

Achieving Approximate Global Optimization of Truth Inference for Crowdsourcing Microtasks

Lizhen Cui^{1,2} Jing Chen¹ · Wei He^{1,2} · Hui Li^{1,2} · Wei Guo^{1,2} · Zhiyuan Su³

Received: 20 January 2021 / Revised: 12 April 2021 / Accepted: 21 April 2021 / Published online: 12 May 2021 © The Author(s) 2021

Abstract

Microtask crowdsourcing is a form of crowdsourcing in which work is decomposed into a set of small, self-contained tasks, which each can typically be completed in a matter of minutes. Due to the various capabilities and knowledge background of the voluntary participants on the Internet, the answers collected from the crowd are ambiguous and the final answer aggregation is challenging. In this process, the choice of quality control strategies is important for ensuring the quality of the crowdsourcing results. Previous work on answer estimation mainly used expectation-maximization (EM) approach. Unfortunately, EM provides local optimal solutions and the estimated results will be affected by the initial value. In this paper, we extend the local optimal result of EM and propose an approximate global optimal algorithm for answer aggregation of crowdsourcing microtasks with binary answers. Our algorithm is expected to improve the accuracy of real answer estimation through further likelihood maximization. First, three worker quality evaluation models are presented based on static and dynamic methods, respectively, and the local optimal results are obtained based on the maximum likelihood estimation method. Then, a dominance ordering model (DOM) is proposed according to the known worker responses and worker categories for the specified crowdsourcing task to reduce the space of potential task-response sequence while retaining the dominant sequence. Subsequently, a Cut-point neighbor detection algorithm is designed to iteratively search for the approximate global optimal estimation in a reduced space, which works on the proposed dominance ordering model (DOM). We conduct extensive experiments on both simulated and real-world datasets, and the experimental results illustrate that the proposed approach can obtain better estimation results and has higher performance than regular EM-based algorithms.

Keywords Microtask crowdsourcing · Quality management · Optimization strategy · Maximum likelihood estimation

Wei He hewei@sdu.edu.cn

> Lizhen Cui clz@sdu.edu.cn

Jing Chen 201934822@mail.sdu.edu.cn

Hui Li lih@sdu.edu.cn

Wei Guo guowei@sdu.edu.cn

Zhiyuan Su suzhiyuan@inspur.com

- ¹ School of Software, Shandong University, Jinan, China
- ² Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China
- ³ State Key Laboratory of High-end Server & Storage Technology, Beijing, China

1 Introduction

In recent years, crowdsourcing has attracted extensive attention from both the industry and academia. It can help solve tasks that are intrinsically easier for humans than for computers by leveraging the intelligence of a large group of people [1]. Such tasks are usually intelligent and computer-hard, which cannot be effectively addressed by existing machine-based approaches, such as entity resolution, sentiment analysis, image recognition, and so on [2]. Currently, there are many successful crowdsourcing platforms, such as Upwork, Crowdflow, and Amazon Mechanical Turk (AMT). In the crowdsourcing platforms, requesters can publish tasks, which are accepted and performed by workers. Crowdsourcing tasks are classified from multiple dimensions in some current studies. Bhatti et al. [3] classify tasks as micro-, complex, creative, and macro-tasks. Microtasks can be performed in short amounts of time by individual workers. Complex tasks require skills, knowledge, and computational efforts to solve a problem and usually can be decomposed into smaller sub-tasks. Creative tasks are related to idea generation, creative design, or co-creation. Finally, macrotasks are non-decomposable tasks and cannot be divided into smaller subproblems, require expert knowledge, skills and often involve collaboration among workers. In this paper, we focus on microtasks, which are independent and can be completed by a single worker in short amounts of time. These tasks have fine granularity and do not require workers to have specific expertise. There are many examples of microtask crowdsourcing, for example, translating text fragments, reading verification codes, rating, and ranking. Microtasks are simple for humans but difficult for computers. In the actual microtask crowdsourcing system, workers are not completely reliable, they may make mistakes or deliberately submit wrong answers. To obtain a more valid answer, tasks are usually assigned to more than one worker, each of whom performs the task independently (called a redundancy-based strategy). The answers given by different workers are then aggregated to infer the correct answer (called truth) of each task. This is a fundamental problem called Truth Inference that has been extensively studied in existing crowdsourcing which determines how to effectively infer the truth of each task. Specifically, we focus on truth inference for binary tasks in microtasks, that is, each task only has yes/on choices, which have important application value in crowdsourcing. For example, in a query such as "Do the two videos belong to the same theme?", the expected answers of the form "yes/no" where yes is denoted by 1 and no is denoted by 0.

Truth inference is significant for controlling the quality of crowdsourcing and is a crucial issue for crowdsourcing platforms to get the correct answers. The truth inference algorithms proposed in existing works can be divided into two main categories, direct computation, and optimization methods. The direct computation estimates task truth directly based on worker answers without modeling workers or tasks. The optimization methods model the worker or task in advance then defines an optimization function to express the relationship between the worker responses and the task truth, and finally derive an iterative method to compute parameters collectively. In the optimization methods, the task truth and other parameters are mainly calculated iteratively until convergence using the EM algorithm, which is a classical and effective method for estimating the truth values of unknown variables. However, two limitations of EM hinder its effectiveness in this application scenario: EM-based algorithms are highly dependent on initialization parameters; using EM to estimate the maximum likelihood can only get the local optimal results, which often get stuck in undesirable local optima [4].

Global optimization is the most ideal result, but there are difficulties in its implementation. The most intuitive method to obtain the globally optimal result is to find the global maximum likelihood values of all possible mappings from tasks to answers, to find the most likely true answers. Consider a simple example, if we have 50 binary tasks, then the full number of task-answer sequences for these tasks is 2^{50} , an exceedingly large number. However, considering the large-scale operation in the context of crowdsourcing, the number of calculations required increases exponentially with the increase of tasks and workers. Therefore, it is often intractable to obtain these global optimal quality management technologies.

In this paper, global optimization cannot be achieved but we are not satisfied with the local optimal results derived by traditional optimization methods. We compromise between the inaccessible global optimization and the local optimum, and further truth discovery on the local optimum results derived from optimization-based methods, and propose an iterative optimization method to obtain an approximate global optimum solution. By modeling the worker quality, a likelihood function is constructed to capture the relationship between worker quality and task truth. We then prune the local optimum which is derived by iteratively converging the likelihood optimization function using the EM framework to construct the dominance ordering model (DOM). As a result, we narrow the search scope and reduce the mapping space. Then, a Cut-point neighbor detection algorithm is designed to iteratively search the response with the maximum likelihood-based on our model until convergence [5], approaching the optimal solution without increasing largescale computation.

To sum up, the main contributions of this paper include the following four points:

- We present three different worker quality evaluation models to obtain local optimal results using maximum likelihood estimation, namely worker quality confusion matrix model, worker quality probability parameter model, and dynamic worker quality evaluation model;
- We construct a pruning strategy-based dominance ordering model (DOM) based on the local optimal results, which is composed of worker responses and worker categories (i.e., task-response sequence), and reduces the space of potential task-response sequence while retaining the dominant sequence;
- We propose a Cut-point neighbor detection algorithm on the constructed DOM model to find the task-response sequence with the maximum likelihood within the dominance ordering model (DOM) by iterative search;
- 4. We perform extensive experiments to compare the results obtained by our algorithm with the local optimum results obtained by the EM algorithm on a variety

of metrics. The experimental results show that our algorithm significantly outperforms EM-based algorithms in both simulated data and real-world data.

The remaining of this paper is organized as follows. Section 2 discusses the related works. Section 3 describes our concept definitions and illustrates some symbols with an example. Section 4 describes the estimation model of worker quality based on static and dynamic. We describe the iterative optimization method in Sect. 5 and present our experimental results in Sect. 6. Finally, we conclude our work in Sect. 7.

2 Related Work

In existing crowdsourcing, it is common for multiple workers to be assigned the same task and the answers given by different workers are aggregated to infer the truth for each task. Since the crowd (called workers) may produce lowquality or even noisy answers, the problem of truth inference has been widely studied in existing crowdsourcing to tolerate low-quality workers and to infer high-quality results from noisy answers [6].

To solve this problem, a simple and straightforward idea is majority voting (MV), which treats the truth of each task as the answer given by the majority of employees [7-12]. But the MV strategy presupposes that every worker has an equal vote, which is not possible in real life, where there are differences among workers. Some are ordinary workers, some are experts, some choose answers randomly, and even malicious employees give wrong answers. Therefore, it is necessary to capture the quality of each worker, and it is wiser to trust the answers given by highly qualified workers. Some jobs require workers to complete a small number of tasks with ground truth (called golden tasks) before they can answer the task, known as "qualification test" [13]. This method is used to assess a worker's ability in advance. This can detect and eliminate some cheaters or malicious workers before the worker answers the task. Test questions can also be randomly mixed into common tasks to test the quality of the worker, which gives a more realistic picture of the worker's actual ability and is known as "hidden test" [14, 15]. Liu et al. [16] obtained the accuracy of the worker's answers to the tasks by adding test questions and then used Bayesian theory to obtain the true answers to the tasks based on the quality of the worker and the answers to obtain the true answers to the final task. However, both methods have some limitations, for example: in gualification tests, many workers do not want to complete "extra" tasks without compensation, and in hidden tests, even adding test questions may not improve the quality of workers.

Based on the above problems, existing works [15, 17–21] propose optimization methods to solve them. The basic idea of the optimization method is to use a custom optimization function to capture the relationship between the worker quality and the true value of the task. Then, an iterative approach is used until convergence finally inferring the true answer of worker quality and task. Most of the works use the EM framework to iteratively compute the unknown parameters. The difference between these works is that they model task difficulty and worker quality differently and construct different optimization functions to express the relationship between the two sets of parameters. Some works model the quality of each worker as a single probability value using a real number $q_w \in [0, 1]$ [14, 15]. The higher q_w is, the worker w has higher ability to correctly answer tasks, which is consistent with the Worker Quality Probability Parameter Model we describe in Section 4.2. Li et al. [18] extend a wider range of worker quality probability with $q_w \in (-\infty, +\infty)$. In addition, worker quality can be characterized by confusion matrices [19, 20]. The approach proposed by Dawid et al. [22] first used confusion matrices to model worker quality. Venanzi et al. [21] extended the DS model by assuming a fixed number of worker clusters and that workers in the same cluster have similar confusion matrices, rather than modeling individual confusion matrix modeling. Imamura et al. [17] proposed a broader class of crowdsourcing models including the DS model as a special case that enables it to handle worker clusters, which is more practical than the DS model in a real crowdsourcing setting. A minimax error rate under more practical setting is also derived, and the correctness of the theoretical error analysis is verified by numerical calculations. Similarly, we mainly use the confusion matrix to express the worker quality in this paper, with the difference that we simplify the representation of the confusion matrix for computational convenience, as described in Sect. 4.1.

Besides, some works [23-27] use probabilistic models for truth-value inference of crowdsourcing tasks. Li et al. [23] proposed a Bayesian model (BWA) with conjugate before solving the classification problem of crowdsourcing labels, extending from discrete binary classification tasks to multiclass classification tasks, and a direct inference is performed using expectation-maximization (EM). Kurup et al. [24] proposed an iterative probabilistic model-based approach for crowdsourcing task aggregation. The quality of workers was estimated using a predictive model with the expertness, reliability, and task easiness of the workers as parameters, which used true answers as latent variables. Expectation-maximization (EM) is used to estimate the parameters and hidden variables that provide maximum likelihood. Li et al. [25] similarly focus on probabilistic truth discovery models and reconstruct them as a geometric optimization

problem. Based on sampling techniques and a few other ideas, the first $(1 + \epsilon)$ -approximation solution is achieved.

Some works focused on theoretical guarantees. Das Sarma et al. [4] proposed a technique for global optimal quality management, finding the maximum likelihood item ratings and worker quality estimates. They made two limiting assumptions: (1) all workers have the same quality; (2) the number of workers answering each question is fixed. These assumptions are too restrictive in reality. Our approach obtains results by modeling worker quality from different perspectives (Sect. 4) and using the most commonly used EM framework to iteratively maximize the likelihood function. We do not provide theoretical guarantees, but we find an approximate global maximum likelihood mapping on the locally optimal results further using our proposed iterative algorithm under more relative assumptions (i.e., workers can have different values of quality and each task can get a different number of answers).

3 Problem Description

3.1 Framework Overview

The overall procedure of the proposed optimization method is illustrated in Fig. 1. To begin with, requesters publish microtasks on the crowdsourcing platform and microtasks are assigned to workers. Workers then accept and answer the tasks, and submit answers back to the platform. Based on the collected worker answers to the tasks, we estimate the true answers for the tasks and worker quality using EM with different prior models. Local optimal results of task answers and worker quality are obtained. Then, workers are ranked into different quality categories based on the local optimal results of worker quality. After that, a dominance ordering model (DOM) based on known worker classification and worker response (i.e., task-response sequence) is constructed. The vertices of the model with lower probability are pruned, which narrows the search scope and reduces the mapping space. Then, a Cut-point neighbor detection algorithm is designed to iteratively search for the task-response sequence with the maximum likelihood in our model until convergence. Finally, the approximate global optimal results are obtained.

3.2 Problem Definition

We start with the introduction of some symbols in Table 1 and then combined it with an example of image annotation to make a specific description of some symbols.

3.2.1 Task Question and Option

Consider a group of tasks $\{t\}^n$ with a total number of *n*. These tasks are completed by a group of workers $\{w\}^m$ with a total number of *m*. Work *w* completes task *t* with *k* options $\{1, 2, 3, ..., k\}$. Each worker can answer multiple different tasks, and each task can be accomplished by multiple different workers. Each task has a correct answer z_t (that is, one of the *k* options is the true answer).



Fig. 1 Overall procedure of the proposed approach

L. Cui et al.

Table 1 Notation table

Symbol	Explanation
t	Task
w	Worker
r_t^w	Responses from worker w to task t
z_t	The true answer of task
р	Worker response probability matrix
$p_{i,j}$	The probability of the worker choosing i when the answer to the task is j
N_t^w	The number of times worker w responds to task t
q_w	Worker quality
L	Overall likelihood
DOM	Dominance ordering model
DOG	Dominance ordering graph
d_{n*}	Distance between the nth class worker and the best worker

3.2.2 Task Response

 r_t^w as the response of worker w to task t. For the binary problem studied in this paper, each r_t^w has a value of 0 or 1.

3.2.3 Worker Response Probability Matrix

For the binary problem studied in this paper, a worker response probability matrix of $p = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$ is considered. p_{11} and p_{21} , respectively, represent the probability that $r_t^w = 0$ and $r_t^w = 1$ when the real answer of the task is 0. p_{12} and p_{22} respectively represent the probability that $r_t^w = 0$ and $r_t^w = 1$ when the real answer of the task is 1. The whole matrix is described by a pair of values (e_0, e_1) , in which e_0 is worker false positive (FP) rates (i.e., p_{21} value) and e_1 is false negative (FN) rates (i.e., p_{12} value).

3.2.4 Overall Likelihood

Assuming that each worker answers the question independently. The likelihood value of t_1 is the product of the probability that the worker who answers the task t_1 makes the correct response. *L* is the overall likelihood value of a set of tasks. It is the product of the likelihood value of each task in the task set. Its calculation formula is $L = L(t_1) \times L(t_2) \times ... \times L(t_n)$.

3.2.5 Task-response Sequence

The task-response sequence is constructed by combining workers response and workers category. A worker's response to a binary task is 1, denoted by Y; similarly, a response to a binary task is 0, denoted by N, which is not related to the task truth value and whether the worker's response to the task is correct or not.

3.2.6 Distance

We calculate the distance between workers in plane rectangular coordinate system. The quality of workers is represented by his/her error rate (e_0, e_1) , which is a point in the coordinate system. The distance between workers is expressed by the Euclidean distance between two points in a two-dimensional plane. The best worker quality is (0, 0), which is the origin.

3.2.7 Maximum Likelihood Problem

For a pool of workers $\{w\}^m$, there are a sequence of tasks $\{t\}^n$ where *n* is the number of tasks, and each task is associated with a latent true answer z_t which is picked from *I* different options. Tasks $\{t\}^n$ will receive responses $\{r\}_t^w$ from workers $\{w\}^m$. Here, we call a function *f* that assigns options to tasks a mapping. So there are 2^n mappings for the filtering problem. Assuming independence of worker responses. We calculate the probability of the task-option mapping as allover likelihood *L*. We maximize the allover likelihood *L* to evaluate true answers $\{z\}^t$ for tasks and the performance of each worker.

Example 1 In this example, there are a group of image annotation tasks $\{t_1, t_2, \dots, t_n\}$. All of which are binary task problems with two options $\{0, 1\}$. We take t_1, t_2 as an example to illustrate, we assume that $z_{t_1} = 1$, $z_{t_2} = 0$. A group of 10 workers{ $w_1, w_2, w_3, \dots, w_{10}$ } respond to these tasks and the error rates of ten workers is (0.1, 0.3), (0.2, 0.2), (0.3, 0.2), (0.3, 0.6), (0.4, 0.4), (0.5, 0.5), (0.6, 0.4), (0.7, 0.7), (0.8, 0.6), (0.7, 0.7), (0.8, 0.6), (0.8(0.6), (0.9, 0.5). We determine the category to which each worker belongs based on the distance between the worker and the origin. In this example, we classify workers into three categories, which are characterized by numbers 1, 2, and 3, respectively. The workers who have a high-quality rank in the front. We divide the distance between the worst worker [that is, the error rate (1, 1)] and the origin on average into three intervals. The distance between the first class workers and the origin is within $[0,\sqrt{2}/3]$ (that is $d_{1*} \in$ $[0,\sqrt{2/3}]$). Similarly, $d_{2*} \in [\sqrt{2/3}, 2\sqrt{2/3}]$ and $d_{3*} \in [2\sqrt{2/3}]$,1]. According to the calculation, we determine the category to which each worker belongs. Workers choose tasks to answer, and different tasks will receive different numbers of answers. Here t_1 and t_2 receive 2 responses. Table 2 shows several workers' responses received by t_1, t_2 .

We take the task-response sequence of t_1 as Y1N1 and taskresponse sequence of t_2 as Y2N2, for example, to calculate **Table 2** Example of workers responses to t_1 and t_2

Task	Task responses			
	First class workers	Second class workers	Third class workers	answer sequence
<i>t</i> ₁	$r_{t_1}^{w_1} = 1, r_{t_1}^{w_3} = 1$	-	_	Y1Y1
	$r_{t_1}^{w_2} = 1$	$r_{t_1}^{w_2} = 1$	-	Y1Y1
	$r_{t_1}^{w_1} = 1$	_	$r_{t_1}^{w_8} = 1$	Y1Y3
	_	$r_{t_1}^{w_5} = 1$	$r_{t_1}^{w_9} = 1$	Y2Y3
	-	$r_{t_1}^{W_6} = 1$	$r_{t_1}^{W_{10}} = 0$	Y2N3
	_	_	$r_{t_1}^{w_8} = 1, r_{t_1}^{w_{10}} = 1$	Y3Y3
	$r_{t_1}^{w_1} = 1, r_{t_1}^{w_2} = 0$	-	_	Y1N1
<i>t</i> ₂	_	$r_{t_2}^{w_4} = 1, r_{t_2}^{w_7} = 0$	_	Y2N2
	$r_{t_2}^{w_3} = 0$	$r_{t_0}^{W_5} = 1$	_	Y2N1
	_	$r_{t_0}^{W_6} = 0$	$r_{t_0}^{w_9} = 1$	Y3N2
	_	-	$r_{t_0}^{\frac{1}{W_8}} = 0, r_{t_0}^{\frac{1}{W_9}} = 0$	N3N3
	-	$r_{t_{0}}^{w_{7}}=0$	$r_{t_2}^{W_{10}} = 0$	N2N3
	$r_{t_2}^{w_1} = 0$	_	$r_{t_0}^{\frac{1}{W_8}} = 0$	N1N3
	$r_{t}^{W_{2}} = 0$	$r_{t}^{w_{4}} = 0$	_	N1N2

the likelihood value. From Table 2, w_1 and w_2 answer t_1 , their error rate is (0.1, 0.3) and (0.2, 0.2), respectively, so their probability of answering the t_1 correctly is 0.7 and 0.8, respectively. So we can get $L(t_1) = 0.7 \times 0.8 = 5.6 \times 10^{-1}$. Similarly, we can get $L(t_2) = 0.7 \times 0.4 = 2.8 \times 10^{-1}$. If n = 2 in task set, only t_1 and t_2 are included, then the overall likelihood of $L = L(t_1) \times L(t_2) = 1.568 \times 10^{-1}$.

4 Worker Quality Evaluation Model

Due to the different abilities and knowledge background of workers, the quality of employees varies greatly. To get a more valid estimation answer, an accurate worker quality evaluation model is very important. In this section, the quality of employees is modeled, and then the true answer of the task is inferred based on the quality of the workers and the answers of the workers. In this section, we model the quality of workers using static and dynamic methods, respectively. Three different models are used to model worker quality, including the worker quality confusion matrix model, worker quality probability parameter model, and dynamic worker quality evaluation model. The worker quality confusion matrix model and worker quality probability parameter model assess the quality of the workers based on the static method, which refers to the use of a fixed value to express the quality of workers. To describe the quality characteristics of workers more accurately, a dynamic worker quality evaluation model is also proposed. The dynamic worker quality evaluation model uses a function distribution to express the relationship between worker quality and task difficulty.

4.1 Worker Quality Confusion Matrix Model

Consider a worker response probability matrix p, of the size I * I (I is the number of task options). Thus, p(i, j) is the probability that a worker rates a question with true value j as having option i. Given responses r_t^w of worker w to the tasks and the tasks' true answers z_t , we can get the worker quality matrix. For binary tasks, we can use (e_0, e_1) to represent the quality of workers. N_t^w represents the number of times worker w responds to task t. We can calculate the worker error rates e_0 and e_1 as follows:

$$e_0^w = \frac{\sum_{t=1}^t N_t^w (r_t^w = 1 \land z_t = 0)}{\sum_{t=1}^t N_t^w (z_t = 0)}$$
(1)

$$e_1^{w} = \frac{\sum_{t=1}^{t} N_t^{w} (r_t^{w} = 0 \land z_t = 1)}{\sum_{t=1}^{t} N_t^{w} (z_t = 1)}$$
(2)

Binary tasks ask workers to select "*T*" or "*F*" for each claim, then an example confusion matrix for *w* is $p = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$, where $q_{2,1} = 0.3$ means that if the truth of a task is "*F*," the probability that the worker answers the task as "*T*" is 0.3. Similarly, $q_{1,2} = 0.2$ means that if the truth of a task is "*T*," the probability that the worker answers the task as "*F*" is 0.2.

Example 2 Suppose we are given 6 binary tasks $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ with true answers $A = \{z_1, z_2, z_3, z_4, z_5, z_6\} = \{1, 1, 2\}$

1, 1, 0, 0, 0}, the responses of worker *w* are $R = \{1, 1, 0, 0, 1, 1\}$, then we can evaluate false positive and false negative rates for worker *w* : $e_0 = 2/3$; $e_1 = 1/3$.

Given data about tasks $\{t\}^n$, workers $\{w\}^m$ and responses $\{r\}_t^w$. We calculate the likelihood *L* based on the probability error rate, i.e., (e_0, e_1) of each worker. In the confusion matrix model, we focus on binary tasks (with fixed two choices) and model each worker as a confusion matrix q_w with size 2 × 2. From Eqs. 1 and 2, the worker *w*'s answer follows the probability $p = \begin{bmatrix} 1 - e_1 & e_1 \\ e_0 & 1 - e_0 \end{bmatrix}$. The target is to optimize the likelihood function *L*, where

$$L = \prod_{t=1}^{t} \prod_{w=1}^{w} (e_0^w)^{N_t^w} (r_t^{w=1 \wedge z_t=0}) (1 - e_0^w)^{N_t^w} (r_t^{w=0 \wedge z_t=0}) (e_1^w)^{N_t^w} (r_t^{w=0 \wedge z_t=1}) (1 - e_1^w)^{N_t^w} (r_t^{w=1 \wedge z_t=1})$$
(3)

and it applies the EM algorithm to devise two iterative steps.

4.2 Worker Quality Probability Parameter Model

Here, we model the quality of workers only by a binary parameter. Each worker's quality is modeled as worker probability $q_w \in [0, 1]$. Then

$$p(q_w = \text{reliable}) = \frac{\sum_{t=1}^t N_t^w(r_t^w = z_t)}{\sum_{t=1}^t N_t^w}$$
(4)

which means that the probability worker w correctly answers a task is q_w , and worker w incorrectly answers a task is $1 - q_w$.

Given the data of tasks $\{t\}^n$, workers $\{w\}^m$ and responses $\{r\}_t^w$. We calculate the likelihood *L* based on the reliability of each worker.

$$L = \prod_{t} \prod_{w} [p(q_{w} = \text{reliable})]^{N_{w}^{t}(r_{t}^{w} = z_{t})}$$

$$[1 - p(q_{w} = \text{reliable})]^{N_{w}^{t}(r_{t}^{w} \neq z_{t})}$$
(5)

we apply the EM framework and iteratively updates q_w and r_t^w to approximate its optimal value.

Base on the responses $\{r\}_t^w$ of a set of workers $\{w\}^m$ to a set of tasks $\{t\}^n$, a EM algorithm is utilized to maximize the likelihood function. To estimate the parameters of the two static worker quality models, and infer the true answer to the task. The core process of the EM algorithm has one E-step and one M-step step.

E-step Estimate the probability of the correct answer to the task based on the value of the worker quality model parameters obtained by M-step.

M-step Maximize the expectation of the likelihood function and re-estimate the value of the worker model parameters based on the estimated value of the task answer derived from the E-step.

4.3 Dynamic Worker Quality Evaluation Model

The dynamic model extends the above two models in task model. Rather than assuming that each task is the same, it models each task *t*'s difficulty $\beta_t \in [0, 1]$ (the higher, the more difficult). Here $\beta_t = 0$ means the task is very simple that almost all workers could provide a correct response to it. $\beta_t = 1$ means the task if so confusing that even the most reliable worker cannot give the response for sure. We model the quality of a worker using α_w where $\alpha_w \in [0, 1]$. Here $\alpha_w = 1$ means the worker is always correct in responding and $\alpha_w = 0$ means the worker is always incorrect. Then, it models the worker's answer as

$$p(r_t^w = z_t | \alpha_w, \beta_t) = \frac{1}{1 + e^{-(\alpha_w - \beta_t)}}.$$
(6)

We calculate the likelihood *L* based on the quality of the worker and the difficulty of the task.

$$L = \prod_{w} \prod_{t} p(r_t^{w} | z_t, \alpha_w, \beta_t).$$
⁽⁷⁾

We use maximum likelihood estimation to estimate the parameters of the worker model and the true answers to the task. We assume that all workers' responses are independent, then the target is to maximize the likelihood function.

5 Optimization Strategy

In general, the technique for global optimal quality management is to find the task-response sequence with the maximum likelihood from all possible mappings from tasks to results. However, this technique is intractable due to the large-scale operation involved in a typical crowdsourcing setting. To address this problem, we first estimate the true answers for the tasks and worker quality using EM with different prior models. We can rank workers into different quality categories according to the estimates of their quality. After that, we propose a dominance ordering model (DOM) based on known task-response sequences. A Cutpoint neighbor detection algorithm is designed to search for the task-response sequence with the maximum likelihood in our model and the task-response sequences with lower probability are pruned. The results of our search algorithm are then used as a new input to update worker categorizations and the dominance ordering model until convergence.

5.1 Classifying Workers

We classify workers into c categories according to their quality, with workers having high quality ranked in the front. In the case of a small number of workers, each worker may be classified into one category. Here, we classify workers to simplify our model, and the quality of each worker still varies in our final result. We calculate the distance between a worker and the best worker, then classify them according to this distance value. To show the difference between workers more accurately, we set the distance between any two adjacent categories to be equal. Figure 2 shows an example of classifying workers.

Example 3 In this example, we classify workers into ten categories. We use the confusion matrix to model worker quality, and the entire matrix can be described with just two values $[e_0, e_1]$. The error rates of worker 1 and worker 2 is [0.3, 0.45] and [0.25, 0.2], respectively. We can calculate the distance between these two workers and the best worker with error rates [0, 0] in this example. Then, we set the difference between two adjacent worker categories as $\sqrt{2}/10$. So worker 1 is in the fourth class and worker 2 is in the third class.

Workers may have different levels of expertise and the results collected from the crowd are inherently noisy and



Fig. 2 Example of ten worker categories

ambiguous. In order to find the correct task-answer mapping with the maximum likelihood, we propose a dominance ordering model.

After worker classification, we can construct the dominance ordering model. In this paper, we focus on the filtering problem. Workers' responses are in the form of "Y/N". In this way, "Y1" denotes that a worker in the first category answered "yes" to a task, and "N1" denotes that a worker in the first category answered "no" to the task. Each task may be accomplished by multiple workers and each worker may answer multiple different tasks. For example, task *t* received three responses (1, 1, 0) from different workers; the second worker belongs to the fourth category of workers, and the third worker belongs to the seventh category of workers. Then, the response set for task *t* is denoted as $Y_1Y_4N_7$.

We observe that response sets are in an inherent ordering. For the tasks with a same number of responses, we sort the response sets by the level of expertise of workers. For example, there are two tasks t_1 and t_2 , each with three responses of *yes*. Responses of t_1 are from workers belonging to the first, third, and fourth categories, whereas responses of t_2 are from workers belonging to the first, third, and fifth categories. Then, the response set of t_1 is ordered higher than that of t_2 , Y1Y3Y4 dominates Y1Y3Y5 (i.e., Y1Y3Y4-> Y1Y3Y5). In this model, the probability of answer Y is smaller and smaller from top to bottom.

Definition 1 (*Dominance Ordering*) The response set for each vertex contains one or several elements from $\{Y1 \dots Yc, Nc \dots N1\}$. Vertex v_1 dominates vertex v_2 if and only if one of the following conditions is satisfied:

- 1. v_1 and v_2 contain the same number of "1" and "0" responses in total, and at least one response of "1" in v_1 is answered by a worker with higher quality than any worker answering "1" in v_2 , or at least one response of "0" in v_1 is answered by a worker with lower quality than any worker answering "0" in v_2 .
- 2. v_1 contains more "1" responses and fewer "0" responses than v_2 .

5.2 The Dominance Ordering Model (DOM)

For tasks receiving the same number of responses, their responses are constructed in the same dominance ordering graph (DAG). In addition, the DAG that receive different numbers of responses are integrated in order to handle the problem that tasks receive different numbers of responses at the same time. For the DAG where tasks receive an even number of responses, we set up a central layer where the response sets of vertices are characterized by the same



Fig. 3 Example of our model with a maximum of three workers

number and worker classes to responses yes and no (e.g., Y1Y2N1N2). However, for the DAG where tasks receive an odd number of responses, we set a virtual center layer, which is composed of the edges equal to the starting point and the end point. We integrate each DAG into our model through the central layer and the vertices with the same distance from the central layer have similar dominance. Figure 3 shows an example of DOM.

Example 4 In this example, the workers sets are divided into three categories, and each task receives up to three responses from workers. DAG (a), (b), and (c) in Fig. 3 represent tasks with 1, 2, and 3 responses, respectively. As shown in Fig. 3,

the central layer exists in DAG (b), while (a) and (c) only contain a synthetic central layer.

5.3 Cut-point Neighbor Detection Algorithm

In this section, we describe the process of Cut-point neighbor detection as illustrated in Algorithm 1.

Definition 2 (*Cut-point*) In our model, the probability that the answer to a question is 1 (*Yes*) decreases from top to bottom. Here, we define the Cut-point as a mapping that divides the vertices in our model into two partitions. The vertices above the Cut-point are mapped to 1, and those below the Cut-point are mapped to 0.

After constructing the dominance ordering model (DOM), we search for the maximum likelihood mapping. The search begins from the starting Cut-point which is generated by the EM algorithm. We will constantly adjust the position of the Cut-point to find a sequence with maximum likelihood. First, we find the vertices closest to the Cut-point and put them into a vertex-set. We then replace the answers of the vertices in this set (that is, vertices whose answers are 1 will be changed to 0 and vice versa). In the first round of replacement, we replace the answers of the vertices one by one. Then, recalculate the overall likelihood of the taskresponse sequence. If the likelihood increases, the changes are retained, and these vertices are removed from the vertexset. Otherwise, restore the answers of vertices to before the replacement operation. In the second round of replacement, we replace answers of any two vertices in the vertex set. In round s, any vertices in the vertex set are replaced. Here, we set a stop value γ for the rounds s to control the number of computations. This process is illustrated by the example in Fig. 4.

5.3.1 Iteration of Cut-point Neighbor Detection

Our Cut-point neighbor detection algorithm eventually produces a new task-response sequence and the quality of the workers. We utilize the results as input to update the workers' classification and the position of the tasks in our model. Then, a further search is performed until convergence (i.e., the difference of the final likelihood value of two iterations is 0 or below a predefined threshold). This process is illustrated in Algorithm 2. In each iteration, we increase the number of worker categories to make tasks at the same vertex more similar. In this way, we can find a task-response sequence with a higher likelihood effectively.

5.4 Discussion of Cut-point Neighbor Detection

In this section, we describe our algorithm for finding the sequence with approximate global maximum. A naive global optimal algorithm could be to scan all workers' responses, calculating for each task-response sequence and the likelihood L. However, the number of all task-response sequence



Fig. 4 Example of the proposed Cut-point neighbor detection

is exponential. Given a set of tasks $\{t\}^n$, we can assign a value of 0 or 1 to any of them, resulting in 2^n different task-response sequences. This makes the naive algorithm prohibitively expensive.

The complexity introduced by worker classes is governed by the number of worker classes. Fewer worker categories imply fewer task-response sequence and hence, a lower complexity, while fine worker classification imply a higher complexity. For instance, in the limiting case where we assume every worker belongs to the same category, we do not classify workers, and the dominance ordering graph (DAG) correspondingly has no edges. Therefore, the complexity of our algorithm, which scales linearly with the number of mappings, is $\mathcal{O}(2^k)$, where k is the number of worker responses per task. Note that although this is exponential in the maximum number of worker responses per task, typical values of k are a very small constant for most practical applications. At the other extreme, when the different worker classes are highly constrained, the dominance ordering graph (DAG) can be reduced to a single chain. In this extreme, the resulting complexity of our algorithm is $\mathcal{O}(\binom{k+1}{k})$.

In this paper we infer task truth and worker quality by maximizing the likelihood function. Maximum likelihood estimation is an optimization method that evaluates task truth and model implied parameters given the set of worker responses to the task. Standard techniques for solving this estimation problem typically involve the use of expectation-maximization (EM). The EM algorithm is only guaranteed to obtain a local maximum by iterating an expectation step and a maximization step several times until convergence. To our knowledge, however, current applications and researches based on EM provide no theoretical guarantees. Since our work continues on local optimization results derived based on EM methods, it is also difficult to provide theoretical guarantees. Instead, we can provide experimentbased probabilistic guarantees on the estimated true value of the tasks and demonstrate the effectiveness of our method through experimental results.

Algorithm 1 Cut-point Neighbour Detection

```
Input: data; f;
 Output: new_L(new likelihood); new_f(mapping corresponding to new_L); new_q(Worker quality
corresponding to new_{-}f);
 1: Construct V; E = Dominance Ordering Model(DOM);
 2: function MAIN(f, data)
 3 \cdot s \leftarrow 1 \cdot
 4: while (s < \gamma) do
       new_{-L} \leftarrow VertexSelection(s, DOM, L);
 5
 6:
       if new_{-}L > L then
 7:
          f \leftarrow new_{-}f:
 8.
          L \leftarrow new_{-}L
 9:
          s \leftarrow 1;
10:
       else
11:
          s + +
12:
       end if
13<sup>.</sup> end while
14: return new_L; new_f; new_q;
15: function VertexSelection(s, DOM, L)
16:
    if (s = 0) then
17:
       calculate new_L:
18:
       return new_L;
19:
    else
20:
       for vertex in vertexset do
21:
           change answer of tasks in vertex;
22:
          if (answer = 1) then
23:
             answer \leftarrow 0;
24:
           else
25:
             answer \leftarrow 1:
26:
           end if
27.
          new_L \leftarrow VertexSelection(s-1, DOM, L)
28.
          if new_L > L then
29:
             return new_L:
30.
          end if
       end for
31:
32: end if
```

Algorithm 2 Iteration of Cut-point Neighbour Detection

Input: data, f (from EM Algorithm) Output: L^* (maximum of likelihood); f^* (mappings corresponding to likelihood*); q^* (Worker quality corresponding to f*); 1: while not converged do 2: update c, DOM; 3: $(new_L, new_f, new_q) \leftarrow MAIN(f, data)$ 4: end while 5: return $L^*; f^*; q^*$

6 Experiments

In this section, we evaluate the performance of our iterative optimization strategy (labeled as IOS_EM) on synthetic and real rating data and compare it against the EM algorithm (labeled as BAS_EM). We report our performance in terms of overall likelihood and other parameters of interest. Here, we discuss two experiments, one based on simulative data and another based on real-world data, and analyze the results to draw conclusions.

Note that here we assume that worker identity is unknown and each worker is an independent individual. Arbitrary workers could answer different tasks - our goal is to characterize the behavior of the worker population as a whole. We expect the overall likelihood value to be larger and our assumptions on worker independence to be closer to the truth.

6.1 Experiment 1: Synthetic Data Experiments

In this section, we describe our experiments based on synthetic data. Here, we choose the estimation prior model of worker quality modeled by a confusion matrix. For the binary task we study, the number pair (e_0, e_1) is used to simplify the representation of worker quality confusion matrix model. We generate the data based on this model and compare our method against the EM algorithm in terms of (a) overall likelihood, (b) accuracy of answer predictions, and (c) accuracy of worker quality predictions.

6.1.1 Dataset

To generate the ground truth answers for a set of tasks, given a fixed selectivity u, we assign a ground truth value of 1 with a probability of u and 0 with a probability of (1-u) for each task. Then, we generate a distinct worker response probability matrix for each worker, with the only constraint that most of the workers (more than 90%) are better than random (workers' error rate e_0 and e_1 are < 0.5). We then generate worker responses based on these matrices.

6.1.2 Judgment Indicators

Overall likelihood As mentioned in Sect. 3, the overall likelihood is the product of each task's likelihood values in a task set.

Accuracy of answer predictions It is actually error rate (ER). It is the ratio of the number of mislabeled tasks to the total number of tasks.

Accuracy of worker quality predictions The Average Euclidean Distance (AED) between the estimated worker quality and the actual worker quality is used to measure



Fig. 5 Synthetic Data Experiment. Overall likelihood: a Setting 1; b Setting 2; and c Setting 3

306



Fig. 6 Synthetic Data Experiment. Accuracy of answer predictions: a Setting 1; b Setting 2; and c Setting 3

the accuracy of worker quality predictions. The smaller the value, the more accurate our assessment of worker quality.

6.1.3 Experimental Process and Result

We compare our algorithm with the EM algorithm which is also settling the maximum likelihood problem. BAS_EM takes an initial estimate or guesses for worker error rates as a parameter. Here, we experiment with the initialization of e_0 and $e_1 = 0.5$. We set task number m = 500, and we vary the selectivity u, and the number of worker responses per task k.

We perform experiments under three data settings: Setting 1, each task receives k responses and m = k (m is the total number of workers); Setting 2, each task receives k responses and m > k; Setting 3, each task receives a different number of responses and m > k.

Overall likelihood. Figure 5 shows the likelihoods of taskresponse sequence returned by our algorithm and BAS_EM instances on a varied number of workers, for three data settings. The y-axis is in log scale, with a higher value being more desirable. In Fig. 5a, there are 3–10 workers in each data and each worker has completed all tasks (500 tasks), and in Fig. 5b, there are 10 workers in each data and each task receives a different number of responses (x-axis), so each worker completed 150–500 tasks. Contrast this to Fig. 5c, here, each task receives different responses (less than workers) and the total number of workers (x-axis) is varying in each data. We observe that our strategy has a significant improvement in likelihood values when the information given to BAS_EM is sparser.

Accuracy of answer predictions. In Fig. 6, we plot the error rate(ER) of task ground truth estimations each of the algorithms estimate task answer incorrectly (a lower score is better). Here, again, our strategy estimates true values of tasks with higher accuracy than BAS_EM.

Accuracy of worker quality predictions. To evaluate the estimated worker quality against the actual one, we plot the Average Euclidean Distance (AED) between our estimated matrix and the actual one (a lower score is better) in Fig. 7. We observe that our strategy's estimations are closer to the actual probability matrix than all the BAS_EM, and when the likelihood value improves larger, the estimation result is better.



Fig. 7 Synthetic Data Experiment. Accuracy of worker quality predictions: a Setting 1; b Setting 2; and c Setting 3



Fig. 8 Real Data Experiment. Likelihood and error rate of answer estimation for (a), (b) and (c)

6.1.4 Summary

For all metrics, our strategy outperforms BAS_EM. It should be noted that our algorithm has more obvious advantages in the third case and is closer to the actual situation. The third setup (i.e., each task receives a different number of responses) is common in real crowdsourcing markets.

6.2 Experiment 2: Real-world Data Experiments

In this section, we describe our results on a real-world dataset. We evaluate our method with three different estimation prior models: (A) workers' quality represented by a confusion matrix, (B) workers' quality represented by a binary parameter, and (C) workers' quality and task difficulty, and compare our method versus the BAS_EM in terms of overall likelihood and ground truth of task estimations.

6.2.1 Dataset

Our dataset is a sentiment analysis dataset, which corresponds to a collection of more than ten thousand sentences extracted from the movie review website *RottenTomatoes*. It contains a set of 5000 tasks responded by 203 workers. From this collection, a random subset of 5000 sentences was selected and published on Amazon Mechanical Turk for annotation. Given the sentences, the workers were asked to provide the sentiment polarity (positive or negative). We have ground truth *yes/no* answers for each task, but we do not know the real worker quality.

6.2.2 Judgment Indicators

Overall likelihood It is the product of each task's likelihood values in a task set. Its effectiveness is shown by a line graph in Fig. 8.

Accuracy of answer predictions It refers to the error rate (ER), and a lower score is better. Its experimental results are represented in a bar chart in Fig. 8.

6.2.3 Experimental Process and Result

To evaluate the performance of our strategy based on EM(IOS_EM), we vary the size of data by randomly selecting a fixed number of labels from all the data and compare the estimates of the answer (the yes/no answers) with the given ground truth.

Overall likelihood Figure 8 shows the likelihoods of task-response sequence returned by IOS_EM and different BAS_EM on a varied number of labels. The *y*-axis is on a log scale, with a higher value being more desirable. And Table 3 shows the difference between our method and different BAS_EM in terms of likelihood and error rate of ground truth estimations. In Fig. 8, we observe that our method can significantly improve the likelihood value with the three different estimations prior model.

Table 3 Difference oflikelihood and error rate for (a)and (b)

Label number	D-likelihood (a)	D-ER (a) (%)	D-likelihood (b)	D-ER (b) (%)
10000	+ 138.168	- 0.679	+ 233.171	- 1.116
12000	+ 281.449	- 1.200	+ 247.675	- 1.095
15000	+ 269.892	- 1.099	+ 207.998	- 0.753
17000	+ 289.584	- 0.987	+ 231.146	- 0.826
20000	+ 297.262	- 0.901	+ 407.979	- 1.022
22000	+ 308.145	- 0.880	+ 205.360	- 0.500
25000	+ 57.137	- 0.140	+ 9.194	- 0.040
27000	+ 235.244	- 0.600	+ 59.636	- 0.160
25000 27000	+ 50.145 + 57.137 + 235.244	-0.140 -0.600	+ 203.300 + 9.194 + 59.636	- 0.040 - 0.160

Accuracy of answer predictions In Fig. 8, we also compare the error rate(ER) of label estimations with BAS_EM. We plot the error rate of task true answer estimations of each algorithm. Here, again, our method improves the accuracy of the answer estimations while improving the likelihood value.

7 Conclusion

Truth inference has a strong impact in crowdsourcing and it is a fundamental issue of current research. Most of the work has used a custom optimization function to capture the relationship between worker's quality and ground truth and then executed an iterative algorithm until convergence to infer the true answer to the task and other implicit variables finally. The EM framework is commonly employed to derive the implied parameters in the likelihood function in most of the works, but the EM algorithm only provides a locally optimal solution and highly limited by initial parameters. In this paper, we first model the worker quality from three different views and construct likelihood functions and then go further truth discovery based on the local optimum results obtained using the EM algorithm rather than stopping. We construct a dominance ordering model (DOM) based on the local optimum results and design a Cut-point neighbor detection algorithm to improve the estimation of task truth by increasing the overall likelihood value. The DOM constructed based on the pruning strategy not only serves as a platform for the iterative search algorithm but also greatly reduces the space of potential mappings to be considered. The experimental results show that the approximate global optimal results we obtain based on pruning search are better than the local optimal results obtained by simply executing the EM algorithm.

This paper focuses on the binary problems in microtasks. Now there are more and more multiple problems and open-ended crowdsourcing problems. In future research, we hope that the ideas described in this paper can be extended to solve more general types of tasks, such as multiple choice tasks, numeric tasks, and others, and are not limited to truth inference for binary tasks.

Acknowledgements This work is partially supported by National Key R&D Program No.2017YFB1400100, SDNFSC No.ZR2018MF014, SPKR&DP No.ZR2019LZH008.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

- Tong Y, Zhou Z, Zeng Y, Chen L, Shahabi C (2020) Spatial crowdsourcing: a survey. VLDB J 29(1):217–250. https://doi.org/ 10.1007/s00778-019-00568-7
- Lu J, Li W, Wang Q, Zhang Y (2020) Research on data quality control of crowdsourcing annotation: a survey. In: IEEE International conference on dependable, autonomic and secure computing, International conference on pervasive intelligence and computing, International conference on cloud and big data computing, International conference on cyber science and technology congress, DASC/PiCom/CBDCom/CyberSciTech 2020, Calgary, AB, Canada, 17–22 August 2020, pp 201–208. https://doi.org/10. 1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00044
- Bhatti SS, Gao X, Chen G (2020) General framework, opportunities and challenges for crowdsourcing techniques: a comprehensive survey. J Syst Softw 167:110611. https://doi.org/10.1016/j. jss.2020.110611
- Das Sarma A, Parameswaran A, Widom J (2016) Towards globally optimal crowdsourcing quality management: the uniform worker setting. In: Proceedings of the 2016 international conference on management of data, pp 47–62. https://doi.org/10.1145/ 2882903.2882953
- Cui L, Chen J, He W, Li H, Guo W (2020) A pruned DOMbased iterative strategy for approximate global optimization in crowdsourcing microtasks. In: Wang X, Zhang R, Lee Y, Sun L, Moon Y (eds.), Web and Big Data—4th international joint conference, APWeb-WAIM 2020, Tianjin, China, 18–20 September 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol 12317, pp 779–793. https://doi.org/10.1007/978-3-030-60259-8_ 57
- Zheng Y, Li G, Li Y, Shan C, Cheng R (2017) Truth inference in crowdsourcing: is the problem solved? Proc VLDB Endow 10(5):541–552. https://doi.org/10.14778/3055540.3055547
- Cao CC, She J, Tong Y, Chen L (2012) Whom to ask?: jury selection for decision making tasks on micro-blog services. Proc VLDB Endow 5(11):1495–1506
- Franklin MJ, Kossmann D, Kraska T, Ramesh S, Xin R (2011) Crowddb: answering queries with crowdsourcing. In: Proceedings of the ACM SIGMOD international conference on management of data, SIGMOD 2011, Athens, Greece, 12–16 June 2011, pp 61–72. https://doi.org/10.1145/1989323.1989331
- Kuncheva LI, Whitaker CJ, Shipp CA, Duin RP (2003) Limits on the majority vote accuracy in classifier fusion. Pattern Anal Appl 6(1):22–31. https://doi.org/10.1007/s10044-002-0173-7
- Marcus A, Karger RD, Madden S, Miller R, Oh S (2012) Counting with the crowd. Proc VLDB Endow 6(2):109–120. https://doi.org/ 10.14778/2535568.2448944
- 11. Park H, Pang R, Parameswaran AG, Garcia-Molina H, Polyzotis N, Widom J (2012) Deco: a system for declarative

crowdsourcing. Proc VLDB Endow 5(12):1990–1993. https:// doi.org/10.14778/2367502.2367555

- Yan T, Kumar V, Ganesan D (2010) Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In: Proceedings of the 8th international conference on mobile systems, applications, and services (MobiSys 2010), San Francisco, California, USA, 15–18 June 2010, pp. 77–90. https://doi. org/10.1145/1814433.1814443
- Khattak FK, Salleb-Aouissi A (2011) Quality control of crowd labeling through expert evaluation. In: Proceedings of the NIPS 2nd workshop on computational social science and the wisdom of crowds, vol 2, p 5
- Demartini G, Difallah DE, Cudré-Mauroux P (2012) Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st international conference on World Wide Web, pp 469–478
- Liu X, Lu M, Ooi BC, Shen Y, Wu S, Zhang M (2012) CDAS: a crowdsourcing data analytics system. Proc VLDB Endow 5(10):1040–1051. https://doi.org/10.14778/2336664.2336676
- Liu X, Lu M, Ooi BC, Shen Y, Wu S, Zhang M (2012) Cdas: a crowdsourcing data analytics system. Proc VLDB Endow 5(10):1040–1051
- 17. Imamura H, Sato I, Sugiyama M (2018) Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. In: Dy, JG, Krause A (eds.), Proceedings of the 35th international conference on machine learning, ICML 2018, Proceedings of machine learning research, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018, vol 80, pp 2152–2161
- Li Q, Li Y, Gao J, Zhao B, Fan W, Han J (2014) Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In: Dyreson CE, Li F, Özsu MT (eds.), International conference on management of data, SIGMOD 2014, Snowbird, UT, USA, 22–27 June 2014, pp. 1187–1198. ACM. https://doi.org/10.1145/2588555.2610509
- Ipeirotis PG, Provost F, Wang J (2010) Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD workshop on human computation, pp 64–67
- Raykar VC, Yu S, Zhao LH, Jerebko A, Florin C, Valadez GH, Bogoni L, Moy L (2009) Supervised learning from multiple

experts: whom to trust when everyone lies a bit. In: Proceedings of the 26th annual international conference on machine learning, pp 889–896

- Venanzi M, Guiver J, Kazai G, Kohli P, Shokouhi M (2014) Community-based Bayesian aggregation models for crowdsourcing. In: Chung C, Broder AZ, Shim K, Suel T (eds.), 23rd international World Wide Web conference, WWW'14, Seoul, Republic of Korea, 7–11 April, pp 155–164. https://doi.org/10. 1145/2566486.2567989
- Dawid AP, Skene AM (1979) Maximum likelihood estimation of observer error-rates using the EM algorithm. J R Stat Soc Ser C (Applied Statistics) 28(1):20–28
- 23. Li Y, Rubinstein BIP, Cohn T (2019) Truth inference at scale: a Bayesian model for adjudicating highly redundant crowd annotations. In: Liu L, White RW, Mantrach A, Silvestri F, McAuley JJ, Baeza-Yates R, Zia L (eds.), The World Wide Web conference, WWW 2019, San Francisco, CA, USA, 13–17 May, pp 1028–1038. https://doi.org/10.1145/3308558.3313459
- Kurup AR, Sajeev GP (2019) Aggregating unstructured submissions for reliable answers in crowdsourcing systems. In: 9th international symposium on embedded computing and system design, ISED 2019, Kollam, India, 13–14 December, pp 1–7. https://doi.org/10.1109/ISED48680.2019.9096224
- Li S, Xu J, Ye M (2020) Approximating global optimum for probabilistic truth discovery. Algorithmica 82(10):3091–3116. https://doi.org/10.1007/s00453-020-00715-5
- Wu M, Li Q, Wang S, Hou J (2019) A subjectivity-aware algorithm for label aggregation in crowdsourcing. In: Qiu M (ed.), 2019 IEEE international conference on computational science and engineering, CSE 2019, and IEEE international conference on embedded and ubiquitous computing, EUC 2019, New York, NY, USA, 1–3 August, pp 373–378. https://doi.org/10.1109/CSE/EUC.2019.00077
- Patwardhan M, Sainani A, Sharma R, Karande S, Ghaisas S (2018) Towards automating disambiguation of regulations: using the wisdom of crowds. ACM/IEEE international conference, pp 850–855. https://doi.org/10.1145/3238147.3240727