User Participation Game in Collaborative Filtering

Lei Xu*, Chunxiao Jiang*, Yan Chen†, Yong Ren*, and K. J. Ray Liu†
*Department of Electronic Engineering, Tsinghua University, Beijing, 100084, P. R. China
†Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA
E-mail: l-xu08@mails.tsinghua.edu.cn, {jchx, reny}@tsinghua.edu.cn, {yan, kjrliu}@umd.edu

***Abstract*—Collaborative filtering (CF) is widely used in recommendation systems. A user can get good recommendations only when both the user himself/herself and other users actively participate, i.e. providing sufficient rating data. However, due to the rating cost, rational users tend to provide as few ratings as possible. Therefore, there exists a trade-off between the rating cost and 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 rating data are used for recommendation, we design a behavior rule which allows users to achieve a SE via iteratively rating items. Experimental results based on real data 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 too many ratings. The SE analysis of the proposed game in this paper is helpful for designing mechanisms to encourage user participation.**

***Index Terms*—collaborative filtering, game theory, satisfaction equilibrium, behavior rule.**

I. INTRODUCTION

Recommendation system has been successfully applied in a variety of applications. Among the approaches to building recommendation systems, collaborative filtering (CF) [1] is most widely adopted. The basic idea of CF is to utilize the information provided by users to match users with similar interests and to make corresponding recommendations. Therefore, user participation is of vital importance for the success of CF. Conventionally, information provided by users is represented as a user-item rating matrix $\mathbf{R} = [r_{ij}]_{N\times M}$. The value of the rating r_{ij} shows user i's preference for item j. The major task of the recommendation server is to predict the missing values in the matrix where users did not give their preferences for certain items.

Generally, a user assigns ratings to items after he/she has obtained experience of the items. Since rating items incurs some cost, such as time cost and privacy cost, the user may only rate part of the items that he/she has experienced. As a result, the rating matrix may be quite sparse, which inevitably impairs the recommendation quality [2]. Moreover, as the name *collaborative filtering* suggests, whether a user can get accurate recommendations depends not only on the ratings provided by the user himself/herself, 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 by rating only a few items. In such a case, it is natural to employ game theory [3] to model the interactions among users in a CF system.

In this paper, we build a game theoretical model to study users' rating behaviors in a CF-based recommendation system. Halkidi *et al.* [4] also presented a game theoretical approach to address the trade-off between privacy preservation and recommendation quality. Chen and Liu [5] provided a general game analysis of human behaviors in social networks. Different from these studies, we model the interactions among users as a satisfactory game with incomplete information, i.e., each user only has the knowledge of his/her own ratings and recommendations, while others' ratings cannot be observed. Meanwhile, the CF algorithm adopted by the recommendation server is also unknown to users. To analyze the game with incomplete information, we apply the notion of *satisfaction equilibrium* (SE) which was originally introduced by Ross and Chaib-draa [6], [7]. To the best of our knowledge, this is the first time that SE is applied to study of CF. A game is said to be in SE when all players simultaneously satisfy their individual constrains. In the context of CF, a user's expectation for recommendation quality is seen as his/her constrain. Recommendation quality can also be seen as an intrinsic motivation for users to rate items. Hence, different from the work in [8], [9] where incentive mechanisms were designed to encourage user participation, in this paper we focus on the equilibrium of the proposed satisfactory game and try to answer the following question: without external incentives, whether and how all users can get satisfying recommendations by rating the appropriate amount of items. Based on the characteristics of recommendation system, we design a behavior rule which allows users to achieve the SE via iteratively choosing items to rate.

The rest of the paper is organized as follows. Section II describes the system model, and Section III presents the satisfactory game formulation. Section IV introduce the behavior rule for learning satisfaction equilibrium. Experiment results are shown in Section V and conclusion is drawn in Section VI.

II. SYSTEM MODEL

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

978-1-4799-7088-9/14/$31.00 ©2014 IEEE

a rating vector $\mathbf{r}_i = (r_{i1}, r_{i2}, \cdots, r_{iM})$ 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 $s_j \notin \tilde{S}_i$ ($j = 1, \cdots, M$). Usually, a high value of r_{ij} implies user i has strong preference for item s_j.

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

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

where $f_{ij}^{CF}(\mathbf{R})$ means that the predicted rating is determined by both the CF algorithm and the whole ratings.

After receiving the recommendation vector, the user evaluates the recommendation quality by judging whether the recommendation matches his/her interest. Let $\mathbf{p}_i = (p_{i1}, \cdots, p_{iM})$ denote user i's interest, where p_{ij} represents user i's true preference for item s_j ($j = 1, \cdots, M$). We assume $0 \leq p_{ij} \leq r_{\max}$ and define $p_{ij} = r_{ij}$ for $s_j \in \tilde{S}_i$. The quality of $\hat{\mathbf{r}}_i$ is evaluated by a user-specific function $g_i : \mathbb{R}^M \to \mathbb{R}$. For example, $g_i(\hat{\mathbf{r}}_i)$ can be defined as follows:

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

A large $g_i(\hat{\mathbf{r}}_i)$ implies high similarity between $\mathbf{r}_i$ and $\mathbf{p}_i$, namely high recommendation quality. Recommendation of high quality is the most important incentive for a user to actively participate in the rating process. From (1) and (2), we can see that the recommendation quality obtained by one user is affected by other users' ratings. Thus, users in a CF system interact with each other via providing ratings to the RS. In the following, we will use satisfactory game to formulate the interaction among users.

III. 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. It is assumed that all users share the same action space, i.e. for any $i \in \mathcal{N}$, there is $\mathcal{A}_i = \left(A^{(1)}, \cdots, A^{(K)}\right)$, where $K = 2^{|S|} - 1$ ($|S|$ denotes the cardinality of S), $A^{(k)} \subseteq S$ ($k = 1, \cdots, K$) and $A^{(k)} \neq \varnothing$. When choosing an action, each user follows his/her own probability distribution over the action space. We use $\boldsymbol{\pi}_i = \left(\pi_i^{(1)}, \cdots, \pi_i^{(K)}\right)$ to denote the distribution, where $\pi_i^{(k)} \triangleq \Pr\left(a_i = A^{(k)}\right)$ represents the probability that user i chooses the action $A^{(k)}$.

Given an action profile $\mathbf{a} = (a_1, \cdots, a_N) \in \mathcal{A}$ ($\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$), 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 CF algorithm is specified, we introduce a mapping $h_i : \mathcal{A} \to \mathbb{R}$ to show the influence of users' actions on recommendation quality:

$$g_i(\hat{\mathbf{r}}_i) = h_i(\mathbf{a}) = h_i(a_i, \mathbf{a}_{-i}) , \tag{3}$$

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

As mentioned in Section I, rating items incurs some cost. The more items the user rates, the higher cost he/she has to pay. Let $c_i(a_i)$ denote the cost paid by user i when he/she 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

Intuitively, if the user i himself/herself or other users rate more items, the rating matrix will become less sparse, and user i can get better recommendation. When every user has rated all the items he/she has experienced, every user can obtain the best recommendation quality $\Gamma_i^{\max}$:

$$\Gamma_i^{\max} = h_i(S_1, S_2, \cdots, S_N) = \max_{\mathbf{a} \in \mathcal{A}} h_i(\mathbf{a}) . \tag{4}$$

However, due to the rating cost, not every user would rate all the items he/she has experienced, which means $\Gamma_i^{\max}$ is difficult to achieve. Suppose that each user i has a relatively low expectation Γ_i ($\Gamma_i < \Gamma_i^{\max}$) for the recommendation quality. As long as $h_i(\mathbf{a}) \geq \Gamma_i$, user i will be *satisfied*. Given the actions of other users, user i may choose some action to make himself/herself satisfied. We use $f_i(\mathbf{a}_{-i})$ to denote the set of such actions:

$$f_i(\mathbf{a}_{-i}) = \{a_i \in \mathcal{A}_i : h_i(a_i, \mathbf{a}_{-i}) \geq \Gamma_i\} , \tag{5}$$

where the mapping $f_i : \mathcal{A}_{-i} \to 2^{\mathcal{A}_i}$ is usually called *correspondence* [10].

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

$$\hat{G}_{CF} = \left(\mathcal{N}, \{\mathcal{A}_i\}_{i \in \mathcal{N}}, \{f_i\}_{i \in \mathcal{N}}\right) . \tag{6}$$

This formulation of game is called *satisfaction form*, which was first introduced by Perlaza et al. [10] to model the QoS (Quality-Of-Service) provisioning problem.

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 *satisfaction equilibrium* (SE) [10]:

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

We have assumed that for all $i \in \mathcal{N}$, there is $\Gamma_i < \Gamma_i^{\max}$, hence the action profile $\mathbf{a}^{\max} \triangleq (S_1, S_2, \cdots, S_N)$ is a SE of the proposed game. However, $\mathbf{a}^{\max}$ 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 $\mathbf{a}^+ = \left(a_1^+, \cdots, a_N^+\right)$ which satisifies $\forall i \in \mathcal{N}$, there is $a_i^+ \in f_i\left(\mathbf{a}_{-i}^+\right)$ and $c_i\left(a_i^+\right) \leq c_i(S_i)$.

IV. Learning Satisfaction Equilibrium

In this section, we study the behavior rule that allows users to learn a satisfaction equilibrium. The learning process is essentially an iterative process of information exchange between users and the RS, during which each user chooses his/her actions as follows.

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

$$\pi_i^{(k)}(0) = \begin{cases} \beta_i(0)/\alpha^{c_i\left(A^{(k)}\right)}, & \text{if} \quad A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} . \tag{7}$$

The above equation means that user prefers to choose the low-cost action. The parameter $\alpha > 1$ shows how much the user cares about the cost. The normalization factor $\beta_i(0)$ is defined as follows:

$$\beta_i(0) = \frac{1}{\sum\limits_{k:\ A^{(k)} \subseteq S_i} \alpha^{-c_i\left(A^{(k)}\right)}} . \tag{8}$$

After every user has chosen his/her 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 ($n = 1, 2, \cdots$), user i evaluates $\hat{\mathbf{r}}_i(n-1)$ to see whether it is satisfying. We use a binary variable $v_i(n-1)$ to indicate the evaluation result:

$$v_i(n-1) = \begin{cases} 1, & \text{if} \quad g_i(\hat{\mathbf{r}}_i(n-1)) \geq \Gamma_i \\ 0, & \text{otherwise} \end{cases} . \tag{9}$$

According to $v_i(n-1)$, user i updates the probability distribution $\boldsymbol{\pi}_i(n) = \left(\pi_i^{(1)}(n), \cdots, \pi_i^{(K)}(n)\right)$ and then chooses an action $a_i(n)$. One thing should be pointed out is 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 server can still compute recommendations for him/her 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 there is $a_i(n) \supseteq a_i(n-1)$.

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

$$\pi_i^{(k)}(n) = \begin{cases} \sigma_i(n-1), & \text{if } A^{(k)} = a_i(n-1) \\ \beta_i(n)/\alpha^{c_i\left(A^{(k)}\right)}, & \text{if } a_i(n-1) \subset A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} , \tag{10}$$

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

$$\sigma_i(n-1) = \frac{|a_i(n-1)|}{|S_i|} . \tag{11}$$

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/she is not satisfied with current recommendation. The normalization factor $\beta_i(n)$ is defined as follows:

$$\beta_i(n) = \frac{1}{\sum\limits_{k:\ a_i(n-1) \subset A^{(k)} \subseteq S_i} \alpha^{-c_i\left(A^{(k)}\right)}} . \tag{12}$$

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 $k \in \{1, \cdots, K\}$, $\pi_i^{(k)}(n)$ is now defined as follows:

$$\pi_i^{(k)}(n) = \begin{cases} \mu_i, & \text{if } A^{(k)} = a_i(n-1) \\ \beta_i(n)/\alpha^{c_i\left(A^{(k)}\right)}, & \text{if } a_i(n-1) \subset A^{(k)} \subseteq S_i \\ 0, & \text{otherwise} \end{cases} , \tag{13}$$

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

$$\beta_i(n) = \frac{1 - \mu_i}{\sum\limits_{k:\ a_i(n-1) \subset A^{(k)} \subseteq S_i} \alpha^{-c_i\left(A^{(k)}\right)}} . \tag{14}$$

After every user has chosen his/her action, the RS computes the recommendations based on the rating matrix $\mathbf{R}(n)$ and returns $\hat{\mathbf{r}}_i(n)$ to user i. Then the learning process then 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 rules converges to a SE $\mathbf{a}^+ = (a_1(n_s), \cdots, a_N(n_s))$.

Regarding the convergence of the behavior rule, we provide the following proposition.

Proposition 1: The proposed behavior rule converges to a SE of the game $\hat{G}_{CF} = \left(\mathcal{N}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{f_i\}_{i\in\mathcal{N}}\right)$, if both the following two conditions are satisfied:

(a) for any $\mathbf{a} = (a_1, a_2, \cdots, a_N) \in \mathcal{A}$, $\mathbf{a}' = (a'_1, a'_2, \cdots, a'_N) \in \mathcal{A}$, if $\forall i \in \mathcal{N}$, there is $a_i \subseteq a'_i$, and $\exists j \in \mathcal{N}$, $a_j \subset a'_j$, then for all $i \in \mathcal{N}$, there is $h_i(a_i, \mathbf{a}_{-i}) \leq h_i(a'_i, \mathbf{a}'_{-i})$;

(b) $0 < \mu_i < 1$.

The first condition implies that, after one iteration, every user gets closer to his/her expected result, as long as there is at least one user provides new ratings in the iteration. The second condition implies that during the learning process, whether a user is satisfied or not, he/she may continue to rate items. The two conditions together ensure that as the iterative process proceeds, the recommendation quality perceived by each user gradually improves, even though the improvement may temporarily stagnate for one or more iterations. In other words, a user who hasn't got satisfying recommendation at this moment will eventually be satisfied at some moment. Therefore, with the two conditions, convergence of the behavior rule can be guaranteed.

TABLE I
EXPERIMENT RESULTS.

$\{\eta_i\}_{i=1}^{500}$	run ID	1	2	3	4	5	6	7	8	9	10
$\eta_i = 0.5$	n_{stop}	14	15	17	16	16	13	16	15	15	13
	$\bar{\sigma}_i$	0.462	0.467	0.486	0.480	0.468	0.445	0.472	0.462	0.464	0.449
$\eta_i = 0.8$	n_{stop}	452	556	447	418	450	395	462	483	391	373
	$\bar{\sigma}_i$	0.824	0.848	0.829	0.825	0.825	0.820	0.832	0.832	0.819	0.823
10%: $\eta_i = 0.8$, 90%: $\eta_i = 0.5$	n_{stop}	218	275	166	272	278	392	411	379	295	370
	$\bar{\sigma}_i$	0.790	0.811	0.769	0.805	0.809	0.835	0.839	0.835	0.815	0.829

V. EXPERIMENT RESULTS

A. Dataset and Parameter Setting

To validate the SE learning method, we conduct experiments by using the Jester dataset [11], which contains ratings of 100 jokes from 24983 users. We randomly choose 500 users from the users who have rated all the 100 jokes and use their ratings $\mathbf{R} = [r_{ij}]_{500\times 100}$ to run simulations. Each row of $\mathbf{R}$ is deemed as the corresponding user's interest vector $\mathbf{p}_i$. For each user i, we assume $|S_i| = 70$ and randomly remove 30% of the user's ratings. The resulting matrix is denoted by $\mathbf{R}_{full}$. In all experiments, we set $\alpha = 1.2$, and define $c_i(a_i) = |a_i|$, $\mu_i = 0.9$ for each user. Based on $\mathbf{R}$ and $\mathbf{R}_{full}$, we compute $\Gamma_i^{\max}$ by employing a user-based collaborative filtering algorithm. Then we set $\Gamma_i = \eta_i \Gamma_i^{\max}$, where $0.5 \le \eta_i < 1$ (we assume that user's expectation is no lower than half of the best). We have tested 3 groups of $\{\eta_i\}_{i=1}^{500}$: (i) $\eta_i = 0.5$ for all users; (ii) $\eta_i = 0.8$ for all users; (iii) $\eta_i = 0.8$ for randomly chosen 50 users, and $\eta_i = 0.5$ for the rest. Given a group of $\{\eta_i\}_{i=1}^{500}$, we run the learning method for 10 times. During each run, the iterative process stops when all users are satisfied or the number of iterations reaches 10000. At the beginning of each iteration n, we record the number of satisfied users $N_S(n)$. After each run, we record the number of iterations n_{stop} and the average of the rating completeness $\bar{\sigma}_i \triangleq \frac{1}{500}\sum_{i=1}^{500} |a_i(n_{stop})|/|S_i|$.

B. Resutls

Table I shows the experiment results. As we can see, when all users have moderate expectations for the recommendation quality ($\eta_i = 0.5$), a SE can be reached after about 15 iterations and each user only needs to rate less than half of the items that he/she has experienced. When all users have high expectations ($\eta_i = 0.8$), the learning process becomes much longer. Usually more than 400 iterations are required, and a user has to rate more than 80% of the items that he/she has experienced. When most of the users have moderate expectations and only a few users (10%) expects recommendations of high quality, the learning process is also much longer than the case that all users have moderate expectations.

To better understand the learning results, we draw the set of $N_S(n)$ in Fig. 1. As we can see, in the third setting (depicted by green circles), after only 15 iterations, 90% of the users are already satisfied. During the first 15 iterations, $N_S(n)$ grows at almost the same rate with that of the first setting (depicted by red circles). However, in the third setting, it takes much longer time to achieve the high expectations of the rest users. This is mainly because most of the users are satisfied and

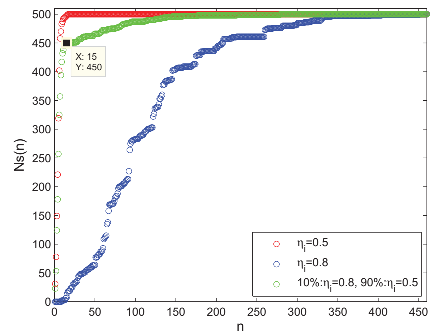

Fig. 1. The change of the number of satisfied users. Red circles represent the result of one run in a setting where $\eta_i = 0.5$ for all $i \in \mathcal{N}$; blue circles represent the result of one run in a setting where $\eta_i = 0.8$ for all $i \in \mathcal{N}$; green circles represent the result of one run in a setting where $\eta_i = 0.8$ for 10% of the users and $\eta_i = 0.5$ for the rest.

they prefer rating no more items, and for those unsatisfied users, the recommendation results only improve a little after one iteration.

From Table I we can also observe that in the third setting, $\bar{\sigma}_i$ generally exceeds 0.8, which is similar to the case when all users have high expectations. This result implies that to satisfy the high expectations of a small amount of users, even users with moderate expectations have to pay high cost, i.e. rating many items, but in return these users can also get recommendations that are much better than they expected. Based on the experiment results, we can conclude that when all users have moderate expectations for recommendation quality, users can achieve a SE with low cost.

VI. CONCLUSION

In this paper, we formulated the interaction among users in a CF system as a game in satisfaction form. To learn the satisfaction equilibrium of the game, we proposed a behavior rule that a user iteratively updates the probability distribution over his/her action space and gradually rate more items. Experiment results demonstrate the feasibility of the proposed behavior rule.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China under Grant No. 61371079, 61273214, 61271267 and 61471025, National Basic Research Program of China under Grant No. 2013CB329105, and a Postdoctoral Science Foundation funded project.

References

[1] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.

[2] M. Grčar, D. Mladenič, B. Fortuna, and M. Grobelnik, "Data sparsity issues in the collaborative filtering framework," in *Proceedings of the 7th International Conference on Knowledge Discovery on the Web: Advances in Web Mining and Web Usage Analysis*, ser. WebKDD'05. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 58–76.

[3] R. Gibbons, *A primer in game theory*. Harvester Wheatsheaf Hertfordshire, 1992.

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

[5] Y. Chen and K. Liu, "Understanding microeconomic behaviors in social networking: An engineering view," *Signal Processing Magazine, IEEE*, vol. 29, no. 2, pp. 53–64, March 2012.

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

[7] ——, "Learning to play a satisfaction equilibrium," in *Workshop on Evolutionary Models of Collaboration*, 2007.

[8] J.-S. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, March 2010, pp. 60–68.

[9] Y. Gao, Y. Chen, and K. Liu, "On cost-effective incentive mechanisms in microtask crowdsourcing," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014, to appear.

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

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