Worker Recommendation with High Acceptance Rates in Collaborative Crowdsourcing Systems

Mingchu Li[1,2], Xiaomei Sun[1,2], Xing Jin[1,2](✉), and Linlin Tian[1,2]

[1] School of Software Technology, Dalian University of Technology, Dalian 116620, China
{mingchul,linlint}@dlut.edu.cn, sunxiaomeijob@mail.dlut.edu.cn, jinxingdlut@gmail.com

[2] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China

Abstract. Crowdsourcing has emerged as a popular Internet-based collaboration paradigm, in which tasks published by requesters can be economically and efficiently accomplished by crowd workers. To ensure the quality of service (QoS) provided by crowd workers, requesters are more likely to assign tasks to trustworthy workers, therefore, trust have played an important role in the design of worker recommendation mechanisms in crowdsourcing systems. Most existing studies focus on the trust that requesters place on workers, however, which would suffer the low-acceptance problem because crowd workers would refuse to participate in tasks published by low-trustworthy requesters with a great probability. In order to address the low-acceptance problem, in this paper, by using biased matrix factorization, we proposed a novel worker recommendation mechanism which can evaluate mutual trust relationship between requesters and workers. And also, to accurately measure the matching degree between tasks and workers, a comprehensive and practical task matching mechanism has been presented by incorporating time matching, skill matching, payment matching, and location matching. Finally, extensive simulations and real data experiments highlight the performance of our proposed worker recommendation mechanism.

Keywords: Collaborative crowdsourcing · Trust · Worker recommendation · Matrix factorization · Task matching

1 Introduction

In recent years, numerous websites and applications have been widely applied in various fields to realize the great potential of the crowdsourcing, including Wikipedia[1], Amazon Mechanical Turk (AMTurk)[2], Gigwalk[3], etc. In

[1] https://www.wikipedia.org/.
[2] https://www.mturk.com/.
[3] http://www.gigwalk.com/.

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 53–74, 2019.
https://doi.org/10.1007/978-3-030-12981-1_4

crowdsourcing systems, participators are required to work collaboratively to accomplish a crowdsourcing task published by a requesters.

One important aspect of crowdsourcing systems is worker recruitment, which directly affects the success of crowd tasks. There are numerous literatures available studying on worker recruitment, which can be generally classified into two methodologies: billboard and worker recommendation.

Billboard. Tasks are distributed on specialized crowdsourcing platforms that crowd workers can participate in their preferred tasks on a first-come-first-served basis. Due to the simplicity of being implemented in practice, the billboard methodology has been widely used in industry filed, for example, AMTurk and Gigwalk. However, the billboard methodology suffers the following two potential disadvantages.

(1) Long time waiting. In large-scale crowdsourcing platforms, a worker need to spend much time on picking a proper task to participate when confronting with numerous and varied tasks. Subsequently, requesters would take a long time to recruit enough workers, which is not allowed for real-time tasks. As illustrated in [17], less than 15% tasks can be completed within one hour in AMTurk.
(2) Low quality service. Due to the open and dynamic nature of crowdsourcing systems, in which there exists various workers with diverse abilities and reliability properties. To guarantee the quality of service (QoS) provided by workers, reputation systems have been naturally introduced in. However, in practice, the deployment of reputation systems encounter the challenge of reputation inflation. As illustrated in [5], workers' acceptance rates (one form of reputation value) on AMTurk are often above 97%, regardless of workers' performances. Therefore, in the billboard methodology, requesters would receive low QoS due to the fact that the truth ability and reliability of workers cannot be clearly figured out.

Worker Recommendation. To address the above two potential disadvantages in billboard, worker recommendation [1–3,6,13,22,24,26,31,34] has been proposed as the alternative methodology for worker recruitment, in which proper workers will be automatically recommended basing on requesters' requirements. According to existing literatures, the worker recommendation based approach normally consists two parts: task matching and trust evaluation. Task matching part assesses the probability of a worker to complete a specific task, in terms of skills, payments, etc. Trust evaluation part describes the confidence that a requester places on a worker to finish a task expectably. To ensure the QoS, high-trustworthy workers are more likely to be selected as participants.

In bilateral rating crowdsourcing systems, for example, Zhubajie[4] (a famous crowdsourcing platform in China), requesters and workers will rate each other after transactions. Let us consider the following scenario: (1) Requester A trusts worker B by giving a positive rating for her acceptable performance; (2) Due

[4] http://www.zbj.com/.

to the delay in payment, B would place a distrust on A by giving a negative rating. Regarding to a new crowd task published by A, B would be unceasingly recommended to A as a potential participant with great probability by using traditional worker recommendation mechanisms which only consider the trust that requesters place on workers. However, due to the distrust that worker B places on requester A, B would refuse to take the task, which will lead a low-acceptance problem of crowd tasks. In order to address the low-acceptance problem, the trust evaluation from workers to requesters should also be studied, however, which has not been considered before in the domain of crowd worker recruitment, to the best of our knowledge.

Our major contributions in this paper are listed as follows.

– We propose a comprehensive task matching mechanism by incorporating time matching, skill matching, payment matching, and location matching. To our best of knowledge, no other previous work has considered in all aspects.
– Basing on biased matrix factorization [12], we firstly propose three methods (Method 1, Method 2 and Method 3) to evaluate the mutual trust relationships between requesters and workers.
– We propose a top-K worker recommendation framework for worker recruitment by combining the proposed task matching mechanism and trust evaluation methods, which also consider the cold start problem.
– We conduct extensive simulations and experiments over a large public dataset to assess the performance of our proposed worker recommendation framework. Results demonstrate that our methods outperform all baselines in terms of trust evaluation and acceptance rates.

The remainder of this paper is organized as follows. In Sect. 2, we give an overview of related work on worker recommendation methodology. In Sect. 3, we present the system model in detail. In Sect. 4, we propose a top-K worker recommendation framework. In Sect. 5, we provide simulations and real data experiments to illustrate the performance of the proposed methods. In Sect. 6, we conclude this paper.

2 Related Work

We particularly review the related work on worker recommendation methodology in terms of task matching and trust evaluation.

Task Matching. In [3], workers and tasks are matched based on an underlying taxonomy that workers' interests are taken into account. Goel et al. [6] consider heterogeneous set of tasks requiring certain skills, and each worker has certain expertise and interests, and then a bipartite graph has been used to represent all the possible assignments between the workers and the tasks. Ye et al. [31] present a context-aware task matching model by incorporating task type and reward amount. Yuen et al. [34] propose a TaskREC framework based on a unified probabilistic matrix factorization to predict the worker-task rating.

Kurve et al. [13] consider the task difficulty and worker skill in the assignment of crowd tasks. In [22], Schnitzer et al. reveal that task similarities act as a key parameter for building a task matching mechanism. Tong et al. [24] study the online mobile micro-task allocation problem in spatial crowdsourcing by considering the time matching and location matching. In CrowdAdvisor [1] framework, five metrics are defined in worker recommendation: Personal Characteristics, Freelancer-Job compatibility, Freelancer-Client compatibility, Freelancer-Team compatibility and Freelancer Motivation. Yuan et al. [33] observe that people usually exhibit different levels of busyness at different contexts in task assignment, and then propose a model to predict people's interruptibility intensity. By using the fuzzy c-Means algorithm, Alsayasneh et al. [2] match workers with personalized and diverse tasks. [20] and [27] consider the timeliness in task matching. In the spatial crowdsourcing scenario, Wang et al. [26] propose an effective heuristic methods to solve the multi-objective optimization in which task coverage is maximized and incentive cost is minimized.

However, these above researches don't study the trust relationship between requesters and workers.

Trust Evaluation. To ensure the QoS provided by crowd worker, reputation and trust mechanisms have been widely used. In [32], Yu et al. propose a social welfare optimizing reputation-aware decision-making approach for task assignment. EndorTrust [28] has been proposed as a reputation system which not only assess but also predict the trustworthiness of contributions without wasting workers' effort. Ye et al. [30] extend their previous work [31] by considering the trust that requesters place on workers, in which random walker algorithm has been adopted. To address the issue of assign reliable workers to nearby tasks in spatial crowdsourcing, Hassan and Curry [7] use semi-bandit learning to reduce the uncertainty of worker reliability. Xiang et al. [29], propose a trust-based mixture of Gaussian processes (GP) model to yield accurate estimations in the presence of various types of misbehaving workers, in which a Bayesian trust framework has been developed to maintain and update trustworthiness of crowd workers. In [9], a new comprehensive model for computing reputation scores of workers has been proposed by considering direct and indirect evaluations expressed by requesters.

However, these above researches don't consider the trust that workers place on requesters, hence they will confront with the low-acceptance problem of crowd tasks.

3 System Model

In this paper, we consider a bilateral rating crowdsourcing system with M requesters and N workers. And each crowd task need multiple workers to participate in, for example, monitoring the traffic [4].

3.1 Task Features

Heterogeneous tasks are studied in our crowdsourcing model, which can be characterized by the following six attributes.

Time Tolerance. Different tasks may have different time tolerance, e.g., some of them need to be finished immediately [23,24]. We use $T_{tolerance} \in [0,1]$ to represent the time tolerance of a certain task, smaller value of $T_{tolerance}$ intuitively denotes the more urgent need to be accomplished. Different from [23,24], we explicitly compute the value of $T_{tolerance}$ by using the following formula:

$$T_{tolerance} = \frac{2}{1+e^{t_{now}-t_{finish}}} - 1,$$

where t_{now} and t_{finish} stand for the published time of a task and the deadline for calling for workers, respectively, hence, we have $t_{now} < t_{finish}$. The published time can be automatically obtained once a certain task has been distributed, and the deadline would be directly described by the task owner.

Task Type. Heterogeneous crowd tasks [1,3,6,22] are considered in this paper. It is assumed that there are k categories of skills in our model. We use a vector $T_{type} = [t_1, t_2, \ldots, t_k]$ to represent the type of a certain task. Element t_i has a binary value: 1 means the i-th skill is required to finish the given task, otherwise, 0 means the i-th skill is not required.

Skill Level. For a specific task type, heterogeneous skill levels should also be studied [13,18]. We use $T_{level} = [l_1, l_2, \ldots, l_k]$ to represent the requirement of skill level for a certain task. Each element l_i has a continuous value from 0 to 1, the higher value of skill level intuitively carries the greater demand of proficiency in i-th skill, e.g., $l_i \geq 0.9$ means that an expert is required.

In sum, $T_{requirement} = [r_1, r_2, \ldots, r_k]$ has been defined to represent the requirement of skill and skill level of a certain task, simultaneously, where $r_i = t_i * l_i$.

Task Payment. Every worker who finishes a certain task will gain a payment p from the task owner [1,3,5,26,31,34,37]. The payment p will be directly described by the requester once the task is published.

Geographical Location. Regarding to a certain task in spatial crowdsourcing systems [7,23,24,26], the attribute of geographical location should be considered, which has been represented as $T_{geography} = [g_1, g_2]$, where g_1 and g_2 correspond to the longitude and latitude coordinates, respectively.

Number of Workers. Generally, for the crowdsourcing paradigm, multiple workers are needed to complete a complex crowd task [17,18,24,26,31,37]. And in this paper, we use a variable T_n to represent the required number of workers for a certain task.

3.2 Worker Features

Subsequently, heterogeneous worker is studied in our crowdsourcing model, which can be characterized by the following five attributes.

Degree of Busyness. Intuitively, a busier worker would participate in a new task with a smaller probability. Therefore, to develop an efficient worker recommendation mechanism, the degree of busyness for each worker should be evaluated [25,33]. Different from pervious studies, the busyness of a worker is explicitly presented. We use a variable $W_{busyness} \in [0, 1]$ to represent the degree of busyness of a certain worker. The higher value of $W_{busyness}$ implies the much busier the worker will be. Intuitively, a busier worker would participate in a new task with a smaller probability. The degree of busyness can be calculated as:

$$W_{busyness} = \frac{1}{1 + e^{-\alpha_1(n_{task} - \beta_1)}},$$

where n_{task} stands for the number of tasks that the worker has participated in. Parameters α_1 and β_1 are introduced to control the value of $W_{busyness}$. Specially, n_{task} can be set as ∞, if a worker is not online.

Worker Skill. Corresponding to the heterogeneous crowd tasks, the heterogeneity of worker skill is studied [1,3,6,22]. We use $W_{skill} = [s_1, s_2, \ldots, s_k]$ to represent the skill ability of a certain worker. Each element s_i has a binary value: 1 means the i-th skill is owned by the given worker, otherwise, s_i will be fixed as 0.

Skill Proficiency. Regarding to a specific skill, a worker could be an elementary, intermediate, advanced, or expert player [1,21]. We use $W_{proficiency} = [p_1, p_2, \ldots, p_k]$ to represent the proficiency level in each skill for a certain worker. Element p_i has a continuous value from 0 to 1, the higher value of p_i illustrates the greater ability in the use of i-th skill. Based on the data of history transactions, we can formulate p_i as:

$$p_i = \frac{n_j^*}{n_j} \cdot \frac{1}{1 + e^{-\alpha_2(n_j - \beta_2)}},$$

where n_j and n_j^* stand for the number of times to use the j-th skill within previous crowd tasks and the number of acceptance times for using the j-th skill, respectively. Obviously, $\frac{n_j^*}{n_j}$ means the acceptance rate in the use of the j-th skill. And $\frac{1}{1+e^{-\alpha_2(n_j-\beta_2)}}$ is monotonically increasing with the increase of n_j. Therefore, we get that p_i is proportional to the acceptance rate and practical experience.

In sum, $T_{ability} = [a_1, a_2, \ldots, a_k]$ is defined to represent the skill and skill proficiency of a certain worker, simultaneously, where $a_i = s_i * p_i$.

Bid. We use $W_{bid} = [b_1, b_2, \ldots, b_k]$ to represent the expected bid of a certain worker [1,3,5,26,31,34,37], its element b_i means the expected reward that will be obtained from a requester by using the i-th skill during a crowd task.

Geographical Location. Each worker in our crowdsourcing model has the information of geographical location [7,23,24,26], which can be represented as $W_{geography} = [gw_1, gw_2]$, where gw_1 and gw_2 denote the longitude and latitude coordinates of a certain worker, respectively.

3.3 Trust Relationship

Trust from Requesters to Workers. Generally, in crowdsourcing systems, after worker j finishes a task published by requester i, j would get a positive rating from i, if i is satisfied with j's performance, otherwise, j will get a negative rating. Based on the ratings provided by requesters, we can compute the trust from requesters to workers. Subsequently, a matrix $\mathrm{RW} = [rw_{i,j}]_{M,N}$ can be built to represent the trust relationship from requesters to workers, and its element $rw_{i,j}$ denotes the specific trust that requester i places in worker j. The detailed method to compute the trust can be found in the Eigentrust [10,16].

Trust from Workers to Requesters. Due to the open nature of crowdsourcing systems, there also may exist untrustworthy requesters who may refuse to offer payments to workers or be extremely critical on workers' performances, etc. To protect the benefits of workers, the reliability of requesters should also be considered.

In a bilateral rating crowdsourcing system, with the similar procedure in building RW matrix, $\mathrm{WR} = [wr_{j,i}]_{N,M}$ matrix can be constructed basing on the feedbacks from workers to requesters over transactions, its entry $wr_{j,i}$ denotes the specific trust that worker j places on requester i.

4 Worker Recommendation Framework

In this section, we discuss the proposed worker recommendation framework in detail in terms of task matching, trust evaluation and top-K recommendation.

4.1 Task Matching

To pick proper workers for a certain task, we firstly propose a comprehensive task matching mechanism by incorporating time matching, skill matching, payment matching, and location matching.

Time Matching. Naturally, to satisfy the time requirement of requesters in completing tasks, an urgent task should be matched with an idle worker.

Given the time tolerance feature $T_{tolerance}$ of a certain task t and the busyness feature $W_{busyness}$ of a certain worker w. As discussed in Sects. 3.1 and 3.2, the higher value of $T_{tolerance}$ means the task can wait more time to be performed, and the lower value of $W_{busyness}$ means the worker has more available time to take a new task. Therefore, the time matching can be qualitatively defined as: $T_{tolerance} - W_{busyness}$. Intuitively, it could be assumed that w can finish t in time if $T_{tolerance} - W_{busyness} > 0$, otherwise, t is not suitable for w to participate in. By normalizing $T_{tolerance} - W_{busyness}$, we can model the time matching as:

$$M_{time} = \frac{1}{1 + e^{-\alpha_3(T_{tolerance} - W_{busyness} + \beta_3)}}, \tag{1}$$

where parameters α_3 and β_3 are used to regulate the value of M_{time}.

Skill Matching. To improve the QoS of crowd tasks, a complex task should be assigned to skilled workers.

$T_{requirement}$ and $W_{ability}$ are given as the skill requirement of a certain task t and the skill ability of a certain worker w, respectively. Considering the fact that there may exist similarity between two different skills, e.g., a worker could has the probability of coding *Python* program if she/he is skilled in writing *C++* program. To have a comprehensive view of workers' ability, we propose a matrix $Corr = [c_{i,j}]_{k,k}$ to represent the similarity of any two skills, its entry $c_{i,j}$ denotes the similarity between i-th and j-th skill. How to build the $Coor$ matrix is out of scope of this paper, one feasible way is to use the clustering technology [14,15,35,36].

We assume a_i as the explicit ability of a certain worker, and a_j is assumed as the implicit ability which has not been used before. Basing on the similarity between tasks, a_j can be updated as:

$$a_j^* = \max_i a_i \cdot c_{i,j}.$$

Thus, a comprehensive ability $W_{ability}^c = [a_1^c, a_2^c, \ldots, a_k^c]$ of worker w over k skills can be proposed, where $a_i^c = a_i$ for explicit ability, and $a_j^c = a_j^*$ for implicit ability. Intuitively, the skill matching m_{skill} can be defined as:

$$m_{skill} = \frac{W_{ability}^c \cdot T_{type}}{T_{requirement} \cdot T_{type}},$$

the higher value of m_{skill} implies the better matching degree. Particularly, $m_{skill} = 0$ means totally mismatching, while $m_{skill} = 1$ means completely matching. By normalizing m_{skill}, we get the skill matching degree as:

$$M_{skill} = \frac{1}{1 + e^{-\alpha_4(m_{skill} - 1 - \beta_4)}}. \tag{2}$$

The term $m_{skill} - 1$ has the similar function as $T_{tolerance} - W_{busyness}$ plays in Eq. (1), a well skill matching M_{skill} will be obtained if $m_{skill} - 1 \geq 0$, otherwise, worker w is regarded as not very suitable for task t.

Payment Matching. The amount of payment plays an important role that workers would like to receive payments from requesters that are not less than their expectation.

Given the task payment feature p of a certain task t and the bid feature W_{bid} of a certain worker w. We use $b = T_{type} \cdot W_{bid}$ to represent the total expected bid of worker w for task t. Intuitively, the difference value $p - b$ can be used to measure the payment matching, the higher value of the difference illustrates the greater probability of w to participate in t. However, this definition of payment matching suffers some vulnerability. Let us consider the following two scenarios.

Scenario 1: The payment $p = 99$ for task t, and the total bid $b = 100$ for worker w.
Scenario 2: The payment $p = 9$ for task t, and the total bid $b = 10$ for worker w.

Obviously, the payment matching in Scenario 1 is better than in Scenario 2, however, $p - b$ obtains the same value 1 in these two scenarios. To address this issue, we revise the payment matching by introducing the total expected bid as the denominator:

$$m_{payment} = \frac{p-b}{b}.$$

By normalizing $m_{payment}$, the skill matching degree can be formally defined as:

$$M_{payment} = \frac{1}{1 + e^{-\alpha_5 (m_{payment} - \beta_5)}}. \tag{3}$$

Location Matching. To decrease the cost of moving for involved workers, location matching should be considered.

Given the geographical location $T_{geography} = [g_1, g_2]$ of a certain task and $W_{geography} = [gw_1, gw_2]$ of a certain worker, we can use Euclidean distance to measure the distance between $T_{geography}$ and $W_{geography}$:

$$d = \sqrt{(g_1 - gw_1)^2 + (g_2 - gw_2)^2},$$

the smaller value of d demonstrably illustrates the higher matching in location, and then the location matching $M_{location}$ can be formulated as:

$$M_{location} = e^{-\alpha_6 \cdot d}, \tag{4}$$

which has a continuous value from 1 to 0 with the increasing of d.

Personalized Matching Degree. Different requesters may have different priorities in task matching. For example, requesters who have high demand in QoS would assign more weight to skill matching. In order to propose a personalized matching degree, we introduce a weight vector $W = [w_1, w_2, w_3, w_4]$ which can be directly defined by the task owner, its entries w_1, w_2, w_3 and w_4 denote the weight of M_{time}, M_{skill}, $M_{payment}$ and $M_{location}$, respectively. By using the harmonic mean equation, the personalized matching degree can be obtained:

$$\begin{gathered} M = \frac{1}{\frac{w_1}{M_{time}} + \frac{w_2}{M_{skill}} + \frac{w_3}{M_{payment}} + \frac{w_4}{M_{location}}} \\ s.t. \sum_i w_i = 1 \end{gathered} \tag{5}$$

4.2 Trust Evaluation

In order to address the low-acceptance problem in traditional worker recommendation mechanisms, in this paper, we firstly develop a novel worker recommendation mechanism by further considering the trust that workers place on requesters.

Background on Matrix Factorization
Due to the data sparsity that each requester have only interacted with a small fraction of the whole workers, and vice versa, therefore, it is a critical problem to inference the trust relationship between unknown requesters and workers. As we know, matrix factorization (MF) [8,12] is one of the most effective recommendation techniques to predict users' unknown preferences from the observed users' preferences, which can be intuitively used to solve the trust evaluation problem in this paper.

Given the RW matrix of dimensions $|M| \times |N|$, the principle of MF based approach is to decompose RW into two low-dimensional matrices: requester-feature matrix $P \in \mathbb{R}^{d \times M}$ and worker-feature matrix $Q \in \mathbb{R}^{d \times N}$, simultaneously. And then, the trust that requester i places on worker j can be predicted by the inner product of requester-specific vector p_i and worker-specific vector q_j, i.e., $\widehat{rw}_{i,j} = p_i^T \cdot q_j$, where $p_i \in \mathbb{R}^d$ and $q_j \in \mathbb{R}^d$ are the i-th and j-th column of P and Q, respectively. By considering requester/worker biases, the basic MF model can be extended to the biasedMF [12]:

$$\widehat{rw}_{i,j} = \mu + b_i + b_j + p_i^T \cdot q_j, \tag{6}$$

where μ is the average trust value in RW, b_i and b_j are the biases of requester i and worker j. We use $\Omega(RW)$ to denote the locations of observed trust rating in RW matrix. The parameters p_i, q_j, b_i and b_j can be learned by minimizing a loss function as follows:

$$\mathcal{L}_{RW} = \frac{1}{2} \sum_{(i,j) \in \Omega(RW)} (\widehat{rw}_{i,j} - rw_{i,j})^2 + \Theta(q_i, p_j, b_i, b_j), \tag{7}$$

where $\Theta(q_i, p_j, b_i, b_j)$ means the regularization term:

$$\Theta(q_i, p_j, b_i, b_j) = \frac{\lambda}{2}(\sum_i ||q_i||_F^2 + \sum_j ||p_j||_F^2 + \sum_i b_i^2 + \sum_j b_j^2).$$

where λ is a parameter to control model complexity and to avoid over-fitting. Moreover, with the consideration of implicit ratings, the biasedMF model can be extended to SVD++ [11] which can well improve the predictive accuracy. However, SVD++ is very costly in computing, to balance the accuracy and the run time, in this paper, biasedMF is adopted to design the worker recommendation system.

Matrix Factorization on Both RW and WR Matrices
By considering both RW and WR matrices, we propose a novel worker recommendation mechanism which evaluates mutual trust relationship between

requesters and workers, and then address the low-acceptance problem of crowd task.

Depending on the number of shared common feature spaces, our proposed worker recommendation mechanism can categorized into three methods:

Method 1: RW and WR share no common feature space.
Method 2: RW and WR share one common feature space. Without loss of generality, RW and WR are assumed to share a common requester-feature space.
Method 3: RW and WR share two common feature spaces.

Method 1. In this case, as shown in Fig. 1, RW and WR share no common feature space, therefore, we can have the following four feature matrices:

$P \in \mathbb{R}^{d \times M}$: the requester feature matrix of the RW matrix.
$Q \in \mathbb{R}^{d \times N}$: the worker feature matrix of the RW matrix.
$U \in \mathbb{R}^{d \times M}$: the requester feature matrix of the WR matrix.
$V \in \mathbb{R}^{d \times N}$: the worker feature matrix of the WR matrix.

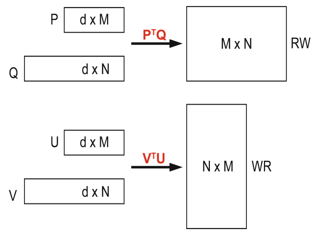

Fig. 1. Method 1.

Similar with the biasedMF model on RW matrix, regarding to WR matrix, the trust that worker k places on requester l can be predicted as:

$$\widehat{wr}_{k,l} = \eta + b_k + b_l + v_k^T \cdot u_l, \tag{8}$$

where η is the average trust value in WR matrix, b_k and b_l are the biases of worker k and requester l. And then, the parameters u_l, v_k, b_l and b_k can be learned by minimizing a loss function as follows:

$$\mathcal{L}_{WR}^{(1)} = \frac{1}{2} \sum_{(k,l) \in \Omega(WR)} (\widehat{wr}_{k,l} - wr_{k,l})^2 + \Theta(u_l, v_k, b_l, b_k). \tag{9}$$

By combining $\mathcal{L}_{RW}$ and $\mathcal{L}_{WR}^{(1)}$, we get an overall loss function for both RW and WR matrices:

$$\mathcal{L}^{(1)} = \mathcal{L}_{RW} + \mathcal{L}_{WR}^{(1)}. \tag{10}$$

In order to obtain a local minimization of the given loss function, we perform the following gradient descents:

$$\begin{aligned}
\frac{\partial \mathcal{L}^{(1)}}{\partial b_i} &= \sum_{j \in R(i)} e_{i,j}^{RW} + \lambda b_i, & \frac{\partial \mathcal{L}^{(1)}}{\partial b_j} &= \sum_{i \in R^+(j)} e_{i,j}^{RW} + \lambda b_j \\
\frac{\partial \mathcal{L}^{(1)}}{\partial b_k} &= \sum_{l \in W(k)} e_{k,l}^{WR} + \lambda b_k, & \frac{\partial \mathcal{L}^{(1)}}{\partial b_l} &= \sum_{k \in W^+(l)} e_{k,l}^{WR} + \lambda b_l \\
\frac{\partial \mathcal{L}^{(1)}}{\partial p_i} &= \sum_{j \in R(i)} e_{i,j}^{RW} q_j + \lambda p_i, & \frac{\partial \mathcal{L}^{(1)}}{\partial q_j} &= \sum_{i \in R^+(j)} e_{i,j}^{RW} p_i + \lambda q_j \\
\frac{\partial \mathcal{L}^{(1)}}{\partial u_l} &= \sum_{k \in W^+(l)} e_{k,l}^{WR} v_k + \lambda u_l, & \frac{\partial \mathcal{L}^{(1)}}{\partial v_k} &= \sum_{l \in W^(k)} e_{k,l}^{WR} u_l + \lambda v_k
\end{aligned} \tag{11}$$

Regarding to RW matrix, $R(i)$ denotes the set of workers that requester i has rated, and $R^+(j)$ denotes the set of requesters who have rated worker j. Regarding to WR matrix, $W(k)$ denotes the set of requesters that worker k has rated, and $W^+(l)$ denotes the set of workers who have rated requester l. We use $e_{i,j}^{RW} = \widehat{rw}_{i,j} - rw_{i,j}$ and $e_{k,l}^{WR} = \widehat{wr}_{k,l} - wr_{k,l}$ to denote the trust prediction error from requester i to worker j and from worker k to requester l, respectively.

Method 2. In this case, as shown in Fig. 2, RW and WR share a common requester-feature space, therefore, we can have the following three feature matrices:

$P \in \mathbb{R}^{d \times M}$: the requester feature matrix of the RW and WR matrices.
$Q \in \mathbb{R}^{d \times N}$: the worker feature matrix of the RW matrix.
$V \in \mathbb{R}^{d \times N}$: the worker feature matrix of the WR matrix.

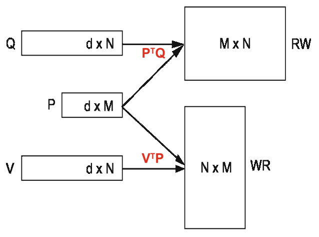

Fig. 2. Method 2.

Subsequently, the trust that worker k places on requester i can be predicted as:

$$\widehat{wr}_{k,i} = \eta + b_k + b_i + v_k^T \cdot p_i. \tag{12}$$

By minimizing a loss function as follows, the parameters p_i, v_k, b_i and b_k can be learned:

$$\mathcal{L}^{(2)}_{WR} = \frac{1}{2} \sum_{(k,i)\in\Omega(WR)} (\widehat{wr}_{k,i} - wr_{k,i})^2 + \Theta(p_i, v_k, b_i, b_k). \tag{13}$$

By considering both $\mathcal{L}_{RW}$ and $\mathcal{L}^{(2)}_{WR}$, the overall loss function can be formulated as:

$$\mathcal{L}^{(2)} = \mathcal{L}_{RW} + \mathcal{L}^{(2)}_{WR}. \tag{14}$$

Similarly, gradient descents can be used to achieve the minimization of the given loss function:

$$\begin{aligned}
&\frac{\partial \mathcal{L}^{(2)}}{\partial b_i} = \sum_{j\in R(i)} e^{RW}_{i,j} + \sum_{k\in W^+(i)} e^{WR}_{k,i} + \lambda b_i \\
&\frac{\partial \mathcal{L}^{(2)}}{\partial b_j} = \sum_{i\in R^+(j)} e^{RW}_{i,j} + \lambda b_j, \qquad \frac{\partial \mathcal{L}^{(2)}}{\partial b_k} = \sum_{l\in W(k)} e^{WR}_{k,l} + \lambda b_k \\
&\frac{\partial \mathcal{L}^{(2)}}{\partial p_i} = \sum_{j\in R(i)} e^{RW}_{i,j} q_j + \sum_{k\in W^+(i)} e^{WR}_{k,i} v_k + \lambda p_i \\
&\frac{\partial \mathcal{L}^{(2)}}{\partial q_j} = \sum_{i\in R^+(j)} e^{RW}_{i,j} p_i + \lambda q_j, \qquad \frac{\partial \mathcal{L}^{(2)}}{\partial v_k} = \sum_{i\in W(k)} e^{WR}_{k,i} p_i + \lambda v_k
\end{aligned} \tag{15}$$

Method 3. In this case, as shown in Fig. 3, RW and WR share two common spaces, therefore, we can have the following two feature matrices:

$P \in \mathbb{R}^{d\times M}$: the requester feature matrix of the RW and WR matrices.
$Q \in \mathbb{R}^{d\times N}$: the worker feature matrix of the RW and WR matrices.

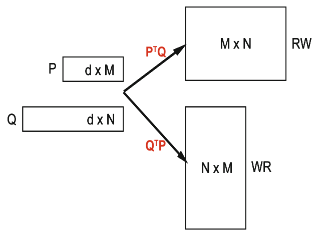

Fig. 3. Method 3.

By using the biasedMF model, regarding to WR matrix, the trust that worker j places on requester i can be predicted as:

$$\widehat{wr}_{j,i} = \eta + b_j + b_i + q_j^T \cdot p_i. \tag{16}$$

The parameters b_i, b_j, p_i and q_j can be learned by minimizing the following loss function:

$$\mathcal{L}_{WR}^{(3)} = \frac{1}{2} \sum_{(j,i)\in\Omega(WR)} (\widehat{wr}_{j,i} - wr_{j,i})^2 + \Theta(p_i, q_j, b_i, b_j). \tag{17}$$

With the consideration of both $\mathcal{L}_{RW}$ and $\mathcal{L}_{WR}^{(3)}$, we get an overall loss function as:

$$\mathcal{L}^{(3)} = \mathcal{L}_{RW} + \mathcal{L}_{WR}^{(3)}. \tag{18}$$

By using gradient descents, we can obtain a local minimization of the given loss function:

$$\begin{aligned}
\frac{\partial\mathcal{L}^{(3)}}{\partial b_i} &= \sum_{j\in R(i)} e_{i,j}^{RW} + \sum_{j\in W^+(i)} e_{j,i}^{WR} + \lambda b_i \\
\frac{\partial\mathcal{L}^{(3)}}{\partial b_j} &= \sum_{i\in R^+(j)} e_{i,j}^{RW} + \sum_{i\in W(j)} e_{j,i}^{WR} + \lambda b_j \\
\frac{\partial\mathcal{L}^{(3)}}{\partial p_i} &= \sum_{j\in R(i)} e_{i,j}^{RW} q_j + \sum_{j\in W^+(i)} e_{j,i}^{WR} q_j + \lambda p_i \\
\frac{\partial\mathcal{L}^{(3)}}{\partial q_j} &= \sum_{i\in R^+(j)} e_{i,j}^{RW} p_i + \sum_{i\in W(j)} e_{j,i}^{WR} p_i + \lambda q_j
\end{aligned} \tag{19}$$

By using the above three methods, we can predict the trust relationship between requesters and workers, overall, the trust that requester i places on worker j can be described as:

$$rw_{i\to j} = \begin{cases} rw_{i,j} & \textit{if } i \textit{ has a direct trust with } j \\ \widehat{rw}_{i,j} & \textit{otherwise.} \end{cases} \tag{20}$$

And the trust that the worker j places on requester i can be described as:

$$wr_{j\to i} = \begin{cases} wr_{j,i} & \textit{if } j \textit{ has a direct trust with } i \\ \widehat{wr}_{j,i} & \textit{otherwise.} \end{cases} \tag{21}$$

4.3 Top-K Recommendation

Regarding to a certain task published by requester i, we can compute a recommendation score for any participant worker j by combining task matching and trust evaluation as:

$$rs_j = M_j \cdot rw_{i\to j} \cdot wr_{j\to i}. \tag{22}$$

An then, the top-K workers with the highest scores will be recommended to requester i. In order to address the cold start problem, with fixed 10% probability, newcomer workers will be recommended to participated in crowd tasks.

5 Performance Evaluation

In this section, we have implemented simulations and real data experiments to illustrate the performance of the proposed top-K worker recommendation mechanism. Our codes have been built on *Surprise*[5] which is a Python scikit for building and analyzing recommender systems.

5.1 Experimental Settings

Trust Evaluation. Epinions[6] is a well-known publicly data set to illustrate who-trust-whom, which contains: 131828 users and 841372 edges. Each edge has two values: 5 means trust and 1 means distrust. From the original Epinions data set, we randomly extract a subgraph which contains 2000 different users, and 1000 users of them are regarded as requesters, while the rest users are regarded as workers. By only considering weighted edges between requesters and workers, RW and WR can be respectively built. We use 5-fold cross-validation for learning and testing. And each measurement is averaged over 10 instances.

Top-K Worker Recommendation. To conduct our experiments on the proposed top-K worker recommendation mechanism, the task features and worker features are indispensably needed. However, such data sets have not been presented to the public by any existing crowdsourcing platform, as a result, we generate synthetic crowd transactions to evaluate the performance of our mechanisms in terms of acceptance rate.

Table 1. Parameters setting

Parameter	Commentate	Value
M	# of requesters	1000
N	# of workers	1000
k	# of the skills	10
T_n	# of required workers for a certain task	20
α_i	Regulate the value of task matching	1
β_i	Regulate the value of task matching	-1.38
w_i	Weight of task matching	0.25

The base settings that apply for simulations are summarized in Table 1. Considering that there should be a high matching degree (e.g., 80%) if the task feature is exactly matched with the worker feature, hence, α_i and β_i are fixed to be 1 and -1.38 as shown in Table 1. In our simulations, time tolerance $T_{tolerance}$ for

[5] https://pypi.org/project/scikit-surprise/1.0.3/.
[6] https://snap.stanford.edu/data/soc-sign-epinions.html.

a certain task follows an uniform distribution $U(0, 1)$. Same with other studies [19,29], normal distributions are adopted to realize the settings of the rest task and worker features, e.g., T_{type} and W_{skill}.

The crowdsourcing system is modeled as a discrete time slot system. At the start of each time slot, the features of workers and requesters will be re-initialized. And within one time slot, we assume that each requester will publish a crowd task with a certain probability, e.g., 10%. And the result is averaged over 100 time slots.

5.2 Experimental Results

Results in Trust Evaluation

Baseline Methods. Regarding to trust evaluation, we compare the performance of our proposed mechanisms with the following approaches.

- UserKnn: An user based collaborative filtering method which implements the prediction based on the preferences of similar users.
- ItemKnn: An item based collaborative filtering method which implements the prediction based on item rating similarity.
- Non-biased MF: A basic matrix factorization method.
- Biased MF: An extension of non-biased MF by considering users' biases.

Metrics. Mean absolute error (MAE) and root-mean-square error (RMSE) have been used as the evaluation metrics.

Results. Firstly, we take Method 1 as the example to study the impact of λ on predictive accuracy. As shown in Fig. 4, we tune the parameter λ in the range $\{0.01, 0.02, 0.04, 0.08, 0.16\}$ while fixing $d = 10$, and it is observed that Method 1 achieves the best predictive accuracy with RMSE $= 0.875$ when $\lambda = 0.02$ is adopted. The result clearly illustrates that a proper value of λ can improve the performance on predictive accuracy. Similar results can also be obtained with different settings of d or different methods.

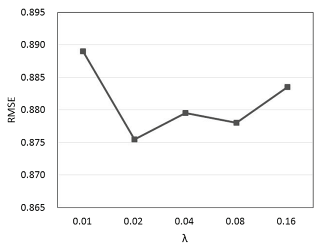

Fig. 4. The impact of λ on predictive accuracy [$d = 10$].

Then, we study the impact of d on predictive accuracy. The result is presented in Fig. 5 in terms of RMSE when Method 1 has been adopted. We tune the parameter d in the range $\{5, 10, 15, 20, 25\}$ while fixing $\lambda = 0.02$. It is indicated that the optimal setting of d is 10, under which case Method 1 achieves the best performance. Moreover, the other settings of λ or methods have similar performance trends.

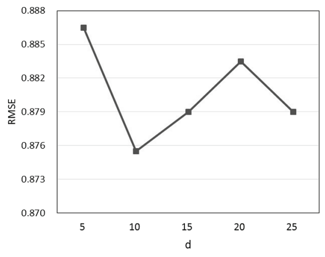

Fig. 5. The impact of d on predictive accuracy [$\lambda = 0.02$].

Finally, we report the results of our proposed three methods compared to the baselines. The results are obtained based on: $\lambda = 0.02$ and $d = 10$. As shown in Table 2, it is illustrated that our algorithms (Method 1, Method 2, and Method 3) obviously outperform the baselines in terms of RMSE and MAE. Specifically, it is interesting to find that (1) Method 1 achieves the best performance among the three methods, and (2) Method 3 performs better than Method 2. A reasonable explanation for the first finding is that: Method 1 has four explicit feature spaces which can efficiently recover the observed RW and WR matrices. The possible reason for the second finding could be that: Method 3 which simultaneously and unbiasedly consider the RW and WR matrices would has a better prediction than Method 2 which biasedly learns the common requester-feature space.

Table 2. Performance comparison in trust evaluation

	RMSE	MAE
Method 1	0.875	0.475
Method 2	0.975	0.531
Method 3	0.939	0.522
UserKnn	1.092	0.577
ItemKnn	1.082	0.582
Non-biased MF	1.496	1.051
Biased MF	1.047	0.559

Results in Top-K Worker Recommendation

Baseline Methods. The top-K worker recommendation mechanism is a general framework, in which the trust evaluation part can adopt UserKnn, ItemKnn, Non-biased MF and Biased MF, respectively, as the baseline methods.

Metric. Considering the scenario that a worker j has been recommended to a requester i. A valid recommendation instance can be defined as: (1) i accepts to employ j as the participant, meanwhile, (2) j accepts to take i as the employee. And then, the acceptance rate (AR) of a worker recommendation method can be calculated as:

$$AR = \frac{\#\ of\ valid\ recommendation\ instances}{\#\ of\ total\ recommendation\ instances}.$$

In this paper, we adopt the AR as the evaluation metric.

Results. A valid recommendation instance is determined by two factors in our crowdsourcing model: task matching and mutual trust evaluation. An overall matching degree which is derived from Eq. (5) is used as the criterion to measure whether a certain worker is suitable for a specifical task. In experiments, we use the overall matching degree to represent the probability that a certain worker is successfully matched with a published task in respect of task matching. Further, to evaluate the mutual trust between requesters and workers, we introduce a threshold value θ to measure the condition of trust establishment among users: (A) Requester i will trust worker j if $\widehat{rw}_{i,j} \geq 5 - \theta$; (B) Conversely, worker j will trust requester i if $\widehat{wr}_{j,i} \geq 5 - \theta$. In experiment, the mutual trust between requester i and worker j is assumed to be established if condition (A) and condition (B) are both satisfied. Table 3 shows the result of performance comparison in terms of AR, where we tune the parameter θ in the range of $\{4, 2, 1, 0.5, 0.25\}$. It is observed that: (1) AR would gradually decrease with the decreasing value of θ because valid recommendation instances are more difficult to happen if the condition of trust establishment becomes more critical; (2) Due to the better performances in trust evaluation, our proposed algorithms (Method 1, Method 2 and Method 3) evidently have higher values of AR than the baselines especially for the smaller value of θ.

Which part contribute more? In the proposed top-K worker recommendation framework, two parts have been included: task matching and mutual trust evaluation. Definitely, it is necessary to explore which part contributes more on the overall performance. In the following, we introduce:

- OTM: A top-K worker recommendation framework that **O**nly consists the **T**ask **M**atching part.
- OTE: A top-K worker recommendation framework that **O**nly consists the mutual **T**rust **E**valuation part.

The results are presented in Fig. 6 while fixing $\theta = 1$. Obviously, the worker recommendation mechanisms that consider both task matching and trust evaluation significantly perform better than the mentioned compared methods. And,

Table 3. Performance comparison in term of acceptance rate

	$\theta = 4$	$\theta = 2$	$\theta = 1$	$\theta = 0.5$	$\theta = 0.25$
Method 1	0.915	0.844	0.801	0.721	0.547
Method 2	0.915	0.843	0.796	0.696	0.492
Method 3	0.915	0.843	0.794	0.701	0.500
UserKnn	0.915	0.818	0.733	0.603	0.258
ItemKnn	0.915	0.820	0.739	0.615	0.302
Non-biased MF	0.915	0.754	0.589	0.395	0.250
Biased MF	0.915	0.829	0.745	0.624	0.436

OTM with $AR = 0.569$ performs better that OTE with $AR = 0.185$, which illustrates that task matching part plays an more important role than trust evaluation part in the design of worker recommendation mechanisms. Similar results will also be obtained with the different settings of θ.

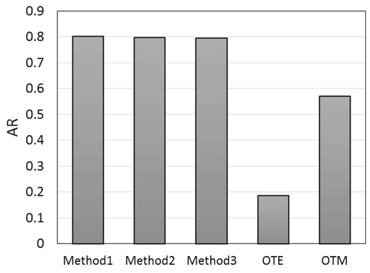

Fig. 6. Performance comparison with OTM and OTE [$\theta = 1$].

6 Conclusion

In this paper, based on biased matrix factorization, we specifically and firstly consider the mutual trust between requesters and workers which can address the low-acceptance problem in traditional worker recommendation mechanisms. Further, a comprehensive task matching mechanism has been proposed by incorporating time matching, skill matching, payment matching, and location matching, no other previous work has considered in all aspects. Extensive simulations and experiments over a large public dataset demonstrate that our proposed worker recommendation methods outperform all baselines in terms of trust evaluation and acceptance rates.

Limitations and Future Work

This paper has the following fundamental limitations.

(i) Requesters and workers considered in this paper are assumed to honestly feedback their opinions, however, which is not true in real-life crowdsensing systems. In our future work, we will further study the reliability of ratings provided by the requesters and workers.

(ii) The fundament of our proposed top-K worker recommendation mechanism is to calculate an overall recommendation score, however, it is difficult to determine the weight of each feature in practice. In our future work, we will use multi-objective optimization to extend this work.

Acknowledgements. The paper is supported by National Nature Science foundation of China under grant Nos.: 61572095 and 61877007.

References

1. Abhinav, K., Dubey, A., Jain, S., Virdi, G., Kass, A., Mehta, M.: CrowdAdvisor: a framework for freelancer assessment in online marketplace. In: ICSE, pp. 93–102. IEEE (2017)
2. Alsayasneh, M., et al.: Personalized and diverse task composition in crowdsourcing. TKDE **30**(1), 128–141 (2018)
3. Difallah, D.E., Demartini, G., Cudré-Mauroux, P.: Pick-a-crowd: tell me what you like, and I'll tell you what to do. In: WWW, pp. 367–374. ACM (2013)
4. Farkas, K., Nagy, A.Z., Tomás, T., Szabó, R.: Participatory sensing based real-time public transport information service. In: PERCOM Workshops, pp. 141–144. IEEE (2014). https://doi.org/10.1109/PerComW.2014.6815181
5. Gaikwad, S.N.S., et al.: Boomerang: rebounding the consequences of reputation feedback on crowdsourcing platforms. In: UIST, pp. 625–637. ACM (2016)
6. Goel, G., Nikzad, A., Singla, A.: Matching workers expertise with tasks: incentives in heterogeneous crowdsourcing markets. In: NIPS (2013)
7. ul Hassan, U., Curry, E.: Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning. Expert Syst. Appl. **58**, 36–56 (2016)
8. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: SIGIR, pp. 549–558. ACM (2016)
9. Jabeur, N., Karam, R., Melchiori, M., Renso, C.: A comprehensive reputation assessment framework for volunteered geographic information in crowdsensing applications. Ubiquit. Comput. 1–17 (2018)
10. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: WWW, pp. 640–651. ACM (2003)
11. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. TKDD **4**(1), 1–24 (2010)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Kurve, A., Miller, D.J., Kesidis, G.: Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. TKDE **27**(3), 794–809 (2015)

14. Liu, H., Zhang, X., Zhang, X.: Possible world based consistency learning model for clustering and classifying uncertain data. Neural Netw. **102**, 48–66 (2018)
15. Liu, H., Zhang, X., Zhang, X., Cui, Y.: Self-adapted mixture distance measure for clustering uncertain data. Knowl.-Based Syst. **126**, 33–47 (2017)
16. Lu, K., Wang, J., Li, M.: An Eigentrust dynamic evolutionary model in P2P file-sharing systems. Peer Peer Netw. Appl. **9**(3), 599–612 (2016)
17. Pu, L., Chen, X., Xu, J., Fu, X.: Crowdlet: optimal worker recruitment for self-organized mobile crowdsourcing. In: INFOCOM, pp. 1–9. IEEE (2016)
18. Qiao, L., Tang, F., Liu, J.: Feedback based high-quality task assignment in collaborative crowdsourcing. In: AINA, pp. 1139–1146. IEEE (2018)
19. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
20. Safran, M., Che, D.: Real-time recommendation algorithms for crowdsourcing systems. Appl. Comput. Inform. **13**(1), 47–56 (2017)
21. Schall, D., Satzger, B., Psaier, H.: Crowdsourcing tasks to social networks in BPEL4People. World Wide Web **17**(1), 1–32 (2014)
22. Schnitzer, S., Neitzel, S., Schmidt, S., Rensing, C.: Perceived task similarities for task recommendation in crowdsourcing systems. In: WWW, pp. 585–590 (2016)
23. Song, T., et al.: Trichromatic online matching in real-time spatial crowdsourcing. In: ICDE, pp. 1009–1020. IEEE (2017)
24. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.: Online mobile micro-task allocation in spatial crowdsourcing. In: ICDE, pp. 49–60. IEEE (2016)
25. Wang, J., Wang, F., Wang, Y., Zhang, D., Lim, B.Y., Wang, L.: Allocating heterogeneous tasks in participatory sensing with diverse participant-side factors
26. Wang, L., Yu, Z., Han, Q., Guo, B., Xiong, H.: Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks. TMC **17**(17), 1637–1650 (2018)
27. Wang, Y., Tong, X., He, Z., Gao, Y., Wang, K.: A task recommendation model for mobile crowdsourcing systems based on dwell-time. In: BDCloud-SocialCom-SustainCom, pp. 170–177. IEEE (2016)
28. Wu, C., Luo, T., Wu, F., Chen, G.: EndorTrust: an endorsement-based reputation system for trustworthy and heterogeneous crowdsourcing. In: GLOBECOM, pp. 1–6. IEEE (2015)
29. Xiang, Q., Zhang, J., Nevat, I., Zhang, P.: A trust-based mixture of Gaussian processes model for reliable regression in participatory sensing. In: IJCAI, pp. 3866–3872 (2017)
30. Ye, B., Wang, Y.: CrowdRec: trust-aware worker recommendation in crowdsourcing environments. In: ICWS, pp. 1–8. IEEE (2016)
31. Ye, B., Wang, Y., Liu, L.: Crowd trust: a context-aware trust model for worker selection in crowdsourcing environments. In: ICWS, pp. 121–128. IEEE (2015)
32. Yu, H., Shen, Z., Miao, C., An, B.: A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems. In: AAMAS, pp. 1315–1316 (2013)
33. Yuan, F., Gao, X., Lindqvist, J.: How busy are you?: Predicting the interruptibility intensity of mobile users. In: CHI, pp. 5346–5360. ACM (2017)
34. Yuen, M.C., King, I., Leung, K.S.: TaskRec: a task recommendation framework in crowdsourcing systems. NPL **41**(2), 223–238 (2015)

35. Zhang, X., Liu, H., Zhang, X.: Novel density-based and hierarchical density-based clustering algorithms for uncertain data. Neural Netw. **93**, 240–255 (2017)
36. Zhang, X., Liu, H., Zhang, X., Liu, X.: Novel density-based clustering algorithms for uncertain data. In: AAAI, pp. 2191–2197 (2014)
37. Zhang, X., Xue, G., Yu, R., Yang, D., Tang, J.: Truthful incentive mechanisms for crowdsourcing. In: INFOCOM, pp. 2830–2838. IEEE (2015)