EndorTrust: An Endorsement-based Reputation System for Trustworthy and Heterogeneous Crowdsourcing

Chunchun Wu*, Tie Luo†, Fan Wu*, Guihai Chen*
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
†Institute for Infocomm Research, A*STAR, Singapore
Email: bubble_chun@sjtu.edu.cn, luot@i2r.a-star.edu.sg, {fwu, gchen}@cs.sjtu.edu.cn

Abstract—**Crowdsourcing is a new distributed computing paradigm that leverages the wisdom of crowd and the voluntary human effort to solve problems or collect data. In this context, trustworthiness of user contributions is of crucial importance to the viability of crowdsourcing. Prior mechanisms either do not consider the trustworthiness or quality of contributions or have to assess it only after workers' submission of contributions, which results in irreversible effort expenditure and negative player utilities. In this paper, we propose a reputation system, EndorTrust, to not only assess but also predict the trustworthiness of contributions without wasting workers' effort. The key approach is to explore an inter-worker relationship called endorsement to improve trustworthiness prediction using machine learning methods, while also taking into account the heterogeneity of both workers and tasks.**

I. Introduction

Crowdsourcing is a new problem-solving model that solicits solutions for various tasks from the crowd, particularly online labor markets. The widely available and easily accessible network of labors removes the barrier of operating formal-employment based businesses and of building extensive corporate infrastructure, yet may achieve higher cost-efficiency and flexibility. Furthermore, crowdsourcing has attracted extensive attention and been exploited to develop numerous kinds of applications, such as environment monitoring [1], traffic condition observation [2], and communication quality measurement [3]. These applications open the door for new innovative research and will significantly revolutionize our daily lives.

However, crowdsourcing faces serious challenges as well, among which two crucial ones are incentive and trustworthiness. The incentive issue refers to motivating workers to continually participate in tasks. To this end, many incentive mechanisms [4]–[8] were proposed and they rely on monetary rewards. However, adopting monetary incentives in practice often faces the challenge that it may require a secure and reliable transaction system and may need to assure users to disclose their bank account information to the platform. Some other work [9] does not use monetary incentive but allocates service to uses based on their contribution, but there is still a limitation which also exists in other prior work

including [4]–[8]. That is, task requesters typically only want to pay those workers whose contributions (e.g., solutions, data) have met a certain quality requirement or expectation, but requesters can only assess the quality *after* the workers submit their contributions. Therefore, some workers will receive no payment because they fail to meet the quality requirement, which leads to negative utilities since they have already exerted irreversible effort. This clearly constitutes a *disincentive* to workers.

The second issue, trustworthiness, refers to ensuring the quality of solution or data. This is particularly relevant since workers in crowdsourcing possess very different skill levels and diverse abilities, and there may even be malicious users who put sand in the wheels. In fact, crowdsourcing suffers from many malicious behaviors such as fake customer reviews, social network spam, and collusive campaigns [10]. However, assessing the trustworthiness or quality of contributions happens after submission of contributions, which incurs irreversible effort expenditure and negative utilities as mentioned above. Therefore, the challenge is whether we can design a mechanism to assess the quality or trustworthiness of contributions without ex-ante expenditure.

In this paper, we propose a reputation system, EndorTrust, to not only assess but also predict the trustworthiness of contributions without ex-ante expenditure. We use reputation as a metric to determine how trustworthy a worker is, as well as a motivation for workers to contribute high quality data so as to earn reputation. The novelty of this work lies in the following.

1) We explore an inter-worker relationship called *endorsement* [11] to improve our trustworthiness prediction as well as assessment. Endorsement casts a social-network perspective on workers rather than treating workers individually as in traditional crowdsourcing. To the best of our knowledge, we are the first to leverage connections among workers to design a trustworthy crowdsourcing system.

2) To overcome negative player utilities caused by paying irreversible contribution effort, our mechanism sets out to predict the trustworthiness of contributions rather than requiring workers to contribute first. Our prediction method is based on a most superior machine learning approach and we adapt it to crowdsourcing for the first time.

3) We take into account the heterogeneity of both workers and tasks, where the worker heterogeneity refers to that workers possess different skills or expertise and the task heterogeneity refers to that tasks require different sets of skills and skill levels. Such heterogeneity commonly exists in crowdsourcing but is largely overlooked by

prior work.

4) We associate each endorsement relationship with a degree that dynamically varies according to workers' achievements and performance. Particularly, we model the variation based on the general evolution of human relationship.

The rest of the paper is organized as follows. After briefly reviewing related work in Section II, we present an overview of our system model in Section III. This is then followed by our prediction method detailed in Section IV, and a feedback and adjustment process described in Section V. Then we evaluate the performance of EndorTrust in Section VI, and conclude in Section VII.

II. RELATED WORK

According to a market survey report [12], the U.S. consumers have low (8%) trust in online information sources, which manifests the importance of trustworthiness of information. Indeed, trust analysis has been an actively studied area covering a broad range of issues such as source reputation, information channel robustness, fact finding, data lineage and provenance [13].

Recently, crowdsourcing emerged as a new problem-solving paradigm and has attracted substantial attention from the research community [4]–[9] and led to various real-life or laboratory applications [1]–[3]. Zhang and Schaar [14] presented a generic framework to analyze and design incentive protocols based on social norms for crowdsourcing platforms, with an emphasis on addressing the free-riding problem. Dua *et al.* [15] and Gilbert *et al.* [16] verified data integrity received from user devices within a trusted platform module (TPM). However, TPM-enabled devices requires hardware change and hence the solution is not widely deployed. Huang *et al.* [17] proposed a reputation system that employs the Gompertz function to compute device reputation score as a reflection of the trustworthiness of the contributed data. The work by Luo *et al.* [11] is the first to address both incentive and trustworthiness issues in the context of crowdsourcing (and networked computing in general), by exploiting the incumbent social relationships among people. Peng *et al.* [18] incorporated data quality into incentive mechanism design for crowdsensing in order to motivate participants to perform high-quality sensing, while Tham *et al.* [19] proved that starting from random contribution behaviors, the *quality of contributed service* can converge to a good market equilibrium under a game-theoretical setting.

However, all the above prior work either does not consider the trustworthiness of contributed data or only assesses data quality *after* receiving the contributions, which results in workers paying unnecessary irreversible efforts and thereby receiving negative utilities.

Our work fills this gap by proposing a reputation system to predict, as well as assess, the trustworthiness of contributions. Reputation systems have been generally shown (e.g. see [20] for a survey) to be are an effective tool in a wide range of problems. More importantly, we reinforce our trustworthiness prediction by leveraging an inter-worker relationship called endorsement and using machine learning techniques. In the vast literature, only a handful of works [11], [21] consider the connections among crowdsourcing workers and they have not been applied to trustworthiness prediction.
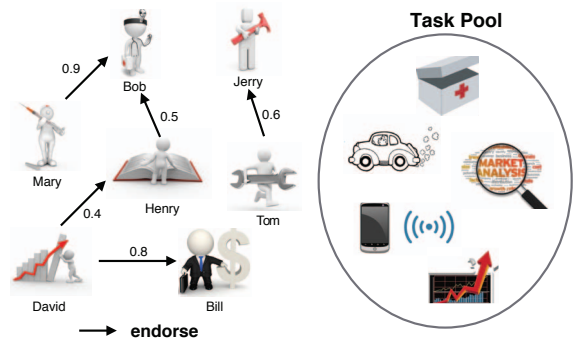

Fig. 1. An illustrative crowdsourcing platform with endorsement links

III. SYSTEM OVERVIEW

A crowdsourcing platform is depicted in Fig.1. A population of workers who possess a diverse range of skills participate in a variety of tasks that require different domain-specific knowledge or expertise. For example, Yahoo! Answers has 25 categories of tasks (questions) ranging from "Computer & Internet" to "Travel" to "Family & Relationships" to "Health" [22].

Workers are connected to one another via endorsement links which represent a trust or support relationship. Every endorsement link has a weight between 0–1 that indicates how much confidence the endorser has in the endorsee. For example in Fig.1, Mary and Henry both endorse Bob, but Mary is a nurse who may have worked with Bob (a doctor) for a long time. So Mary puts a high degree of endorsement (e.g. 0.9) on Bob. On the other hand, Henry may know of Bob only from some TV news or online articles, and thus he sets a moderate degree of endorsement (e.g. 0.5). Consequently, the worker population can be represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathcal{D})$ where $\mathcal{N}$ is the set of workers, $\mathcal{L}$ is the set of endorsement links, and $\mathcal{D}$ is the set of weights. A link $l_{ij} \in \mathcal{L}$ represents that i endorses j for some $i, j \in \mathcal{N}$, and a weight $d_{ij} \in \mathcal{D}$ represents the degree of endorsement link l_{ij}.

The workflow of a crowdsourcing system that uses our endorsement-based reputation system, EndorTrust, is outlined in Algorithm 1. When a requester posts a task, workers who are interested express their willingness to join in by signing up for this task. Then, Lines 6–11 predict the *trustworthiness of contribution* (ToC) for each of the applicants, where two determining factors are each applicant's own reputation and the endorsement impact from his endorsers (Line 10). The underlying prediction method is explained in detail in Section IV. Next, at Lines 12–14, the requester selects the top n workers in terms of their ToC predicted above, to be the qualified workers to perform the task.

After the task is completed, the requester shall give each worker i a rating r_{ik} based on their respective performance in the task. This rating will cause the *reputation score* (RS) of each worker to be updated as follows:

$$RS_i = \frac{\sum_{k \in CoT_i} r_{ik}}{|CoT_i|} \cdot \log(1 + |CoT_i|) \qquad (1)$$

where CoT_i is the collection of tasks performed by i so far (ever since he registered on this platform). The $\log$ term serves the purpose of favoring long-term well-performing

Algorithm 1: Workflow of EndorTrust

Input: The directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \mathcal{D})$.

1 Initialize reputation scores RS_i for every worker;

2 **while** *The platform is in service* **do**

3   A requester posts a task k;

4 $m \leftarrow$ number of workers needed;

5   a set of workers W_k sign up;

6   **foreach** $i \in W_k$ **do**

7   **foreach** j *who endorses* i **do**

8   Predict e_{jk} as j's expertise in task k, according to (8);

9   **end**

10 $ToC_{ik} = RS_i + \sum_{j \in N_i^{in}} RS_j \cdot d_{ji} \cdot e_{jk}$;

11   **end**

12   Sort workers in non-descending order of ToC_{ik}: $ToC_{1k} \geq ToC_{2k} \geq \cdots \geq ToC_{|W_k|k}$;

13 $n \leftarrow min(m, |W_k|)$;

14   The requester selects first n workers in W_k;

15   **foreach** i *who just participated in task* k **do**

16   Requester gives rating r_{ik};

17 $RS_i = \frac{\sum_{k \in CoT_i} r_{ik}}{|CoT_i|} \cdot \log(1 + |CoT_i|)$;

18   **end**

19   Update $\mathcal{D}$;

20 **end**

---

workers to short-term well-performing workers. These happen at Lines 15–18.

Finally, at Line 19, each worker's endorsers will have their degree of endorsement adjusted. This is a feedback and adjustment procedure which will be detailed in Section V.

IV. TRUSTWORTHINESS PREDICTION

We predict the trustworthiness of contribution (ToC) for a worker based on two factors: (i) the worker's reputation score, which reflects his historical performance and reliability, and (ii) the endorsement impact from his endorsers, which consists of the following three elements. First, the reputation scores of his endorsers. Clearly, being endorsed by someone with a higher reputation score has a stronger effect on boosting one's ToC. Second, the degree of endorsement matters. Those who trust you more should be treated differently from those who trust you less. Third, the heterogeneity of tasks (in terms of expertise requirements) and that of workers (in terms of skill profiles). This is important but overlooked by the vast literature, when putting both of them together. To illustrate how this heterogeneity affects the endorsement impact, take Fig.1 as an example again. Suppose a requester posts a task seeking for medical help and Bob (a doctor) is interested and signs up. As Bob is endorsed by both Mary and Henry, should we treat the endorsement impacts from Mary and Henry equally (assuming that they have the same reputation score and the same degree of endorsement)? The answer is no. Since Mary is a nurse and possesses relevant domain knowledge required by this task, she is a good "attestant" as to whether Bob is qualified for the task. On the other hand, Henry is a writer and hence his endorsement would not be so effective on Bob with respect to this particular (medical) task.

Taking all the above factors into account, we predict the

ToC for worker i with respect to task k as

$$ToC_{ik} = RS_i + \sum_{j \in N_i^{in}} RS_j \cdot d_{ji} \cdot e_{jk} \qquad (2)$$

Where e_{jk} is worker j's expertise in performing task k, and N_i^{in} is the set of all the endorsers of worker i.

A. Assessing Expertise via Machine Learning

It is challenging to assess one's expertise on a particular task, i.e., e_{jk} as in (2). This is because the worker may not have performed this particular task before and hence there is no precise historical information to refer to. However, chances are that he may have taken some "similar" tasks and we could "infer" his expertise on this current task.

This motivated us to take a machine learning approach known as *collaborative filtering* [23], which is one of the most effective approaches for building *recommender systems*. Collaborative filtering analyzes relationships between users and interdependencies among products to recommend new user-product associations. A major appealing feature of collaborative filtering is that it can generate more accurate latent features of data than manually collected profiles [23].

The intuition behind collaborative filtering recommendation is that a user's preference for an item is based on ratings of "neighboring items" (i.e., other products that tend to get similar ratings when rated by the same user), and that like-minded users can complement each other's ratings [24]. This assumption also holds in our case. Accordingly, we assume that workers with similar expertise would have similar performance, and a worker would perform similarly on similar tasks.

There are many collaborative filtering methods. In particular, we employ *matrix factorization* [25], which is one of the most superior collaborative filtering methods, to realize our prediction of ToC. Matrix factorization performs extremely well in recommender systems, and has become popular in recent years due to its flexible scalability and predictive accuracy.

To apply this method, we maintain a *rating matrix* with one dimension representing workers and the other representing tasks. An entry r_{ik} in the matrix stands for the performance rating of worker i in task k (given by the task requester after i completes task k). Note that not all r_{ik} have values.

Matrix factorization maps both workers and tasks to a joint latent factor space, where each worker i is characterized by a vector $\vec{p_i}$ and each task k is characterized by a vector $\vec{q_k}$, and worker-task interactions are modeled as inner products in the space. Essentially, the latent factor space tries to explain ratings by characterizing both tasks and workers using factors. For example, a question-and-answer forum may describe a task (i.e., a question) according to its relevance to categories such as fashion, healthcare, education, and electronics, which constitutes factors in $\vec{q_k}$. Similarly, the vector $\vec{p_i}$ describes worker i in terms of his performance with respect to the same factors. Thus, the inner product $\vec{q_k}^T \vec{p_i}$, a linear combination of these factors, captures the interaction between worker i and task k, and is used to approximate worker i's performance in task k, leading to the estimate rating:

$$\hat{r}_{ik} = \vec{q_k}^T \vec{p_i}. \qquad (3)$$

Note that the latent factor vectors $\vec{p_i}$ and $\vec{q_k}$ are not manually assigned but automatically inferred from workers' relevant past performance. This is achieved by minimizing the *regularized squared error* on the set of known ratings κ:

$$\text{argmin}_{\vec{q_k},\vec{p_i}} \sum_{(i,k)\in\kappa} (r_{ik} - \vec{q_k}^T \vec{p_i})^2 + \lambda(||\vec{q_k}||^2 + ||\vec{p_i}||^2). \quad (4)$$

The first term is to fit the model to previously obtained ratings. The second term is to avoid overfitting because our goal is to generalize observed ratings to a model that predicts future, unknown ratings. So we regularize the parameters by penalizing on magnitudes. The parameter λ is a constant that controls the extent of regularization, and is usually determined by cross-validation.

We solve this optimization problem using *stochastic gradient descent* [26], which is an effective method well adopted by both academic and industrial communities [27]–[29]. This stochastic gradient descent algorithm loops through all known ratings in the training data (i.e. κ). For each given rating r_{ik}, a prediction $\hat{r}_{ik} = \vec{q_k}^T \vec{p_i}$ is made, and it calculates the associated prediction error:

$$\delta_{ik} := r_{ik} - \hat{r}_{ik}. \quad (5)$$

Then it modifies the vectors $\vec{p_i}$ and $\vec{q_k}$ by moving in the opposite direction of the gradient, yielding:

$$\vec{q_k} \leftarrow \vec{q_k} + \gamma \cdot (\delta_{ik} \cdot \vec{p_i} - \lambda \cdot \vec{q_k}) \quad (6)$$
$$\vec{p_i} \leftarrow \vec{p_i} + \gamma \cdot (\delta_{ik} \cdot \vec{q_k} - \lambda \cdot \vec{p_i}) \quad (7)$$

where γ controls the rate of descenting.

Finally, after solving for $\vec{p_i}$ and $\vec{q_k}$, we can determine the expertise of each endorser j in task k, for all j, k:

$$e_{jk} = \begin{cases} r_{jk} & \text{If } (j,k) \in \kappa, \\ \vec{q_k}^T \vec{p_j} & \text{Otherwise.} \end{cases} \quad (8)$$

Thus, before a worker actually performs a task, Eqn. (2) predicts the trustworthiness of contribution for the worker on the task, by combining his own reputation and the endorsement impact from his endorsers with respect to the task.

V. FEEDBACK AND ADJUSTMENT

After a task is completed, the requester gives each worker a rating based on their respective performances. This will affect not only their reputation score (see (1)), as can be expected in a typical reputation system, but also how much their endorsers will continue to "trust" these workers, which is characterized by the degree of endorsement in our model.

To determine how to adjust the degree of endorsement, we draw an analogy to human relationship which generally evolves in the following manner: a relationship develops slowly between strangers, and then slightly accelerates when they become acquainted with each other; subsequently, it makes steady progress as they cultivate sort of a friendship, and after a sufficiently long period of time, it reaches certain intimacy. We use the *generalized logistic function* [30] to model this evolution process. This function has a customizable S-shape curve, which well captures the progression from slow start to acceleration to steady incline and eventually approaching a climax:

$$D_{ji}(R_i) = A_{ji} + \frac{B_{ji} - A_{ji}}{(1 + Qe^{-K(R_i-M)})^{1/v}} \quad (9)$$

where A is the lower asymptote, and B is the upper asymptote. K determines the growth rate, while $v > 0$ affects near which asymptote maximum growth occurs. Q depends on the value $D(0)$. M indicates the input value of maximum growth if $Q = v$. These parameters can be manipulated to customize one's expected S-shape curve. The celebrated *sigmoid function* [31] is a special case of generalized logistic function, which is shown in Fig. 2 and defined by the formula:
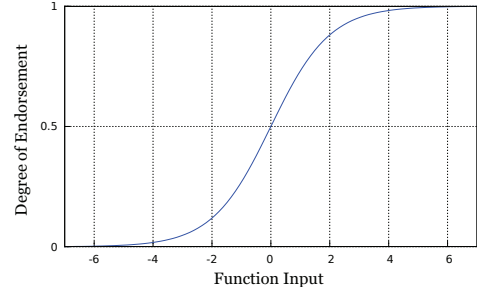
$$S(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

Fig. 2. Sigmoid Function

The output of (9), D_{ji} which is between 0 and 1, represents the degree of endorsement from worker j to worker i. As mentioned before, endorsement implies a sense of trust or support that may come from real-life social acquaintance, and thus may have a large discrepancy. When a new endorsement link l_{ji} is established, it is accompanied with an initial degree of endorsement $\bar{d}_{ji}$, which represents how trustworthy the endorser perceives the endorsee in the first place. Apart from that, the endorser also provides his tolerance level c_{ji} indicating to what (maximum) extent the endorser would allow the initial trust to change. Then we elaborate parameters A_{ji} and B_{ji}, the lower and upper asymptotes of D_{ji}, respectively, which are determined by

$$A_{ji} = \max\{0, \bar{d}_{ji} - c_{ji}\} \quad (11)$$
$$B_{ji} = \min\{1, \bar{d}_{ji} + c_{ji}\} \quad (12)$$

As for the input of the generalized logistic function, R_i is worker i's overall performance by aggregating historical information, which is calculated as follows:

$$R_i = \sum_{k \in CoT_i} \alpha^{t_0-t_k} \cdot (r_{ik} - \frac{\sum_{j \in CoW_k} r_{jk}}{|CoW_k|}) \quad (13)$$

where CoT_i is the collection of tasks performed by worker i, and CoW_k is the collection of workers who have ever participated in task k. The discounting factor, $\alpha^{t_0-t_k}$, where t_0 is current time and t_k is the finished time of task k, prioritizes recent information over the past. The weight $0 < \alpha \leq 1$, reduces the impact of past data and achieves ageing. $\frac{\sum_{j \in CoW_k} r_{jk}}{|CoW_k|}$ is the average performance of workers who attend task k. When worker i's performance is above average, he will be considered well-behaved. Otherwise, he will disappoint his endorsers (reducing the degree of endorsement), and be considered not worthy of confidence.

Moreover, relationship between humans tends to be built up gradually through many trustworthy transactions, while

torn down rapidly after even a handful of disappointed behaviors [17]. The building rate and breaking rate of relationship between humans are asymmetrical. This feature can be easily facilitated by adopting different ageing weights for well-behaved workers and ill-behaved workers respectively. The building weight α_{well} applies to workers whose performance is above average while the breaking weight α_{ill} applies otherwise. That is,

$$\alpha = \begin{cases} \alpha_{well} & \text{If } r_{ik} > \frac{\sum_{j \in CoW_k} r_{jk}}{|CoW_k|} \\ \alpha_{ill} & \text{Otherwise.} \end{cases} \quad (14)$$

Note that, $\alpha_{ill} > \alpha_{well}$, so that the summation in Eq.13 is dominated by negative terms (when worker's performance is poor). Therefore, worker i needs to work harder and pay more effort to offset his past ill-behaved performance, or in other words, to compensate for hurting his endorsers' "feelings".

VI. Performance Evaluation

In this section, we conduct a series of simulations to evaluate the performance of EndorTrust. The benchmark used for comparison is a typical reputation system which uses reputation scores as the only selection criteria (instead of our predicted ToC which combines both reputation score and endorsement impact).

In order to simulate the rating generation (which in reality is done by task requesters), we associate (i) each task with a *demand vector* that represents the task's demand on different skills, and (ii) each worker with a *competence vector* that represents the worker's competence in the same set of skills. Note that although these two vectors are semantically the same as the latent factor vectors $\vec{p_i}$ and $\vec{q_k}$, they are only used in simulations to model the ground truth in order to generate ratings, whereas $\vec{p_i}$ and $\vec{q_k}$ are used in actual crowdsourcing systems to compute the expertise e_{ik}.

Thus, a rating r_{ik} can be generated by a dot product of worker i's competence vector and task k's demand vector, upon which we add a small Gaussian perturbation to simulate the randomness of human (requester) behaviors. In our simulation, there are 20 workers and 50 tasks with their respective competence vectors and demand vectors randomly generated with a dimension of 10. Both vectors are normalized such that the rating is between 0 and 1. The variance of the Gaussian distribution is set to 0.01.

In order to generate the endorsement relationships, we assume that the willingness to endorse someone depends on the competence of the endorsee and the familiarity between the endorser and the endorsee. We use the mean of each worker's competence vector to represent his competence, and the *cosine similarity* of two worker's competence vectors to approximate their familiarity. Cosine similarity is a widely used measure of similarity between two vectors, defined as

$$sim(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2}} \quad (15)$$

where $\vec{A} = \{a_1, a_2, ..., a_n\}$ and $\vec{B} = \{b_1, b_2, ..., b_n\}$. The above similarity score ranges from -1, meaning exactly opposite, to 1, meaning exactly the same, with 0 indicating orthogonal (irrelevant). Thus, we generate endorsement links using a probability of worker i endorsing worker j that is

proportional to worker j's competence and the familiarity between i and j.

A. Requester Satisfaction

In our first set of experiments, we examine *requester satisfaction*, reflected by the average rating in each round, where a round is determined by the completion of all the 50 tasks. We run simulation for a total of 50 rounds to observe the variation of requester satisfaction. The results are shown in Fig. 3, where we see that EndorTrust achieves higher requester satisfaction than the Benchmark. This is attributed to the advantage that EndorTrust can identify competent workers more accurately.
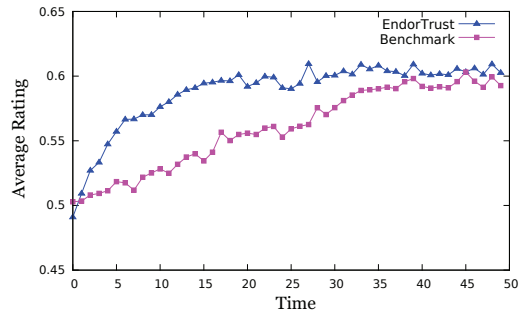

Fig. 3. Requester satisfaction.

B. Discrimination Problem

The discrimination problem is a thorny issue in reputation communities: When a new user join a community, he may never get a chance to be selected to perform a task because others have had sufficient time to accumulate high reputation scores; to make it worse, this situation will persist because he has no chance to perform tasks to increase his reputation score. EndorTrust can break this vicious circle through endorsement: new workers who are indeed trustworthy can be endorsed by others through social acquaintance [11], which helps increase his ToC and hence the chance of being selected.
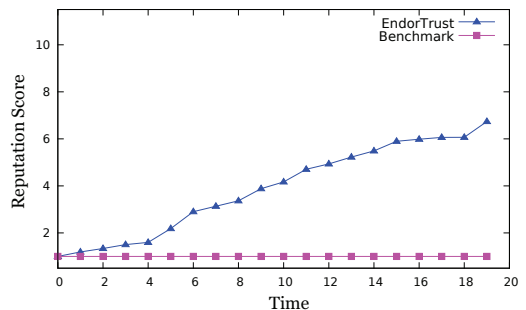

Fig. 4. Reputation score of a new worker.

To examine this problem, we add a new worker with low (initial) reputation score and a high competence vector, after the first 50 rounds of simulations as described in Section VI-A, during which other workers have gained medium to high reputation scores. The simulation results are presented in Fig. 4. We see that in EndorTrust, the new worker does get opportunities to undertake tasks and thereby build up his reputation. This is because other workers endorse him with

the probability described earlier in Section VI. On the other hand, he is never recognized in Benchmark to contribute to crowdsourcing even though he is a worker with good potential. This demonstrates that, by leveraging endorsements, EndorTrust is able to overcome the discrimination problem which traditional reputation systems suffer from.

C. Cold-start Problem

Cold-start problem is another long-lived obstacle for reputation systems. For a new system, it cannot draw any inference for any user because it has not yet gathered sufficient information. As a result, a requester has to randomly select workers to perform tasks. This situation only gradually gets improved after many rounds when sufficient historical information is accumulated. In our experiments, we observe that, as depicted in Fig. 5, EndorTrust enables task requesters to quickly identify qualified workers through the endorsement impacts; but on the other hand, the traditional reputation system Benchmark has to go through the cold-starting period slowly because of lacking historical information of workers.
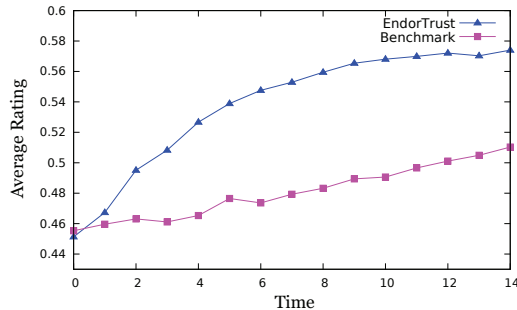

Fig. 5.   Requester satisfaction in the starting period.

VII.   CONCLUSION

In this paper, we leverage an endorsement relationship among workers to design a reputation system for trustworthy crowdsourcing, called EndorTrust, which takes into account the heterogeneity of both workers and tasks. The system predicts the trustworthiness of contributions using a collaborative filtering method, matrix factorization, by which we avoid expenditure of workers' irreversible effort and help requesters select the most qualified workers to perform tasks. Our mechanism also provides a solution to the discrimination problem and the cold-start problem generally existing in reputation systems.

REFERENCES

[1] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *KDD*, 2013.

[2] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *MobiSys*, 2011, pp. 127–140.

[3] F.-J. Wu and T. Luo, "WiFiScout: A crowdsensing wifi advisory system with gamification-based incentive," in *MASS*, 2014.

[4] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *MOBICOM*, 2012.

[5] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *INFOCOM*, 2014.

[6] S. Chawla, J. D. Hartline, and B. Sivan, "Optimal crowdsourcing contests," in *SODA*, 2012.

[7] I. Krontiris and A. Albers, "Monetary incentives in participatory sensing using multi-attributive auctions," *International Journal of Parallel, Emergent and Distributed Systems*, June 2012.

[8] T. Luo, S. S. Kanhere, S. K. Das, and H.-P. Tan, "Optimal prizes for all-pay contests in heterogeneous crowdsourcing," in *MASS*, 2014.

[9] C. K. Tham and T. Luo, "Fairness and social welfare in service allocation schemes for participatory sensing," *Computer Networks*, vol. 73, pp. 58–71, 2014.

[10] G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Serf and turf: crowdturfing for fun and profit," in *WWW*, 2012.

[11] T. Luo, S. S. Kanhere, and H.-P. Tan, "Sew-ing a simple endorsement web to incentivize trustworthy participatory sensing," in *SECON*, 2014.

[12] A. Enright, "Consumers trust information found online less than offline messages," in *Internet Retailer*, 2010.

[13] M. Gupta and J. Han, "Heterogeneous network-based trust analysis: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 1, pp. 54–71, 2011.

[14] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *INFOCOM*, 2012.

[15] A. Dua, N. Bulusu, W.-C. Feng, and W. Hu, "Towards trustworthy participatory sensing," in *USENIX*, 2009.

[16] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall, "Toward trustworthy mobile sensing," in *HotMobile*, 2010.

[17] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: the case for a reputation system in participatory sensing," in *MSWiM*, 2010.

[18] P. Dan, W. Fan, and C. Guihai, "Pay as how good you do: A quality based incentive mechanism for crowdsensing," in *MobiHoc*, 2015.

[19] C. K. Tham and T. Luo, "Quality of contributed service and market equilibrium for participatory sensing," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 829–842, 2015.

[20] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[21] H. Amintoosi and S. S. Kanhere, "A reputation framework for social participatory sensing systems," *Mobile Networks and Applications*, vol. 19, no. 1, pp. 88–100, 2014.

[22] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: everyone knows something," in *WWW*, 2008.

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

[24] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995.

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

[26] W. A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique," *Signal Processing*, vol. 6, no. 2, pp. 113–133, 1984.

[27] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds.   Springer, 2010, pp. 177–186.

[28] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *ICML*, 2004.

[29] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *SIGKDD*, 2011.

[30] F. Richards, "A flexible growth function for empirical use," *Journal of experimental Botany*, vol. 10, no. 2, pp. 290–301, 1959.

[31] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation*.   Springer, 1995, pp. 195–201.