommendation quality, hence more time is needed. As for the rating completeness, $\mu = 0.9$ means the user may provide more ratings after he is satisfied, thus by the time an SE is reached, the user may have rated much more items than he needs to. While $\mu = 1$ means the user prefers to rate the minimum number of items necessary to get satisfactory recommendations, thus when an SE is achieved, the average rating completeness is lower than that of $\mu = 0.9$.

When most users have moderate expectations for the recommendation quality ($\eta_i = 0.5$) and a small portion of users have much higher expectations ($\eta_i = 0.85$), an SE can still be reached when $\mu = 0.9$, although the convergence time is much longer than that when all users have moderate expectations, and the average rating completeness is close to that when all users have high expectations. This result implies that in order to meet the high expectations of a few users, users with moderate expectations have to rate much more items after they are satisfied. When $\mu = 1$, satisfied users no longer rate more items. Hence, after the majority of users have been satisfied, those unsatisfied users can hardly improve the recommendation quality. For example, as shown in Table I, under the third setting of η_i, the 1% of users who have high expectations are still unsatisfied after 10 000 iterations. From the corresponding $\bar{\sigma}_i$ we can learn that most users just rate "enough" number of items to meet their moderate expectations, while the amount of their ratings is far from enough to achieve the expectations of the rest 72 users.

To better understand the influence of the minority high expectations on the learning results, we take a detailed look at the results on Jester data set and draw the sets of $|\mathcal{N}_S(n)|$ corresponding to different settings in Fig. 5. As shown in Fig. 5(a), in a setting where $\eta_i = 0.85$ for 20% of the users (depicted by magenta circles), after 15 iterations, nearly 80% of the users are already satisfied. During the first 15 iterations, $|\mathcal{N}_S(n)|$ grows at almost the same rate with that of the setting where $\eta_i = 0.5$ for all users (depicted by red circles). After most users are satisfied, the growth rate of $|\mathcal{N}_S(n)|$ decreases. This is because the satisfied users prefer rating no more items, and for those unsatisfied users the recommendation results only

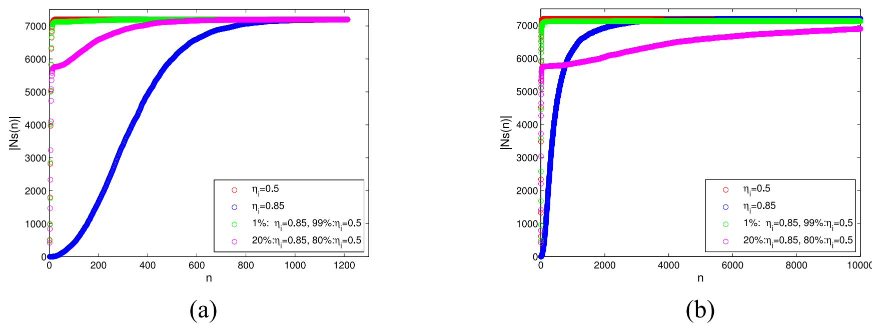


Fig. 5. Change of the number of satisfied users. Simulations are conducted on Jester data set. (a) $\mu = 0.9$. (b) $\mu = 1$.

improve a little after one iteration. Consequently, many more iterations are required to achieve the expectations of the rest users. Similar results can be observed in Fig. 5(b).

The simulation results coincide with our intuition about the SE in a CF system: when all users have moderate expectations for recommendation quality, an SE can be realized in low cost, that is, every user can get satisfactory recommendations without rating many items. From the results shown in Tables I and II, we can get some general insight about the convergence conditions of the learning algorithm. Next we will conduct another group of simulations to verify the analysis presented in Section VI.

C. Relationship Between Recommendation Quality and Rating Completeness

The theoretical analysis we presented in Section VI is based on some assumptions (see Assumptions 1 and 2). Before we verify the convergence conditions, we first conduct some experiments on Jester data set to validate the rationality of the assumptions. For each user i, we utilize $\mathbf{R}'$ to construct a group of rating matrices $\{\mathbf{R}_{i,k}\}$. Each matrix $\mathbf{R}_{i,k}$ corresponds to a certain pair of σ_i and σ_{-i}, where $\sigma_i \in \{(1/70), (2/70), \ldots, (70/70)\}$ and $\sigma_{-i} \in \{(1/100), (2/100), \ldots, (100/100)\}$. For example, given $\sigma_i = (5/70)$ and $\sigma_{-i} = (10/100)$, we randomly choose five nonzero ratings from the ith row of $\mathbf{R}'$ and set them to 0, then we randomly set $(1-[10/100])\times 100\%$ of the nonzero ratings in other rows to 0. We apply the user-based CF algorithm [see (2)] to $\mathbf{R}_{i,k}$, and evaluate the recommendation results based on (4) to get $h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)$. By drawing $\{(\sigma_i, \sigma_{-i}, h(\sigma_i, \sigma_{-i}; \mathbf{p}_i))\}$ in a 3-D space, we can get a plot of $h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)$ corresponding to the user i. Fig. 6 shows an example. From Fig. 6(a), we can see that the recommendation quality improves when σ_i or σ_{-i} increases. This result confirms Assumption 1. From the contour plot shown in Fig. 6(b) we can observe that, given a proper value of $h(\sigma_i, \sigma_{-i}; \mathbf{p}_i)$, there is an approximate quadratic relationship between σ_i and σ_{-i}. Assumption 2 is proposed based on the this observation. Experiment results of other users can also support the two assumptions.

D. Convergence Test

We implement the simplified learning algorithm described in Section VI and conduct simulations on Jester data set to verify the convergence conditions proposed in the Appendix.

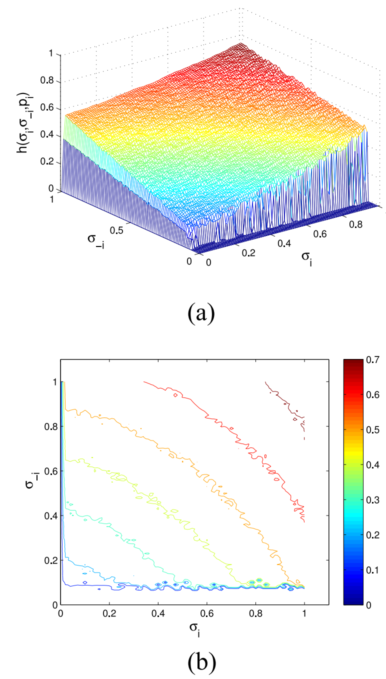


Fig. 6. Illustration of the relationship between recommendation quality and rating completeness. (a) Fitted surface is obtained by interpolating the experiment results $\{(\sigma_i, \sigma_{-i}, h(\sigma_i, \sigma_{-i}; \mathbf{p}_i))\}$ corresponding to one user. (b) Contour plot of (a).

As described in Assumption 2, we randomly divide users into two groups $\mathcal{N}_A$ and $\mathcal{N}_A$, and we set η_A to 0.5. The two parameters ρ_N and η_B are set in a following way: given $\rho_N \in \{0.01, 0.10, 0.20\}$, we calculate the corresponding θ_B according to (41), then we set $\eta_B = \theta_B - 0.05$, $\eta_B = \theta_B$, and $\eta_B = \theta_B + 0.05$, respectively. To prove that the algorithm can converge when all users have high expectations, we also test the setting $\eta_A = \eta_B = \theta_B + 0.05$. Given a group of (ρ_N, η_A, η_B), we run the simplified learning algorithm ten times. The iterative process stops at iteration n_{stop} when one of the following conditions is met: 1) $|\mathcal{N}_S(n_{\text{stop}})| = N$ and 2) $\forall i \in \mathcal{N}_{US}(n_{\text{stop}})$, $\sigma_i(n_{\text{stop}}) = 1$.

Let N_S denote the number of satisfied users at the end of the learning process, namely $N_S = |\mathcal{N}_S(n_{\text{stop}})|$. Table III shows the simulation results. As we can see, given η_A and ρ_N, an SE can always be reached when $\eta_B = \theta_B - 0.05$. When $\eta_B = \theta_B$, some users in $\mathcal{N}_B$ can be satisfied and some cannot. This result is slightly different from the theoretical analysis presented in

TABLE III
SIMULATIONS RESULTS OF THE SIMPLIFIED LEARNING ALGORITHM

ρ_N	η_A	θ_B	η_B	runID	1	2	3	4	5
0.01	0.5	0.792	0.742	n_{stop}	68	67	68	67	68
				N_S	7200	7200	7200	7200	7200
			0.792	n_{stop}	70	70	70	70	70
				N_S	7178	7178	7177	7176	7176
			0.842	n_{stop}	70	70	70	70	70
				N_S	7130	7130	7131	7130	7129
$\eta_A = \eta_B = 0.842$				n_{stop}	65	65	65	65	65
				N_S	7200	7200	7200	7200	7200
0.1	0.5	0.807	0.757	n_{stop}	69	68	68	68	69
				N_S	7200	7200	7200	7200	7200
			0.807	n_{stop}	70	70	70	70	70
				N_S	7132	7130	7139	7141	7141
			0.857	n_{stop}	70	70	70	70	70
				N_S	6642	6618	6627	6630	6619
$\eta_A = \eta_B = 0.857$				n_{stop}	66	66	65	66	66
				N_S	7200	7200	7200	7200	7200
0.2	0.5	0.824	0.775	n_{stop}	69	68	68	69	69
				N_S	7200	7200	7200	7200	7200
			0.825	n_{stop}	70	70	70	70	70
				N_S	7133	7146	7139	7142	7130
			0.875	n_{stop}	70	70	70	70	70
				N_S	6368	6400	6383	6398	6387
$\eta_A = \eta_B = 0.875$				n_{stop}	66	67	67	66	67
				N_S	7200	7200	7200	7200	7200

Section VI, where (38) implies that an SE can be achieved when $\eta_B \leq \theta_B$. We think the reason for the inconsistence between theoretical analysis and simulation results is that the relationship between users' rating completeness and recommendation quality does not exactly accord with assumption we have made in (30). When $\eta_B = \theta_B + 0.05$, most of the users in $\mathcal{N}_B$ cannot be satisfied. From Table III we can also see that when users have similar high expectations, the simplified learning algorithm can converge to an SE before users have rated all the item they have experienced ($n_{\text{stop}} < |S_i|$). These results demonstrate that in a CF system, whether an equilibrium can be achieved via users' spontaneous participation depends on whether the users are homogeneous in the sense that they expect same recommendation quality.

E. Incentive Mechanism

From the SE learning results presented in Section VII-B we can see that, when different users have similar expectations, the recommendation quality solely can motivate users to provide enough ratings to the RS, so that the server can generate satisfying recommendations for all users. In this case, external incentive for user participation is not necessary. However, if there are significant differences among users' expectations and users no longer rate more items after they are satisfied, then the SE cannot be achieved via the proposed learning algorithm. In such a case, some kind of external incentive (e.g., monetary rewards) is required to encourage users to provide more ratings.

As described in Section VII-A1, the cost of choosing action $A^{(k)}$ is defined as $c_i(A^{(k)}) \triangleq |A^{(k)}|$. Suppose that the RS pays $b(A^{(k)}) \triangleq \kappa |A^{(k)}|$ to the user as a reward, where the parameter $\kappa \in (0, 1)$ denotes the monetary reward that the user can get by rating one item. Paying rewards to users can be seen as a way to reduce the rating costs of users. More specifically, when user i gets a reward $b(A^{(k)})$, the actual rating cost he pays is $c_i(A^{(k)}) - b(A^{(k)})$. According to Algorithm 1, actions with low cost are preferred by users, hence users may rate more items if they are rewarded. Besides, motivated by the monetary rewards, users will continue to rate items even if they are satisfied with current recommendations. To formulate this intuition, we make a small modification to Algorithm 1: at each iteration, if the user is satisfied, i.e., $v_i(n-1) = 1$, the probability that the user keeps previous action is defined as $\mu \triangleq 1 - \kappa$. Since $0 < \kappa < 1$, there is $\mu < 1$. As we have verified in Section VII-B, an SE can always be achieved when $\mu < 1$.

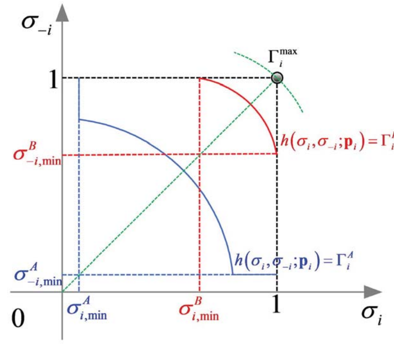


Fig. 7. Relationship between σ_i and σ_{-i} with respect to a given Γ_i.

To evaluate the performance of the modified learning algorithm, we conduct simulations on Jester data set. Parameters of the algorithm are set in the same way as we've done before, and the reward parameter κ is set to 0.01, 0.1, 0.5, and 0.9, respectively. Again, to reduce the influence of randomness, we run the algorithm five times for each setting. Simulation results are shown in Table IV. By comparing Tables I and IV we can see that, when the RS pays rewards to users, the learning algorithm converges to the equilibrium at a fast speed. The higher the rewards are, the faster the algorithm converges. For example, when there are only 1% of users who have high expectations for recommendation quality, if no reward is offered and $\mu = 0.9$, at least 400 iterations are required to reach an SE; if the RS adopts a reward mechanism and sets $\kappa = 0.1$, the value of μ is still 0.9, but this time an SE can be achieved after about 200 iterations.

The simulation results indicate that the recommendation sever can push the interactions among users toward SE by offering rewards to users. The reward mechanism proposed above is quite simple. A more elaborate incentive mechanism, where the differences in rating cost and expectation for recommendation quality among users are considered, should be developed. We will investigate this problem in future work.

VIII. CONCLUSION

In this papers', we formulated the interaction among users in a CF system as a game in satisfaction form. To learn the SE of the game, we proposed a behavior rule that a user iteratively updates the probability distribution over the action space and gradually rate more items. We have analyzed the convergence of the proposed rule under some simplifying assumptions. By conducting simulations on real data sets, we have demonstrated that when users have similar expectations

TABLE IV
SIMULATION RESULTS OF SE LEARNING WITH REWARDS

		1%: $\eta_i = 0.85$, 99%: $\eta_i = 0.5$					20%: $\eta_i = 0.85$, 80%: $\eta_i = 0.5$				
	runID	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
$\kappa = 0.01$	n_{stop}	**102**	**172**	**184**	**201**	**102**	**401**	**302**	**303**	**304**	**304**
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.939	0.888	0.891	0.935	0.938	0.96	0.964	0.973	0.98	0.979
$\kappa = 0.1$	n_{stop}	**183**	**205**	**201**	**201**	**202**	**301**	**302**	**304**	**303**	**302**
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.859	0.931	0.885	0.886	0.899	0.904	0.914	0.931	0.923	0.914
$\kappa = 0.5$	n_{stop}	**15**	**19**	**16**	**16**	**18**	**21**	**18**	**20**	**20**	**21**
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.884	0.907	0.889	0.89	0.904	0.916	0.903	0.911	0.912	0.915
$\kappa = 0.9$	n_{stop}	**3**	**3**	**3**	**3**	**3**	**4**	**4**	**4**	**5**	**4**
	N_S	7200	7200	7200	7200	7200	7200	7200	7200	7200	7200
	$\bar{\sigma}_i$	0.925	0.926	0.922	0.923	0.925	0.962	0.963	0.963	0.982	0.963

for the recommendation quality, an SE can be achieved via users' spontaneous rating behaviors.

The game-theoretic analysis we presented in this paper may provide some implications to the study of user behaviors in collaborative systems. The derived convergence conditions may also be helpful to the design of incentive mechanisms. In future work, we would like to investigate how to utilize both the intrinsic motivation and external incentives to encourage user participation.

APPENDIX
CONVERGENCE CONDITIONS OF THE SIMPLIFIED LEARNING ALGORITHM

To derive specific convergence conditions from (29), we make the following assumption as a complementary to Assumption 2: for all $i \in \mathcal{N}$, given $\eta_i \in \{\eta_A, \eta_B\}$, the relationship between σ_i and σ_{-i} can be formulated as

$$\sigma_i^2 + \sigma_{-i}^2 = 2\eta_i^2 \tag{30}$$

where $(1/M_0) \leq \sigma_i \leq 1$, $(1/M_0) \leq \sigma_{-i} \leq 1$.

In above assumption, the quadratic relationship between σ_i and σ_{-i} is proposed based on simulation results on real data set (see Section VII-C). As shown in Fig. 7, the threshold $\sigma_{-i,\min}$ is now defined in the following way.

1) If $\eta_A < \eta_B \leq (1/\sqrt{2})$, then for all $i \in \mathcal{N}_B$, $\sigma_{-i,\min} = (1/M_0)$. Considering that $n_A \geq 1$, (29) implies that

$$\frac{1}{M_0} > \frac{|\mathcal{N}_B| - 1}{N-1} + \frac{N - |\mathcal{N}_B|}{N-1} \cdot \frac{1}{M_0}. \tag{31}$$

Then we can get

$$(M_0 - 1)(|\mathcal{N}_B| - 1) < 0. \tag{32}$$

Because $M_0 \geq 1$ and $|\mathcal{N}_B| \geq 1$, above inequality does not hold. Therefore, when $\eta_A < \eta_B \leq (1/\sqrt{2})$, it is impossible that $k_B < k_{\min}$, which means the algorithm must converge.

2) If $(1/\sqrt{2}) < \eta_B < 1$, then for all $i \in \mathcal{N}_B$, there is

$$\sigma_{-i,\min} = \sqrt{2\eta_B^2 - 1}. \tag{33}$$

From $z_i(n_A) = Z_F$ we can get

$$\frac{n_A}{M_0} < \sqrt{2\eta_B^2 - 1}. \tag{34}$$

On the other hand, for any user $i \in \mathcal{N}_A$, the following inequality holds:

$$[\sigma_i(n_A - 1)]^2 + \left[\sigma_{-i}(n_A - 1)\right]^2 \geq 2\eta_A^2 \tag{35}$$

where $\sigma_i(n_A - 1) = \sigma_{-i}(n_A - 1) = (n_A/M_0)$, then we get

$$\frac{n_A}{M_0} \geq \eta_A. \tag{36}$$

From (29), (33), (34), and (36) we can get

$$\sqrt{2\eta_B^2 - 1} > \frac{|\mathcal{N}_B| - 1}{N-1} + \frac{N - |\mathcal{N}_B|}{N-1}\eta_A. \tag{37}$$

Based on the above discussions, we can provide the following proposition.

Proposition 1: The simplified learning algorithm can converge to an SE of the game $\hat{G}_{\text{CF}} = (\mathcal{N}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{f_i\}_{i\in\mathcal{N}})$ if Assumption 2 holds and one of the following two conditions holds.

1) $\eta_A < \eta_B \leq (1/\sqrt{2})$.
2) $(1/\sqrt{2}) < \eta_B < 1$ and $\sqrt{2\eta_B^2 - 1} \leq [(|\mathcal{N}_B| - 1)/(N - 1)] + [(N - |\mathcal{N}_B|)/[N-1]]\eta_A$.

To better understand the influence of η_A, η_B, and $|\mathcal{N}_B|$ on the convergence of the learning algorithm, we make following discussions.

1) Given $\rho_N \triangleq (|\mathcal{N}_B|/N)$, according to (37), the simplified algorithm cannot converge if the following condition holds:

$$\eta_B > \sqrt{\frac{1}{2}\left[\frac{\rho_N N - 1}{N-1} + \frac{(1-\rho_N)N}{N-1}\eta_A\right]^2 + \frac{1}{2}}. \tag{38}$$

We use θ_B to denote the right side of (38). Fig. 8 illustrates how θ_B changes with η_A under different settings of ρ_N. As we can see, given ρ_N, θ_B grows with η_A, but the growth rate of θ_B is lower than that of η_A. This means as expectations of most users become higher (larger η_A), even if users do not make significant difference on their expectations, there may be some users who can never be satisfied. From Fig. 8, we can also observe that for a given η_A, θ_B increases with ρ_N. This implies that as more users have high expectations (larger ρ_N), users can expect higher recommendation quality (larger η_B).

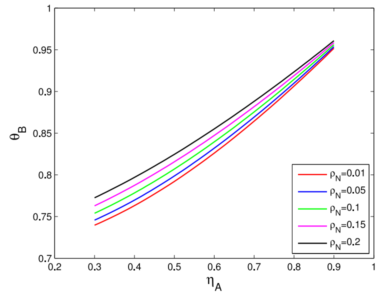


Fig. 8. Illustration of the relationship between θ_B and η_A. Given ρ_N and η_A, the simplified learning algorithm cannot converge to an SE if $\eta_B > \theta_B$. We set $0.3 \le \eta_A \le 0.9$ and $N = 10\,000$ to compute θ_B.

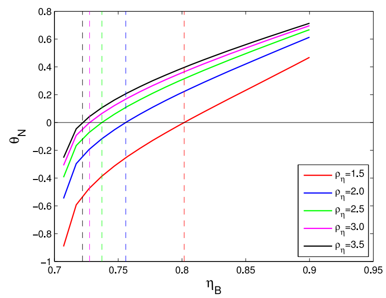


Fig. 9. Illustration of the relationship between θ_N and η_B. Dotted line marks $\eta_{B,\min}$ corresponding to a given ρ_η. Given ρ_η, the simplified learning algorithm cannot converge to an SE if $\eta_B > \eta_{B,\min}$ and $\rho_N < \theta_N$. We set $(1/\sqrt{2}) < \eta_B \le 0.9$ and $N = 10\,000$ to compute θ_N.

2) Given $\rho_\eta \triangleq (\eta_B/\eta_A)$, the simplified algorithm cannot converge if $(1/\sqrt{2}) < \eta_B < 1$ and

$$\frac{|\mathcal{N}_B|}{N} < \frac{(N-1)\sqrt{2\eta_B^2 - 1} + 1 - N\frac{\eta_B}{\rho_\eta}}{N\left(1 - \frac{\eta_B}{\rho_\eta}\right)}. \tag{39}$$

We use θ_N to denote the right side of above inequality. The inequality implies $\theta_N > 0$ from which the following two conditions can be derived:

$$N > \frac{\sqrt{2}}{\sqrt{2} - \frac{1}{\rho_\eta}} \tag{40}$$

$$\eta_B > \frac{-\frac{2N}{\rho_\eta} + \sqrt{\Delta_f}}{2\left[2(N-1)^2 - \frac{N^2}{\rho_\eta^2}\right]} \tag{41}$$

where $\Delta_f = (2N/\rho_\eta)^2 + 4[2(N-1)^2 - (N^2/\rho_\eta^2)](N^2 - 2N + 2)$. We use $\eta_{B,\min}$ to denote the right side of (41). Fig. 9 illustrates how θ_N changes with η_B. As we can see, given ρ_η, θ_N increases with η_B, which implies when users with high expectation expect higher recommendation quality, there should be more such users so that an SE can be achieved. Given η_B, θ_N increases with ρ_η, which implies that as the difference on expected recommendation quality between two types of users becomes wider, more users can have high expectations. These implications are consistent with those we get from Fig. 8.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[2] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, p. 4, Jan. 2009.

[3] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, "Information security in big data: Privacy and data mining," *IEEE Access*, vol. 2, pp. 1149–1176, 2014.

[4] M. Grčar, D. Mladenič, B. Fortuna, and M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *Proc. 7th Int. Conf. Knowl. Disc. Web Adv. Web Min. Web Usage Anal. (WebKDD)*, Chicago, IL, USA, 2006, pp. 58–76.

[5] D. Rafailidis and P. Daras, "The TFC model: Tensor factorization and tag clustering for item recommendation in social tagging systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 673–688, May 2013.

[6] M. Mao, J. Lu, G. Zhang, and J. Zhang, "Multirelational social recommendations via multigraph ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4049–4061, Dec. 2017.

[7] Y. Ren, G. Li, J. Zhang, and W. Zhou, "Lazy collaborative filtering for data sets with missing values," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1822–1834, Dec. 2013.

[8] B. Li, X. Zhu, R. Li, and C. Zhang, "Rating knowledge sharing in cross-domain collaborative filtering," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1068–1082, May 2015.

[9] P. Symeonidis, "ClustHOSVD: Item recommendation by combining semantically enhanced tag clustering with tensor HOSVD," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1240–1251, Sep. 2016.

[10] Y. Gao, Y. Chen, and K. J. R. Liu, "On cost-effective incentive mechanisms in microtask crowdsourcing," *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 1, pp. 3–15, Mar. 2015.

[11] Y.-H. Yang, Y. Chen, C. Jiang, C.-Y. Wang, and K. J. R. Liu, "Wireless access network selection game with negative network externality," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5048–5060, Oct. 2013.

[12] L. Xu *et al.*, "Energy efficient D2D communications: A perspective of mechanism design," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7272–7285, Nov. 2016.

[13] R. Gibbons, *A Primer in Game Theory*. Hertfordshire, U.K.: Harvester Wheatsheaf, 1992.

[14] L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. J. R. Liu, "User participation game in collaborative filtering," in *Proc. IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, Atlanta, GA, USA, Dec. 2014, pp. 263–267.

[15] M. Halkidi and I. Koutsopoulos, "A game theoretic framework for data privacy preservation in recommender systems," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer-Verlag, 2011, pp. 629–644.

[16] Y. Chen and K. J. R. Liu, "Understanding microeconomic behaviors in social networking: An engineering view," *IEEE Signal Process. Mag.*, vol. 29, no. 2, pp. 53–64, Mar. 2012.

[17] Y. Meng, C. Jiang, L. Xu, Y. Ren, and Z. Han, "User association in heterogeneous networks: A social interaction approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9982–9993, Dec. 2016.

[18] S. M. Perlaza, H. Tembine, S. Lasaulce, and M. Debbah, "Quality-of-service provisioning in decentralized networks: A satisfaction equilibrium approach," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 2, pp. 104–116, Apr. 2012.

[19] S. Ross and B. Chaib-Draa, "Satisfaction equilibrium: Achieving cooperation in incomplete information games," in *Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2006, pp. 61–72.

[20] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[21] R. M. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," in *Proc. Cup Workshop 13th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, 2007, pp. 7–14.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, May 2001, pp. 285–295.

[23] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[24] T. Kandappu, A. Friedman, R. Boreli, and V. Sivaraman, "PrivacyCanary: Privacy-aware recommenders with adaptive input obfuscation," in *Proc. IEEE 22nd Int. Symp. Model. Anal. Simulat. Comput. Telecommun. Syst.*, Paris, France, Sep. 2014, pp. 453–462.

[25] N. Polatidis, C. K. Georgiadis, E. Pimenidis, and H. Mouratidis, "Privacy-preserving collaborative recommendations based on random perturbations," *Expert Syst. Appl.*, vol. 71, pp. 18–25, Apr. 2017.

[26] J. Parra-Arnau, D. Rebollo-Monedero, and J. Forné, "Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems," *Entropy*, vol. 16, no. 3, pp. 1586–1631, 2014.

[27] P. De Meo, G. Quattrone, G. Terracina, and D. Ursino, "An XML-based multiagent system for supporting online recruitment services," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 4, pp. 464–480, Jul. 2007, doi: 10.1109/TSMCA.2007.897696.

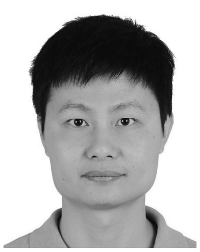
[28] H. Blanco and F. Ricci, "Acquiring user profiles from implicit feedback in a conversational recommender system," in *Proc. 7th ACM Conf. Recommender Syst. (RecSys)*, Hong Kong, 2013, pp. 307–310. [Online]. Available: http://doi.acm.org/10.1145/2507157.2507217

Lei Xu received the Ph.D. degree in electronic engineering from Tsinghua University (THU), Beijing, China, in 2015. She was a Post-Doctoral Fellow with the Department of Computer Science and Technology, THU from 2015 to 2017. Her current research interests include privacy issues in data mining, game theory, and information sharing networks.

Chunxiao Jiang (S'09–M'13–SM'15) received the B.S. degree (Highest Hons.) in information engineering from Beihang University, Beijing, China, in 2008 and the Ph.D. degree (Highest Hons.) in electronic engineering from Tsinghua University, Beijing, in 2013.

From 2013 to 2016, he was a Post-Doctoral Fellow with the Department of Electronic Engineering, Tsinghua University, during which he visited the University of Maryland at College Park, College Park, MD, USA, and the University of Southampton, Southampton, U.K.

Dr. Jiang was a recipient of the IEEE Globecom Best Paper Award in 2013, the IEEE GlobalSIP Best Student Paper Award in 2015, the IEEE IWCMC Best Paper Award in 2017, and the IEEE Communications Society Young Author Best Paper Award in 2017.



Yan Chen (SM'14) received the bachelor's degree from the University of Science and Technology of China, Hefei, China, in 2004, the M.Phil. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2007, and the Ph.D. degree from the University of Maryland at College Park, College Park, MD, USA, in 2011.

He was with Origin Wireless Inc., Greenbelt, MD, USA, as a Founding Principal Technologist. Since 2015, he has been a Full Professor with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include multimedia, signal processing, game theory, and wireless communications.

Dr. Chen was a recipient of multiple honors and awards, including the Best Student Paper Award at the PCM in 2017, the Best Student Paper Award at the IEEE ICASSP in 2016, the Best Paper Award at the IEEE GLOBECOM in 2013, the Future Faculty Fellowship and Distinguished Dissertation Fellowship Honorable Mention from the Department of Electrical and Computer Engineering in 2010 and 2011, the Finalist of the Dean's Doctoral Research Award from the A. James Clark School of Engineering, University of Maryland in 2011, and the Chinese Government Award for Outstanding Students Abroad in 2010.

Yong Ren (SM'12) received the B.S., M.S., and Ph.D. degrees in electronic engineering from the Harbin Institute of Technology, Harbin, China, in 1984, 1987, and 1994, respectively.

He was a Post-Doctoral Fellow with the Department of Electronics Engineering, Tsinghua University, Beijing, China, from 1995 to 1997, where he is currently a Professor with the Department of Electronics Engineering and the Director of the Complexity Engineered Systems Laboratory. He holds 12 patents, and has authored or co-authored over 100 technical papers in the behavior of computer network, P2P network, and cognitive networks. His current research interests include complex systems theory and its applications to the optimization and information sharing of the Internet, Internet of Things and ubiquitous network, cognitive networks, and cyber-physical systems.

Dr. Ren has serves as a Reviewer of *IEICE Transactions on Communications*, *Digital Signal Processing*, *Chinese Physics Letters*, the *Chinese Journal of Electronics*, the *Chinese Journal of Computer Science and Technology*, and the *Chinese Journal of Aeronautics*.

K. J. Ray Liu (F'03) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1983, and the Ph.D. degree from the University of California, Los Angeles, CA, USA, in 1990, both in electrical engineering.

He was named a Distinguished Scholar-Teacher with the University of Maryland at College Park, College Park, MD, USA, in 2007, where he is Christine Kim Eminent Professor of information technology. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of information and communications technology with recent focus on future wireless technologies, network science, and information forensics and security.

Dr. Liu was a recipient of the IEEE Signal Processing Society 2014 Society Award, IEEE Signal Processing Society 2009 Technical Achievement Award, the Highly Cited Researcher Award by Thomson Reuters, the teaching and research recognitions from the University of Maryland, including University-Level Invention of the Year Award and College-Level Poole and Kent Senior Faculty Teaching Award, the Outstanding Faculty Research Award, and the Outstanding Faculty Service Award, all from A. James Clark School of Engineering, and various best paper awards. He is the Director-Elect of IEEE Board of Director. He was the President of IEEE Signal Processing Society, where he has served as the Vice President Publications and the Board of Governor. He has also served as the Editor-in-Chief of the *IEEE Signal Processing Magazine*. He is a fellow of AAAS.