Reputation Management in Crowdsourcing Systems

Mohammad Allahbakhsh*, Aleksandar Ignjatovic*, Boualem Benatallah*
Seyed-Mehdi-Reza Beheshti*, Elisa Bertino†, and Norman Foo*

*The University of New South Wales, Sydney, NSW 2052, Australia
{mallahbakhsh, ignjat, boualem, sbeheshti, norman}@cse.unsw.edu.au

†Purdue University, West Lafayette, Indiana, USA
bertino@cs.purdue.edu

***Abstract*—Worker selection is a significant and challenging issue in crowdsourcing systems. Such selection is usually based on an assessment of the reputation of the individual workers participating in such systems. However, assessing the credibility and adequacy of such calculated reputation is a real challenge. In this paper, we propose a reputation management model which leverages the values of the tasks completed, the credibility of the evaluators of the results of the tasks and time of evaluation of the results of these tasks in order to calculate more dependable quality metrics for workers and evaluators. The model has been implemented and experimentally validated.**

***Index Terms*—Reputation, Degree of Fairness, Crowdsourcing**

I. Introduction

Crowdsourcing involves receiving, incorporating and consolidating contributions from a large crowd with varied levels of expertise [6]. The people who own crowdsourcing task are called *requesters* and the people who do the tasks are called *workers*. Due to lack of enough information, lack of expertise, dishonesty of workers or evaluators, bias in user interests and many more reasons [8], [1], quality of contributions in crowdsourcing tasks is always under question. The overall quality of the outcome of a crowdsourced task depends on the quality of the workers, the processes which govern the task creation, selection of workers, coordination of sub-tasks including reviewing intermediary outcomes, aggregation of individual contributions, etc.

Using reputation as an indicator of community-wide judgment on workers' trustworthiness is a a popular method for evaluation of the quality of workers in existing crowdsourcing platforms [2], [12], [7], [5]. Various information sources are used for calculating reputation such as feedbacks received from community members [5], [7], artifacts generated by workers [2], and task properties like credit paid for the task and the time in which the task has been done [7]. Moreover, a wide variety of approaches are used for reputation calculation such as deterministic approaches, fuzzy techniques, belief mode, bayesian approaches, etc [12].

Regardless of the process of calculating reputation scores and the information items included in this process, credibility of the calculated reputations is a real challenge in crowdsourcing systems. Dishonest evaluators may try to manipulate reputation of workers in various ways [16]. They may cast random evaluations on workers' contributions regardless of the quality. More complicated, they may behave well in some time intervals to build up a good reputation and then treat workers badly in some other time intervals alternatively to hide from being detected. Also, they may promote themselves by treating well a large number of workers and then attack a few numbers without being identified as a dishonest evaluator. Malicious manipulation of reputation can lead to inadequate worker selection which directly affects the quality of the obtained contributions. Also, such manipulation can harm community members, leaving them vulnerable to deceptive evaluators. For example, in online markets like Amazon Mechanical Turk loosing reputation results in decreasing chance of getting further jobs, possibly causing unfair loss of income. In both existing crowdsourcing platforms, as well as in research prototypes, this issue is not fully addressed [17].

To address this issue, we propose a reputation management model which adequately takes into account the trustworthiness of the evaluators, the time of evaluation and the credit paid for tasks. We analyze behavior of evaluators in smaller time intervals to identify alternatively changing behaviors. We also check the pairwise relations between evaluators and workers to detect self promoting people who just mistreat a small number of workers. We also use majority consensus to detect outliers to decrease their impact on the reputation of the workers. Unfair evaluations are broadly divided into two categories [17]: (i) *individual* and (ii) *collaborative (collusion)*. Our model identifies individual unfair evaluations and even some straightforward collaborative attacks effectively. But it is still vulnerable to some kinds of more complicated collaborative attacks [16].

The unique contributions of the paper are as follows:

1) We propose a new metric called *degree of fairness* to show how fair evaluators have been, when evaluating contributions of workers. We use majority consensus on the trustworthiness of the workers as an indicator to show how close the evaluator's opinion is to community consensus.
2) We propose a graph data model for better understanding, representing and analyzing worker evaluation in

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250499

crowdsourcing systems. This model allows representing evaluators, workers, evaluations cast on the quality of the workers, pairwise trust and degree of fairness.

3) We propose an algorithm for computing reputation ranks for workers. The algorithm uses pairwise trust and degree of fairness ranks for building a reputation rank for every worker.
4) Our experimental results confirm that our model is robust against unfair or inaccurate evaluations, to an extent surpassing two most commonly used existing methods (eBay and PageRank).

The remainder of the paper is organized as follows. In section (II) we study related work. In section (III) we formulate the problem. In section (IV) our graph data mode is proposed. We calculate local quality metrics in section (V) and pairwise quality metrics in section (VI). In section (VII) we show how we calculate community-wide reputation of workers. In section (VIII) we show evaluation results and we conclude in section (IX).

II. Related Work

There are two categories of studies related to our research.

Crowd enhanced platforms: The Amazon Mechanical Turk[1] is a general-purpose online marketplace suitable for doing simple crowdsourcing tasks called Human Intelligence Task (HIT). There is no any metric called reputation in MTurk but there are some other metrics showing the trustworthiness of the workers like ratio of submitted HITs which have been accepted . There are no mechanisms in MTurk for detecting unfair evaluations, hence workers are highly vulnerable against misbehavior.

eBay[2] is another crowd enhanced system in which people sell and buy goods. People evaluate each other when they involve in transactions and based on these feedback, a reputation is built for the person as a seller or a buyer or both. As we show here, eBay reputation model is also vulnerable to unfair evaluations.

StackOverflow[3] is a question answering web site. Users in StackOverflow can ask questions, answer to questions asked by others and vote on the quality of the questions or answers. Regarding received votes, a reputation score is calculated for every member. There are no means for identifying unfair evaluations in StackOverflow and users can easily manipulate reputations calculated for workers.

Research Tools and Prototypes: Noor and Sheng [11] have proposed a trust management framework for cloud environments but it is very similar to reputation concept in crowdsourcing era and their idea is close to our work. They propose a credibility model for identifying unfair evaluations by using the concept of majority. They calculate an experience degree for every consumer evaluating services and apply it to aggregation of his votes to eliminate votes form dishonest evaluators. The problem is that people sometimes are fair with most of the people while they are unfair just with a few number of people. In this case the unfairness of the evaluator will not be detected due to large number of fair votes. The other problem with Noor et.al model is that time and credit are not considered in calculation of trust and also experience of the customer.

PageRank [13] is one of the most popular reputation management algorithms which employs the reputation of evaluators in calculating the reputation of workers. The votes given by highly reputable people are more important than votes of low reputable evaluators in PageRank model. This model is used by Google to rank web pages in the internet. As we show in this work, PageRank does not employ any means for identifying unfair evaluations and is weak against unfair evaluations. It also does not consider time in the reputation calculations.

EigenTrust [9] is a popular trust model which is built based on the PageRank algorithm and tries to solve the problem of unfair evaluations. EigenTrust supposes that there are some pre-trusted users in the system in which we can trust and it is evident that this assumption is not applicable to most of the existing crowdsourcing systems like question answering systems or online marketplaces. Also, It has been shown that EigenTrust is not robust against unfair evaluations [15].

III. Problem Formulation

Overview. Let us assume that in a crowdsourcing system N_W workers denoted by $W = \{w_j : 0 \leq j \leq N_W\}$ contribute to tasks. We also assume that N_R members (either workers or requesters) have evaluated at least one contribution in the system. We call this group, *Evaluators* and denote them by $R = \{r_i : 0 \leq i \leq N_R\}$. Suppose that contribution of worker w_j at time stamp k has been evaluated by evaluator r_i. This evaluation is denoted by $e_{ij}(k)$. These evaluations will be taken into account for determining reputation of w_j and fairness of r_i. We assume that $e_{ij}(k)$ is a real number in a fixed range $[1, M]$, $(M > 1)$. M is a system dependant constant and is different in various systems. e.g., it is 5 in Amazon online market[4]. $e_{ij}(k) = M$ means full trust and $e_{ij}(k) = 1$ means distrust.

The capabilities and trustworthiness of the workers or evaluators may change in time, so we consider time as a factor in calculations. We believe that recent feedbacks should have bigger impacts in reputation ranks than older ones. To apply time, we divide life time of the system which is in fact the time distance between the first and last evaluations cast in the system, to equal time intervals and analyze the behavior of workers in those intervals independently. The size of the time interval is dependent to the nature of the system, arrival rate of the evaluations and formulations used for calculation of quality metrics.

Notations. In the following we list the notations to be used in the subsequent sections.

[1] http://www.mturk.com

[2] http://www.ebay.com

[3] http://www.stackoverflow.com

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

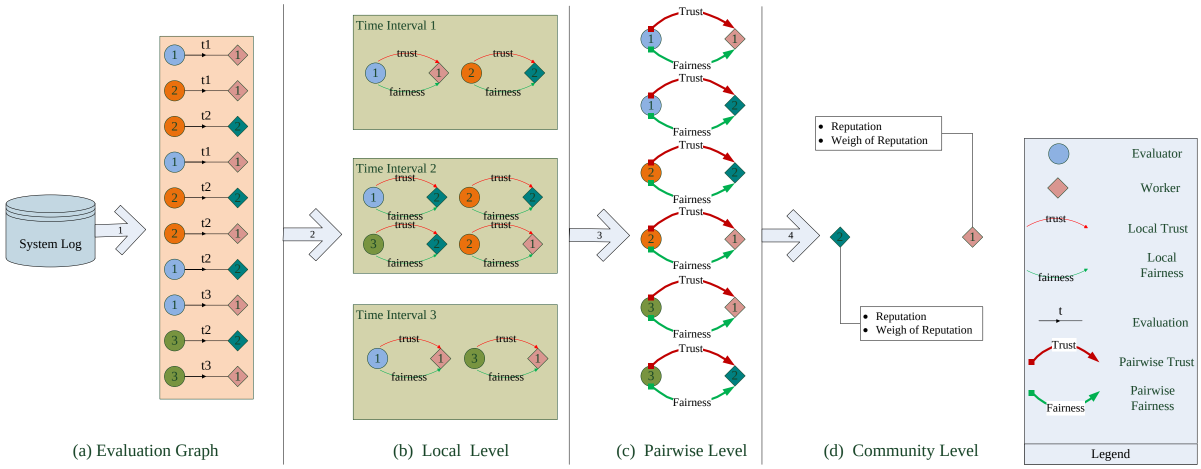


Fig. 1. The process of calculating reputations.

- $R = \{r_i\}$ is the set of all evaluators. We also denote set of evaluators who evaluated worker w_j in time interval t by $R(j, t)$. The R_j denotes set of all evaluators who have evaluated contributions of w_j at least once.
- $W = \{w_j\}$ is the set of all workers. We also denote set of workers whom have been evaluated by evaluator r_i in time interval t by $W(i, t)$. Moreover, $W(i)$ is the list of workers whose contributions have been evaluated by r_i at least once.
- $INT = \{t_n\}$ is the set of all time intervals. Every t_n has a starting time and ending time. The time intervals are disjoint and have no intersections. Also, every time interval has an index which is the number of time intervals i.e., we order time intervals based on their starting and ending times and assigns every time interval an index starts from 1 and increases by 1 for consequent time intervals. We denote index of time interval t_n by I_{t_n}.
- $D_j(t)$ is the set of the time instances (evaluation time stamps) in which the worker w_j has been evaluated in time interval t. Also, we denote set of time instances in which the worker w_j has been evaluated by evaluator r_i in time interval t by $D_{ij}(t)$. We suppose that in every time instance, at most one evaluation can happen. So, we use $|D_j(t)|$ as the number of evaluations on w_j in time interval t and $|D_{ij}(t)|$ as the number of evaluations given by r_i on w_j in time interval t.

Example Scenario. Voting is one of the popular crowdsourcing tasks [18]. In this kind of tasks, the opinions of the crowd are collected to help requesters making better decisions. In Wikipedia[5], the voting process is used to elect administrators[6]. Every registered user can nominate herself or another user for being an administrator in Wikipedia and initiate and election. The other users participate in the election and cast their votes on the eligibility of nominee to be an administrator in the Wikipedia. If the majority of the users recognize her eligible, she will become a Wikipedia administrator. In this crowdsourcing task, the requester is the nominator, the worker is the nominee, evaluators are voters, the task is evaluating the eligibility of the nominee for being and administrator in Wikipedia and contribution is the nominee's request.

We use the the log of Wikipedia Adminship Election[7] which is collected by Leskovec et. al for behavior prediction in online social networks [10], referred in the following as WIKILog. WIKILog contains about $2,800$ elections with around $100,000$ total votes and about $7,000$ users participating in the elections either as a voter or a nominee. We will use the WIKILog in the paper to demonstrate how it is possible to use the proposed framework to calculate people reputations in crowdsourcing systems. For example, we will show how it is possible to: (i) calculate local and pairwise trust between evaluators and workers; (ii) calculate the local and pairwise degree of fairness between evaluators and workers; and (iii) calculate reputation of the workers.

IV. Data Model

We model crowdsourcing entities (mainly evaluators and workers) in log of a crowdsourcing system and their relationships as a directed graph $G = (V, E)$ where V is a set of nodes representing entities and E is a set of directed edges representing relationships between nodes. There are two types of entities in the model: **Worker** and **Evaluator**. Every entity is identified by a unique ID. Every worker entity represents a particular worker $w_i \in W$ and every evaluator entity represents an evaluator $r_i \in R$.

Moreover, there are three types of relationships between entities: *Evaluation*, *Trust* and *Fairness*.

Evaluation relationship represents the result of an evaluation action which an evaluator has performed on the quality of the contributions of a worker. Attributes of evaluation

[5] http://www.wikipedia.org/

[6] http://en.wikipedia.org/wiki/Wikipedia:Requests_for_adminship

[7] http://snap.stanford.edu/data/wiki-Elec.html

relationship are: the *score* which is given as the quality assessment result to the worker, the *time* in which the evaluation has happened and the *credit* which has been paid for the corresponding task.

Trust relationship is a pairwise relationship between an evaluator and a worker to show in what extent the evaluator, trusts the worker. The trust relationship can show trust between two entities in a specific time interval or life time of the system. A trust relationship has three main attribute: a *trust value*, the *weight of trust* value, and the *level* of calculated trust which can be *local* (for a time intervals) or *pairwise* (for life time of the system).

Fairness relationship is a pairwise relationship between an evaluator and a worker to show how fair the evaluator has behaved when she has been evaluating contributions of the worker. Similar to trust relationships, the fairness relationship also can show fairness of evaluators either in a particular time interval or life time of the system. A fairness relationship has two main attribute: *fairness value* and the *level* of calculated degree of fairness which can be *local* (for a time intervals) or *pairwise* (for life time of the system).

Following, we describe how we build the evaluation graph, establish necessary relationships between nodes and add some attributes to nodes to calculate reputation scores and assign them to workers. The overall process is illustrated in Figure 1.

Step 1: Preprocessing. The aim of preprocessing the crowdsourcing log is to generate a graph by considering the set of workers and evaluators in the log as nodes of the graph, and evaluation relationships between them encoded as edges between nodes. In order to preprocess crowdsourcing log, we perform the following two steps: (i) we generate graph nodes (i.e., evaluators and workers) by extracting evaluations and their attributes from the log and form the set of graph nodes (vertices), one for each person (but with no relations between nodes); and (ii) we generate evaluation relationships between nodes one for each evaluation action. We use the querying framework proposed in our previous work [4] to analyze the graph and formulate the relationships between any pairs of nodes in the graph (see Figure 1(a)).

Step 2: Local Level Calculations. In the second step, we use the equation proposed in section (V) to build local trust and fairness relations between nodes. The Local Pairwise Trust and its corresponding weight are used as attributes of the local trust relationships and Local Pairwise Fairness as the attribute of the fairness relation (Figure 1(b)). The level attribute of trust and fairness relationships in this step are set to '*local*'.

Step 3: Pairwise Level Calculations. In the third step, we establish pairwise trust and fairness relationships between nodes. This relations are established with '*pairwise*' as their level value. In addition to level attribute, the other attributes of a trust relationship are τ_{ij} as pairwise trust calculated by Equation 8 and its corresponding weight (Equation (9)). Also, the second attribute of pairwise fairness relationship is φ_{ij} calculated as pairwise degree of fairness using Equation (10). See Figure 1(c).

Step 4: Community Wide Calculations. In the last step, we us pairwise trusts and fairness relationships to build a reputation rank for every worker, as illustrated in Figure 1(d). We add reputation rank as a new attribute to worker entities. The reputation rank is supported by a weight as another attribute of the worker to show how dependable is the calculated reputation rank.

V. Local Quality Metrics

In local level, two quality metrics characterize the relation between evaluators and the workers: *Local Pairwise Trust* and *Local Pairwise Degree of Fairness*.

A. Local Pairwise Trust (LPT)

In every time interval, evaluators may evaluate contributions of workers. We build a trust relationship between every pair of requesters and workers in each time interval and call it *Local Pairwise Trust (LPT)*. We define LPT between evaluators and workers in time period t as follows:

Definition 1: We define Local Pairwise Trust, $T_{ij}(t)$, intended to be a measure for showing how much r_i has been trusting w_j in time interval t, as follows:

$$T_{ij}(t) = \frac{\sum_{k \in D_{ij}(t)} e_{ij}(k)}{|D_{ij}(t)|} \tag{1}$$

T_{ij} is the average of all evaluations received from r_i on the contributions of w_j in time interval t. We also calculate a weight for the LPT to show how dependable it is. We use credit e.g, monetary value paid for performing the task to weight the LPT. This is done due to the fact that the trust ranks built on the feedbacks received for high credit tasks are more dependable than ranks built on low credit tasks.

Definition 2: Suppose that $c(i, j, l)$ is the monetary credit paid for a task done by w_j on time instance k and has been evaluated by r_i. Also, assume that function h is a strictly increasing function that defines how $c(i, j, l)$ must be considered in the weight of the local pairwise trust rank. The Weight of Local Pairwise Trust is denoted by $WT_{ij}(t)$ and is calculated as follows:

$$WT_{ij}(t) = \frac{\sum_{k \in D_{ij}(t)} h(c(i, j, l))}{|D_{ij}(t)|} \tag{2}$$

In fact, $WT_{ij}(t)$ is the average of the credits paid for tasks done by w_j and evaluated by r_i in time interval t. in our experiments $h(x) = x$, i.e., $h(c(i, j, k)) = c(i, j, k)$.

B. Local Pairwise Fairness (LPF)

Reputation Management Systems (RMS) must be robust against unfair evaluations. While helping requesters find high quality workers, an RMS must protect workers against unfair evaluators as well. Degree of fairness metrics proposed in this paper address this problems.

Majority consensus has been widely used as a measure for finding outliers and dishonest evaluators [11], [16]. In other words, majority of evaluators provide a realistic and dependable evaluation of the performance of a worker. So, we use the majority consensus as a measure for checking the credibility of evaluations provided by evaluators on the contributions of workers.

We calculate a *Local Pairwise Fairness (LPF)* between every pair of related evaluators and workers in each time interval. For example if an evaluator has assessed contributions of n workers in a particular time interval, we will create n local degree of fairness relations one for each evaluated worker. LPF between r_i and w_j in time interval t is denoted by $F_{ij}(t)$. LPF shows how credible the local trust rank calculated between an evaluator and a worker is; i.e., we use $F_{ij}(t)$ to show how dependable the trust feedbacks that r_i has given to w_j are.

We calculate LPF in four steps. At first, we calculate the average of all evaluations given to a particular worker, say w_j in time interval t using Equation (3).

$$\overline{e_j(t)} = \frac{\sum_{(l \in R(j,t) \wedge k \in D_j(t))} e_{lj}(k)}{|D_j|} \tag{3}$$

In the second step, we calculate the average of all evaluations given to w_j by r_i in time interval t using Equation (4).

$$\overline{e_{ij}(t)} = \frac{1}{|D_{ij}(t)|} \sum_{k \in D_{ij}} e_{ij}(k) \tag{4}$$

In the third step, we calculate the average distance of all evaluations given to a worker from $\overline{e_j(t)}$ as we show in Equation (5).

$$AD_j = \sqrt{\frac{\sum_{i \in R(j,t)} (\overline{e_{ij}} - \overline{e_j})^2}{|D_j(t)|}} \tag{5}$$

Finally, we define and calculate local pairwise fairness of relations between evaluators and workers regarding the Equations (3), (4) and (5).

Definition 3: Suppose that the r_i has assessed contributions of w_j in the time interval t. The *Local Pairwise Fairness* between r_i and w_j shows how fairly r_i has evaluated the contributions of w_j in time interval t and is denoted by $F_{ij}(t)$. The $F_{ij}(t)$ is calculated as follow:

$$F_{ij}(t) = \begin{cases} \frac{\overline{e_j} - AD_j - \overline{e_{ij}}}{M} & \text{if } \overline{e_{ij}} < (\overline{e_j} - AD_j) \\ 1 & \text{if } (\overline{e_j} - AD_j) \leq \overline{e_{ij}} \leq (\overline{e_j} + AD_j) \\ \frac{\overline{e_{ij}} - (\overline{e_j} + AD_j)}{M} & \text{if } (\overline{e_j} + AD_j) < \overline{e_{ij}} \end{cases} \tag{6}$$

According to Equation (6), the LPFs fall in $\overline{e_j} \pm AD_j$ are considered as being trustworthy and dependable but the averages that fall out of that range are considered as low credible and their impact on trustworthiness of the worker are decreased dramatically. The $F_{ij}(t)$ shows how close to the majority consensus the judgment of r_i about w_j's trustworthiness in time interval t is. We use LPF to reduce the effect of evaluations generated by outliers.

VI. Pairwise Quality Metrics

We build reputation of workers based on the local and pairwise relations between evaluators and workers (see Figure 1). We have calculated local pairwise quality metrics in section (V). We call them local because they are calculated in a single time interval. In this section we build two global pairwise metrics which are the building blocks of reputation scores. These metrics are *Pairwise Trust* and *Pairwise Degree of Fairness*.

A. Pairwise Trust

Pairwise Trust (PT) is an indicator for showing how an evaluator trusts a particular worker. We use a modified version of the model proposed in our previous work [7] for calculating pairwise trust ranks. Pairwise trust is the aggregation of all local pairwise trust ranks between the evaluator and the worker. In addition to local pairwise trust ranks, we involve index of the time interval of LPT in the calculation of pairwise trusts as well. The more recent the LPT is, the bigger impact it should have in the pairwise trust [7]. To apply time, we define a constant called q, ($q \geq 1$). The value assigned to q determines how fast the importance of an LPT decreases as the time progresses, and is system dependant. Suppose that the 'half life' of the importance of LPTs in a system is θ i.e. the importance of an LPT after θ time intervals decreases to a half of its original value. Because time intervals are relatively short, in comparison with life time of the system, we suppose half life is an integer number greater than or equal to 2 i.e. $\theta \geq 2$. The constant q is then calculated using the Equation (7).

$$q = 2^{1/\theta} \tag{7}$$

In Section (VIII) we will show the impact of different values of *'q'* on the calculated trustworthiness tanks.

Definition 4: We define *Pairwise Trust* rank between an evaluator r_i and a worker w_j to show in what extent r_i trust w_j. Pairwise trust is denoted by τ_{ij} and is calculated as follows:

$$\tau_{ij} = \frac{\sum_{t \in INT} T_{ij}(t) \times q^{I_t}}{\sum_{t \in INT} q^{I_t}} \tag{8}$$

We also calculate a weight for the pairwise trust rank to show how dependable is the calculated trust rank. The weight of τ_{ij} is denoted by ω_{ij} and is calculated using Equation (9).

$$\omega_{ij} = \frac{\sum_{t \in INT} WT_{ij}(t) \times q^{I_t}}{\sum_{t \in INT} q^{I_t}} \tag{9}$$

B. Pairwise Degree of Fairness

Pairwise Degree of Fairness (LPF) shows how fair an evaluator has been while evaluating contributions of a worker. Same as pairwise trust in section (VI-A), we use index of time intervals and local pairwise fairness degrees to calculate the pairwise degree of fairness. LPFs are numbers in range $[0, 1]$. The value 0 for LPF means the evaluator has been completely unfair to the worker and value 1 implies being completely fair. We choose the smallest LPF of an evaluator in relation with each worker as their pairwise degree of fairness to discourage them from being unfair. To increase impact of recent activities, we use constant q which is defined in Equation (7).

Definition 5: We define *Pairwise Degree of Fairness* between an evaluator r_i and a worker w_j to show how fair r_i

Algorithm 1 Worker's Reputation Calculation

```
Input: Set of all pairwise trusts $\tau_{ij}$, Set of the weight of all pairwise trusts $\omega_{ij}$, Set of all workers and Set of all pairwise degrees of fairness.

Output: $P$ as the set of all reputation scores ($\{\rho_j\}$) and $\Omega$ as their corresponding weights ($\{\omega_j\}$)
for all $w \in W$ do
    $r = 0$
    $w = 0$
    $T_w \leftarrow$ All Trust ranks on Worker w ($T_w \subset \{\tau_{ij}\}$)
    $Wt \leftarrow$ Weight of all trust ranks in $Tr_w$ ($Wt \subset \{\omega_{ij}\}$)
    $F \leftarrow$ All pairwise degrees of fairness on w ($F \subset \{\varphi_{ij}\}$)
    for all $\tau \in T_w$ do
        wupdate = $Wt_\tau * F_\tau$
        $r = r + \tau * wupdate$
        $w = w + wupdate$
    end for
    $P[w] = r/w$
    $\Omega[w] = w$
end for
return $P$ and $\Omega$
```

has been in time when she has been evaluating contributions of w_j. Pairwise degree of fairness is denoted by φ_{ij} and is calculated as follows:

$$\varphi_{ij} = \min(F_{ij}(t) \times q^{I_t}), \text{ where } t \in INT \tag{10}$$

VII. Reputation of the Workers

Reputation of a worker is an indicator of community-wide judgment on worker's performance. Therefore, for building reputation of a worker we have to aggregate judgement of all evaluators on the quality of the worker i.e., aggregating all pairwise trusts between evaluators and the worker.

Definition 6: We define *Reputation* of worker w_j denoted by ρ_j as the community wide judgement of trustworthiness of w_j and calculate it as follows:

$$\rho_j = \sum_{l \in R_j} \frac{\omega_{lj} \times \varphi_{lj}}{\sum_{l \in R_j} \omega_{lj}\varphi_{lj}} \tau_{lj} \tag{11}$$

Equation (11) shows that the reputation is an aggregation of the pairwise trust ranks that a worker received from all evaluators, prorated by their corresponding degree of fairness, which is reflected in the value of the corresponding multiplier $\omega_{lj} \times \varphi_{lj} / \sum_{l \in R_j} \omega_{lj} \times \varphi_{lj}$. Such multiplier takes into account the weight of the trust rank ω_{ij} and the corresponding degree of fairness φ_{ij}. The denominator $\sum_{l \in R_j} \omega_{lj} \times \varphi_{lj}$ re-normalizes the sum, making ρ_j a weighted average of individual trust ranks. This method of calculating trust ranks reflects our intuition that different evaluators have different credibility levels which should reflect their overall behavior in the system.

Also, our model distinguishes between the reputation scores that are built based on high number of evaluations received from fair evaluators and reputation scores built based on few number of evaluations received from unfair evaluators. This is possible by providing a corresponding weight for every calculated reputation score.

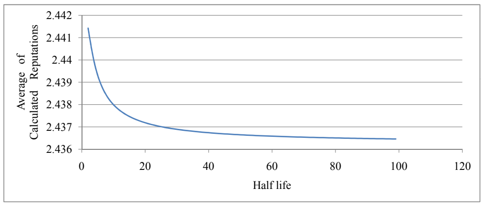


Fig. 2. The impact of half life value on average of calculated reputations.

Definition 7: We define the Weight of Reputation and denote it by Ω_j to show how dependable is the calculated reputation rank for w_j . We consider the weight of pairwise trust ranks and degree of fairness ranks involved in calculating the reputation to compute its weight. The Ω_j is calculate as follows:

$$\Omega_j = \sum_{l \in R_j} \omega_{lj}\varphi_{lj} \tag{12}$$

Equation 12 shows that the weight of reputation is calculated by aggregating weight of trust ranks received from all involved evaluators weighted by the pairwise degree of fairness between every evaluator and the worker. The Algorithm 1 shows the process of calculating reputation scores and their corresponding weights in a simple algorithmic manner.

VIII. Implementation and Evaluation

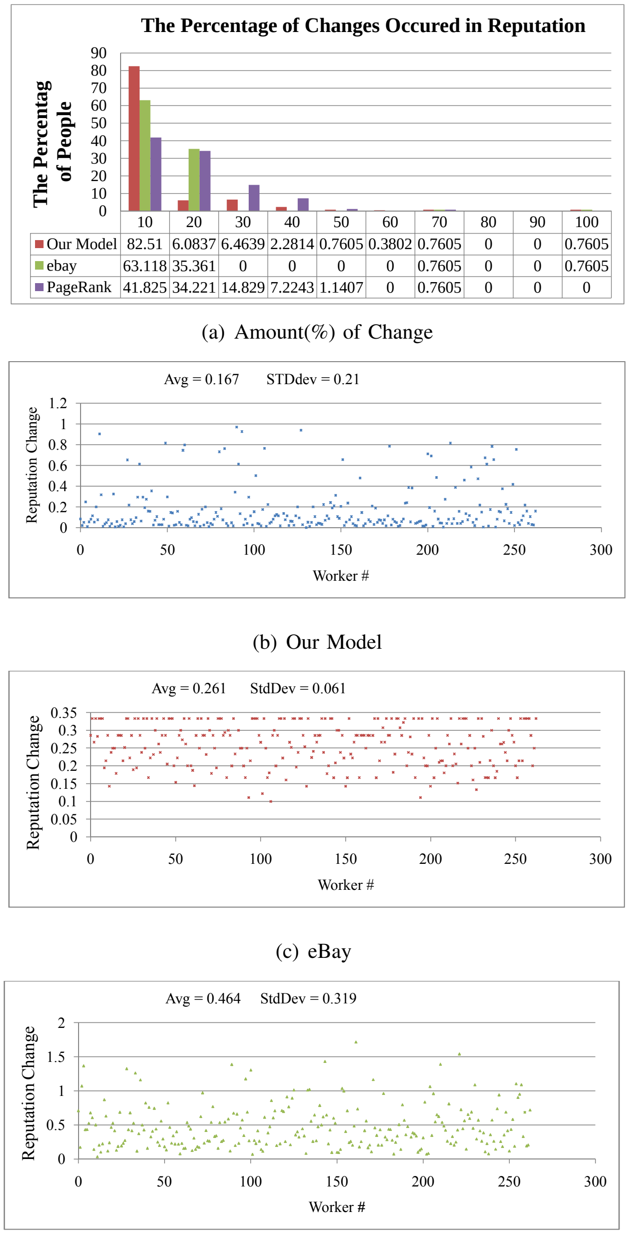
Implementation. We have proposed our model using a graph data model. To implement it, we have used a graph processing language proposed in our previous work [4], [3]. Previously in [4] we proposed a query language for graph analysis called FPSPARQL. FPSPARQL is a folder-enabled extension for SPARQL which helps users group related nodes, apply queries on them and save them for further use. SPARQL [14] is an RDF query language, standardized by the World Wide Web Consortium, for semantic web. In our recent work [3], we enhanced FPSPARQL by adding features for online analytical processing on graphs. We use the recent version of FPSPARQL to calculate trust, fairness and reputations scores in our model. To do so, we also need to select a value for constant *'q'*. Regarding Equations (8) and (9), the bigger the half life, the smaller the q and consequently trust ranks are. We ran an experiment with different values for q and observed how the average of calculated reputation ranks changes for different values of half life. The results are shown in Figure 2. We choose the smallest possible value i.e. 2 for half life to maximize trust ranks. We also map votes in WIKILog from {-1,0,1} to {1,2,3} to make them compatible with our model.

Evaluation. To evaluate our model, we compare it with two popular reputation calculation models. The first one is normal averaging model which is widely used in existing crowdsourcing systems. The models used in Amazon, eBay, and lots of other online communities or markets is normal averaging. In normal averaging model reputation is the average of all votes cast on the quality of he worker's contributions. The second model is adaptive averaging model in which the votes cast by people are weighted by reputation of the voter. The Google PageRank model [13], EigenTrust [9] and our previous work [7] are examples of adaptive averaging model. We have chosen PageRank as the base of all these reputation models to compare with our model.

To assess robustness of our model against unfair evaluations, in the first step we apply all three models to WIKILog and calculate a reputation rank for each worker in every model. Then, we add some noises to the dataset to check robustness of the models against unfair evaluations. As noise, we add reasonable amount of 20% unfair votes on all workers. To check robustness of models we manipulate reputation of workers by supporting all untrustworthy workers (workers with normal average reputation less than 2) by adding votes with value of 3. We also attack all trustworthy workers (workers with normal average reputation greater or equal than 2) by adding votes with value of 1. Then we calculate reputation scores again and analyze changes happened in the reputation of workers. Figure 3(a) shows how reputation of workers have changed after adding noises to the dataset. The horizontal axis of the chart is the amount of changes happened in the reputation of workers and the vertical axis is the fraction of workers who have experienced that amount of the change in their reputation scores.

As shown in Figure 3(a), 82.5% of the workers in our model experienced changes less than 10% in their reputation. This fraction is only 63.1% for eBay and 41.8% for PageRank. So, in comparison with eBay and PageRank, our model is more robust against manipulations of reputation by unfair evaluations.

Figures 3(b) to 3(d) show the distribution of changes in the reputation of workers in these three models. We note that there are some users whose reputations in our model have not changed but in others have. To better compare the changes, we have chosen just workers whose reputations in our model has changed and compared it with other models. As shown in Figure 3(b) the changes in our model are distributed with an average of 0.167 and a standard deviation of 0.21. The average and the standard deviation for eBay are 0.261 and 0.061 (Figure 3(c)) and 0.464 and 0.319 for PageRank respectively (see Figure 3(d)). This implies that changes in the reputation of the workers in our model are mostly in range of $[0, 0.377]$. For eBay most of the changes fall in the range of $[2.0, 0.322]$ and for PageRank in the range of $[0.145, 0.783]$. It shows that the changes in the reputation scores due to such attacks in our model is significantly lower than in the other two; thus, our model is more robust against unfair evaluations.



	10	20	30	40	50	60	70	80	90	100
🟥 Our Model	82.51	6.0837	6.4639	2.2814	0.7605	0.3802	0.7605	0	0	0.7605
🟩 ebay	63.118	35.361	0	0	0	0	0.7605	0	0	0.7605
🟪 PageRank	41.825	34.221	14.829	7.2243	1.1407	0	0.7605	0	0	0

(a) Amount(%) of Change

(b) Our Model

(c) eBay

(d) PageRank

Fig. 3. Changes happened in reputations in three models

Our model also provides weights for calculated reputations. As shown in Table I, the reputations calculated in eBay and PageRank models (and consequently all other similar models like Amazon and EigenTrust) are just one scalar value and it is very hard to judge the credibility of such calculated reputation or to compare two reputation ranks just using their values. But in our model every reputation comes with a corresponding weight showing the credibility of such reputation. This makes it easier to compare workers even when they have similar reputations. For example reputation of 'borisblue' in Table I is 2.96 and higher than reputation of 'elonka' which is 2.91. In terms of reputation values, 'borisblue' is more trustworthy than 'elonka' but by looking at the corresponding weights of their reputations we realize that we can trust 'elonka' more than 'borisblue' because the weight of reputation of 'borisblue', i.e., 1.032, is more than eighteen times smaller than the weight of reputation of 'elonka' which is 19.123. Thus, the reputation rank of 'elonka' is far more credible

UserID	UserName	Our Model		eBay Reputation	PageRank Reputation
		Reputation	Weight		
1	taoster	0.25	2.0	2.0	3.0
2	anthony	1.0	0.1250	1.0	1.5
....	....	....	....	....	....
948	borisblue	2.96	1.032	2.66	2.99
....	....	....	....	....	....
1038	elonka	2.91	19.123	2.76	2.92
....	....	....	....	....	....

TABLE I
SAMPLES OF CALCULATED REPUTATIONS.

than the reputation rank of ‘borisblue’, making ’elonika’ a preferred worker, despite its lower reputation. Thus, annotating reputation with a corresponding weight which indicates the credibility of such rank helps minimize the risk of choosing workers sub-optimally due to unreliable reputation ranks.

IX. DISCUSSION AND CONCLUSION

In this paper we have proposed a model for reputation management in crowdsourcing environments. We have introduced an analytic model for calculating a more dependable reputation rank of workers by taking into account the time, the credit amount and more importantly the credibility of the evaluators. We have also proposed a model for calculating a degree of fairness of evaluators. We use degree of fairness to distinguish the honest evaluators from dishonest ones who cast unfair votes. We have validated our model using experimental evaluations and compared the robustness of our model with two commonly used methods (eBay and PageRank). The presented results show that our model is more robust against manipulating the reputation of workers by unfair evaluations than eBay and PageRank.

In the real world, workers may involve in many crowdsourcing tasks, and our method very effectively utilizes this fact to make it harder to manipulate workers’ reputations by dishonest and unfair evaluations. The more activities the worker has, the more evaluations are needed to create a major change in her reputation. So, the experienced users that have lots of activities will benefit from more robust reputation scores. For the novice workers or workers with few activities, because of the small number of evaluations that build up their reputation scores, it is easier to manipulate their reputations by unfair evaluations. However, when the overall number of the activities of the user increases in time, those unfair evaluations will be detected and gradually the reputation of the worker will be corrected and unfair evaluations will be essentially ignored by our method for calculating reputation of workers.

As future work, we plan to extend our model to identify colluding groups and protect workers against collaborative unfair evaluation. We are also in the process of building a flexible people evaluation tool based on our model which can be seamlessly integrated with the existing crowdsourcing platforms.

REFERENCES

[1] E. Agichtein et. al. Finding high-quality content in social media. In *WSDM ’08*, pages 183–194, New York, NY, USA, 2008. ACM.

[2] D. Alfaro et. al. Reputation systems for open collaboration. *Commun. ACM*, 54:81–87, August 2011.

[3] S.-M.-R. Beheshti, B. Benatallah, H. R. M. Nezhad, and M. Allahbakhsh. A framework and a language for on-line analytical processing on graphs. In *Web Information System Engineering (WISE), 13th International Conference, Paphos, Cyprus*, 2012.

[4] S. Beheshti et. al. A query language for analyzing business processes execution. In *BPM*, pages 281–297, 2011.

[5] L. CABRAL and A. HORTACSU. The dynamics of seller reputation: Evidence from ebay*. *The Journal of Industrial Economics*, 58(1):54–78, 2010.

[6] J. Howe. The rise of crowdsourcing. *Wired*, June 2006.

[7] A. Ignjatovic, N. Foo, and C. T. Lee. An analytic approach to reputation ranking of participants in online transactions. In *The 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 587–590, Washington, DC, USA, 2008. IEEE Computer Society.

[8] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP ’10, pages 64–67, New York, NY, USA, 2010. ACM.

[9] S. Kamvar et. al. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.

[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *The 19th international conference on World wide web*, pages 641–650, New York, NY, USA, 2010. ACM.

[11] T. H. Noor and Q. Z. Sheng. Trust as a Service: A Framework for Trust Management in Cloud Environments. In *The 12th International Conference on Web and Information Systems (WISE’11)*, Sydney, Australia, October 2011.

[12] Z. Noorian and M. Ulieru. The state of the art in trust and reputation systems: A framework for comparison. *Journal of theoretical and applied electronic commerce research*, 5(2):97–117, 2010.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[14] E. Prud’hommeaux and A. Seaborne. Sparql query language for rdf (working draft). Technical report, W3C, March 2007.

[15] l. Qiao et. al. An empirical study of collusion behavior in the maze p2p file-sharing system. In *Proceedings of the 27th International Conference on Distributed Computing Systems*, ICDCS ’07, pages 56–, Washington, DC, USA, 2007. IEEE Computer Society.

[16] Y. Sun and Y. Liu. Security of online reputation systems: The evolution of attacks and defenses. *Signal Processing Magazine, IEEE*, 29(2):87 –97, march 2012.

[17] G. Swamynathan, K. Almeroth, and B. Zhao. The design of a reliable reputation system. *Electronic Commerce Research*, 10:239–270, 2010. 10.1007/s10660-010-9064-y.

[18] M.-C. Yuen, I. King, and K.-S. Leung. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT),2011 IEEE Third International Confernece on Social Computing (SocialCom)*, pages 766 –773, oct. 2011.