Incentive Mechanism Design for Truth Discovery in Crowdsourcing with Copiers

Lingyun Jiang, Xiaofu Niu, Jia Xu, *Member, IEEE,* Dejun Yang, *Senior Member, IEEE,* Lijie Xu

Abstract—Crowdsourcing has become an effective tool to utilize human intelligence to perform tasks that are challenging for machines. Many truth discovery methods and incentive mechanisms for crowdsourcing have been proposed. However, most of them cannot deal with the crowdsourcing with copiers, who copy a part (or all) of data from other workers. This paper aims at designing crowdsourcing incentive mechanism for truth discovery of textual answers with copiers. We formulate the problem of maximizing the social welfare such that all tasks can be completed with the least confidence for truth discovery and design an three-stage incentive mechanism. In contextual embedding and clustering stage, we construct and cluster the content vector representations of textual crowdsourced answers at the semantic level. In truth discovery stage, we estimate the truth for each task based on the dependence and accuracy of workers. In reverse auction stage, we design a greedy algorithm to select the winners and determine the payment. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the proposed mechanisms achieve computational efficiency, individual rationality, truthfulness, and guaranteed approximation. Moreover, our truth discovery methods show prominent advantage in terms of precision when there are copiers in the crowdsourcing systems.

Index Terms—crowdsourcing; truth discovery; incentive mechanism; Bayesian analysis; semantic analysis

1 Introduction

Crowdsourcing is a distributed problem-solving paradigm, in which a crowd of undefined size is engaged to solve complex or large-scale tasks through an open platform. Many famous knowledge repositories, such as Wikipedia, Zhihu and Freebase were created by workers, who contributed knowledge on a wide variety of topics. In recent years, crowdsourcing has become an effective tool and has been widely applied to many fields, such as video analysis [1], knowledge discovery [2], Smart Citizen [3], and conducting human-robot interaction studies [4].

Many crowdsourcing applications require integrating data from multiple workers, each of which provides a set of values as "facts". However, "facts and truth really don't have much to do with each other" [6]. Different workers may provide conflicting values, some being true while some being false. To provide data with high accuracy to the requesters, it is critical for the truth discovery systems to resolve conflicts and discover true values.

The crowdsourcer extracts and aggregates crowdsourced information in order to discover the truth, where the accuracy of crowdsourcing data is fundamentally important. In crowdsourcing, the accuracy of data can be largely affected by the expertise and willingness of individual workers [7]. Particularly, the workers with different knowledge and personal effort levels usually submit data with different accuracy. Furthermore, the rational workers always strategically minimize their efforts when performing the tasks, decreasing the accuracy of data.

Typically, we often expect the true value provided by more workers than any false one, so we can apply voting [8] and take the value provided by the majority of the workers as the truth. The main drawback of this approach is that they treat the reliability of each worker equally.

Unfortunately, the behavior of copying between workers is common in practice, especially when the crowdsourcing tasks require the textual answers, such as film review and sentence translation. This is because it needs more efforts to provide the textual answers than the numeric data or the choices from predefined options. On the other hand, most of the information is about some static aspects of the world, such as the authors and publishers of books, directors, actors and actresses of movies, and the presidents of a company in past years. In this scenario, the workers may copy, crawl, or aggregate the data previously submitted by others, and submit the copied data. For example, [9] collected data on Manhattan restaurants from 12 web sources from 1/2009 to 3/2009. There are 5269 restaurants mentioned by at least 2 data sources. Among them, there are 280 closed restaurants that they still provide in their lists. The behavior of copying cannot be avoided by the compliance rules because submitting the same data with others does not imply the copying behavior directly.

Existence of copying behavior will make most of the existing truth discovery methods ineffective [10-14] since they assume that the workers are independent. For example, as shown in Table 1, there are five workers, who provide the director's names of five films, and only worker 1 provides all correct data. However, since the names provided by worker 4 and worker 5 are copied from worker 3 (with certain errors during copying), the naive voting method will consider them as the majority, making wrong decisions of the truth for *ET*, *Godzilla*, and *Totoro*.

In this paper, we aim to develop an integrated solution to solve the following two issues: given the conflicting values provided by the workers with copiers, how to estimate the true value? Further, how to incentivize the strategic workers with high accuracy?

As illustrated by Fig. 1, we model the crowdsourcing process as a sealed reverse auction. First, the platform publicizes a set of

- *L. Jiang, X. Niu, J. Xu and L. Xu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China. E-mail: jianglingyun, 1017041103, xujia , ljxu@njupt.edu.cn.*
- *D. Yang. is with Colorado School of Mines, Golden, CO 80401. E-mail: djyang@mines.edu.*

Corresponding author: Jia Xu. E-mail: xujia@njupt.edu.cn
This is an extended and enhanced version of the paper [5] that appeared in IEEE ICDCS 2019. This research was supported in part by NSFC grants 61872193, 61872191 and 62072254, and NSF grants 1717315.

TABLE 1
An example of crowdsourcing with copiers

Tasks \ Workers	1	2	3	4	5
The Lord of the Rings	*Peter Jackson*	*James Cameron*	*Peter Jackson*	*Peter Jackson*	*Luc Besson*
ET	*Steven Spielberg*	*James Cameron*	*Roland Emmerich*	*Roland Emmerich*	*James Cameron*
Avatar	*James Cameron*	*James Cameron*	*James Cameron*	*James Cameron*	*James Cameron*
Godzilla	*Roland Emmerich*	*Peter Jackson*	*Hayao Miyazaki*	*Hayao Miyazaki*	*Hayao Miyazaki*
Totoro	*Hayao Miyazaki*	*Hayao Miyazaki*	*Luc Besson*	*Luc Besson*	*Luc Besson*

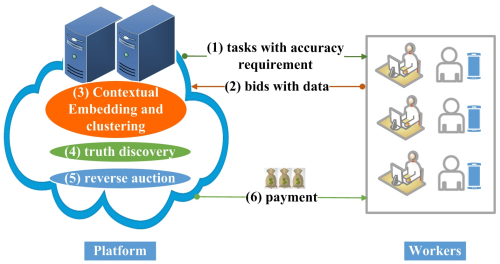


Fig. 1. Reverse auction based crowdsourcing process

tasks, and each task has an accuracy requirement. The workers who are interested in performing the crowdsourcing tasks can bid with the data. The platform then preprocesses the crowdsourcing answers through contextual embedding and clustering. Afterwards, the platform executes the truth discovery for each task. Meanwhile, the accuracies of workers are estimated in the truth discovery process. Finally, the platform selects a subset of workers as winners and determines the payment to winners based on the bid price and accuracies of workers.

We aim to present an *Incentive Mechanism for Crowdsourcing with Copiers* (*IMC*^{2}), which is a three-stage incentive mechanism, consisting of Stage 1: contextual embedding and clustering; Stage 2: truth discovery; Stage3: reverse auction. In the contextual embedding and clustering stage, *IMC*^{2} constructs and clusters the content vector representations of crowdsourced answers. In the truth discovery stage, *IMC*^{2} performs the *Dependence and Accuracy based Truth Estimation* (*DATE*) and returns the accuracy of workers at the same time. Stage 1 and Stage 2 together is termed as *S-DATE* (*Semantic-oriented DATE*). In the reverse auction stage, *IMC*^{2} selects the winners and determinates the payment to the workers.

The problem of designing truthful incentive mechanism for the truth discovery in crowdsourcing with copiers is very challenging. First, for the textual data of crowdsourcing, we need to analyze the semantics to obtain the value of the data before estimating the true value. Second, we do not know how workers obtain their data. Thus, we need to detect copiers from a snapshot of data. It is challenging to detect the copiers because submitting the same data with others does not imply the copying behavior directly. Third, if any two workers submit the same data, it is not obvious which one is the copier only based on a snapshot of data. This means that we should calculate the bidirectional dependence. Furthermore, the effective method of accuracy calculation is needed for the copiers since the copiers may contribute to the truth discovery by submitting the combination of the manual data and copied data. Finally, the workers may take a strategic behavior by submitting dishonest bid price to maximize their utility.

The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to design the incentive mechanism, which stimulates the strategic workers to reach the least confidence for truth discovery at the semantic level in the crowdsourcing.
- To obtain the values of textual crowdsourcing answers, we propose the method of semantic preprocessing, which employs language representation model and adaptive clustering algorithm.
- We propose a truth discovery algorithm, which takes both the dependence and accuracy of workers into consideration, for the crowdsourcing with copiers.
- We model the *Social Optimization Accuracy Coverage* (*SOAC*) problem and design a reverse auction mechanism to solve the *SOAC* problem. We show that the designed mechanism satisfies the desirable properties of computational efficiency, individual rationality, truthfulness, and guaranteed approximation.

The rest of the paper is organized as follows. We review the state-of-art research in Section 2. Section 3 models the *SOAC* problem, defines the dependence, and lists some desirable properties. Section 4 presents the detailed design of contextual embedding and clustering. Section 5 presents the detailed design of truth discovery. Section 6 presents the detailed design of reverse auction. Section 7 presents the detailed analysis of proposed incentive mechanism. We present the performance evaluation in Section 8 and conclude this paper in Section 9.

2 Related Work

2.1 Truth Discovery in Crowdsourcing

Many methods of truth discovery for crowdsourcing have been proposed in the literature. Miao *et al.* proposed the first privacy-preserving truth discovery framework called *PPTD* [10], which relies on the threshold homomorphic cryptosystem to protect the confidentiality of workers' values and weights. Tang *et al.* proposed non-interactive privacy-preserving truth discovery to protect workers' data while enabling truth distillation [11]. Xiao *et al.* proposed *BUR* protocol [12], which can recruit nearly the minimum number of workers while ensuring that the total accuracy of each task is no less than a given threshold. Wu *et al.* designed an unsupervised learning approach to quantify the workers' data qualities and long-term reputations [13]. Wu *et al.* proposed a novel approach to against strategic Sybil attack, called *TDSSA* [15], which runs extended truth discovery and probabilistic task assignment in a batch mode to periodically promote tasks completed by high-quality workers as new golden tasks. Li *et al.* studied truth discovery problem from biased crowdsourced answers and designed three fairness enhancing methods, namely *Pre-TD*, *FairTD*, and *Post-TD*, for truth discovery [16]. Wang *et al.* proposed an edge computing-based privacy-preserving truth discovery mechanism, *PrivSTD* [17], for streaming crowdsourced data to realize high accuracy of discovered truth while protecting the privacy of workers. However, none of these studies considers the incentive to the workers.

Jin *et al.* proposed an integrated framework for multi-requester mobile crowdsourcing, *CENTURION* [14], consisting of a data

aggregation mechanism and a double auction-based incentive mechanism. However, the data aggregation mechanism largely depends on the reliability level of workers, which is usually unknown in advance. Sun *et al.* proposed a contract-based personalized privacy-preserving incentive mechanism for truth discovery in crowdsourced question answering systems, *PINTION* [18], which provides personalized payments for workers with different privacy demands as a compensation for privacy cost, while ensuring accurate truth discovery. However, *PINTION* is only effective for binary-choice question answering. Moreover, both [14] and [18] do not consider the copying behavior. Our early work proposed a truth discovery mechanism, *DATE*, in crowdsourcing with copiers [5]. However, *DATE* can only be applied for numeric crowdsourcing data or the choices from predefined options. *DATE* is ineffective for textual crowdsourcing answers. This is because different textual answers may have similar meaning, which can be regarded as with same value, and *DATE* cannot recognize the semantics of textual answers. This study extends the capability of *DATE* for the processing of text data.

2.2 Quality-aware Crowdsourcing Incentive Mechanisms

Various quality-aware incentive mechanisms have been proposed for crowdsourcing systems. Jin *et al.* proposed *INCEPTION* [19], a framework that integrates the incentive, data aggregation, and data perturbation. Wang *et al.* studied the problem of measuring workers' long-term quality and proposed *MELODY* [20]. Wen *et al.* proposed an incentive mechanism based on a *Quality Driven Auction* [21], where the worker is paid based on the quality of sensed data. Jin *et al.* designed an incentive mechanism based on reverse combinatorial auctions, and incorporate the *Quality of Information* (*QoI*) of workers into the incentive mechanism [22]. Xu *et al.* stated that choosing the compatible users to perform cooperative tasks can improve the quality and success rates of mobile crowd sensing service, and proposed truthful incentive mechanisms for the mobile crowd sensing system, where each task needs to be performed by a group of compatible users [23]. Recently, some online quality-aware incentive mechanisms have been proposed. Miao *et al.* proposed a probabilistic model to measure the quality of tasks and a hitchhiking model to characterize workers' behavior patterns. They modeled the task assignment as a quality maximization problem and derive a polynomial-time online assignment algorithm [24]. Gao *et al.* formulated an optimization problem to maximize the amount of high-quality sensing data under a limited task budget. They presented a quality-aware incentive mechanism for online scenarios [25]. The proposed incentive mechanism allows the platform to provide selected participants with an extra bonus according to task completion level and their previous performance. However, none of these studies considers the dependence of workers.

Overall, there is no off-the-shelf mechanism in the literature that considers both dependence and accuracy of workers.

3 System Model

3.1 Reverse Auction Model

We consider a crowdsourcing system consisting of a platform and a set $W = \{1, 2, ..., n\}$ of n workers, who are interested in performing the crowdsourcing tasks. The platform resides in the cloud. The platform publicizes a set $T = \{t_1, t_2, ..., t_m\}$ of m tasks and wants to discover the truth for each task from the crowdsourcing data. Each task $t_j \in T$ has an accuracy requirement Θ^j, which is the least confidence to discover the truth for t_j.

Each worker $i \in W$ submits a triple $B_i = (T_i, b_i, D_i)$, where $T_i \subseteq T$ is the task set it is willing to perform, and b_i is its bid price that worker i wants to charge for performing T_i. Each T_i is associated with the cost c_i, which is the private information and known only to worker i. Different from most crowdsourcing systems [26-29], each worker sends its data (answers) D_i of task set T_i to the platform at the same time. Let $\mathbf{D} = (D_1, D_2, ..., D_n)$ be the crowdsourcing data of all workers.

Given the task set T, the bid profile $\mathbf{B} = (B_1, B_2, ..., B_n)$, and the accuracy requirement profile $\mathbf{\Theta} = (\Theta^1, \Theta^2, ..., \Theta^m)$, the platform calculates the estimated truth $\mathbf{et} = (et^1, et^2, ..., et^m)$ for each task, the winner set $S \subseteq W$ and the payment p_i for each winner $i \in S$. We define the utility of any worker i as the difference between the payment and its real cost:

$$u_i = p_i - c_i \tag{1}$$

Since we consider the workers are selfish and rational individuals, each worker can behave strategically by submitting a dishonest bid price to maximize its utility.

The utility of the platform is:

$$u_0 = V(S) - \sum_{i \in S} p_i \tag{2}$$

where $V(S)$ is the value of the platform obtained from the winners.

The social welfare is defined as the total utility of the platform and all workers:

$$u_{social} = u_0 + \sum_{i \in W} u_i = V(S) - \sum_{i \in S} c_i \tag{3}$$

We consider an incentive mechanism $\mathcal{M}$ consisting of a truth estimation function, a winner selection function and a payment function. The truth estimation function estimates the truth $\mathbf{et}$ for all tasks and returns an accuracy matrix $\mathbf{A} = \{A_i^j\}_{n \times m}$, where A_i^j is the accuracy of worker i for task t_j, $i \in W$, $t_j \in T$. The winner selection function outputs the subset of workers $S \subseteq W$. The payment function returns a vector $\mathbf{p} = (p_1, p_2, ..., p_n)$ of payments to all workers.

The objective of our incentive mechanism is maximizing the social welfare subject to the constraint that each task can satisfy the accuracy requirement. We consider that there is no incremental value of the platform on winners if all of the tasks can be completed by the workers in S with accuracy no less than the accuracy requirement, that is, the value of $V(S)$ is constant if the accuracy constraints of all tasks can be satisfied. This assumption is reasonable for crowdsourcing systems as shown in [26, 30] because the platform already has enough confidence for truth discovery. In this case, the problem of maximizing the social welfare is equivalent to the problem of minimizing the social cost (total cost of winners). We refer to this problem as the *Social Optimization Accuracy Coverage* (*SOAC*) problem, which can be formulated as follows:

$$\textbf{(SOAC):} \quad \textit{Minimize} \sum_{i \in S} c_i \cdot x_i \tag{4}$$

$$\textbf{s.t.} \sum_{i \in W} A_i^j \cdot x_i \geq \Theta^j, \ \forall t_j \in T \tag{5}$$

$$x_i \in \{0, 1\}, \ \forall i \in W \tag{6}$$

where x_i is the binary variable for each worker $i \in W$. Let $x_i = 1$ if i is a winner; otherwise, $x_i = 0$.

The constraint (5) represents the accuracy coverage for each task and ensures that the total accuracy of all winners for this task is no less than the accuracy requirement. To ensure the confidence level of truth discovery collected from workers, sufficient amount of collective effort is necessary for each task. Note that quantifying the integrated crowd quality (accuracy in this paper) is still a challenging problem in crowdsourcing, since correlation or interaction may exist among answers. Existing works all make assumption for integrated quality of workers, such as additive assumption made in [20, 22]. In this paper, we also take the additive assumption, in which the integrated accuracy of workers is defined as the total accuracy of workers. On the other hand, for the task that none of the workers capable to execute it with high-accuracy, collective efforts of multiple workers are necessary to ensure crowdsourcing quality. Constraint (5) ensures that the tasks can be performed by the workers with integrated crowd quality.

3.2 Dependence Model of Workers

We take the dependence of workers into consideration in order to reduce the impact of copiers on truth estimation.

Definition 1. (Dependence of workers) *We say that there exists a dependence between any two workers i and i' if they derive the same part of their data directly from the other worker.*

An independent worker provides all values independently. It may provide some erroneous values because of incorrect knowledge of the real world, mis-spelling, etc. We use $i \perp i'$ to represent that workers i and i' are independent.

A copier copies a part (or all) of data from other workers (independent workers or copiers). Let r be the probability that a value provided by a copier is copied. The value of r can be estimated from the historical crowdsourcing data (see parameter setting in subsection 8.2). When there is no historical crowdsourcing data, the value of r depends on the quality requirement of specific crowdsourcing tasks and the trust in the workers. The higher the quality requirement is or the lower the trust in the workers, the more the workers are selected in order to satisfy the accuracy requirement of platform. The copier can copy from multiple workers by union, intersection, etc. In addition, a copier may revise some of the copied values or add additional values. Such revised and added values are considered as independent contributions of the copier. For any two workers i and i', we denote i depending on i' by $i \rightarrow i'$.

To make the computation tractable, we assume that the dependence of workers satisfies the following properties:

- **Independent copying:** The dependence of any pair of workers is independent of the dependence of any other pair of workers.
- **No loop dependence:** The dependence relationship between workers is non-transitive.
- **Uniform false-value distribution:** For each task, an independent worker has the same probability of providing each of multiple false values of the task (we will remove this assumption later).

3.3 Desirable Properties

We list the desirable properties of designed incentive mechanism:

- **Computational efficiency:** An incentive mechanism is computationally efficient if the truth estimation **et**, the winner set S, and the payment vector **p** can be computed in polynomial time.
- **Individual Rationality:** Each winner will have a non-negative utility while bidding its true cost.
- **Truthfulness:** An incentive mechanism is truthful if reporting the true cost is a weakly dominant strategy for all workers. In other words, no worker can improve its utility by submitting a false cost, no matter what others submit.
- **Social Optimization:** We attempt to find the optimal solution or approximation algorithm for *SOAC* problem.

The importance of the first two properties is obvious, because they together assure the feasibility of the incentive mechanism. The third property is indispensable for guaranteeing the compatibility. Being truthful, the incentive mechanism can eliminate the fear of market manipulation and the overhead of strategizing over others for the workers. The last property guarantees that the incentive mechanism can have a guaranteed approximation ratio to the optimal solution.

We list the frequently used notations in Table 2.

TABLE 2
Frequently used notations

Symbol	Description
W, n	worker set, number of workers
T, m	task set, number of tasks
$\mathbf{\Theta}$, Θ^j	accuracy requirement profile, accuracy requirement of task t_j
T_i, t_j	task set of worker i, task j
num^j	number of false values of task t_j
b_i, c_i	bid price of worker i, cost of worker i
$\mathbf{D}$, D_i	data of all workers, data of worker i
$\mathbf{B}$, B_i	bid profile, bid of worker i
et, et^j	estimated truth of all tasks, estimated truth of task t_j
$\mathbf{p}$, p_i	payment profile, payment of worker i
u_i, u_0, u_{social}	utility of worker i, utility of the platform, social welfare
S, $V(S)$	winner set, value of winners
$\mathbf{A}$, A_i^j	accuracy matrix, accuracy of worker i for task t_j
W_v^j	set of workers who provide value v for task t_j
T^s	set of tasks on which i and i' provide the same true value
T^f	set of tasks on which i and i' provide the same false values
T^d	set of tasks on which i and i' provide different values
P_s^j	probability that i and i' provide the same true value for task t_j
P_f^j	probability that i and i' provide the same false value for task t_j
P_d^j	probability that i and i' provide different values on task t_j
$I_v^j(i)$	probability that worker i provides value v of task t_j independently
r	copy probability
ε	initial accuracy
α	priori probability of dependence
φ	maximum number of iterations

4 Contextual Embedding and Clustering

In this section, we construct the feature for clustering crowdsourced answers. And the feature, content vector representation of the answers, is constructed with the contextual embedding based on *BERT* (*Bidirectional Encoder Representations from Transformers*) [31]. Unlike [32] (using unidirectional language models for pre-training) and [33] (using a shallow concatenation of independently trained left-to-right and right-to-left LMs), the

Transformer-based *BERT* is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers, and the *BERT* model further advances the model performance by introducing pre-training and has reached the state-of-the-art performance on many related NLP tasks. Based on the content vector representation of the answers, we use *KANN-DBSCAN* (*K-Average Nearest Neighbor Density-Based Spatial Clustering of Applications with Noise*) [34] to cluster the answers adaptively.

4.1 Contextual Embedding

The *BERT* model architecture is shown in Fig.2, where E_i, $i = 1, 2, ..., N$, represents input embedding of a single word, H_i, $i = 1, 2, ..., N$ represents the hidden layer, and Trm represents a feature extractor based on the attention mechanism, namely *Transformer* model [35]. After the attention matrix and attention weighting in *Transformer*, each word in the sequence contains the information before and after the word. Therefore, one word token may have a different embedding depending on its intended meaning in the sentence. After the attention mechanism and weighting, the current word is re-expressed by all other words in this sentence.

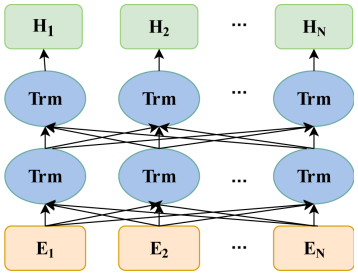


Fig. 2. *BERT* model architecture

Given a sequence of input embeddings, the output contextual embedding is composed by the input sequence with different attention at each position. The attention weight is calculated as:

$$Attention_{(DQ,DK)} = Softmax(\frac{DQ \cdot DK^T}{\sqrt{d_k}}) \quad (7)$$

where DQ, DK are query and key matrix of input embeddings, d_k is the length of a query or key vector. Multiple parallel groups of such attention weights, also referred as attention heads, make it possible to attend to information at different positions.

The pre-training process of *BERT* consists of two different tasks: *Masked Language Model* (*MLM*) and *Next Sentence Prediction* (*NSP*). The purpose of *MLM* is to make *BERT* be a universal model for different crowdsourcing tasks. The purpose of *NSP* is to let the *BERT* understand the connection between any two sentences. To obtain a vector representation of each answer, we usually fine-tune the trained *BERT* model.

4.2 Clustering

In this subsection, we cluster the high-dimensional vector representations of answers from *BERT* for each task. Since the number of values is unknown, we use *KANN-DBSCAN* to determine the parameters adaptively without requirement for the distribution of crowdsourced data sets. Algorithm 1 illustrates *KANN-DBSCAN*.

We consider there are n^j answers for any task $t_j \in T$, and the corresponding vector representations is $\overline{\mathbf{D}^j}$. Let $e^j_{k,k'} = dist(d^j_k, d^j_{k'})$ be the cosine distance of any two vectors $d^j_k, d^j_{k'} \in \overline{\mathbf{D}^j}$ (Lines 2-6), and the distance matrix is $\{e^j_{k,k'}\}^{n^j \times n^j}$. For each row of $\{e^j_{k,k'}\}^{n^j \times n^j}$, we sort $e^j_{k,k'}$ in nondecreasing order (Lines 7-9).

Algorithm 1 : KANN-DBSCAN

Input: vector representations of answers $\overline{\mathbf{D}}$
Output: cluster set C^j for each task $t_j \in T$

1: **for each** $t_j \in T$ **do**
2: **for** $k = 1$ **to** n^j **do**
3: **for** $k' = 1$ **to** n^j **do**
4: $e^j_{k,k'} = dist(d^j_k, d^j_{k'})$;
5: **end for**
6: **end for**
7: **for** $k = 1$ **to** n^j **do**
8: sort $e^j_{k,k'}$ in nondecreasing order, $\forall k' = 1, 2, ..., n^j$;
9: **end for**
10: **for** $k = 1$ **to** n^j **do**
11: $Eps^j_k \leftarrow \frac{\sum_{k'=1}^{n^j} e^j_{k',k}}{n^j}$;
12: $MinPts^j_k \leftarrow \frac{\sum_{k'=1}^{n^j} |N_{Eps}(Eps^j_k, d^j_{k'})|}{n^j}$;
13: **end for**
14: $mark \leftarrow 0$;
15: **for** $k = 1$ **to** $n^j - 1$ **do**
16: $N^k_c \leftarrow |DBSCAN(Eps^j_k, MinPts^j_k)|$;
17: $N^{k+1}_c \leftarrow |DBSCAN(Eps^j_{k+1}, MinPts^j_{k+1})|$;
18: **if** $N^k_c == N^{k+1}_c$ **then**
19: $mark \leftarrow 1$;
20: **end if**
21: **if** $mark \neq 0$ **and** $N^k_c \neq N^{k+1}_c$ **then**
22: $k_0 \leftarrow k$;
23: **break**;
24: **end if**
25: **end for**
26: $C^j \leftarrow DBSCAN(Eps^j_{k_0}, MinPts^j_{k_0})$;
27: **end for**

Then we generate n^j pairs of parameters. Specifically, we compute the average distance Eps^j_k for each column of $\{e^j_{k,k'}\}^{n^j \times n^j}$ (Line 11). We define $N_{Eps}(Eps^j_k, d^j_{k'})$ as the set of answers with distance no more than Eps^j_k from $d^j_{k'} \in \overline{\mathbf{D}^j}$, i.e., $N_{Eps}(Eps^j_k, d^j_{k'}) = \{d^j_{k''} | dist(d^j_{k'}, d^j_{k''}) < Eps^j_k, \forall d^j_{k''} \in \overline{\mathbf{D}^j} \backslash \{d^j_{k'}\}\}$. We calculate the parameter $MinPts^j_k$, which is the average size of $N_{Eps}(Eps^j_k, d^j_{k'})$, for every Eps^j_k (Line 12). Thus, we generate a list of parameters of Eps^j_k and $MinPts^j_k$.

Next, we determine which pair of parameters is chosen in *DBSCAN* (Lines 14-25). The function $DBSCAN(Eps^j_k, MinPts^j_k)$ returns the cluster set through *DBSCAN* with parameters Eps^j_k and $MinPts^j_k$. Let *mark* indicate whether the number of clusters does not change. We first iterate every pair of parameters in order, until the number of clusters is not change. Then we go on the iteration, and choose the maximum of k (denoted as k_0) such that the size of $DBSCAN(Eps^j_k, MinPts^j_k)$ does not change. The final parameters are $Eps^j_{k_0}$ and $MinPts^j_{k_0}$. Finally, we call $DBSCAN(Eps^j_{k_0}, MinPts^j_{k_0})$ to obtain the cluster set C^j of task t_j.

The function $DBSCAN(Eps^j_k, MinPts^j_k)$ works as follows:

(1) Choose arbitrary unvisited answer $d^j_{k'}$ and find $N_{Eps}(Eps^j_k, d^j_{k'})$.

(2) If $|N_{Eps}(Eps^j_k, d^j_{k'})| \geq MinPts^j_k$, answer $d^j_{k'}$ and $N_{Eps}(Eps^j_k, d^j_{k'})$ generate a new cluster together. Recursively pro-

cess all unvisited answers in the current cluster in the same way to expand the cluster.

(3) If $|N_{Eps}(Eps_k^j, d_{k'}^j)| < MinPts_k^j$, mark answer $d_{k'}^j$ as the noise answer.

(4) For other unvisited answers, repeat step (1) to step (3) until all answers are belong to a cluster or are marked as noise answers.

(5) All noise answers generate clusters alone.

(6) Return the cluster set.

5 Truth Discovery

After the contextual embedding and clustering stage, the textual crowdsourced answers have been clustered. The answers in the same cluster are considered to have the same value. In this section, we present our truth discovery algorithm, *DATE*, to estimates the truth from conflicting values submitted by the workers. *DATE* follows three steps (the details will be shown in subsection 5.1, 5.2, and 5.3, respectively) illustrated by Fig. 3 iteratively until the estimated truth does not change or the number of iterations exceed the maximum number of iterations φ.

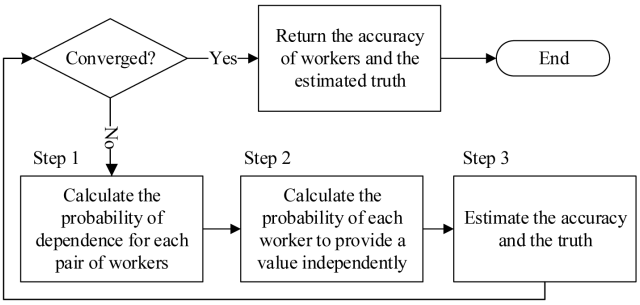


Fig. 3. Workflow of *DATE*

5.1 Dependence of Workers

We consider two types of workers: independent workers and copiers. For any pair of workers $i \in W$, $i' \in W$, $i \neq i'$, we apply Bayesian analysis to compute the probability that i and i' are dependent given the observation of data set $\mathbf{D}$ (also called value set of clustered answers). For this purpose, we need to compute the probability of the observed data, conditioned on the dependence or independence of these two workers.

We define three task sets: T^s, the set of tasks on which i and i' provide the same true value; T^f, the set of tasks on which they provide the same false values; T^d, the set of tasks on which they provide different values. Initially, the true value can be obtained through the voting mechanism on data set $\mathbf{D}$ for each task. In the following iterations, the true value will be determined based on the estimated truth **et**.

We first consider the situation where the two workers are independent. Since there is only one true value, the probability that i and i' provide the same true value for task t_j, denoted by P_s^j for convenience, is

$$P_s^j = P(t_j \in T^s |\ i \perp i') = A_i^j \cdot A_{i'}^j \tag{8}$$

where A_i^j and $A_{i'}^j$ are the accuracies of i and i' for task t_j, respectively. We set $A_i^j = \varepsilon$ for $\forall i \in W$, $\forall t_j \in T$, where $\varepsilon \in (0, 1)$ is the initial accuracy. We will refine the accuracy gradually in the later rounds of *DATE*.

According to the assumption of *uniform false-value distribution* made in subsection 3.2, any independent worker has the same probability of providing each false value. Without loss of generality, we consider that each task t_j has $(num^j + 1)$ different values, that is, there are one true value and num^j false values. Then, the probability that any worker i provides one false value for task t_j is $\frac{1-A_i^j}{num^j}$. Thus, the probability that i and i' provide the same false value for taskt_j, denoted by P_f^j, is

$$\begin{aligned} P_f^j &= P(t_j \in T^f |\ i \perp i') \\ &= num^j \cdot \frac{1 - A_i^j}{num^j} \cdot \frac{1 - A_{i'}^j}{num^j} = \frac{(1 - A_i^j) \cdot (1 - A_{i'}^j)}{num^j} \end{aligned} \tag{9}$$

Then, the probability that i and i' provide different values on task t_j, denoted by P_d^j, is

$$P_d^j = P(t_j \in T^d |\ i \perp i') = 1 - P_s^j - P_f^j \tag{10}$$

Thus, the conditional probability of observing $\mathbf{D}$ is

$$P(\mathbf{D}|i \perp i') = \prod_{t_j \in T^s} P_s^j \cdot \prod_{t_j \in T^f} P_f^j \cdot \prod_{t_j \in T^d} P_d^j \tag{11}$$

We next consider the situation where i and i' are with the dependence. When i copies from i' (similar for i' copying from i), we have

$$P(t_j \in T^s |\ i \to i') = A_{i'}^j \cdot r + P_s^j \cdot (1 - r) \tag{12}$$

$$P(t_j \in T^f |\ i \to i') = (1 - A_{i'}^j) \cdot r + P_f^j \cdot (1 - r), \tag{13}$$

$$P(t_j \in T^d |\ i \to i') = P_d^j \cdot (1 - r) \tag{14}$$

Thus, the conditional probability of observing $\mathbf{D}$ is

$$\begin{aligned} &P(\mathbf{D}|i \to i') \\ &= \prod_{t_j \in T^s} [A_{i'}^j \cdot r + P_s^j \cdot (1 - r)] \\ &\cdot \prod_{t_j \in T^f} [(1 - A_{i'}^j) \cdot r + P_f^j \cdot (1 - r)] \cdot \prod_{t_j \in T^d} [P_d^j \cdot (1 - r)] \end{aligned} \tag{15}$$

We compute $P(i \to i'|\mathbf{D})$ accordingly:

$$\begin{aligned} &P(i \to i'|\mathbf{D}) \\ &= \frac{P(\mathbf{D}|i \to i')P(i \to i')}{P(\mathbf{D}|i \to i')P(i \to i') + P(\mathbf{D}|i \perp i')P(i \perp i')} \\ &= [1 + (\frac{1\text{-}\alpha}{\alpha}) \cdot \prod_{t_j \in T^s} \frac{P_s^j}{A_{i'}^j \cdot r + P_s^j \cdot (1 - r)} \\ &\cdot \prod_{t_j \in T^f} \frac{P_f^j}{(1 - A_{i'}^j) \cdot r + P_f^j \cdot (1 - r)} \cdot (\frac{1}{1 - r})^{|T^d|}]^{-1} \end{aligned} \tag{16}$$

where $P(i \to i')$ is the a priori probability that worker i and i' are dependent. Let $P(i \to i') = \alpha$, $P(i \perp i') = (1 - \alpha)$, $0 < \alpha < 1$ be the default values for every pair of workers initially. The priori probabilities will be iteratively refined in the later rounds of *DATE*.

Note that the probability of i and i' providing the same true or false value is related to the directions of dependence. By applying the Bayesian rule, we can compute the directed probabilities for any pair of workers.

5.2 Probability of Providing the Value Independently

We have described how to detect the dependence of any pair of workers. However, it is possible that a copier provides some of the values independently. For example, copy answers for some tasks and provide the answers independently for other tasks. In this case, it will be inappropriate to ignore the contribution of these values. Thus, we describe how to obtain the probability that any worker provides the value independently in this subsection.

Note that the probability of dependence in (16) is based on the whole data collected. To estimate the truth for each task, we should calculate the probability of providing each possible value independently. Obviously, it would take exponential time to enumerate all possible dependence for each value between all pairs of workers.

To make the computation scalable, we need a polynomial time algorithm. The basic idea is to calculate the probability of providing each possible value v by considering the worker one by one for every task. For convenience, let D^j be the set of values of any task $t_j \in T$. Let W_v^j be the set of workers who provide value v for any task $t_j \in T$. The goal is to calculate the probability of any worker i to provide each possible value v of any task t_j independently, denoted as $I_v^j(i)$. For each task $t_j \in T$ and $v \in D^j$, we define an ordered set $\overline{W_v^j}$, and put the workers in W_v^j into $\overline{W_v^j}$ one by one. For each worker $i \in W_v^j$, $i \notin \overline{W_v^j}$, we compute the probability for i based on the dependence on the workers in $\overline{W_v^j}$.

This method is not precise because if any worker i depends only on workers in $W_v^j \backslash \overline{W_v^j}$ but some of those workers in $W_v^j \backslash \overline{W_v^j}$ depend on the workers in $\overline{W_v^j}$, our estimation still considers that the worker i provides the value independently. To minimize such error, we hope that both the probability that worker i depends on the workers in $W_v^j \backslash \overline{W_v^j}$ and the probability that the workers in $W_v^j \backslash \overline{W_v^j}$ depend on the workers in $\overline{W_v^j}$ be the lowest. Thus, we take advantage of the greedy algorithm and consider workers in such order: In the first round, we select a worker $i_0 \in W_v^j$ with the highest dependence probability, and make this worker as the first one in ordered set $\overline{W_v^j}$; In the later rounds, we select the worker that has the maximal dependence probability on one of the previously selected workers. This process ends when all workers are considered.

Thus, the probability that the worker i provides value v of task t_j independently is

$$I_v^j(i) \leftarrow \prod_{i' \in \overline{W_v^j}} (1 - r \cdot P(i \rightarrow i' | \mathbf{D})) \tag{17}$$

5.3 Accuracy and Truth Estimation

We next compute the accuracy of each worker. A straightforward way is to compute the fraction of true values provided by the worker. However, we do not know what the true values are exactly. Let D_i^j be the value of any task $t_j \in T$ provided by worker i. We denote $P^j(v)$ as the probability that v is true for any task $t_j \in T$. Then A_i^j is equal to $P^j(D_i^j)$:

$$A_i^j = P^j(D_i^j) \tag{18}$$

Now we compute $P^j(v)$ by considering both how many workers provide the value and the accuracies of these workers. We begin with the case where all workers are independent. For the observation D^j provided by each worker $i \in W^j$, where W^j is the set of workers who perform task t_j, we first compute the probability of D^j conditioned on v being true. This probability represents that the workers in W_v^j provide the true value, and the other workers in W^j provide one of the false values.

$$P(D^j | v \text{ is true}) = \Pi_{i \in W_v^j} A_i^j \cdot \Pi_{i \in W^j \backslash W_v^j} \frac{1 - A_i^j}{num^j} \tag{19}$$

Among the values in D^j, there is one and only one true value. Applying the a-priori belief of each value being true is the same, denoted by β. We then have

$$P(D^j) = \sum_{v \in D^j} (\beta \cdot \Pi_{i \in W_v^j} A_i^j \cdot \Pi_{i \in W^j \backslash W_v^j} \frac{1 - A_i^j}{num^j}) \tag{20}$$

Applying the Bayesian rule, we have

$$\begin{aligned} P^j(v) = P(v \text{ is true} | D^j) &= \frac{P(D^j | v \text{ is true}) P(v \text{ is true})}{P(D^j)} \\ &= \frac{\Pi_{i \in W_v^j} \frac{num^j \cdot A_i^j}{1 - A_i^j}}{\sum_{v' \in D^j} \Pi_{i \in W_{v'}^j} \frac{num^j \cdot A_i^j}{1 - A_i^j}} \end{aligned} \tag{21}$$

For the truth discovery, if a worker i copies a value v from other workers, we should ignore i when considering v as the truth. Thus, we adopt $\sum_{i \in W_v^j} A_i^j \cdot I_v^j(i)$ as the support counts of value v for any task $t_j \in T$, and find the value with the maximal support counts in D^j as the final estimated truth.

The whole process of *DATE* is illustrated in Algorithm 2.

5.4 Extension to Nonuniform False-value Distribution

So far, we have presented *DATE* with the assumption of *uniform false-value distribution*. However, the false values of a task may not be uniformly distributed. For example, in the minds of most people, the capital of *Australia* is *Sydney*, but in fact, *Canberra* is its capital. The probability of false value of *Sydney* will be larger than other false values. In this subsection, we extend *DATE* to *nonuniform false-value distribution*.

Let $f(h)$, $h \in [0, 1]$, be the percentage of false values whose distribution probability is h; thus, $\int_0^1 f(h)dh = 1$. Then, the probability that two false-value providers provide the same value is $\int_0^1 h^2 f(h)dh$ instead of $(\frac{1}{num^j})^2 \cdot num^j$. We revise (9) as:

$$P_f^j = P(t_j \in T^f | \ i \perp i') = (1 - A_i^j) \cdot (1 - A_i^j) \cdot \int_0^1 h^2 f(h)dh \tag{22}$$

Similarly, we need to revise formula (19) as follows:

$$\begin{aligned} &P(D^j | v \text{ is true}) \\ &= \Pi_{i \in W_v^j} A_i^j \cdot \Pi_{i \in W^j \backslash W_v^j} (1 - A_i^j \cdot e^{\int_0^1 \ln \ df(h) \cdot |W^j \backslash W_v^j|}) \end{aligned} \tag{23}$$

6 Reverse Auction Design

We first analysis the hardness of *SOAC* problem.

Theorem 1. *The SOAC problem is NP-hard.*

Proof: We consider a special case of *SOAC* problem, where the accuracy requirements for all tasks are the same and sufficiently close to zero. This means that any task $t_j \in T$ can be completed upon there is any worker $i \in W$ with $A_i^j > 0$. In this way, the problem can be simplified as selecting a subset $S \subseteq W$ with minimum total cost such that the workers in S can perform every task in T. Since each worker can bid for a subset of T with a cost, this special problem is actually an instance of the *Weighted Set*

Algorithm 2 : DATE

Input: worker set W, task set T, data set $\mathbf{D}$, copy probability r, initial accuracy ε, priori probability of dependence α, maximum number of iterations φ
Output: estimated truth $\mathbf{et}$, accuracy matrix $\mathbf{A}$

1: **for each** $i \in W$ **do**
2: **for each** $t_j \in T$ **do** $A_i^j \leftarrow \varepsilon$;
3: **for each** $i' \in W$, *s.t.* $i' \neq i$ **do**
4: $P(i \rightarrow i') \leftarrow \alpha, P(i \perp i') \leftarrow (1-\alpha)$;
5: **end for**
6: **end**
7: $\mathcal{K} \leftarrow 0$;
8: **while** $\mathbf{et} \neq \mathbf{et}'$ **and** $\mathcal{K} \leq \varphi$ **do**
9: **for each** $t_j \in T$ **do**
10: **for each** $v \in D^j$ **do** $\overline{W_v^j} \leftarrow \varnothing$;
11: **end for**
12: $\mathbf{et} \leftarrow \mathbf{et}'$;
// Step1: Calculate the probability of dependence
13: calculate $P(i \rightarrow i'|\mathbf{D})$ for every pair of workers $i, i' \in W, i \neq i'$ through formula (15) with $\mathbf{et}$ and $\mathbf{A}$;
// Step2: Calculate the probability of providing a value independently
14: **for each** $t_j \in T$ **do**
15: **for each** $v \in D^j$ **do**
16: $i_0 \leftarrow \arg \max_{i:i,i' \in W_v^j, i \neq i'} (P(i \rightarrow i'|\mathbf{D}) + P(i' \rightarrow i|\mathbf{D}))$;
17: $\overline{W_v^j} \leftarrow \{i_0\}$;
18: **while** $|\overline{W_v^j}| \neq |W_v^j|$ **do**
19: $i_0 \leftarrow \arg \max_{i:i \in W_v^j \setminus \overline{W_v^j}, i' \in \overline{W_v^j}} P(i \rightarrow i'|\mathbf{D})$;
20: $I_v^j(i_0) \leftarrow \prod_{i' \in \overline{W_v^j}} (1 - r \cdot P(i_0 \rightarrow i'|\mathbf{D}))$;
21: $\overline{W_v^j} \leftarrow \overline{W_v^j} \bigcup \{i_0\}$;
22: **end while**
// Step3: Estimate the accuracy and the truth
23: $P^j(v) \leftarrow \frac{\prod_{i \in W_v^j} \frac{num^j \cdot A_i^j}{1-A_i^j}}{\sum_{v' \in D^j} \prod_{i \in W_{v'}^j} \frac{num^j \cdot A_i^j}{1-A_i^j}}$;
24: **end for**
25: **for each** $i \in W$, *s.t.* $t_j \in T_i$ **do**
26: $A_i^j \leftarrow P^j(D_i^j)$;
27: **end for**
28: $et^j \leftarrow \arg\max_{v \in D^j} \sum_{i \in W_v^j} A_i^j \cdot I_v^j(i)$;
29: **end for**
30: $\mathcal{K} \leftarrow \mathcal{K} + 1$;
31: **end while**

Cover (*WSC*) problem. Since the *WSC* problem is a well-known NP-hard problem, the *SOAC* problem is NP-hard. ■

In fact, there is no $(1-\varepsilon)\ln n$ approximate polynomial time algorithm for *WSC* problem [36]. In addition, we cannot use the off-the-shelf *VCG* mechanism [37] since the truthfulness of *VCG* mechanism requires the optimal social cost. The designed reverse auction follows a greedy approach. As illustrated in Algorithm 3, our reverse auction consists of winner selection phase and payment determination phase.

In the winner selection phase, the workers are sorted according to the effective accuracy unit cost, which is defined as $\frac{b_i}{\sum_{t_j \in T_i} \min\{\Theta^{j'}, A_i^j\}}$ for any worker $i \in W$. In each iteration of the

Algorithm 3 : Reverse Auction

Input: task set T, bid profile B, worker set W, accuracy requirement profile Θ, accuracy matrix $\mathbf{A}$
Output: winner set S, payment $\mathbf{p}$
//Winner Selection Phase
1: $S \leftarrow \varnothing, \Theta' \leftarrow \Theta$;
2: **while** $\sum_{t_j \in T} \Theta^{j'} \neq 0$ **do**
3: $i \leftarrow \arg\min_{k \in W \setminus S} \frac{b_k}{\sum_{t_j \in T_k} \min\{\Theta^{j'}, A_k^j\}}$;
4: $S \leftarrow S \bigcup \{i\}$;
5: **for each** $t_j \in T_i$ **do**
6: $\Theta^{j'} \leftarrow \Theta^{j'} - \min\{\Theta^{j'}, A_i^j\}$;
7: **end for**
8: **end while**
//Payment Determination Phase
9: **for each** $i \in W$ **do** $p_i \leftarrow 0$;
10: **for each** $i \in S$ **do**
11: $W' \leftarrow W \setminus \{i\}, S' \leftarrow \varnothing, \Theta'' \leftarrow \Theta$;
12: **while** $\sum_{t_j \in T} \Theta^{j''} \neq 0$ **do**
13: $i_k \leftarrow \arg\min_{k \in W' \setminus S'} \frac{b_k}{\sum_{t_j \in T_k} \min\{\Theta^{j''}, A_k^j\}}$;
14: $S' \leftarrow S' \bigcup \{i_k\}$;
15: $p_i \leftarrow \max\{p_i, \frac{\sum_{t_j \in T_i} \min\{\Theta^{j''}, A_i^j\}}{\sum_{t_j \in T_{i_k}} \min\{\Theta^{j''}, A_{i_k}^j\}} b_{i_k}\}$;
16: **for each** $t_j \in T_{i_k}$ **do**
17: $\Theta^{j''} \leftarrow \Theta^{j''} - \min\{\Theta^{j''}, A_{i_k}^j\}$;
18: **end for**
19: **end while**
20: **end for**

winner selection phase, we select the worker with minimum effective accuracy unit cost over the unselected worker set $W \setminus S$ as the winner until the winners' accuracy can cover the accuracy requirements of all tasks.

In payment determination phase, for each winner $i \in S$, we execute the winner selection phase over $W \setminus \{i\}$ and denote the winner set as S'. We compute the maximum price that worker i can be selected instead of each worker in S'. We will prove that this price is a critical value for worker i later.

7 Mechanism Analysis

We present the theoretical analysis, demonstrating that IMC^2 can achieve the desired properties.

Lemma 1. *IMC^2 is computationally efficient.*

Proof: For Algorithm 1, the time complexity of *DBSCAN* is $O(n^j log_2 n^j)$ for a single task when using the *R-tree* to build the spatial index. Thus, the running time of *KANN-DBSCAN* is dominated by computing the value of $MinPts_k^j$, which takes $O((n^j)^3)$ time. For all tasks, the time complexity of *KANN-DBSCAN* is $O(m(\max_{t_j \in T} n^j)^3)$.

The running time of Algorithm 2 is dominated by the while loop for sorting the workers in $\overline{W_z^j}$ (Lines 18-22), which takes $O(n^2)$ since there are at most n workers in W_z^j. Since *DATE* executes the sorting for each value of each task, and the maximal number iteration is φ, *DATE* is bounded by $O(\varphi n^2 m \max_{t_j \in T} num^j)$.

For Algorithm 3, finding the worker with minimum effective accuracy unit cost takes $O(nm)$, where computing the value of $\sum_{t_j \in T_k} \min\{\Theta^{j'}, A_k^j\}$ takes $O(m)$. Hence, the while-loop (Lines 2-8) takes$O(n^2 m)$. In each iteration of the for-loop (Lines 10-20), a process similar to Lines 2-8 is executed. Hence the time

complexity of the whole reverse auction is dominated by this for-loop, which is bounded by $O(n^3m)$. ■

Lemma 2. *IMC^2 is individually rational.*

Proof: Let i_k be worker i's replacement which appears in the ith place in the sorting over $W\backslash\{i\}$. Since worker i_k would not be at ith place if i is considered, we have $\frac{b_i}{\sum_{t_j\in T_i}\min\{\Theta^{j'},A_i^j\}} \le \frac{b_{i_k}}{\sum_{t_j\in T_{i_k}}\min\{\Theta^{j'},A_{i_k}^j\}}$. Hence $b_i \le \frac{\sum_{t_j\in T_i}\min\{\Theta^{j'},A_i^j\}}{\sum_{t_j\in T_{i_k}}\min\{\Theta^{j'},A_{i_k}^j\}}b_{i_k} = \frac{\sum_{t_j\in T_i}\min\{\Theta^{j''},A_i^j\}}{\sum_{t_j\in T_{i_k}}\min\{\Theta^{j''},A_{i_k}^j\}}b_{i_k}$, where the equality relies on the observation that $\Theta^{j'} = \Theta^{j''}$ for every $k \le i$, which is due to the fact that $S = S'$ for every $k \le i$. This is sufficient to guarantee $b_i \le \max_{k\in W\backslash S'}\frac{\sum_{t_j\in T_i}\min\{\Theta^{j''},A_i^j\}}{\sum_{t_j\in T_{i_k}}\min\{\Theta^{j''},A_{i_k}^j\}}b_{i_k} = p_i$ ■

Before analyzing the truthfulness of IMC^2, we first introduce the Myerson's Theorem [38].

Theorem 2. *An auction mechanism is truthful if and only if:*

- *The selection rule is monotone: If worker i wins the auction by bidding b_i, it also wins by bidding $b_i' < b_i$;*
- *Each winner is paid the critical value: Worker i would not win the auction if it bids higher than this value.*

Lemma 3. *IMC^2 is truthful.*

Proof: Based on Theorem 2, it suffices to prove that the selection rule of IMC^2 is monotone and the payment p_i for each i is the critical value. The monotonicity of the selection rule is obvious as bidding a lower price cannot push worker i backwards in the sorting. We next show that p_i is the critical value for worker i in the sense that bidding higher p_i could prevent worker i from winning the auction. Note that $p_i = \max_{k\in\{1,\dots,e\}}\frac{\sum_{t_j\in T_i}\min\{\Theta^{j''},A_i^j\}}{\sum_{t_j\in T_{i_k}}\min\{\Theta^{j''},A_{i_k}^j\}}b_{i_k}$. If worker i bids $b_i \ge p_i$, it will be placed after e since $b_i \ge \frac{\sum_{t_j\in T_i}\min\{\Theta^{j''},A_i^j\}}{\sum_{t_j\in T_{i_e}}\min\{\Theta^{j''},A_{i_e}^j\}}b_{i_e}$ implies $\frac{b_i}{\sum_{t_j\in T_i}\min\{\Theta^{j''},A_i^j\}} \ge \frac{b_{i_e}}{\sum_{t_j\in T_{i_e}}\min\{\Theta^{j''},A_{i_e}^j\}}$. Hence, worker i would not win the auction because the first e workers have met the accuracy requirement for each task in T. ■

Then, we provide our analysis about the approximation ratio of IMC^2 using the dual fitting method [39]. The normalized primal linear program **P** has been formulated in equation (4)∼(6). The dual program **D** is formulated in equation (24)∼(27).

$$\textbf{D:}\ \max \sum\nolimits_{t_j\in T}\Theta^j y_j - \sum\nolimits_{i\in W} z_i \tag{24}$$

$$s.t.\ \sum\nolimits_{t_j\in T_i}(A_i^j y_j) - z_i \le b_i,\ \forall i \in W \tag{25}$$

$$y_j \ge 0,\ \forall t_j \in T \tag{26}$$

$$z_i \ge 0,\ \forall i \in W \tag{27}$$

We define any task $t_j \in T$ as alive at any iteration in winner selection phase if its accuracy requirement is not fully satisfied. We define that task t_j is covered by T_i if $t_j \in T_i$ and t_j is alive when worker i is selected. The coverage relationship is represented as $t_j \prec T_i$. Moreover, we define the minimum accuracy as Δv. Suppose when worker i is selected, the residual accuracy requirement profile is $\{\Theta^{1^*},\Theta^{2^*},...,\Theta^{m^*}\}$ and T_i is the $i_j th$ set that covers t_j, the corresponding normalized effective accuracy unit cost in terms of unit accuracy can be represented in equation (28):

$$w(t_j,i_j) = \frac{b_i\Delta v}{\sum_{t_j\in T_i}\min\{\Theta^{j^*},A_i^j\}} \tag{28}$$

We assume that t_j is covered by h_j sets. Then we have $w(t_j,1) \le ... \le w(t_j,h_j)$. We then define two constants $\Omega = \frac{1}{\Delta v}\sum_{t_j\in T}\Theta^j$and $\varepsilon = \max A_i^j\cdot|T_i|\cdot b_i,\ i\in W,\ t_j\in T$.

Lemma 4: *The following pairs $(y_j,\ z_i)$, $t_j \in T$, $i \in W$ are feasible to the dual program **D**.*

$$y_j = \frac{w(t_j,h_j)}{2\varepsilon H_n\Delta v}, \forall t_j \in T,$$

$$z_i = \begin{cases} \frac{\sum_{t_j\prec T_i}\left(\min\{\Theta^{j*},A_i^j\}(w(t_j,h_j)-w(t_j,i_j))\right)}{2\varepsilon H_\Omega\Delta v}, i\in S \\ 0, i\notin S \end{cases}$$

where $H_n = 1+\frac{1}{2}+...+\frac{1}{n}$, $H_\Omega = 1+\frac{1}{2}+...+\frac{1}{\Omega}$.

Proof: Suppose for any worker $i \in W$, there are s_i tasks in T_i. We reorder these tasks in the order in which they are fully covered.

If $i \notin S$, then we have $z_i = 0$. Suppose when the last unit accuracy requirement of t_j is covered, the residual accuracy requirement profile is $\{\Theta^{1+},\Theta^{2+},\ ...,\ \Theta^{m+}\}$, then the total residual accuracy requirement of alive tasks contained by T_i are represented as $\sum_{k=j}^{s_i}\min\{\Theta^{k+},A_k^j\}$. We have

$$w(t_j,h_j) \le \frac{b_i\Delta v}{\sum_{k=j}^{s_i}\min\{\Theta^{k+},A_i^k\}}$$

Therefore, we have

$$\begin{aligned}\sum\nolimits_{j=1}^{s_i}(v_i(t_j)y_j) - z_i &\le \sum\nolimits_{j=1}^{s_i}\frac{v_i(t_j)b_i}{2\varepsilon H_\Omega\sum_{k=j}^{s_i}\min\{\Theta^{k+},A_i^k\}} - 0 \\ &\le \frac{b_i}{H_\Omega}\left(1+\frac{1}{2}+...+\frac{1}{\Omega}\right) \le b_i\end{aligned}$$

If worker $i \in S$, then we assume that when worker $i$is selected as a winner, s_i' tasks in T_i already been fully covered. We have

$$\begin{aligned}&\sum\nolimits_{j=1}^{s_i}(A_i^j y_j) - z_i \\ =&\frac{\sum_{j=1}^{s_i}(w(t_j,h_j)A_i^j)}{2\varepsilon H_\Omega\Delta v} - \frac{\sum_{j=s_i'+1}^{s_i}\min\{\Theta^{j*},A_i^j\}\left(w(t_j,h_j)-w(t_j,i_j)\right)}{2\varepsilon H_\Omega\Delta v} \\ =&\frac{\sum_{j=1}^{s_i'}(w(t_j,h_j)A_i^j)}{2\varepsilon H_\Omega\Delta v} + \frac{\sum_{j=s_i'+1}^{s_i}\min\{\Theta^{j*},A_i^j\}w(t_j,i_j)}{2\varepsilon H_\Omega\Delta v} \\ &+\frac{\sum_{j=s_i'+1}^{s_i}A_i^j - \min\{\Theta^{j*},A_i^j\}w(t_j,h_j)}{2\varepsilon H_\Omega\Delta v} \\ \le&\frac{\sum_{j=1}^{s_i'}(w(t_j,h_j)A_i^j)}{2\varepsilon H_\Omega\Delta v} + \frac{\sum_{j=s_i'+1}^{s_i}\min\{\Theta^{j*},A_i^j\}w(t_j,i_j)}{2\varepsilon H_\Omega\Delta v} \\ =&\sum\nolimits_{j=1}^{s_i'}\frac{A_i^j b_i}{2\varepsilon H_\Omega\sum_{k=j}^{s_i}\min\{\Theta^{k*},A_i^k\}} + \frac{b_i}{2\varepsilon H_\Omega} \\ \le&\frac{b_i}{H_\Omega}\left(\frac{1}{s_i}+...+\frac{1}{s_i-s_i'+1}+1\right) \le b_i\end{aligned}$$

Hence, the pairs $(y_j,\ z_i)$, $t_j \in T, i \in W$ are feasible to the dual program **D**. ■

Lemma 5: *IMC^2 can approximate the optimal solution within a factor of $2\varepsilon H_\Omega$, where $H_\Omega = 1+\frac{1}{2}+...+\frac{1}{\Omega}$.*

Proof: By substituting the dual solution given in Lemma 4 into equation (24), we have

$$\begin{aligned}&\sum\nolimits_{t_j\in T}\Theta^j y_j - \sum\nolimits_{i\in W} z_i \\ =&\frac{\sum_{i\in S}\sum_{t_j\prec T_i}\left(\min\{\Theta^{j*},A_i^j\}\left(w(t_j,h_j)-w(t_j,i_j)\right)\right)}{2\varepsilon H_\Omega\Delta v} \\ &+\frac{\sum_{t_j\in T}\Theta^j w(t_j,h_j)}{2\varepsilon H_\Omega\Delta v} \\ =&\frac{\sum_{i\in S}\sum_{t_j\prec T_i}\min\{\Theta^{j*},A_i^j\}\frac{b_i\Delta v}{\sum_{t_j\in T_i}\min\{\Theta^{j*},A_i^j\}}}{2\varepsilon H_\Omega\Delta v} = \frac{\sum_{i\in S}b_i}{2\varepsilon H_\Omega} \le OPT\end{aligned}$$ ■

The above lemmas together prove the following theorem.

Theorem 3. *IMC^2 is computationally efficient, individually rational, truthful and $2\varepsilon H_\Omega$ approximate.*

8 Performance Evaluation

8.1 Simulation Setup

We first measure the performance of *DATE* and *S-DATE*, and compare it with following three bench mark algorithms:

- *MV* (*Majority Voting* [8]): The truth of each task is the corresponding value that supported by the most workers.
- *ED* (*Enumerate all workers' Dependence*): Enumerate all possible dependence for each worker with others when calculating the probability of providing each possible value independently (Step 2 of *DATE*).
- *NC* (*No Copier* [40]): Consider all workers are independent. All calculations about dependence are not needed in *NC*. This means that *NC* only includes Step 3 of *DATE*. In this case, *DATE* (*S-DATE*) is simplified to follow classic *CRH* [40] framework actually.

Note that we have enhanced *MV* and *NC* by including the stage of contextual embedding and clustering to make them applicable to text data. The *precision* of the truth discovery is calculated as $\frac{\sum_{t_j \in T} g(et^j = et^{*j})}{|T|}$, where et^{*j} is the real truth of task t_j. $g(et^j = et^{*j}) = 1$ if $et^j = et^{*j}$; otherwise, $g(et^j = et^{*j}) = 0$.

Then, we conduct the simulations to evaluate the *Reverse Auction,* and compare it with following algorithms:

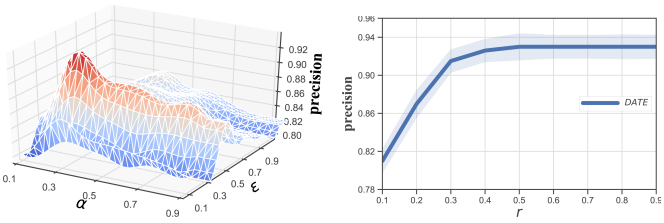
- *GA* (*Greedy Accuracy*): Selects the worker with the highest accuracy greedily, and pays the critical value [38].
- *GB* (*Greedy Bid*): Selects the worker with the lowest bid price greedily, and follows the *Vickrey* payment rule [41].

We set $\varphi = 100$. All the simulations are run on a Centos 7 machine with Intel(R) Xeon(R) CPU E5-2630 2.6GHz and 128 GB memory. Each measurement is averaged over 100 instances.

8.2 Evaluation of *DATE*

If the crowdsourcing answers are numeric data or the choices from predefined options, we can use the *DATE* to calculate the truth of tasks and the accuracy of workers. For *DATE*, we use the data from *Qatar Living Forum* [42] to simulate the crowdsourcing network. The data was collected from survey participants using *Qatar Living Forum* in 2015. It includes 300 questions, 120 workers and 6000 comments. Each comment can be annotated as *"Good"*, *"Bad"* or *"Other"*. The default number of tasks and workers are 300 and 120, respectively. In the simulations, we randomly choose 30 workers from 120 workers as the copiers, and all data of each copier is copied from random one of other 90 workers.

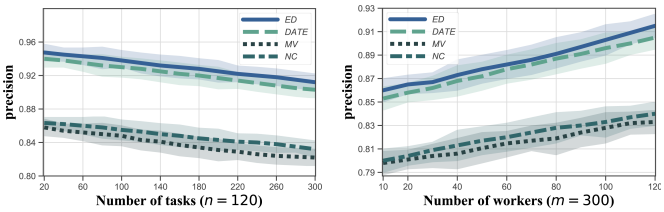
Then, we attempt to find the best setting of ε, α, and r for *DATE* based on data set [43]. We fix $r = 0.2$ as a default value, and vary both ε and α from 0.1 to 0.9. Fig. 4(a) shows that the *precision* fluctuates between 0.82 and 0.92. In our simulations, we set $\alpha = 0.2$ and $\varepsilon = 0.5$ since this setting can obtain the highest *precision* of 0.92. As shown in Fig. 4(b), the *precision* increases significantly when we increase r from 0.1 to 0.4. The *precision* becomes convergent when r is more than 0.4. The setting of r may be influenced largely by the data set adopted, especially, the number of copiers. We set $r = 0.4$ in our simulations.



(a) Precision versus ε, α (b) Precision versus r

Fig. 4. Impact of parameters of *DATE* on *precision*

Fig. 5 compares the *precision* achieved by the *DATE* against the benchmark algorithms. *DATE* can calculate the workers' dependence, thereby obtaining higher *precisions* (more than 0.85 in all cases) than those of *MV* and *NC* (with average improvement 8.4% and 7.4%, respectively). The *precisions* of *DATE* are very close to *ED*. *ED* outperforms *DATE* (with average improvement 0.8%) since it enumerates all possible dependence for each worker with others when calculating the probability of providing each possible value independently. However, as we show later, *ED* takes much more running time than *DATE*. Based on the results of Fig. 5(a), the *precision* decreases when the task increases. In our simulations, we select the tasks based on the index in the increasing order from the data set. In the adopted data set, the tasks with small indices are performed by more workers. This means that fewer values can be used to estimate the truth for the later tasks. Therefore, the *precision* decreases slightly when the number of tasks increases. From Fig. 5(b), we can see that all algorithms obtain the higher *precisions* when the worker increases. This is because the algorithms can estimate the truth from more responses for the tasks.



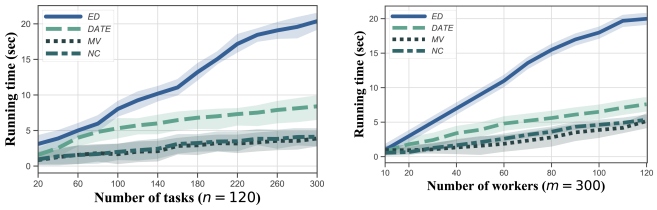
(a) Precision versus tasks (b) Precision versus workers

Fig. 5. Precision of *DATE* with different number of tasks and workers

Fig. 6 depicts the running time of all algorithms. It can be seen that the running time of all algorithms increase with the increase of both tasks and workers. Intuitively, the running time of *ED* increases faster than other algorithms since *ED* calculates all possible dependencies of workers, which leads to the complexity of exponential time. For the setting $n = 120$, $m = 300$, our *DATE* only takes 42.6% of running time comparing with *ED*.

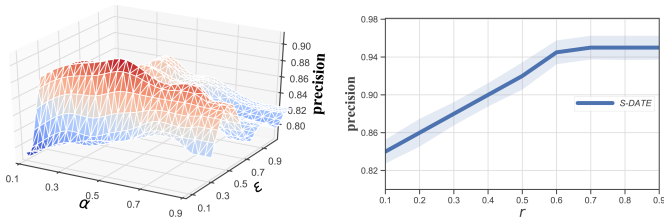
8.3 Evaluation of *S-DATE*

Though the truth discovery framework of *S-DATE* is same as *DATE*, they are designed to process different types of crowdsourcing data. Specifically, *DATE* is ineffective for textual answers. On the other hand, there is no need to use *S-DATE* to process the numeric crowdsourcing data. Thus, we use different data sets for *DATE* and *S-DATE* in the simulations. For *S-DATE*, we use the data from *Consumer Reviews of Amazon Products* [44] to simulate the crowdsourcing network. It includes 130 questions, 300 workers



(a) Running time versus tasks (b) Running time versus workers

Fig. 6. Running time of *DATE*



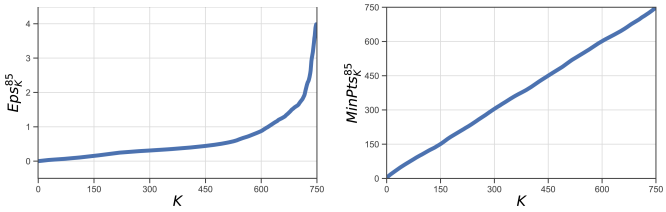
(a) Precision versus ε, α (b) Precision versus r

Fig. 10. Impact of parameters of *S-DATE* on *precision*

and 34660 comments. The default number of tasks and workers are 130 and 300, respectively. In the simulations, we randomly choose 50 workers from 300 workers as the copiers, and all data of each copier is copied from random one of other 250 workers.

We use the *BERT* model [45] trained by Google, with a parameter size of 110M. Before getting the sentence vector, we fine-tuned *BERT* on our crowdsourced dataset. The number of heads of the multi-head mechanism is 12. The number of *Transformer* block is 12. The activation function is gelu [43]. The embedding dimension is 768. The max sequence length is 128. The batch size is 32. Adam [46] optimization method is used for 100 iteration training. The learning rate is 2e-5.

The key of *KANN-DBSCAN* is to determine the parameters *Eps* and *MinPts*. We randomly select a task No.85 with 750 comments, and test different values of K (the index of sorting in line 8 of Algorithm 3, $K = 1, 2, ..., 750$). As seen from Fig. 7, the values of Eps_K^{85} and $MinPts_K^{85}$ increase steadily with K.



(a) Impact of K on *Eps* (b) Impact of K on *MinPts*

Fig. 7. Impact of K on *Eps* and *MinPts*

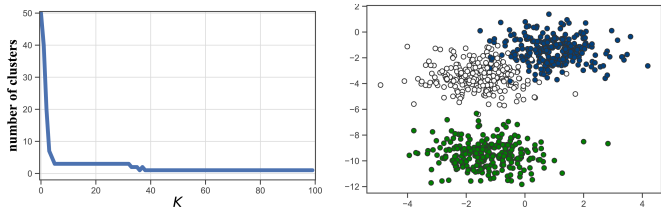


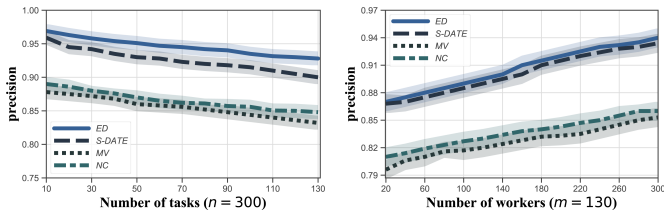
Fig. 8. Impact of K on clustering

Fig. 9. *KANN-DBSCAN* clusters

The relationship between the number of clusters and the K value is shown in Fig. 8. It can be seen that the number of clusters is stable from $K = 6$ to $K = 33$. Based on Algorithm 1, we choose the maximum of K as the optimal value ($K = 33$). Then the corresponding values of *Eps* and *MinPts* can be determined: *Eps*= 0.019, *MinPts*= 44.077. The number of clusters is 3. The final clustering result of this task is shown in Fig. 9, and the F1 score is 0.843.

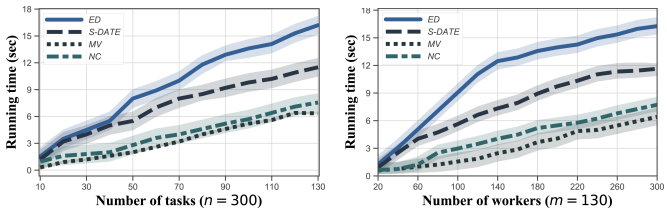
For *S-DATE*, we conduct the similar tests based on data set [43] to find the best settings of ε, α and r, and the results are shown in Fig. 10. We set $\alpha = 0.5$, $\varepsilon = 0.3$ and $r = 0.6$ for *S-DATE*.

We conduct the similar simulations to measure the performance of *S-DATE*. Fig. 11 compares the *precision* achieved by the *S-DATE* against the benchmark algorithms. *S-DATE* can obtain higher *precisions* (more than 0.87 in all cases) than those of *MV* and *NC* (with average improvement 8.1% and 6.9%, respectively). *ED* outperforms *S-DATE* (with average improvement 2.1%). Fig. 12 depicts the running time of all algorithms. *S-DATE* only takes 40.8% of the running time of *ED* when $n = 300$, $m = 130$.



(a) Precision versus tasks (b) Precision versus workers

Fig. 11. Precision of *S-DATE* with different number of tasks and workers



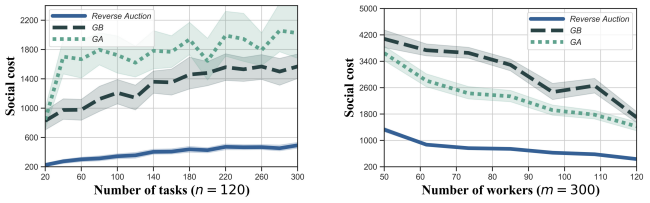
(a) Running time versus tasks (b) Running time versus workers

Fig. 12. Running time of *S-DATE*

8.4 Evaluation of Reverse Auction

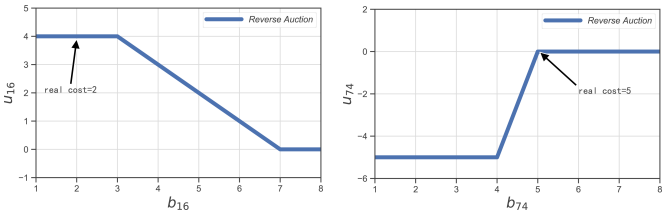
The *Reverse Auction* is based on *S-DATE*. The cost of each worker is selected randomly from the auction dataset [47], which contains 5017 bid prices for *Palm Pilot M515 PDA* from *eBay* workers. The accuracy requirement of each task is uniformly over [2, 4]. The value of each task is uniformly distributed over [5, 8]. We will vary the value of the key parameters to explore the impacts on designed incentive mechanism.

Fig. 13 depicts the social cost of *Reverse Auction*, *GA* and *GB* with different number of tasks and workers. Overall, the social cost increases with increasing tasks since more workers will be selected as winners in order to complete the tasks. On the contrary, the social cost decreases with increasing workers. This is because we can find more workers with high-accuracy and lower bid price to perform the same task. The *Reverse Auction* can obtain the lowest social cost comparing with *GA* and *GB* (with average decrease of



(a) Social cost versus tasks (b) Social cost versus workers

Fig. 13. Social cost

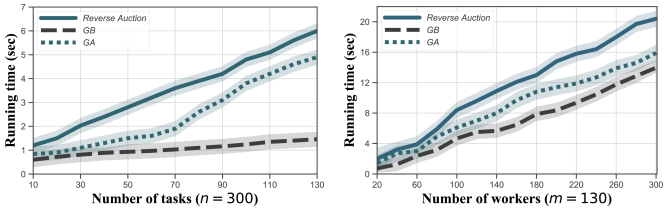


(a) Utility of worker 16 (winner) (b) Utility of worker 74 (loser)

Fig. 17. Truthfulness of *IMC*^{2}

76.2% and 68.4%, respectively) since *Reverse Auction* can output the social cost with guaranteed approximation.

From Fig. 14, we can see that the running time of *Reverse Auction*, *GA*, *GB* increase with the increase both of tasks and workers. This is consistent with our time analysis in Lemma 1. It is not difficult to obtain the time complexity $O(n^3)$ of *GA* and $O(n^2)$ of *GB*, respectively, of which both are lower than $O(n^3m)$ of *Reverse Auction*. Thus, the running time of *GA* and *GB* is lower than *Reverse Auction*.



(a) Running time versus tasks (b) Running time versus workers

Fig. 14. Running time of *Reverse Auction*

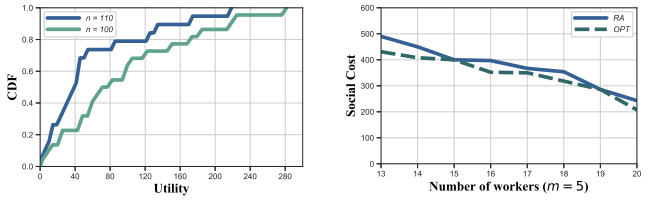


Fig. 15. CDF of utilities

Fig. 16. Comparison with *OPT*

Fig. 15 shows the cumulative distribution function of workers' utilities. We can see that all workers have the nonnegative utilities, verifying the individual rationality of *IMC*^{2}. Specifically, the average utility will decrease if the number of workers increases. This is because the competition among workers increases and the payment will decrease when there are more workers. To show the approximation of *IMC*^{2}, we conduct the small-scale simulations with 5 tasks and at most 20 workers. As shown in Fig. 16, the social cost of *Reverse Auction* is very close to the optimal solution of *SOAC* problem (with average increase of 8.5%).

We verify the truthfulness of *IMC*^{2} by randomly picking two workers and allowing them to bid prices that are different from their true costs. We illustrate the results in Fig. 17. We can see that Worker 16 always obtain its maximum utility of 4 if bidding its real cost $c_{16} = 2$. Accordingly, the loser 74 always obtains nonnegative utility if he/she bids truthfully.

Summary: In the coping environment, our *DATE* can improve the *precision* by 8.4% and 7.4% comparing with *Majority Voting* and *CRH* framework, respectively. For the textual answers, *S-DATE* can improve the *precision* by 8.1% and 6.9% comparing with *Majority Voting* and *CRH* framework, respectively. The *Reverse Auction* can obtain the lowest social cost comparing with *GA* and *GB* (with average decrease of 76.2% and 68.4%, respectively). Moreover, *IMC*^{2} can achieve computational efficiency, individual rationality, truthfulness and guaranteed approximation.

8.5 Case Study

We give example to illustrate how the *IMC*^{2} works. We extract a small data subset from Consumer Reviews of Amazon Products [44] as the case study. There are five goods with accuracy requirement profile $\mathbf{\Theta} = (0.8, 0.9, 0.1, 0.3, 0.8)$ and four customers with bid prices $b_1 = 2$, $b_2 = 3$, $b_3 = 2$ and $b_4 = 2.5$. The tasks (products), workers (customers), review topics, and review texts are shown in Table 3. In this example, customer 4 copied some answers from customer 3 (with certain errors during copying).

Stage 1: Contextual Embedding and Clustering

The values of texts after clustering are given in Table 4.

Stage 2: Truth Discovery

The accuracies of customers in *S-DATE* as well as the estimated truth of *S-DATE*, *ED*, *MV* and *NC* are given in Table 5. According to the estimated truth, the *precisions* of *S-DATE*, *ED*, *MV* and *NC* are 0.8, 0.8, 0.4~0.8 and 0.6, respectively. We can see that *S-DATE* and *ED* can obtain the highest *precision* among all comparison algorithms. If more customers are provided, the *precisions* of *S-DATE* and *ED* will be higher.

Stage 3: Reverse Auction

Winner Selection:

For convenience, we denote the effective accuracy unit cost of any customer i over winner set S as $b_i(S)$.

Round 1: $S = \varnothing$, $\mathbf{\Theta}' = (0.8, 0.9, 0.1, 0.3, 0.8)$

$$b_1(S) = \frac{2}{0.8+0.9+0.1+0.012+0.8} \approx 0.766$$
$$b_2(S) = \frac{3}{0.8+0.002+0.1+0.3+0.8} \approx 1.499$$
$$b_3(S) = \frac{2}{0.003+0.9+0.011+0.3+0.058} \approx 1.572$$
$$b_4(S) = \frac{2.5}{0.003+0.002+0.011+0.3+0.058} \approx 6.684$$

Round 2: $S = \{1\}$, $\mathbf{\Theta}' = (0, 0, 0, 0.288, 0)$

$$b_2(S) = \frac{3}{0.288} \approx 10.417,\ b_3(S) = \frac{2}{0.288} \approx 6.944,$$
$$b_4(S) = \frac{2.5}{0.288} \approx 8.681,\ \text{thus } S = \{1, 3\}.$$

Payment Determination:

TABLE 3
A case study for IMC^2

Method	Product	Topic	Review text			
			Customer 1	Customer 2	Customer 3	Customer 4 (copier)
1	Fire HD 8	function	almost perfect	almost perfect	has limitations	has limitations
2	Amazon Echo Show	price	great price	affordable price	great price	expensive
3	Fire Tablet	quality	best e-book	good, maybe not great	not good	not good
4	Amazon Kindle	size	too small	suitable size	love the size	love the size
5	Kids Edition Tablet	suitableness	works so well for her	great for kids	poor for kids	poor for kids

TABLE 4
Values of review texts after clustering

Product	Customer				Number of values
	1	2	3	4	
1	A	A	B	B	2
2	A	B	A	C	3
3	A	B	C	C	3
4	B	A	A	A	2
5	A	A	B	B	2

For winner 1, winners are 2 and 3.

$$\boldsymbol{\Theta}' = (0.8, 0.9, 0.1, 0.3, 0.8)$$
$$\frac{0.8+0.9+0.1+0.012+0.8}{0.8+0.002+0.1+0.3+0.8} \times b_2 \approx 3.914$$
$$\boldsymbol{\Theta}' = (0, 0.898, 0, 0, 0)$$
$$\frac{0.898}{0.898} \times b_3 = 2, \text{ thus } p_1 = \max\{3.914, 2\} = 3.914.$$

For winner 3, winners are 1 and 4.

$$\boldsymbol{\Theta}' = (0.8, 0.9, 0.1, 0.3, 0.8)$$
$$\frac{0.003+0.9+0.011+0.3+0.058}{0.8+0.9+0.1+0.012+0.8} \times b_1 \approx 0.974$$
$$\boldsymbol{\Theta}' = (0, 0, 0, 0.288, 0)$$
$$\frac{0.288}{0.288} \times b_4 = 2.5, \text{ thus } p_3 = \max\{0.974, 2.5\} = 2.5.$$

9 Conclusion

In this paper, we have designed a three-stage incentive mechanism for truth discovery in crowdsourcing with copiers. In the contextual embedding and clustering stage, we construct and cluster the content vector representations of crowdsourced data to support the textual answers of crowdsourcing. In the truth discovery stage, we calculate the dependence for each pair of workers based on the Bayesian analysis and estimate the truth for each task based on both the dependence and accuracy of workers. In the reverse auction stage, we develop a greedy algorithm to maximize the social welfare such that all tasks can be completed with the least confidence for truth discovery. We have demonstrated that the proposed incentive mechanism achieves computational efficiency, individual rationality, truthfulness, and guaranteed approximation. Moreover, our truth discovery methods show prominent advantage in terms of precision.

References

[1] K. Tahboub, N. Gadgil, J. Ribera, B. Delgado and E. Delp, "An Intelligent Crowdsourcing System for Forensic Analysis of Surveillance Video," in *Proc. SPIE*, 2015, pp. 940701.

[2] J. Fan, M. Lu, B. Ooi, W. Tan and M. Zhang. "A Hybrid Machine-crowdsourcing System for Matching Web Tables," in *Proc. ICDE*, 2014, pp. 976–987.

[3] Smartcitizen. (2020). [Online]. Available: http://www.smartcitizen.me

[4] R. Toris, D. Kent and S. Chernova, "The Robot Management System: A Framework for Conducting Human-robot Interaction Studies through Crowdsourcing," *Journal of Human-Robot Interaction*, vol. 3, pp. 25–49, 2014.

[5] L. Jiang, X. Niu, J. Xu, D. Yang and L. Xu, "Incentivizing the Workers for Truth Discovery in Crowdsourcing with Copiers," in *Proc. ICDCS*, 2019, pp. 1286-1295.

[6] W. Faulkner, (2020). [Online]. Available: https://www.quotes.net/authors/William+Faulkner.

[7] M. Musthag, A. Raij, D. Ganesan, S. Kumar and S. Shiffman, "Exploring Micro-incentive Strategies for Participant Compensation in High-burden Studies," in *Proc. UbiComp*, 2011, pp. 435–444.

[8] Majority. (2020). [Online]. Available: https://en.wikipedia.org/wiki/Majority

[9] X. Dong, L. Berti-Equille and D. Srivastava, "Truth Discovery and Copying Detection in a Dynamic World," in *Proc. VLDB Endowment*, pp. 562–573, 2009.

[10] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao and K. Ren, "Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems," in *Proc. SenSys*, 2015, pp. 183–196.

[11] X. Tang, C. Wang, X. Yuan and Q. Wang, "Non-Interactive Privacy-Preserving Truth Discovery in Crowd Sensing Applications," in *Proc. INFOCOM*, 2018, pp. 1988-1996.

[12] M. Xiao, J. Wu, S. Zhang and J. Yu, "Secret-sharing-based Secure User Recruitment Protocol for Mobile Crowdsensing," in *Proc. INFOCOM*, 2017, pp. 1-9.

[13] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang and G. Chen, "On Designing Data Quality-Aware Truth Estimation and Surplus Sharing Method for Mobile Crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832-847, 2017.

[14] H. Jin, L. Su and K. Nahrstedt, "CENTURION: Incentivizing Multi-requester Mobile Crowd Sensing," in *Proc. INFOCOM*, 2017, pp. 1-9.

[15] Y. Wang, K.Wang and C. Miao, "Truth Discovery against Strategic Sybil Attack in Crowdsourcing," in *Proc. KDD*, 2020, pp. 95-104.

[16] Y. Li, H. Sun and W. Wang, "Towards Fair Truth Discovery from Biased Crowdsourced Answers," in *Proc. KDD*, 2020, pp. 599-607.

[17] D. Wang, J. Ren, Z. Wang, X. Pang, Y. Zhang and X. Shen, "Privacy-preserving Streaming Truth Discovery in Crowdsourcing with Differential Privacy," *IEEE Transactions on Mobile Computing*, 2021, doi: 10.1109/TMC.2021.3062775.

[18] P. Sun, Z. Wang, Y. Feng, L. Wu, Y. Li, H. Qi and Z. Wang, "Towards Personalized Privacy-preserving Incentive for Truth Discovery in Crowdsourced Binary-choice Question Answering," in *Proc. INFOCOM*, 2020, pp. 1133-1142.

[19] H. Jin, L. Su, H. Xiao and K. Nahrstedt, "INCEPTION: Incentivizing Privacy-Preserving Data Aggregation for Mobile Crowd Sensing Systems," in *Proc. MobiHoc*, 2016, pp. 341-350.

[20] H. Wang, S. Guo, J. Cao and M. Guo, "MeLoDy: A Long-term Dynamic Quality-aware Incentive Mechanism for Crowdsourcing," in *Proc. ICDCS*, 2017, pp. 933-943.

[21] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng and X. Shen, "Quality-driven Auction based Incentive Mechanism for Mobile Crowd Sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203-4214, 2014.

[22] H. Jin, L. Su, D. Chen, K. Nahrstedt and J. Xu, "Quality of Information Aware Incentive Mechanisms for Mobile Crowd Sensing Systems," in *Proc. MobiHoc*, 2015, pp. 167–176.

[23] J. Xu, Z. Rao, L. Xu, D.Yang and T. Li, "Incentive Mechanism for Multiple Cooperative Tasks with Compatible Users in Mobile Crowd Sensing

TABLE 5
Estimated truth and accuracy

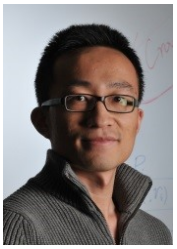
Product	Estimated truth				Ground truth	Accuracy of *S-DATE*			
	S-DATE	*ED*	*MV*	*NC*		Customer 1	Customer 2	Customer 3	Customer 4 (copier)
1	A	A	A or B	B	A	0.997	0.997	0.003	0.003
2	A	A	A	A	A	0.998	0.002	0.998	0.002
3	A	A	C	C	B	0.831	0.158	0.011	0.011
4	A	A	A	A	A	0.012	0.988	0.988	0.988
5	A	A	A or B	A	A	0.942	0.942	0.058	0.058

via Online Communities," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1618-1633, 2020.
[24] X. Miao, Y. Kang, Q. Ma, K. Liu and L. Chen, "Quality-aware Online Task Assignment in Mobile Crowdsourcing," *ACM Transactions on Sensor Networks*, vol. 16, no. 3, pp. 1-21, 2020.
[25] H. Gao, C. Liu, J. Tang, D. Yang, P. Hui and W. Wang, "Online Quality-aware Incentive Mechanism for Mobile Crowd Sensing with Extra Bonus," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2589-2603, 2018.
[26] J. Xu, J. Xiang and D. Yang, "Incentive Mechanisms for Time Window Dependent Tasks in Mobile Crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, 2015.
[27] D. Yang, G. Xue, X. Fang and J. Tang, "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing," in *Proc. MobiCom*, 2012, pp. 173–184.
[28] J. Xu, H. Li, Y. Li, D. Yang and T. Li, "Incentivizing the Biased Requesters: Truthful Task Assignment Mechanisms in Crowdsourcing," in *Proc. SECON*, 2017, pp. 1-9.
[29] J. Xu, C. Guan, H. Wu, D. Yang, L. Xu and T. Li, "Online Incentive Mechanism for Mobile Crowdsourcing based on Two-tiered Social Crowdsourcing Architecture," in *Proc. SECON*, 2018, pp. 1-9.
[30] J. Xu, C. Guan, H. Dai, D. Yang, L. Xu and J. Kai, "Incentive Mechanisms for Spatio-temporal Tasks in Mobile Crowdsensing", in *Proc. IEEE MASS*, 2019, pp.55-63.
[31] J. Devlin, M. Chang and K. Lee, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[32] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, "Improving language Understanding with Unsupervised Learning," *Technical report, OpenAI*, 2018.
[33] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep Contextualized Word Representations," *arXiv preprint arXiv:1802.05365*, 2018.
[34] W. Li, S. Yan, Y. Jiang, S. Zhang and C. Wang, "Research on Method of Self-Adaptive Determination of DBSCAN Algorithm Parameters," *Computer Engineering and Applications*, vol. 55, no. 5, pp.1-7, 2019.
[35] V. Ashish, S. Noam and P. Niki, "Attention Is All You Need," in *Proc. NIPS*, 2017, pp. 5998-6008.
[36] U. Feige, "A Threshold of ln n for Approximating Set Cover," *Journal of the ACM*, vol. 45, pp. 634-652, 1998.
[37] S. De and R. Vohra, "Combinatorial Auctions: A survey," *INFORMS Journal on computing*, vol. 15, no. 3, pp. 284-309, 2003.
[38] R. Myerson, "Optimal Auction Design," *Discussion Papers*, vol. 6, no. 1, pp. 58–73, 1978.
[39] S. Rajagopalan, and V. Vazirani, "Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs," in *Proc. FOCS*, 1993, pp. 322-331.
[40] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan and J. Han, "Resolving Conflicts in Heterogeneous Data by Truth Discovery and Source Reliability Estimation," in *Proc. ACM SIGMOD*, 2014, pp. 1187-1198.
[41] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," *The Journal of finance*, vol.16, no. 1, pp. 8–37, 1961.
[42] SemEval-2015 Task 3. (2020). [Online]. Available: http://alt.qcri.org/semeval2015/task3
[43] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
[44] Consumer Reviews of Amazon Products. (2020). [Online]. Available: https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products
[45] Google-research Bert. (2020). [Online]. Available: https://github.com/google-research/bert
[46] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[47] Modeling online auctions. (2020). [Online]. Available: http://www.modelingonlineauctions.com/datasets

Lingyun Jiang received the Ph.D. degree in Internet of things college from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2017. She is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. Her main research interests include crowdsourcing, opportunistic networks and wireless sensor networks.

Xiaofu Niu received the bachelor's degree in Network Engineering from Hefei University, Hefei, China, in 2017. She is currently pursuing the master's degree in Nanjing University of Posts and Telecommunications. Her research interests are mainly in the mobile crowdsensing and security multiple-party computation.

Jia Xu (*M'15*) received the PhD. degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks.



Dejun Yang (*SM'19*) received the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. Currently, he is an Associate Professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security and privacy problems.

Lijie Xu received his Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Nanjing, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong, from 2011 to 2012. He is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. His research interests include mobile and distributed computing and graph theory algorithms.