A Crowdsourcing Framework for Collecting Tabular Data

Caihua Shan, Nikos Mamoulis, Guoliang Li, Reynold Cheng, Zhipeng Huang, and Yudian Zheng

Abstract—In crowdsourcing, human workers are employed to tackle problems that are traditionally difficult for computers (e.g., data cleaning, missing value filling, and sentiment analysis). In this paper, we study the effective use of crowdsourcing in filling missing values in a given relation (e.g., a table containing different attributes of celebrity stars, such as nationality and age). A task given to a worker typically consists of questions about the missing attribute values (e.g., What is the age of Jet Li?). Although this problem has been studied before, existing work often treats related attributes independently, leading to suboptimal performance. In this paper, we present T-Crowd, which is a crowdsourcing system that considers attribute relationships. Particularly, T-Crowd integrates each worker's answers on different attributes to effectively learn his/her trustworthiness and the true data values. The attribute relationship information is used to guide task allocation to workers. Our solution seamlessly supports categorical and continuous attributes. Our extensive experiments on real and synthetic datasets show that T-Crowd outperforms state-of-the-art methods, improving the quality of truth inference and reducing the monetary cost of crowdsourcing.

Index Terms—Crowdsourcing, Tabular Data, Truth Inference, Task Assignment.

◆

1 INTRODUCTION

Crowdsourcing is an effective way to address computer-hard problems [8], [23], [36], [37], [43] by utilizing numerous ordinary humans (called *workers* or *the crowd*). The general workflow of crowdsourcing is as follows: at first a *requester* proposes a problem, then the problem is transformed into many tasks (i.e., questions), and finally the workers complete the tasks assigned to them and they are given a monetary reward. Crowdsourcing involves two interrelated processes: *truth inference* and *task assignment*. Truth inference refers to addressing noise and errors for inferring the correct value (or truth) for each task from redundant answers [11] [39]. Task assignment refers to selecting appropriate tasks to assign to each incoming worker. Truth inference can be used as a module in task assignment, to estimate the confidence of estimated true values [5] [23].

In this paper, we focus on crowdsourcing *tabular data*, i.e., a collection of related items which are structured in a tabular form and comply to a schema. Each column represents a particular attribute or variable. Each row corresponds to an entity and includes a value for each of the variables. Table 1 illustrates an example about data collection of celebrities; given the name of a celebrity, the goal is to collect the nationality, age, and notability (range from 1 to 5) of the person from the crowd. The bold values shown in Table 1 are the unknown (ground) truth data to be collected from the workers. Each cell of this table can be considered as a task, i.e., a worker may be asked to provide a value for the nationality of a celebrity given his/her name. Our target is to complete an empty or partial-filled table by filling in the cells effectively. Crowdsourcing tabular data finds direct application in database cleaning and integration [15] [28] [29].

Table 1: Ground Truth about Celebrities

	Name	Nationality	Age	Notability
1	Leonardo DiCaprio	**United States**	**42**	**5**
2	Jet Li	**China**	**54**	**4**
3	James Purefoy	**Great Britain**	**53**	**3**

Table 2: Answers to Questions about Celebrities

Worker	Row Id	Nationality	Age	Notability
u_1	1	**United States**	**40**	**5**
	2	**China**	**45**	**3**
u_2	1	**United States**	**42**	**5**
	3	**Great Britain**	**53**	**3**
u_3	2	**China**	**50**	**4**
	3	**United States**	**35**	**2**

Most crowdsourcing systems assume that the set of tasks are homogeneous and independent. However, tasks in tabular data can be *heterogeneous* and *dependent* to each other, which makes effective crowdsourcing on them challenging.

First, the datatypes and domains of different attributes may vary. For example, in Table 1, the task "the nationality of Jet Li?" has a different datatype compared to the task "the age of DiCaprio?" (i.e., categorical vs. continuous). Even attributes of the same datatype may have different domains (e.g., Age vs. Notability). As a result, approaches for integrating the answers of a worker in different homogeneous tasks are not directly applicable. These include the popular EM algorithm [9] for categorical data and data integration

---

- *C. Shan R. Cheng and Z. Huang are with Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: chshan, ckcheng, zphuang@cs.hku.hk*
- *N. Mamoulis is with the Department of Computing Science, University of Ioannina, Ioannina, Epirus, Greece. E-mail: nikos@cs.uoi.gr*
- *G. Li is with the Department of Computing Science, TNList, Tsinghua University, Beijing, China. E-mail: liguoliang@tsinghua.edu.cn*
- *Y. Zheng is with Twitter Inc., San Francisco, California, United States. E-mail: yudianz@twitter.com*

models applied for continuous attributes (GTM [40] and CATD [20]), to be discussed in Section 2. As we will show, applying a different approach for each column does not transfer the knowledge from one datatype to the other, i.e., the estimation of worker quality can be inaccurate due to data sparsity.

Second, in tabular data, there are potential dependencies between rows and columns. The difficulty of a task might depend on the corresponding entity and attribute. As a result, the quality of a worker on a particular task may depend on her quality on other tasks in the same row or column. Take Table 2 as an example, where bold values are the answers of three workers on tasks from Table 1. Note that worker u_3 inputs a wrong nationality of James Purefoy, meaning that she might mistake this celebrity for someone else. Therefore, her answers for the age and notability of the person have high chance to be unreliable, despite the high quality of her input for the second row. This means that when we assign a new task to the coming worker, we should not only consider the worker's inherent quality, but also whether the worker is familiar with the entity (we call it the worker's structure-aware quality). Traditional task assignment methods [7] [18] focus on capturing the former but ignore the latter.

In this paper, we present *T-Crowd*, the first crowdsourcing system that considers heterogeneous and dependent of tabular data in both truth inference and task assignment. T-Crowd processes the submitted answers by each worker to infer a *unified quality* for him or her. T-Crowd seamlessly integrates the worker's answers to questions of different datatypes and domains, addressing consistency and data sparsity issues that would arise from the alternative approach of using different models for different columns. For example, the overall quality of worker u_2 can be regarded better than that of worker u_1 considering their answers to both categorical and continuous values in Table 2. Unified worker quality greatly improves truth inference and task assignment, reducing the total number of tasks to be assigned to workers until all true values can be estimated with high confidence.

T-Crowd captures the importance of tasks (i.e., how confident we are about their value estimates) in the different columns and rows, based on the collected data so far. We also define an *inherent information gain* which is a uniform measure for ranking tasks with respect to a given worker. Then we choose to assign to the worker the tasks with the highest anticipated benefit. In contrast, previous work [15] [29] on crowdsourcing tabular data performs task assignment based on only how many more answers are needed for each task, disregarding worker quality. To further improve performance, we utilize the potential correlations between tasks. We define a *structure-aware information gain* which extends the inherent information gain to also consider as a parameter the previous answers given by the worker on tasks that appear in the same row, when selecting new tasks to assign to him or her.

A preliminary version of this work, which focused on truth inference in crowdsourcing tabular data, appears in [33]. In this paper, in addition to truth inference, we study the task assignment problem. In addition, we evaluate the performance of our proposed task assignment approach on three real datasets and compare it with four competitors. Besides, we add several new and recent competitors for our proposed truth inference algorithms, such as CATD, Zencrowd, TC-onlyCate and TC-onlyCont. The latter two are the constrained versions of T-Crowd that apply only on the categorical or continuous attributes. Finally, we expand our case studies, add experiments with synthetic datasets, and include a comparison to CrowdFill [29].

To summarize, our main contributions are as follows:

- To the best of our knowledge, we are the first to study crowdsourcing tabular data with both heterogeneity and dependency.
- We unify worker quality for all tasks in crowdsourced tabular data, improving the accuracy of truth inference and the performance of task assignment, compared to models that treat each attribute independently.
- Given an incoming worker, we find suitable tasks for him/her based on *inherent information gain*, including the benefit of obtaining additional answers in tasks and the worker's inherent quality. We also extend it to *structure-aware information gain*, which considers the correlation of answer quality between tasks in the same row.
- We evaluate T-Crowd on real datasets; the results demonstrate its superiority over existing alternatives. Compared to previous work, T-Crowd has better truth inference accuracy and converges to the true values of the tasks using only about half of the answers by the workers.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 defines the problem and gives an overview of our system. In Section 4, we present our methodology for truth inference. Our task assignment policy is presented in Section 5. Section 6 includes our experimental evaluation. Finally, we conclude in Section 7.

2 RELATED WORK

Related work falls into two categories: *truth inference* methods used to infer the truth and *task assignment* strategies for an incoming worker.

Truth Inference. The most basic truth inference methods are majority voting for multiple-choice tasks (i.e., categorical data) and taking the median for numerical tasks (i.e., continuous data). These approaches regard all workers as equal, disregarding any differences in their trustworthiness. Methods such as D&S [9], [17] use a confusion matrix to model a worker's quality, and use an Expectation-Maximization (EM) algorithm to infer the truth. More advanced approaches such as TruthFinder [39], Accusim [12], and GLAD [38] improve accuracy using different worker answering models or by considering more parameters, such as a task's difficulty. These methods focus on answering tasks on categorical data. Other methods, such as GTM [40], are designed for continuous crowdsourced data. CRH [21], [22] and CATD [20] are two existing truth inference approaches for both categorical and continuous data. CRH [21] incorporates different distance functions between the answers and the estimated truth to recognize the characteristics of various data types. Specifically, CRH proposes an objective function and minimizes it by updating the estimated true values and source reliability (i.e., worker quality) in turns. CATD [20] considers both source reliability and the confidence interval of the estimation. Additional information of tasks or workers has also been considered in truth inference, such as the latent topics of the tasks [24] and the learn bias of workers [44].

The aforementioned works do not consider tabular data. In Section 4, we present an iterative Expectation-Maximization (EM) truth inference algorithm, which improves the accuracy of truth inference from the answers compared to previous work. The novelty of our work is that we use a probabilistic model for the answers of workers wrt different data types and that we unify workers' quality on categorical data and continuous data explicitly, while methods like CRH design different distance functions for the different data types.

Task Assignment. Online task assignment selects which tasks to assign to each incoming worker, in order to achieve the maximum possible quality for the collected data. In earlier crowdsourcing systems, such as CDAS [23], the candidate tasks are randomly assigned to workers. AskIt [5] is yet another crowdsourcing platform, which assigns the tasks that have the highest uncertainty, again disregarding the quality (or expertise) of the incoming worker for these tasks. CrowdDB [15], Deco [28], and Qurk [25] are extensions of relational database systems that incorporate the crowd's knowledge into query processing. They use answers from the crowd to make up the missing values of query operators. They are similar to our approach in that they collect tabular data; however, they do not focus on the assignment strategy and simply assign random tasks to workers. CrowdFill [29] is a recent system for tabular data, which uses a non-conventional workflow that is not supported by common crowdsourcing platforms such as AMT. In CrowdFill, workers are asked to select and perform tasks from a subset of the table given to them and they can also vote for the answers to these tasks by other workers. Besides, CrowdFill does not estimate worker quality, and does not use properties of tabular data (e.g., attribute dependencies) to assign tasks to workers. Some methods [14], [26], [41] consider the case where the tasks are relevant to different domains and workers are given the tasks that match their domain expertise. In recent work, such as OptKG [7] and CrowdDQS [18], task assignment is modeled by a Markov Decision process or solved by using maximum potential gain, but the application of these models is limited to only multiple-choice tasks (categorical tasks). Other forms of online task assignment, which need explicit workers' collaboration, have been studied in [31], [32]. Different from the above works, our method focuses on crowdsourced tabular data, which is structured and heterogeneous, presenting challenges and opportunities as discussed in the Section 1.

3 PROBLEM DEFINITION

In this section, we formulate the problem and give an overview of T-Crowd. Our goal is to perform crowdsourcing on a two-dimensional table C, defined as follows.

Table 3: Table of Notations

Notation	Description
c_{ij}	cell (task) in the i-th row and j-th column
a_{ij}^u	answer given by worker u for cell c_{ij}
$\mathcal{A}$	the set of all answers, i.e., $\mathcal{A} = \{a_{ij}^u\}$
T_{ij}	distribution of estimated truth for cell c_{ij}
T_{ij}^* ($\widehat{T}_{ij}$)	ground truth (estimated truth) for cell c_{ij}
e_{ij}^u	error of a_{ij}^u with respect to $\widehat{T}_{ij}$
q_u	quality of worker u
α_i (β_j)	difficulty of row i (column j)

Definition 1 (Tabular Data Model). We target the crowdsourcing of a two-dimensional table $C = \{c_{ij}\}$, where $i \in \{1, ..., N\}$ and $j \in \{1, ..., M\}$. C has an *entity* attribute which is the key attribute of the table. Each column is a categorical or a continuous attribute. Each cell c_{ij} represents the value of the i-th entity in the j-th attribute, whose true value (i.e., *truth*, or *ground truth*) is denoted as T_{ij}^*.

Table 1 shows an example of tabular data about celebrities that we want to crowdsource. *Age* and *Notability* are continuous attributes, while *Nationality* is categorical. The entity attribute is *Name*. To obtain the truth for the remaining attributes, we ask the crowd to provide answers.

Definition 2 (Task, Worker, Answer). A task is related to a cell c_{ij} and the workers are asked to answer the task, by providing values for the cell. Let U be a set of workers. A worker $u \in U$ will submit an answer a_{ij}^u, if cell c_{ij} is assigned to u.

For example, to get the age of the second entity, a task provides the name of the second entity and asks workers to input the age. Since workers may have different levels of quality (e.g., some workers are experts, while some are spammers), each task c_{ij} is often assigned to multiple workers and all acquired answers for c_{ij} are aggregated to infer the true value of c_{ij}. Next, we define the two problems that we aim to address in this paper.

Definition 3 (Truth Inference). Given the set of answers $\{a_{ij}^u\}$, by workers $u \in U$ to cells c_{ij}, $i \in \{1, ..., N\}$, $j \in \{1, ..., M\}$, the problem of truth inference is to compute an accurate estimate $\widehat{T}_{ij}$ for each cell c_{ij}'s true value T_{ij}^*.

Definition 4 (Task Assignment). When a worker u requests for a task for C, decide the task to be assigned to u.

Note that existing crowdsourcing platforms, such as the Amazon Mechanical Turk (AMT) [1], support the functionality of dynamically assigning tasks to an incoming worker (e.g., the 'external-HIT' feature in AMT [2]). Table 4 summarizes the notations used in this paper.

System architecture. Figure 1 gives an overview of T-Crowd, our proposed system for crowdsourcing tabular data. A *requester* (e.g., a lifestyle journal) first defines the structure (i.e., schema) of the collected data, such as the datatypes of the columns, and the key attribute. Then the requester publishes tasks to a crowdsourcing platform, e.g., AMT [1]. For an incoming worker u, our Task Assignment module determines one or more cells and assigns the corresponding task(s) to u. This is based on the anticipated

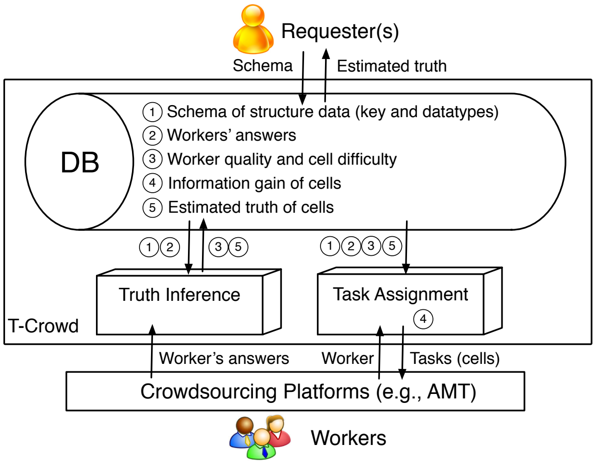


Figure 1: System Architecture

information gain of the different cells by u's answers. Intuitively, the information gain is an estimate of how much more accurate the cells' values become upon collection of u's inputs. When the worker submits an answer a_{ij}^u for a cell c_{ij} to the system, the Truth Inference module infers the estimated truth $\widehat{T}_{ij}$. To facilitate task assignment and truth inference, we also estimate the quality of worker q_u and the difficulty of cells α_i and β_j. Task(s) are assigned to workers and answers are collected until $\widehat{T}_{ij}$ converges (or a budget is exhausted).

4 TRUTH INFERENCE

In this section, we explain how T-Crowd performs truth inference on tabular data. The quality of truth inference for a data cell c_{ij} depends on the quality of workers who answer c_{ij}, and the difficulty of c_{ij}. We first discuss how to model worker quality q_u and cell difficulty $\alpha_i(\beta_j)$ if we already know the truth $\widehat{T}_{ij}$ (Sections 4.1). Then, we show how to infer the true values of cells $\widehat{T}_{ij}$ and these two factors simultaneously by maximizing the likelihood of workers' answers a_{ij}^u (Section 4.2).

4.1 Worker Model

4.1.1 Quality of a Worker

The challenge in modeling worker quality is that attributes may have different datatypes; the answer set of a categorical task is finite and nominal, while that of a continuous task is an integer or a real number. Hence, it is not straightforward to model the quality of a worker using a single parameter. To address this problem, we propose a unified model for both categorical and continuous attributes.

We model the truth of a categorical attribute l^* as an element in a finite unordered set of possible answers $L = \{l_1, l_2, ..., l_{|L|}\}$. An answer from a worker is either correct or wrong depending on whether it is the same as the ground truth. On the other hand, for a continuous attribute, the quality of the answer depends on how close it is to the ground truth. For example, if the age of Jet Li is 54, and a worker answers 53, which is close to the truth, the answer is considered to be a good one.

As discussed, our goal is to use a single parameter q_u to represent the quality of a worker u. For the ease of presentation, we first illustrate how the worker's quality for continuous datatypes can be modeled, and then show how the model can be extended for categorical datatypes.

- For **continuous** datatypes, we model the distribution of the answer given by worker u as a normal distribution: $a_{ij}^u \sim \mathcal{N}(\widehat{T}_{ij}, \phi_u)$:

$$P(a_{ij}^u = x) = \frac{1}{\sqrt{2\pi\phi_u}} \exp\left(-\frac{(x-\widehat{T}_{ij})^2}{2\phi_u}\right), \qquad (1)$$

where $\widehat{T}_{ij}$ is the expected value of c_{ij} and ϕ_u is the variance of u. Intuitively, the higher the quality of a worker is, the smaller the variance will be, as his/her answer should have smaller difference from the truth. Inspired by this, we model $q_u \in [0,1]$ as the probability that the answer from worker u falls into a small range (ϵ) around the truth $\widehat{T}_{ij}$:

$$q_u = P(\; a_{ij}^u \in [\widehat{T}_{ij} - \epsilon, \widehat{T}_{ij} + \epsilon] \;) = \text{erf}(\epsilon/\sqrt{2\phi_u}). \qquad (2)$$

Intuitively, q_u is the area under the normal distribution curve, where ϵ is a general parameter that controls the shape of the area and "erf" is the Gauss error function [4].

- For **categorical** attributes, $q_u \in [0,1]$ indicates the probability that the worker u would correctly answer a task, i.e.,

$$P(a_{ij}^u = z) = (q_u)^{\mathbb{1}_{\{\widehat{T}_{ij}=z\}}} \cdot \left(\frac{1-q_u}{|L|-1}\right)^{\mathbb{1}_{\{\widehat{T}_{ij}\neq z\}}}, \qquad (3)$$

where $\mathbb{1}_{\{.\}}$ is an *indicator function* which returns 1 if the argument is true; 0, otherwise. For example, $\mathbb{1}_{\{5=5\}} = 1$ and $\mathbb{1}_{\{5=3\}} = 0$. Intuitively, worker u has probability q_u to give the correct answer and we evenly distribute the probability $(1 - q_u)$ to the remaining (false) answers. Note that q_u can be expressed as in Equation 2, which means that we can use the same quality measure for categorical and continuous attributes.

4.1.2 Difficulty of a Cell

The answers from workers do not only depend on their expertise, but they are also influenced by the difficulty of tasks. Hence, in our model, the quality of answer a_{ij}^u depends on the quality of worker u, the difficulty β_j of attribute (i.e., column) j, and the difficulty α_i of entity (i.e., row) i.

To incorporate the difficulty of each cell c_{ij} into the worker's quality, we define the variance of his/her answer to a cell c_{ij} as $\phi_{ij}^u = \alpha_i \beta_j \phi_u$. Hence, the variance is positively correlated to the difficulties α_i and β_j, and the inherent variance (ϕ_u) of answers by worker u. Then, following Equation 2, we represent the quality of worker u answering cell c_{ij} as $q_{ij}^u = \text{erf}\left(\epsilon/\sqrt{2\alpha_i\beta_j\phi_u}\right)$. To model the worker's answers on categorical and continuous data, Equations 1 and 3 can be changed accordingly, i.e., by replacing ϕ_u with ϕ_{ij}^u and q_u with q_{ij}^u.

Note that $\widehat{T}_{ij}$, α_i, β_j and ϕ_u are unknown and we discuss how to compute them later. The worker quality q_u (q_{ij}^u) can be calculated directly if we know α_i, β_j, and ϕ_u.

4.2 Inference Process

The objective function of the truth inference problem is to maximize the likelihood of workers' answers, i.e.,

$$\arg\max_{\alpha,\beta,\phi} P(\mathcal{A}|\alpha,\beta,\phi) = \arg\max_{\alpha,\beta,\phi} \sum\nolimits_{\mathcal{T}} P(\mathcal{A},\mathcal{T}|\alpha,\beta,\phi),$$

where $\mathcal{A}$ is the current set of answers by all workers on all cells and $\mathcal{T}$ is a set of all hidden true values, i.e., $\mathcal{T} = \{T_{ij}\}$. T_{ij} denotes the estimated distribution of truth in cell c_{ij}. To optimize this non-convex function, we use the Expectation-Maximization (EM) algorithm [11], which takes an iterative approach. In each iteration of EM, the E-step computes the hidden variables in $\mathcal{T}$, and the M-step computes the parameters α_i, β_j and ϕ_u (q_u). Next, we provide details about the E-step and the M-step.

Expectation Step (E-step). In the E-step, we compute the posterior probabilities of hidden variable $T_{ij} \in \mathcal{T}$ given the values of α, β and ϕ and the observed variable $A_{ij} = \{a^u_{ij}\}, u \in U_{ij}$, i.e., the current answer set of cell c_{ij}.

$$\begin{aligned} &P(\, T_{ij} = z | A_{ij}, \alpha_i, \beta_j, \phi \,) \propto \\ &\prod\nolimits_{u \in U_{ij}} P(a^u_{ij} | T_{ij} = z, \alpha_i, \beta_j, \phi_u) \cdot \mathrm{Prior}(T_{ij} = z). \end{aligned} \tag{4}$$

Based on our defined worker model of $P(T_{ij} = z | A_{ij}, \alpha_i, \beta_j, \phi)$ for different datatypes, the distribution is defined as follows.

(1) For cells c_{ij} of continuous type, we regard that $\mathrm{Prior}(T_{ij} = z)$ follows a normal distribution $\mathcal{N}(\mu^0_j,\ \phi^0_j)$, and $T_{ij} \sim \mathcal{N}(T^{\mu}_{ij},\ T^{\phi}_{ij})$, where T^{μ}_{ij} and T^{ϕ}_{ij} satisfy that

$$\begin{aligned} T^{\mu}_{ij} &= (\sum\nolimits_{u \in U_{ij}} \frac{a^u_{ij}}{\alpha_i \beta_j \phi_u} + \frac{\mu^0_j}{\phi^0_j})\; T^{\phi}_{ij}, \\ T^{\phi}_{ij} &= (\sum\nolimits_{u \in U_{ij}} \frac{1}{\alpha_i \beta_j \phi_u} + \frac{1}{\phi^0_j})^{-1}. \end{aligned}$$

(2) For cells c_{ij} of categorical type, we have

$$P(T_{ij} = z) = \frac{\prod_{u \in U_{ij}} [(q^u_{ij})^{\mathbb{1}_{\{a^u_{ij} = z\}}} (\frac{1 - q^u_{ij}}{|L_j| - 1})^{\mathbb{1}_{\{a^u_{ij} \neq z\}}}]}{\sum_{z \in L_j} \prod_{u \in U_{ij}} [(q^u_{ij})^{\mathbb{1}_{\{a^u_{ij} = z\}}} (\frac{1 - q^u_{ij}}{|L_j| - 1})^{\mathbb{1}_{\{a^u_{ij} \neq z\}}}]},$$

where q^u_{ij} is defined as $\mathrm{erf}\left(\epsilon / \sqrt{2\alpha_i \beta_j \phi_u}\right)$ and L_j is the label set of column j. $\mathrm{Prior}(T_{ij} = z)$ is uniform so it disappears.

Intuitively, the answer given by high quality worker will be trusted more, i.e., given higher weight. To be specific, we estimate the truth distribution T_{ij} by combining the set A_{ij} of workers' answers for c_{ij}. (1) T^{μ}_{ij} can be regarded as a weighted average of answer a^u_{ij} based on the quality $\alpha_i \beta_j \phi_u$. T^{ϕ}_{ij} is a normalized term. (2) Similarly, $P(T_{ij} = z)$ is a normalized product of the qualities q^u_{ij} of the workers whose answer a^u_{ij} is z.

Maximization Step (M-step). In the M-step, we find the values of parameters α, β and ϕ that maximize the expectation of the joint log-likelihood of the observed variable $\mathcal{A}$, as shown below:

$$\begin{aligned} Q(\alpha, \beta, \phi) &= \mathrm{E}_{\mathcal{T}}[\ln P(\mathcal{A}, \mathcal{T} | \alpha, \beta, \phi)] \\ &= \sum_j \sum_i \mathrm{E}_{T_{ij}} \left[\ln \mathrm{Prior}(T_{ij}) + \sum_{u \in U_{ij}} \ln P(a^u_{ij} | T_{ij}, \alpha_i, \beta_j, \phi_u) \right]. \end{aligned} \tag{5}$$

Formula $\mathrm{E}_{T_{ij}}[\sum_{u \in U_{ij}} \ln P(a^u_{ij} | T_{ij}, \alpha_i, \beta_j, \phi_u)]$ is calculated for the different datatypes, as follows.

(1) For cells c_{ij} of continuous type:

$$\sum\nolimits_{u \in U_{ij}} [-\frac{1}{2} \ln(2\pi \alpha_i \beta_j \phi_u) - \frac{(a^u_{ij} - T^{\mu}_{ij})^2 + T^{\phi}_{ij}}{2\alpha_i \beta_j \phi_u}].$$

(2) For cells c_{ij} of categorical type:

$$\begin{aligned} &\sum\nolimits_{z \in L_j} P(T_{ij} = z) \cdot \sum\nolimits_{u \in U_{ij}} \big(\mathbb{1}_{\{a^u_{ij} = z\}} \ln \mathrm{erf}(\frac{\epsilon}{\sqrt{2\alpha_i \beta_j \phi_u}}) \\ &\qquad + \mathbb{1}_{\{a^u_{ij} \neq z\}} \ln \frac{1 - \mathrm{erf}(\frac{\epsilon}{\sqrt{2\alpha_i \beta_j \phi_u}})}{|L_j| - 1} \big). \end{aligned}$$

We apply gradient descent to find the values of α, β and ϕ that locally maximize $Q(\alpha, \beta, \phi)$.

Intuitively, a worker will be of high quality if his/her answers are close to the estimated truth. Thus, we compute a value ϕ_u that maximizes the expectation of the log-likelihood of worker u's answers a^u_{**}. Similarly, we also find an α_i (resp. β_j) that maximizes the expectation of the log-likelihood of answers a^*_{i*} in row i (resp. a^*_{*j} in column j).

Algorithm. By combining the two steps above, we can iteratively update the parameters until convergence. Each T_{ij} is initialized by following the distribution in $\mathrm{Prior}(T_{ij})$. At each iteration, the M-step applies gradient descent to find α_i, β_j and ϕ_u by maximizing Equation 5 and the E-step applies Equation 4. We identify convergence if the differences between the parameter values in subsequent iterations are below a threshold (e.g., 10^{-5}).

Finally we estimate the truth $\widehat{T}_{ij}$ of each cell c_{ij} as:

$$\widehat{T}_{ij} = \begin{cases} T^{\mu}_{ij} & ,\ c_{ij} \text{ is continuous,} \\ \arg\max_{z \in L_j}\ P(T_{ij} = z) & ,\ c_{ij} \text{ is categorical.} \end{cases}$$

Time Complexity. The total cost of the E-step is $\mathcal{O}(l \cdot |\mathcal{A}|)$, where $\mathcal{A}$ is the set of all obtained answers and $l = \max_j(|L_j|)$. In the M-step, one gradient descent needs to compute the gradient of each parameter which takes $\mathcal{O}(l \cdot |\mathcal{A}|)$. If the gradient descent takes v iterations to converge, this step takes $\mathcal{O}(vl \cdot |\mathcal{A}|)$ time in total. Assuming that the algorithm needs w iterations to converge, the total time complexity is $\mathcal{O}(wvl \cdot |\mathcal{A}|)$. In practice, l is constant, and v and w are smaller than 20, thus the time complexity is linear to the number of answers.

Algorithm 1: *Truth Inference Method*

```
Input: workers' answers $a^u_{ij} \in \mathcal{A}$, prior distribution of truth $\mathrm{Prior}(T_{ij})$
Output: truth distribution $T_{ij} \in \mathcal{T}$, worker's quality $\phi_u$, difficulty of
        row $\alpha_i$ and column $\beta_j$
Initialize $T_{ij}$ using $\mathrm{Prior}(T_{ij})$
while true do
    // Step 1: Estimate Worker Quality and Cell Difficulty
    Compute $\alpha_i$, $\beta_j$ and $\phi_u$ maximizing Eq. 5;
    // Step 2: Infer the Truth
    for $1 \leq i \leq N$ do
        for $1 \leq j \leq M$ do
            Obtain $T_{ij}$ by Eq. 4;
    // Check for Convergence
    if Converged then
        break;
return $T_{ij}$, $\alpha_i$, $\beta_j$ and $\phi_u$;
```

5 ONLINE TASK ASSIGNMENT

In this section, we discuss how we select tasks for a worker u. Section 5.1 defines an *inherent information gain* function to measure the utility of assigning a task to the worker, which can handle both categorical and continuous data. The function considers the quality of the worker, the need to obtain more answers for the task, and the task's difficulty. Intuitively, we prefer to assign tasks whose gain of information will be improved the most if the incoming worker answers them. In Section 5.2, we extend this to a *structure-aware information gain* function, which also considers the correlations in the qualities of answers given by the same worker to different cells of the same row.

5.1 Inherent Information Gain

We need a uniform measure for the *utility* (or benefit) of assigning a task (either categorical or continuous) to a worker u with quality q_u. For this purpose we define an *inherent information gain* function, following the steps below. (1) For a categorical cell c_{ij}, the distribution of truth T_{ij} has been computed by $P(T_{ij} = z)$ in Equation 4, which is the probability that label z is correct. Thus, Shannon Entropy [3] can be used to define the uncertainty of task c_{ij}:

$$H_s(T_{ij}) = -\sum\nolimits_{z \in L_j} P(T_{ij} = z) \ln P(T_{ij} = z).$$

(2) For a continuous cell c_{ij}, note that for a continuous distribution, the Differential Entropy [27] is defined as:

$$-\int_{\mathbb{X}} f(x) \ln f(x)\, dx,$$

where $f(x)$ is a probability distribution. Recall that we also define the distribution of truth $T_{ij} \sim \mathcal{N}(T^{\mu}_{ij}, T^{\phi}_{ij})$ of a continuous cell c_{ij} in Equation 4, so its Differential Entropy can be computed as:

$$H_d(T_{ij}) = \frac{1}{2} \ln \left(2\,\pi\, e\, T^{\phi}_{ij}\right).$$

Given the above, we define the *uniform entropy* for task c_{ij}:

$$H(T_{ij}) = \begin{cases} H_d(T_{ij}), & \text{if } c_{ij} \text{ is continuous,} \\ H_s(T_{ij}), & \text{if } c_{ij} \text{ is categorical.} \end{cases}$$

A straightforward approach for task assignment to a worker u is to select the task c_{ij} with the largest uniform entropy. However, this is problematic, as Differential Entropy and Shannon Entropy are not comparable; hence, task assignments may be biased toward one datatype. For example, as pointed out in [27], Differential Entropy can be negative while Shannon entropy is always non-negative. Alternatively, we use Delta Entropy to measure the information gain. Suppose $\mathcal{A}_C$ is the current set of answers we have collected, we can obtain the estimated truth distribution (denoted as $T_{ij,\mathcal{A}_C}$) for each task c_{ij} by the truth inference method presented in Section 4. Specifically, for an incoming worker u, we define the *inherent information gain* of assigning task c_{ij} to her as:

$$IG_q(c_{ij}) = H(T_{ij,\mathcal{A}_C}) - \mathrm{E}_{a^u_{ij}}[H(T_{ij,\mathcal{A}_C \cup \{a^u_{ij}\}})], \qquad (6)$$

where $T_{ij,\mathcal{A}_C \cup \{a^u_{ij}\}}$ is the updated distribution of the estimated truth for task c_{ij} after receiving a new answer a^u_{ij} from u.

By using the inherent information gain measure defined in Equation 6, we alleviate the problem that the domains of the two entropy types are different. If we discretize the range of a continuous random variable X using bins of width Δ, we can compute the Shannon entropy for this new discretized random variable X^{Δ}, and we have the following formula if X's pdf is Riemann integrable:

$$H_s(X^{\Delta}) + \ln \Delta \rightarrow H_d(X),\ as\ \Delta \rightarrow 0.$$

Hence, if Δ is small, $H_d(X_1) - H_d(X_2) \approx H_s(X_1^{\Delta}) - H_s(X_2^{\Delta})$, which means that the subtraction of two differential entropies can be transformed into subtraction of two Shannon entropies. As a result, for cells of different types, $IG(c_{ij})$ is comparable. Algorithm 2 describes the task assignment algorithm in detail.

Algorithm 2: *Online Task Assignment Method*

```
Input: Budget $B$
Output: truth distribution $T_{ij} \in \mathcal{T}$
Initialize each task with several answers from workers
while Budget $B$ is not exhausted do
    // Step 1: Analyze current situation
    Run truth inference to obtain $T_{ij}$, $\alpha_i$, $\beta_j$ and $\phi_u$
    // Step 2: Find task $c^*$ with highest benefit for incoming worker $u$
    for $1 \leq i \leq N$ do
        for $1 \leq j \leq M$ do
            Compute information gain $IG(c_{ij})$ by Eq.6
            if $IG(c_{ij}) > IG(c^*)$ or $c^*$ is not defined then
                $c^* = c_{ij}$
    // Step 3: Collect answers
    Publish task $c^*$ and collect worker $u$'s answer
Run truth inference to obtain the final $T_{ij}$
return $T_{ij}$
```

Computing the distribution of $\mathrm{E}_{a^u_{ij}}[H(T_{ij,\mathcal{A}_C \cup \{a^u_{ij}\}})]$**.** The distribution of an answer a^u_{ij} follows the worker model in Equations 1 and 3 for continuous and categorical tasks, respectively. For a categorical task c_{ij}, the domain of a^u_{ij} is a finite label set, so we use all possible values a^u_{ij} to obtain $T_{ij,\mathcal{A}_C \cup \{a^u_{ij}\}}$ using the inference method described in Section 4. For a continuous task, since the the domain of a^u_{ij} is $\mathcal{R}$, we apply sampling to approximate the value of $T_{ij,\mathcal{A}_C \cup \{a^u_{ij}\}}$. However, it is expensive to run the inference method for each possible answer. To alleviate this problem, we limit the number of iterations per answer, by only updating the parameters related to the answer and keeping the other parameters unchanged. Specifically, for a new answer a^u_{ij}, we locally update the truth distribution T_{ij}, and the qualities of workers who have answered task c_{ij}.

Time Complexity. To compute the benefit for each task c_{ij} (Equation 6), we should first iterate through the possible answers given by the incoming worker and compute a new distribution of truth T_{ij}. The number of possible answers for a categorical task c_{ij} is $|L_j|$ and for a continuous task is the fixed sampling number s_{cont}. Because we approximate the inference method, it only takes $\mathcal{O}(l \cdot |P|)$ where P is the set of parameters we need to update. Let $s = \max(\max_j(|L_j|), s_{\text{cont}})$; the total cost of considering one task for a certain worker is $\mathcal{O}(sl \cdot |P|)$. Then, computing the information gains of all tasks takes $\mathcal{O}(NMsl \cdot |P|)$. Since P includes the truth distribution T_{ij} and the qualities of workers who have answered task c_{ij}, P mainly depends on the average answers per task. Thus, $\mathcal{O}(NMsl \cdot |P|) \approx \mathcal{O}(sl \cdot |\mathcal{A}|)$.

Parallel or distributed computation can be used to accelerate task assignment, as the consideration of the different tasks are independent.

Table 4: Table of Notations in Task Assignment

Notation	Description
e_{ij}^u	error of answer a_{ij}^u with respect to $\widehat{T}_{ij}$
E_j	distribution of error in column j
$P(E_j\|E_k)$	correlation of error between column j and k
w_{jk}	correlation coefficient between column j and k

5.2 Structure-Aware Information Gain

The task assignment approach based on inherent information gain, described in Section 5.1, does not utilize the structural information of table C. We now propose a structure-aware task assignment method. The basic idea is to estimate correlation, i.e., the conditional distribution of the error on a task c_{ij}, given the errors on other tasks c_{i*} in the same row. For this, we consider the answer history of all workers and then use the conditional distribution to obtain a better estimation of the target worker u's error on task c_{ij}.

We have already shown how to estimate the truth $\widehat{T}_{ij}$ for each cell c_{ij} in Section 4. Based on it, we can transform answer a_{ij}^u into error e_{ij}^u. For a continuous attribute, $e_{ij}^u = a_{ij}^u - \widehat{T}_{ij}$, while for a categorical attribute, $e_{ij}^u = \begin{cases} 0 & , a_{ij}^u = \widehat{T}_{ij}, \\ 1 & , a_{ij}^u \neq \widehat{T}_{ij}. \end{cases}$ It is easy to regain answer a_{ij}^u from error e_{ij}^u by reversing the according equation.

We regard $P(E_j|E_k)$ as the correlation of error between column j and k. We estimate $P(E_j|E_k)$ with a maximum likelihood method considering all the answers a_{*j}^* and a_{*k}^* we have collected, which is discussed later. If worker u has answered task c_{ik} before, his/her error for task c_{ij} is recomputed as $P\left(E_j|E_k = e_{ik}^u\right)$. When worker u has answered multiple tasks $L_i^u = \{k|$ worker u answered task c_{ik} on row $i\}$, we need to consider all the observed errors. However, it is not practical to estimate the conditional distribution, given errors from multiple attributes, due to data sparsity. Hence, we consider a linear combination of the correlations, as follows:

$$\frac{\sum_{k\in L_i^u} w_{jk}\cdot P\left(E_j|E_k = e_{ik}^u\right)}{\sum_{k\in L_i^u} w_{jk}} \tag{7}$$

where w_{jk} is the correlation coefficient between attribute j and k:

$$w_{jk} = \frac{(M_j - \overline{M}_j)(M_k - \overline{M}_k)}{\sqrt{(M_j - \overline{M}_j)^2}\sqrt{(M_k - \overline{M}_k)^2}}, \tag{8}$$

where M_j and M_k are the error vector on attribute j and k combined by the pair data $\{(e_{ij}^u, e_{ik}^u)|$ error of answers a_{ij}^u and a_{ik}^u when they are both existed$\}$. $\overline{M}_j$ and $\overline{M}_k$ are also vectors, where each element is the mean of vector M_j and M_k, respectively.

After obtaining the conditional distribution of error e_{ij}^u, we transform error e_{ij}^u into answer a_{ij}^u by reverse operations described above. Then, we calculate $\mathrm{E}_{a_{ij}^u}[H(T_{ij,\mathcal{A}_C\cup\{a_{ij}^u\}})]$ based on new answer distribution a_{ij}^u while $H(T_{ij,\mathcal{A}_C})$ is not changed. Accordingly, the structure-aware information gain $IG_c(c_{ij})$ is calculated using Eq. 6.

Computing the Correlation $P(E_j|E_k)$. Correlation is defined as the conditional probability between column j and k and it is derived from the known errors e_{*j}^* and e_{*k}^*.

(1) Marginal distribution $P(E_j)$. A categorical column is regarded as a Bernoulli distribution while a continuous column is regarded as a normal distribution.

(2) Conditional distribution $P(E_j|E_k)$. Since we have categorical and continuous columns, we have four cases in total. For each case, we use the maximal likelihood method to estimate the parameters in the assumed distribution. We elaborate on these cases below:

i. both j and k are categorical: $P(E_j = 1|E_k = 0)$, $P(E_j = 0|E_k = 0)$, $P(E_j = 1|E_k = 1)$ and $P(E_j = 0|E_k = 1)$ are counted based on the occurrences.

ii. both j and k are continuous: Because errors in continuous columns follow normal distributions, joint distribution $P(E_j, E_k)$ is a bivariate normal distribution. If the mean vector is $\begin{pmatrix}\mu_j\\ \mu_k\end{pmatrix}$ and the covariance matrix is $\begin{pmatrix}\sigma_j^2 & \rho\sigma_j\sigma_k\\ \rho\sigma_j\sigma_k & \sigma_k^2\end{pmatrix}$, the conditional distribution $P(E_j|E_k)$ is also a normal distribution

$$P(E_j|E_k = e_{ik}^u) \sim \mathcal{N}(\mu_j + \frac{\sigma_j}{\sigma_k}\rho(e_{ik}^u - \mu_k),\ (1-\rho^2)\sigma_j^2).$$

iii. column k is categorical and column j is continuous: We assume that the conditional distributions $P(E_j|E_k = 0)$ and $P(E_j|E_k = 1)$ obey normal distributions. We obtain the mean and variance when $E_k = 0$ or $E_k = 1$ separately.

iv. column j is categorical and column k is continuous: Based on the same assumptions as in case (iii), we can estimate $P(E_k|E_j = 0)$ and $P(E_k|E_j = 1)$. Because we also know $P(E_k)$ and $P(E_j)$, the conditional distributions can be calculated using Bayes' theorem

$$P(E_j|E_k = e_{ik}^u) = \frac{P(E_k = e_{ik}^u|E_j)P(E_j)}{P(E_k = e_{ik}^u)}$$

Time Complexity. To compute the correlation $P(E_j|E_k)$, we should iterate through each column and calculate the corresponding conditional distribution. Because there are M columns, the total cost is $\mathcal{O}(M\cdot|\mathcal{A}|)$. The same time is needed to calculate the correlation coefficient W_{jk}. The cost of computing the benefit of each task is the same as that of computing the Inherent Information Gain, which is discussed before. In total, the cost is $\mathcal{O}((M + sl)\cdot|\mathcal{A}|)$.

Assigning Multiple Tasks to Workers. So far we focused on how to select one task to assign to the incoming worker. This does not restrict the applicability of our approach in the case that multiple tasks should be determined and given to the worker as a batch (e.g., as in a HIT on AMT [1]). Suppose that the worker is to be assigned a set $D = \{c_{i_1j_1}, c_{i_2j_2}, \cdots, c_{i_Kj_K}\}$ of K tasks. From the set $\mathcal{A}_\mathcal{D} = \{a_{i_1j_1}^u, a_{i_2j_2}^u, \cdots, a_{i_kj_k}^u\}$ of estimated answers to the tasks by the worker, we can update the distribution of the estimated truth $T_{ij,\mathcal{A}_C\cup\mathcal{A}_\mathcal{D}}$ for each task $c_{ij}\in D$. Then, we can calculate the information gain for D as:

$$IG(D) = \sum\nolimits_{c_{ij}\in D}\left(H(T_{ij,\mathcal{A}_C}) - \mathrm{E}_{\mathcal{A}_\mathcal{D}}[H(T_{ij,\mathcal{A}_C\cup\mathcal{A}_D})]\right). \tag{9}$$

Because the search space of D is $\binom{N\cdot M}{K}$, finding K tasks which maximize $IG(D)$ is expensive. To alleviate the cost, we can apply a greedy approach that iteratively selects the top-K tasks with the largest $IG(c_{ij})$.

Table 5: Statistics of Real-world Datasets

Dataset	#Rows	#Columns	#Cells	# Ans. per Task (Avg)
Celebrity	174	7	1218	5
Restaurant	203	5	1015	4
Emotion	100	7	700	10

6 EXPERIMENTS

This section presents our experimental results. We present the datasets used in Section 6.1. In Sections 6.2 and 6.3, we compare different crowdsourcing solutions in terms of truth inference and task assignment respectively. We perform case studies in Section 6.4. Results on synthetic datasets are shown in Section 6.5. We measure the efficiency in Section 6.6 and do an extra comparison to CrowdFill in Section 6.7. We have implemented a prototype of T-Crowd and other crowdsourcing solutions in Python 2.7, on a Ubuntu server with 8-core Intel(R) Core(TM) i7-3770 CPU @ 1.60GHz cores and 16 GB memory.

6.1 Datasets

We use three real datasets to perform our experiments. Their statistics are shown in Table 5.

Celebrity [6]. This dataset contains information about celebrities. Given a celebrity's picture, workers are requested to provide the following attribute values: name, age, height, nationality, ethnicity, notability, and sentiment of the celebrity in the picture. Name, nationality, and ethnicity are categorical, age, height, notability, and sentiment are continuous. For each entity, the ground truth for name and age are obtained from [6], while that of height, nationality and ethnicity is extracted from IMDb. Notability and Sentiment are the subjective attributes. Our marked ground truth for subjective questions is the answer that agrees with most persons' opinion. The ground truth of sentiment is labeled by three movie experts ranging from 1 to 5 based on facial expressions. The experts followed the measurement of the facial action coding system (FACS) and the emotional facial action system (EMFACS) [13], [16], [35]. This includes 44 action units and combinations of FACS action units represent prototypic expressions of emotion. The positive expressions (happy or relaxed) take 5, the neutral take 3 and the negative ones (sad, fear or angry) take 1. This measurement has been researched in field of facial expression analysis in many years and it can be regarded as objective. Notability is also a value in the range of 1-5 and is obtained from the person's rank in IMDb, i.e., we map 1-200 as 5, 201-400 as 4, 401-600 as 3, 601-800 as 2 and >800 as 1. Since ratings in IMDb are solicited from numerous people, we believe it represents the opinions of the majority.

Restaurant [30]. This dataset contains information about restaurants. Given a review about a certain restaurant, workers are asked to specify the aspect (e.g., food or location), attribute (e.g., price or style), and sentiment (e.g., negative or positive) of the review. They are asked to identify the target (i.e., the restaurant referred by the review) by the starting and end position of its first occurrence in the text. Here, aspect, attribute, and sentiment are categorical; the starting and end positions are continuous. The reviews and the ground truth are obtained from [30].

Emotion [34]. This dataset collects scores for different emotions from a small piece of text. Each worker is asked to give a number in [0,100] for each of the following six emotions: anger, disgust, fear, joy, sadness, and surprise, and a single numeric rating in the interval [-100,100] for her overall (positive or negative) sentiment about the text. Here, all the 7 attributes are continuous. The workers' answers and the ground truth are provided by [34].

For the Celebrity and Restaurant datasets, we collected the workers' answers using AMT [1]. The average number of answers for each task in Celebrity and Restaurant is 5 and 4, respectively, by different workers. We spent $\$0.05$ per HIT where the number of tasks put in a HIT is the same as the number of columns (total cost $\$43.5$ and $\$40.6$, respectively). For Emotion, we use the workers' answers from [34]; each task is answered 10 times. We observed that for all continuous attributes the collected values (excluding spam answers) follow a normal distribution, which is consistent with our assumption in Section 4.1.

6.2 Truth Inference

We select some important other existing solutions based on the guidance from [42], [19] and study the effectiveness of our truth inference approach:

(1) For both categorical and continuous data:

- T-Crowd is our method proposed in Section 4. *TC-onlyCate* and *TC-onlyCont* are the constrained versions of T-Crowd that apply only on the categorical or continuous attributes.
- CRH [21] detects truth from heterogeneous data types by minimizing a loss function.
- CATD [20] detects truth from multi-source data that follows a long-tail distribution along with confidence intervals.

(2) For categorical data only:

- Majority Voting (MV) determines the correct labels based on the majority of answers from workers.
- D&S [9] iteratively estimates each worker's confusion matrix, which is used to infer the correct labels.
- GLAD [38] is a probabilistic approach for crowdsourcing categorical data.
- Zencrowd [10] is a variant of D&S.

(3) For continuous data only:

- Median uses the median of workers' answers as the estimated true value.
- GTM [40] is a truth-finding method specially designed for continuous data.

Effectiveness Measures. We adopt the following measures, proposed in [21], for evaluating the effectiveness of truth inference on categorical and continuous data items:

- Error Rate: For categorical data, we measure the Error Rate by computing the percentage of mismatched values between each method's predicted truth and the ground truth.
- MNAD (Mean Normalized Absolute Distance): It is the root of mean squared distance (RMSE) between

Table 6: Effectiveness of Truth Inference

Method	Celebrity		Restaurant		Emotion
	Error Rate	MNAD	Error Rate	MNAD	MNAD
T-Crowd	**0.0441**	**0.6339**	**0.1855**	**0.5607**	**0.5961**
CRH	0.0460	0.6737	0.1921	0.5835	0.7224
CATD	0.0498	0.7113	0.1954	0.7234	0.6648
Maj. Voting	0.0573	/	0.2003	/	/
EM	0.0620	/	0.2463	/	/
GLAD	0.0498	/	0.1905	/	/
Zencrowd	0.0479	/	0.1872	/	/
TC-onlyCate	0.0498	/	0.1986	/	/
Median	/	0.6998	/	0.6784	0.7026
GTM	/	0.6516	/	0.5871	0.6792
TC-onlyCont	/	0.6400	/	0.5682	**0.5961**

each method's estimated truth and the ground truth. Since different attributes have different scales, we normalize each attribute's RMSE by its own standard deviation and average them.

Effectiveness Comparison. In Table 6, we summarize the effectiveness of truth inference by all methods in terms of Error Rate and MNAD on the three real-world datasets. We observe that our proposed approach T-Crowd is better than all other methods both on categorical data and continuous data. On Celebrity, our method reduces the error rate by 4% on categorical data and the MNAD by 2.7% on continuous data compared to the best result of other methods. The corresponding reductions on Restaurant are 2.6% and 4%. On Emotion, we outperform previous work by 10%. CRH does not have stable performance as it is effective on Celebrity and Restaurant, but ineffective on Emotion. Similarly, CATD is good in terms of error rate but not good in terms of MNAD. Overall, our method is more robust than them.

We also test constrained versions of T-Crowd that apply only on the categorical or only on the continuous attributes. Note that the effectiveness of T-Crowd is better than that of its constrained versions and that the constrained versions are competitive compared to other methods in their class. In summary, T-Crowd outperforms truth inference approaches applied on categorical and continuous data separately. This result demonstrates the benefit of modeling worker quality by a probabilistic model in a unified manner for all datatypes.

6.3 Task assignment

We compare the effectiveness of task assignment by our approach against other crowdsourcing methods.

Competitors. We compare T-Crowd, which uses the truth inference method of Section 4.2 and the task assignment method in Section 5.2 with the following approaches:

- CDAS [23] measures the confidence of the currently estimated values of all tasks based on a quality-sensitive answering model. Each task for which we are already confident is "terminated" and no longer assigned to workers. At each step, CDAS selects at random a non-terminated task to assign to the incoming worker.
- AskIt! [5] uses an entropy-like method to define the uncertainty of each task, and infers the truth by Majority Voting. The task with the highest uncertainty is the next one to be assigned to the incoming worker.
- CRH [21] is an inference method suitable for heterogeneous data. It does not focus on task assignment, hence, tasks are randomly assigned to the incoming workers.
- CATD [20] is an inference method suitable for heterogeneous data, which does not focus on task assignment. Similar to CRH, we collected answers by randomly assigning tasks.

Effectiveness Measures. As in the evaluation of truth inference, we use Error Rate and MNAD to measure task assignment quality. Specifically, for each tested method, we measure the Error Rate and MNAD as a function of the average number of answers collected by task so far. A good method would be able to converge fast with fewer answers per task(i.e., by performing fewer assignments and hence spending less money). Besides, it should achieve a lower true value estimation error when it converges.

End-To-End Comparison. To perform a fair comparison with existing work, we performed experiments on AMT [1] by using the same settings for the different methods (i.e., each task costs the same). We use the 'external-HIT' [2] feature provided by AMT to dynamically assign tasks for the incoming worker. To assess the effectiveness of task assignment, we vary the budget and compare the Error Rate and MNAD of each method under the same budget. To be specific, for each budget, we record the error rate and MNAD on all real datasets as more answers are collected.

Figure 2 shows the experimental results. Naturally, the error rate and MNAD of all assignment policies decrease as more answers are received from the workers and converge to good results after a large number of answers. Askit! uses an entropy-like method, which makes it to prefer continuous tasks first. Thus its MNAD drops fast while the error rate remains high. After selecting all continuous tasks, its error rate starts to drop. Since no task is terminated in the first few iterations, CDAS converges slowly. In addition, since its inference method is simple, the final inferred result is not good compared to that of other methods. CRH and CATD are not probabilistic, which do not use metrics, like entropy or information gain, as the objective for task assignment, so they do not perform as well as T-Crowd. They are superior to Askit! and CDAS because they are more effective in inferring the true values of tasks.

We observe that T-Crowd converges much faster to a low error rate and MNAD compared to the other policies. Specifically, T-Crowd converges to low values before the average number of answers per task is 3 on Celebrity and Restaurant and 6 on Emotion, which shows the effectiveness of our structure-aware information gain measure as an assignment criterion. In addition, due to our superior truth inference method, the values eventually inferred by our framework are better compared to those inferred by the other methods.

6.4 Case Studies

We performed several case studies in order to assess the quality of our system. Due to space constraints, we only report the results on Restaurant. Similar observations can be derived by experimentation on the other datasets.

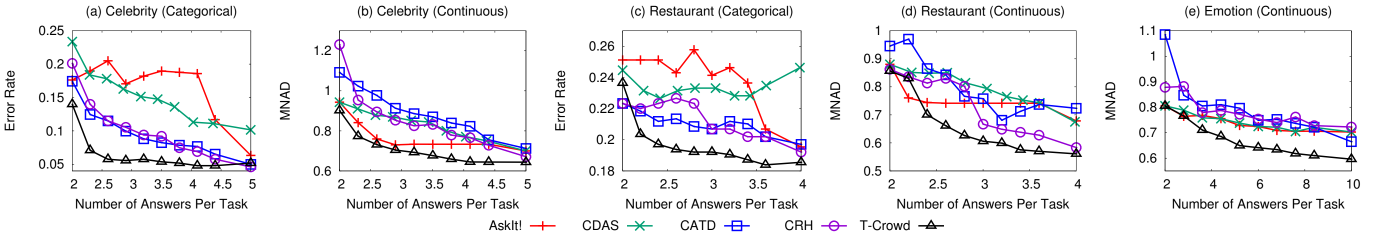


Figure 2: End-To-End System Comparison (Effectiveness)

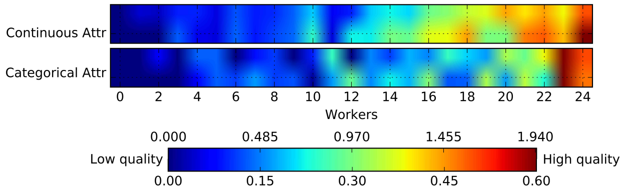
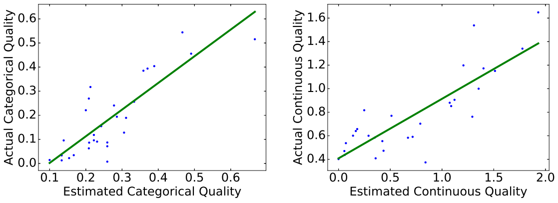


Figure 3: Uniform Worker Quality



(a) categorical (b) continuous

Figure 4: Estimated and Actual Worker Quality

6.4.1 *Worker Quality*

Our first study's goal is to show that (1) each worker's actual quality (computed based on the ground truth) is consistent among different attributes; (2) each worker's estimated quality can be well calibrated to the worker's actual quality.

Consistent Quality for Different Attributes. We collected statistics from the Restaurant dataset to support our assumption in truth inference: a worker has consistent quality over different datatypes of attributes. In Figure 3, we plot a heat map, with the x-axis representing the 25 workers who have given the largest number of answers and the y-axis representing categorical attributes 'Aspect' and 'Sentiment' and continuous attributes 'StartTarget' and 'EndTarget'. Different colors are aligned to standard deviation values (above the colorbar) for continuous attributes and error rates (below the colorbar) for categorical attributes. The color of each pixel represents the average error of answers given by worker u to the tasks on attribute j. For a categorical attribute j, the error is the percentage of wrong answers. For a continuous attribute j, the error is the standard deviation of the differences between the answers and the ground truth. The red color (far right) implies larger error and lower worker quality, while the blue color (far left) means smaller error and better worker quality. Note that the workers have consistent performance for categorical and continuous attributes. In addition, the colors for the same worker are similar regardless the attribute type, i.e., each worker's actual quality is consistent among different attributes.

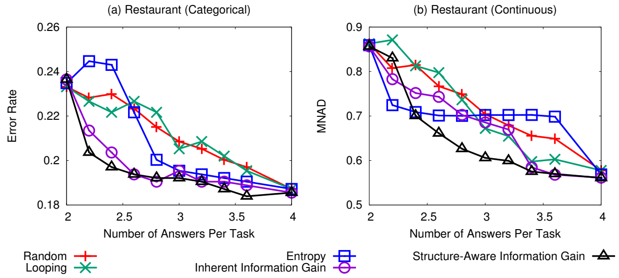


Figure 5: Effectiveness of Assignment Heuristics

Calibration to the Actual Quality. Figure 4 shows that our estimated quality of a worker is close to the actual quality. Each point represents a worker and the x-axis value is the quality estimated by our method while the y-axis value is the actual worker's quality. We also show the result of a linear regression. Observe the strong correlation between our estimation and actual quality; the correlation coefficient is 0.844 for categorical and 0.841 for continuous attributes.

6.4.2 *Assignment Heuristics*

We evaluate the performance of different assignment heuristics. Note that for all of them, we use our inference approach (Section 4.2). The tested heuristics are listed as follows:

- Random: it randomly chooses the task assigned to the worker.
- Looping: it selects the next task in a round-robin manner.
- Entropy: it greedily chooses the next task which has the highest uncertainty (defined as entropy).
- Inherent Information Gain: it proposed in Section 5.1.
- Structure-Aware Information Gain: it proposed in Section 5.2.

Figure 5 presents the Error Rate and MNAD as a function of number of tasks assigned to the workers on Restaurant. The results on the other datasets are similar and omitted for the interest of space. Random and Looping select tasks without considering the answers collected so far, so they converge slowly. Entropy is biased toward selecting continuous tasks over categorical first; hence, this heuristic reduces the MNAD fast, but not the Error Rate. Inherent and Structure-Aware Information Gain consider the continuous and categorical tasks fairly and decrease the Error Rate and MNAD simultaneously. Besides, Structure-Aware Information Gain converges faster than Inherent Information Gain w.r.t. MNAD because it also considers the correlations between attributes. Recall that we use Structure-Aware Information Gain as our default method.

Sentiment Aspect	correct	wrong
correct	589	90
wrong	98	35

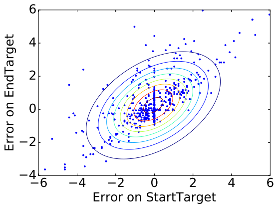


Figure 6: Correlation Among Attributes

6.4.3 *Correlation Among Attributes*

We perform one more experiment to support our assumption that there exist correlations among attributes, by analyzing the answers of workers.

Figure 6 shows the experimental results. In the left part of the figure observe that attributes 'Aspect' and 'Sentiment' have strong correlation. Specifically, if a worker answers 'Aspect' correctly, the probability to answer 'Sentiment' correctly is 86%. However, if a worker answers 'Aspect' wrongly, the probability to answer 'Sentiment' correctly is only 73%. In the right part of the figure, we plot a scatter diagram, with each point representing a worker's error on attributes 'StartTarget' and 'EndTarget'. We use *maximum likelihood estimation* to obtain the joint distribution of errors on these two attributes as described in Section 5.2. We observe a positive correlation between errors on attributes 'StartTarget' and 'EndTarget', which justifies our proposed Structure-Aware Information Gain method that considers correlations among attributes. For example, if the error of 'StartTarget' is 0, the distribution of 'EndTarget' error is $\mathcal{N}(0.28, 0.76)$. However, if the error of 'StartTarget' is 6, the distribution of 'EndTarget' error is $\mathcal{N}(3.75, 0.76)$. In other words, knowing the exact answer of a worker on one attribute can help to predict his/her answer distribution for other attributes better.

6.5 Synthetic Data

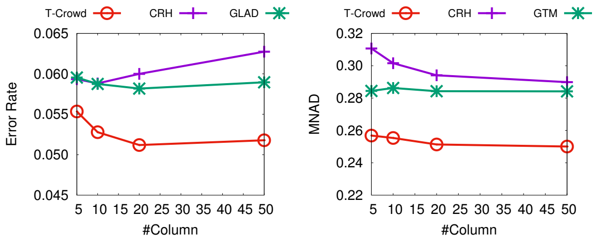
In this section, we use two types of synthetic data, in order to test the performance of our truth inference approach in cases not covered by the real data settings.

6.5.1 *Tests on tables with different properties*

We assess the performance of T-Crowd in terms of truth inference effectiveness by changing the following parameters of our data generator: the number of columns M, the ratio of categorical to the total number columns R and the average difficulty of tasks $\mu\{\alpha_i\beta_j\}$. The default parameters are $M = 10$, $R = 0.5$ and $\mu\{\alpha_i\beta_j\} = 1$. The rest of the settings are as follows:

Worker Sequence and Worker Quality: We use the same number of workers as that in our real experiments for the dataset Celebrity and assume that the workers arrive in the same sequence and that they have the same quality as in the real experiment.

Data and Ground Truth Generation: We implemented a generator for a table that takes as input the number of rows N and columns M, and the datatype and domain range of each column. The number of possible answers in a categorical column is generated from a uniform distribution $U(2, 10)$. The domain of a continuous column is $[0, 1000]$. The ground truth T^*_{ij} of each cell c_{ij} is generated by selecting a value in the corresponding domain randomly.



(a) Categorical Columns (b) Continuous Columns

Figure 7: Effect of the Number of Columns

Workers' Answers: For each worker in sequence, his answer at each cell needs to be generated. The answer a^u_{ij} of each worker u at each cell c_{ij} is created based on the ground truth T^*_{ij} and his quality q_u, based on Eq. 1 and 3.

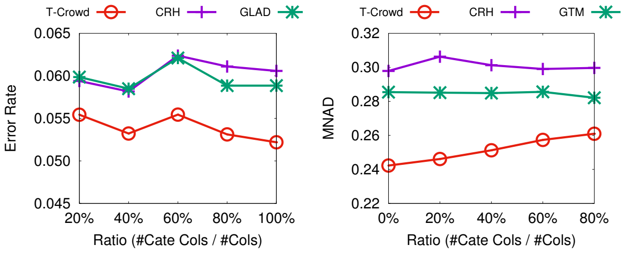
For fairness to all methods , we simulate the assignment strategy used in AMT, i.e., each task gets the same number of answers. For different parameters, we generate new datasets one hundred times and average the results to obtain the error rate and MNAD. We also run other inference methods and found that our method is dominant both on error rate and MNAD.

Results. In the first experiment, we vary the number of columns from 5 to 50. Figure 7 shows that the error rate and MNAD decline gradually when the number of columns increases, showing that T-Crowd infers the quality of each worker and estimates truth more accurate if we have more data. Besides our method is significantly better than the other two approaches. Next, we vary the ratio of categorical attributes from 0% to 100%. Figures 8(a) and Figure 8(b) show that our method's error rate and MNAD do not change much when the ratio varies. Finally, we vary the average difficulty of each cell c_{ij} (i.e., the average $\alpha_i\beta_j$, as defined in Section 4.1.2) from 0.5 to 3. High difficulty implies that the probability that workers answer correctly decreases, hence the error rate and MNAD increase as shown in Figure 9. For easier tasks, our method is significantly better than the others, but when the average difficulty is high, which means that the workers' answers are not credible, all methods perform badly.

6.5.2 *Noise in Workers' Answers*

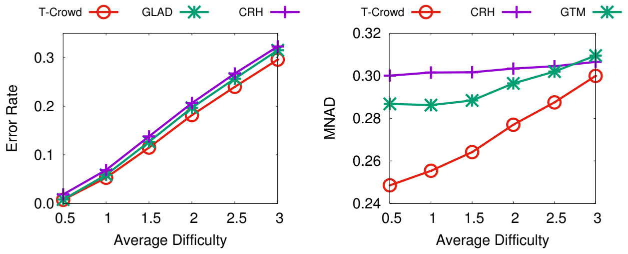
To further demonstrate the advantage of our proposed approach T-Crowd, we conduct simulation experiments by adding noise to the original data collected for Celebrity dataset. We vary the percentage γ of altered original answers by the workers from 10% to 40% (i.e., γ is the percentage of answers with added noise).

For a categorical answer, we randomly select a new label from its domain and replace the original label. For a continuous answer, Gaussian noise is added. We first preprocess this answer by transforming it into its z-score. A new normalized answer is generated by adding the noise which was generated by a Gaussian distribution $\mathcal{N}(0, 1)$. We finally change it to the original scale and obtain the new answer. We randomly choose $NM\gamma$ answers with replacement to add noise and keep the rest the same.



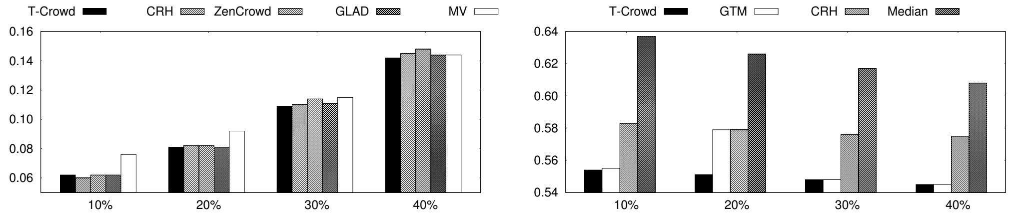
(a) Categorical Columns (b) Continuous Columns

Figure 8: Effect of Ratio of Columns



(a) Categorical Columns (b) Continuous Columns

Figure 9: Effect of Average Difficulty



(a) Error Rate (b) MNAD

Figure 10: Noisy Datasets

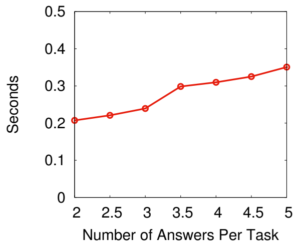
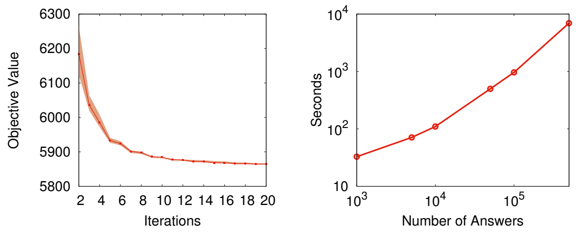


Figure 11: Efficiency of Assignment



(a) Convergence Rate (b) Truth Inference

Figure 12: Efficiency of Inference

For different levels of noise γ, we generate new datasets one hundred times. For each method, we run experiments 3 times to smoothen out possible instabilities. Hence we run in total 300 simulations for each method and average them to obtain the error rate and MNAD for different levels of γ.

Figure 10(a) and 10(b) show the results. The error rate increases while MNAD declines when γ increases. The reason for the decrease of MNAD is that the normalization denominator is the standard deviation of answers in each column. The growth rate of standard deviation is higher than that of RMSE which makes MNAD to decline.

T-Crowd performs well and stably when the level of noise γ increases both in terms of error rate and MNAD. T-Crowd has a very similar error rate and MNAD to CRH and GTM, respectively.

6.6 Efficiency

We first investigate the truth inference cost on Celebrity dataset and then show its running time on a single machine. Figure 12(a) shows the change of the objective value in truth inference at each iteration. Even iteration is the objective value from M step while odd iteration is the objective value from E step. Note that our inference model converges to the estimated value, after only a few iterations.The curve of objective function is different when initial parameters $(\alpha_i, \beta_j, \phi_u)$ are different. We random the parameters several times, average the value and plot the line. We also draw the error bar in the figure to show the minimum and maximum value in each iteration.

Then, we confirm the low cost of truth inference by measuring the throughput of T-Crowd, i.e., how many answers it can process per second. For this purpose, we use synthetic data used in Section 6.5 since the number of answers collected for real data is limited. Figure 12(b) shows that the runtime of T-Crowd is approximately linear to the number of answers; T-Crowd can process approximately 100 answers per second on a single machine. This performance is acceptable, given that the rate of incoming answers is much lower in a real crowdsourcing system. The performance is also consistent with our time complexity analysis at the end of Section 4.2.

Finally, we measure the time required to assign a new task to an incoming worker on the Celebrity dataset (Figure 11). We assume that we already obtained the estimated truth using T-Crowd's truth inference module. We show the running time of computing the *structure-aware* information gain for all candidate tasks each time a new worker arrives. Because it is easy to parallelize task assignments, we run eight processes on our machine. As shown in Figure 11, the assignment cost increases linearly with the average number of answers collected so far for each task. This is consistent to our complexity analysis at the end of Section 5.1, which suggests that the cost is linear to the total number $|\mathcal{A}|$ of answers so far. Still, as the figure shows, new assignments can be conducted in real-time, which is important for a real crowdsourcing platform.

6.7 Comparison to CrowdFill

CrowdFill [29] is a recent crowdsourcing system for tabular data. In CrowdFill, each worker is shown a fragment of a partially-filled table and asked to fill in empty cells, or upvote/downvote the answers entered by other workers. Compared to T-Crowd and the other methods that we have examined, CrowdFill requires the crowdsourcing platform to include additional functions (upvote and downvote operations), which are not currently supported by AMT. Hence, we compare to CrowdFill independently.

Still, to compare the effectiveness of T-Crowd with that of CrowdFill, we conducted an experiment, following the experimental setup of [29]. We collected information about 20 Olympic champions from 5 human workers. Given the picture of an athlete, the objective is to collect information about his/her attributes {*name*, *isRetired*, *#attendedOlympiads*, *#goldMedals*, *currentAge*, *ageInPic*}. Attributes *name* and *isRetired* are categorical and the remaining ones are continuous.

To be fair to both CrowdFill and T-Crowd, each worker was requested to answer questions twice. In the first experiment, workers give their answers independently following the T-crowd setting. The answers are collected and aggregated by T-Crowd to get the final table. In the second experiment, we use Google Docs to simulate CrowdFill's collaborative process. That is, workers can view other workers' answers, and they can choose between filling an empty cell or upvoting/downvoting a completed row. When the number of votes is larger than 2, a row is accepted if its upvotes are more than its downvotes; otherwise, it is rejected and it is offered again to workers to fill in their answers. When all the rows are accepted, we obtain the final table for CrowdFill.

Table 7 shows the error of these two methods. As in the previous experiments, we show the Error Rate for categorical data and RMSE for continuous data. Observe that T-Crowd is more accurate than CrowdFill for continuous attributes and the two methods have similar accuracy for categorical attributes. As opposed to T-Crowd, CrowdFill does not compute and use the unified worker quality for continuous and categorical attributes, which negatively affects its performance on continuous attributes, for which the collected answers are sparser.

Table 7: Error of CrowdFill

	Name	IsRetired	#attended Olympiads	#gold Medals	currentAge	ageInPic
CrowdFill	0.25	0.2	1.24	1.04	4.40	3.22
T-Crowd	0.25	0.15	0.71	0.67	3.17	2.38

7 CONCLUSIONS

In this paper we design a crowdsourcing framework for collecting multi-type tabular data. Most existing methods, which are designed for simple tasks that are all of the same datatype are not effective enough in terms of both truth inference and task assignment. Based on the characteristics of tabular data, we propose a probabilistic truth inference model that unifies worker quality on both categorical and continuous datatypes. Besides, we improve the accuracy of truth inference by considering the variance in the difficulty of different tasks. In addition, we design an information gain function which we use for selecting the tasks to assign to workers, based on the current answers and the workers' quality. We extend this function to consider the correlation in the quality of certain worker's answers for the same entity. Our experiments on three real datasets and synthetic datasets confirm the superiority of our methods, both in truth inference and task assignment compared to the state-of-the-art.

In the future, we plan to conduct experiments with larger tables compared to the ones we have used in Section 6. In addition, we plan to extend our approach to apply on tables for which entities are not known. In this case, entities should also be collected from the crowd. A third direction is the acceleration of truth inference and task assignment by parallel and/or distributed computation. Finally, we will explore the possible improvement of our approach by exploiting the possible correlations between entities (not only attributes), e.g., a worker may be more familiar to celebrities starring in a certain category of films or shows.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for the insightful comments. Guoliang Li was supported by the 973 program of China (2015CB358700), NSF China (61632016, 61472198, 61521002, 61661166012), and TAL education. Reynold Cheng, Caihua Shan, Zhipeng Huang, and Yudian Zheng were supported by the Research Grants Council of Hong Kong (RGC Projects HKU 17229116 and 17205115) and the University of Hong Kong (Projects 104004572, 102009508, 104004129).

REFERENCES

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/.

[2] AMT External Question. http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_ExternalQuestionArticle.html.

[3] Entropy. https://en.wikipedia.org/wiki/Entropy_(information_theory).

[4] ERF Function. http://mathworld.wolfram.com/Erf.html.

[5] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W.-C. Tan. Asking the right questions in crowd data sourcing. In *ICDE*, pages 1261–1264. IEEE, 2012.

[6] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *ECCV*, pages 768–783, 2014.

[7] X. Chen, Q. Lin, and D. Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *ICML*, pages 64–72, 2013.

[8] S. Das, P. S. GC, A. Doan, J. F. Naughton, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, and Y. Park. Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In *SIGMOD*, pages 1431–1446. ACM, 2017.

[9] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[10] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[12] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1):550–561, 2009.

[13] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[14] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pages 1015–1030. ACM, 2015.

[15] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *SIGMOD*, pages 61–72. ACM, 2011.

[16] W. V. Friesen, P. Ekman, et al. Emfacs-7: Emotional facial action coding system. *Unpublished manuscript, University of California at San Francisco*, 2(36):1, 1983.

[17] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *SIGKDD workshop on human computation*, pages 64–67, 2010.

[18] A. R. Khan and H. Garcia-Molina. Crowddqs: Dynamic question selection in crowdsourcing systems. In *SIGMOD*, pages 1447–1462. ACM, 2017.

[19] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced data management: A survey. *TKDE*, 28(9):2296–2319, 2016.

[20] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *PVLDB*, 8(4):425–436, 2014.

[21] 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 *SIGMOD*, pages 1187–1198. ACM, 2014.

[22] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han. Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. *TKDE*, 28(8):1986–1999, 2016.

[23] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.

[24] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *SIGKDD*, pages 745–754. ACM, 2015.

[25] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Crowdsourced databases: Query processing with people. Cidr, 2011.

[26] P. Mavridis, D. Gross-Amblard, and Z. Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *WWW*, pages 843–853, 2016.

[27] J. V. Michalowicz, J. M. Nichols, and F. Bucholtz. *Handbook of differential entropy*. CRC Press, 2013.

[28] H. Park, H. Garcia-Molina, R. Pang, N. Polyzotis, A. Parameswaran, and J. Widom. Deco: A system for declarative crowdsourcing. *PVLDB*, 5(12):1990–1993, 2012.

[29] H. Park and J. Widom. Crowdfill: Collecting structured data from the crowd. In *SIGMOD*, pages 577–588. ACM, 2014.

[30] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval*, pages 27–35, 2014.

[31] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in collaborative crowdsourcing. In *ICDM*, pages 949–954. IEEE, 2015.

[32] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDBJ*, 24(4):467–491, 2015.

[33] C. Shan, N. Mamoulis, G. Li, R. Cheng, Z. Huang, and Y. Zheng. T-crowd: Effective crowdsourcing for tabular data. In *ICDE*. IEEE, 2018.

[34] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263. Association for Computational Linguistics, 2008.

[35] Y.-L. Tian, T. Kanade, and J. F. Cohn. Facial expression analysis. In *Handbook of face recognition*, pages 247–275. Springer, 2005.

[36] V. Verroios, H. Garcia-Molina, and Y. Papakonstantinou. Waldo: An adaptive human interface for crowd entity resolution. In *SIGMOD*, pages 1133–1148. ACM, 2017.

[37] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[38] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

[39] X. Yin, J. Han, and S. Y. Philip. Truth discovery with multiple conflicting information providers on the web. *TKDE*, 20(6):796–808, 2008.

[40] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB*, 2012.

[41] Y. Zheng, G. Li, and R. Cheng. Docs: a domain-aware crowdsourcing system using knowledge bases. *PVLDB*, 10(4):361–372, 2016.

[42] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: is the problem solved? *PVLDB*, 10(5):541–552, 2017.

[43] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng. Qasca: a quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*, pages 1031–1046. ACM, 2015.

[44] H. Zhuang, A. Parameswaran, D. Roth, and J. Han. Debiasing crowdsourced batches. In *SIGKDD*, pages 1593–1602. ACM, 2015.

Caihua Shan received her B.E. degree in Computer Science and Technology from Zhejiang University in 2015. She is currently a PhD student at the Department of Computer Science in The University of Hong Kong. Her research mainly focus on data management in crowdsourcing.

Nikos Mamoulis received his diploma in computer engineering and informatics in 1995 from the University of Patras, Greece, and his PhD in computer science in 2000 from HKUST. He is currently a faculty member at the University of Ioannina. Before, he was professor at the Department of Computer Science, University of Hong Kong. His research focuses on the management and mining of complex data types.

Guoliang Li is currently working as an associate professor in the Department of Computer Science, Tsinghua University, Beijing, China. He received his PhD degree in Computer Science from Tsinghua University, Beijing, China in 2009. His research interests mainly include data cleaning and integration and crowdsourcing.

Reynold Cheng is an Associate Professor of the Department of Computer Science in the University of Hong Kong (HKU). He obtained his PhD from Department of Computer Science of Purdue University in 2005. Dr. Cheng was granted an Outstanding Young Researcher Award 2011-12 by HKU.

Zhipeng Huang received his B.Sc. degree in Machine Intelligence in 2011 from Peking University, China. He is now a PhD student at the Department of Computer Science, University of Hong Kong. His research focuses on the mining of heterogeneous information networks.

Yudian Zheng is currently at Twitter, focusing on building large scale online distributed machine learning systems. His main research interests include data analysis, crowdsourced data management and machine learning. He has published more than 20 papers in leading international conferences/journals, including SIGMOD, KDD, VLDB, TKDE, ICDE, etc.