JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 1231-1246 (2021)
DOI: 10.6688/JISE.202109_37(5).0016

Multi-Factor Influencing Truth Inference in Crowdsourcing

GUANGYUAN ZHANG AND NING WANG+
School of Computer and Information Technology
Beijing Jiaotong University
Beijing, 100044 P. R. China
E-mail: {17120441, nwang}@bjtu.edu.cn

By harnessing human intelligence, crowdsourcing can solve problems that are difficult for computers. A fundamental problem in crowdsourcing is truth inference, which decides how to infer the truth effectively. We propose MFICrowd, a novel truth inference framework which takes multi-factor into account for profiling workers accurately and improving answer accuracy effectively. Based on the diversity degree of task domains and the semantic similarity of candidate answers, we quantify task difficulty for modeling tasks and workers objectively and exactly. By integrating task domains, task difficulty and answer similarity into truth inference, MFICrowd aggregates answers from a group of workers effectively. The comprehensive experimental results on both simulated and real datasets show that our truth inference framework based on multi-factor is effective, and it outperforms existing state-of-the-art approaches in both answer accuracy and time efficiency.

Keywords: crowdsourcing, multi-factor, truth inference, task difficulty, task domains

1. INTRODUCTION

Nowadays, crowdsourcing is used effectively to solve problems that are difficult for computers by human intelligence, such as machine translation [1, 2] and sentiment analysis [3]. On crowdsourcing platforms like Amazon Mechanical Turk [4] and CrowdFlower [5], workers come from all over the world, with different backgrounds, motives and levels of knowledge, so answers submitted by workers cannot be totally trusted. One of the key challenges in crowdsourcing is truth inference, which decides how to aggregate answers from multiple workers to yield high-quality results [6, 7, 8].

The most straightforward method in truth inference is Majority Voting (MV) [9], which selects the option that gets the most votes from workers as the truth. However, when dishonest workers dominate, MV may fall into the worst performance. In fact, workers on crowdsourcing platforms are with different qualities and should be treated distinctly. Intuitively, workers with high quality are more likely to be cautious and can submit correct answers, while malicious workers submit wrong answers deliberately. As a result, worker quality is an indispensable factor in truth inference, and most of researches have attempted

Received September 23, 2019; revised March 31, 2020; accepted May 5, 2020.
Communicated by Chang-Tien Lu.
+Corresponding author.

to evaluate workers or eliminate spammers [10, 11, 12, 13, 14, 15]. Furthermore, to establish truth inference effectively, recent researches have studied on other influencing factors. These methods mainly fall into three categories: (1) Similarity-based methods [16, 17], in which the posterior probability of answers can be calculated more reasonably by considering their similarity; (2) Domain-based methods [7, 18, 19], in which workers qualities are evaluated by considering task domains; (3) Difficulty-based methods [20, 21, 22], in which truth inference distinguish tasks with different levels of difficulty.

Nevertheless, existing approaches lack of a comprehensive consideration of multiple factors such as worker quality, candidate answer similarity, task difficulty and task domains. Especially, task difficulty is determined all by the performance of workers in existing works. In other words, the better the worker answers, the easier the task is. In fact, the difficulty level of a task relies on the task itself and could not be changed by the truth inference process. Besides, for ignoring semantics, answer similarity cannot be evaluated accurately in existing methods.

To address the aforementioned limitations, we propose a **M**ulti-**F**actor oriented Truth **I**nference framework in **Crowd**sourcing, called **MFICrowd**. At first, based on the domain entropy and semantic similarity of candidate answers for a task, we can quantify task difficulty objectively and accurately. Furthermore, we design a valid probabilistic model for truth inference by formulating the relationship between multiple influencing factors and truth inference. Meanwhile, for each worker, MFICrowd evaluates the worker quality dynamically by not only task domain but also task difficulty .Experimental results show the effectiveness of our truth inference framework by leveraging multiple influencing factors for aggregating answers.

Our major contributions are as follows:

1. We are the first to quantify task difficulty based on domain entropy and semantic similarity of candidate answers of a task, which can help model tasks and workers objectively and exactly.

2. We propose a novel truth inference framework MFICrowd, which can integrate multiple factors into truth inference, and generate truths effectively.

3. Experimental results indicate that our difficulty quantification method can reflect actual task difficulty and MFICrowd outperforms existing state-of-art methods in both the result quality and the time efficiency.

2. RELATED WORK

During the past decade years, crowdsourcing has caught much attention and has been widely used in Image labeling, Natural Language Processing, Data Management and other fields. Crowdsourcing platforms are built to provide centralized management of tasks, workers answers. For openness of crowdsourcing platforms and the variety of tasks, truth inference becomes very important in crowdsourcing. How to aggregate answers to yield high-quality results has been studied in a considerable amount of literatures [14, 17-23]. Although the models are different in these studies, they are mainly based on the investigation of quality control and truth inference.

Quality Control. In order to control quality in crowdsourcing, the principle problem is how to model workers and tasks accurately and reasonably. For worker modeling, MV [9] neglects worker quality and treats each worker equally. To overcome the limitations of MV, many researches have explored and improved the estimation of worker quality. PM [23] models different worker with different weight. Furthermore, worker quality is modeled as a fixed value in CDAS [14], while it is modeled as a confusion matrix in D&S [24]. For task modeling, more influence factors have been explored. Similarity-based methods [16, 17] take into account candidate answer similarity of tasks. Domain-based methods [7, 18, 19] model tasks in diverse domains by topic model or the knowledge base. Difficulty-based methods [20, 21, 22] distinguish tasks with different levels of difficulty.

Truth Inference. MV [9] takes the candidate answer submitted by majority workers as the truth, independent of worker quality. In fact, the higher quality of a worker, the more reliable his answers for truth inference. PM [23] incorporates worker weight into answer aggregation. CDAS [14] builds a Bayesian probability model for truth inference, and INQUIRE [25] adopts an incremental truth inference strategy to improve performance. In addition, DASM [17] takes label similarity into consideration, while GLAD [20] exploits the inherent relation between task difficulty and worker quality. Besides, as a worker, the probability that he answers correctly is affected by his expertise in different domains. DOCS [19] tries to make use of domain information during answer aggregation.

Nevertheless, the evaluation of task difficulty in above works relies totally on the performance of workers. In reality, whether a task is difficulty or not is decided by task itself. Also, although some works consider answer similarity in truth inference, they neglect semantic similarity. Furthermore, existing approaches lack of a comprehensive consideration of multiple factors such as worker quality, task difficulty and task domains for truth inference. To the best of our knowledge, we are the first to quantify task difficulty based on domain entropy and semantic similarity of candidate answers of a task. By modeling tasks and workers on the basis of multiple influencing factors respectively, the proposed MFICrowd can fully take advantage of available information and obtain more accurate estimation of the truth.

3. PROBLEM MODELING AND SOLUTION OVERVIEW

3.1 Problem Model

A crowdsourcing platform allows tasks to be performed by a huge number of workers on internet. The platform will receive answers from workers and infer truths for publishers of tasks.

Definition 1 (Task): Let $\mathscr{T}$=$\{t_1, t_2, ..., t_n\}$ be the task set with n requested tasks. Each Task t_i ($1 \leq i \leq n$) being published contains a text description with z candidate answers denoted as $A^{t_i} = \{a_{i1}, a_{i2}, ..., a_{iz}\}$, which can be answered with multiple workers.

Definition 2 (Task Domain): Let $\mathscr{O} = \{o_1, o_2, ..., o_k\}$ be the domain set with k domains. For each task $t_i \in \mathscr{T}$, there exists a Task Domain denoted as a domain vector v^{t_i} = [v_{i1}, v_{i2}, ..., v_{ik}] ($v_{ig} \in [0,1], 1 \leq g \leq k$ and $\sum_{g=1}^{k} v_{ig} = 1$), representing the distribution of relevance between task t_i and k domains in $\mathscr{O}$. The higher v_{ig} indicates that task t_i is more relevant to domain o_g.

Definition 3 (Candidate Answer Similarity): For each task $t_i \in \mathscr{T}$, there exists a set of Candidate Answer Similarity s^{t_i}=$\{s_{i1}, s_{i2}, ..., s_{ib}\}$, where each $s_{ig} \in [0,1](1 \leq g \leq b)$, and each component of s^{t_i} represents the similarity of a pair of candidate answers for t_i (*i.e.*, $|s^{t_i}| = C_z^2$). The higher value of s_{ig} means that it is more difficult to distinguish between a pair of answers.
Definition 4 (Task Difficulty): For each task $t_i \in \mathscr{T}$, Task Difficulty d_i ($d_i \in [0,1]$) reflects the difficulty degree for workers who answer this task. The higher of d_i is, the more difficult for workers to answer task t_i.
Definition 5 (Worker Quality): Let W=$\{w_1, w_2, ..., w_m\}$ denotes a worker set, the Worker Quality of w_j ($w_j \in W$) can be modeled by a quality vector q^{w_j}=$[q_{j1}, q_{j2}, ..., q_{jk}]$, where each $q_{jg} \in q^{w_j}(1 \leq g \leq k)$ reflects the ability for w_j to process tasks in domain o_g.
Definition 6 (Truth): For task $t_i(t_i \in \mathscr{T})$ with candidate answer set $A^{t_i} = \{a_{i1}, a_{i2}, ..., a_{iz}\}$, there exists an inferred Truth $r_i(r_i \in A^{t_i})$ and a ground truth $r_i^*(r_i^* \in A^{t_i})$.

For tasks on crowdsourcing platforms, the ground truth is generally unknown. The aim of truth inference is to infer the final answer as close to the ground truth as possible. As we know, workers are competent for tasks in domains they are familiar with [19]. Besides task domains, we take task difficulty into consideration for evaluation of the worker quality. Intuitively, a worker should be given a higher quality when he correctly answers a difficult task. Furthermore, semantic answer similarity is also considered in our method for distinguishing each candidate answer in more accurate manner. In conclusion, we fully take advantage of available information (task domain, task difficulty, candidate answer similarity) to obtain more accurate estimation of the truth.

3.2 Solution Overview

Fig. 1 illustrates MFICrowd, our truth inference framework to derive simultaneously the difficulty level and the truth for each task, and the quality for each worker. At first, MFICrowd leverages domain information and the semantic similarity of candidate answers to quantify task difficulty. Then, tasks and workers are modeled objectively and exactly based on task difficulty and task domain. Finally, the truth is inferred by calculating the posterior probability of each candidate answer based on *Bayesian decision algorithm*, during which multiple factors are integrated comprehensively and effectively. There are three core modules in MFICrowd as follows:

- **Difficulty Quantification:** This module aims to quantify task difficulty by incorporating the implicit information of domains and candidate answers of each task. Based on domain entropy of a task which captures the task domain dispersion and the semantic similarity between candidate answers, it quantifies the difficulty of the task objectively.

- **Quality Control:** This module models worker quality in diverse domains. The quality of a worker is changing during the entire mission, and it is updated by the module based on domains and difficulty of a task once the truth inference is completed.

- **Truth Inference:** To infer truths effectively, this module aggregates workers' answers in light of multiple factors such as the worker quality and task information about difficulty, domains and candidate answers.

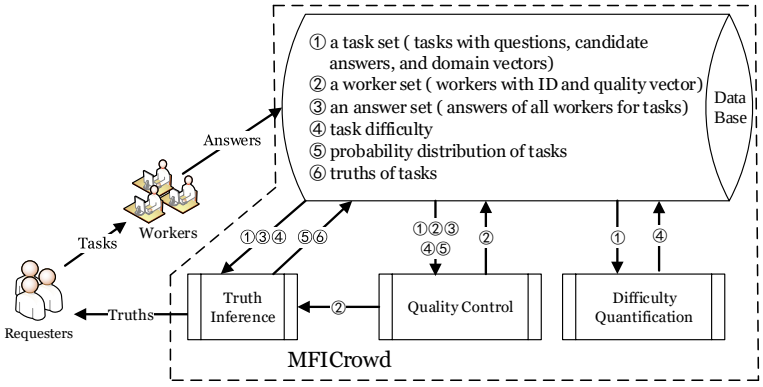

Fig. 1. MFICrowd framework.

4. TRUTH INFERENCE BASED ON MULTIPLE FACTORS

4.1 Quantification of Task Difficulty

For task difficulty, existing researches basically treat it as a parameter varied with iterations [20, 21, 22], so task difficulty is varied with the completion situation of the task. Some factors, such as completion situation and completion time of a task, are faced with high uncertainty and complexity when cheating or laggard workers dominate. In contrast, we study on how to make use of the information contained in task itself to quantify task difficulty objectively.

Intuitively, when a task involves multiple domains, the more diverse the domains are, the more difficult the task will be. Meanwhile, if the similarity of candidate answers is higher, the task is more difficult because the candidate answers are hard to distinguish by workers. We now propose a novel method to quantify task difficulty in three steps.

Step 1: Estimating Domain Entropy Given a task t_i and its domain vector [19], the domain entropy $e_i \in [0,+\infty]$, which reflects the degree of domain diversity of t_i can be derived as follows:

$$e_i = \sum_{g=1}^{k} -v_{ig} \cdot \log v_{ig}. \tag{1}$$

A greater e_i means that the domains of task t_i is more diverse, *i.e.*, the task is strongly related to more than one domain.

Step 2: Calculating Semantic Similarity of Answers We utilize *Word2Vec* [26] to measure the semantic similarity between a pair of candidate answers of a task. Let $word(a_{iy})$=[ω_{y1}, ω_{y2}, ..., ω_{yh}] be the word vector of a candidate answer a_{iy} ($a_{iy} \in A^{t_i}$), the *Euclidean distance* of each pair of candidate answers a_{ix} and a_{iy} of task t_i can be calculated as follows:

$$sim(a_{ix}, a_{iy}) = dist(word(a_{ix}), word(a_{iy})) = \sqrt{\sum_{g=1}^{h} (\omega_{xg} - \omega_{yg})^2}. \tag{2}$$

For *Word2Vec* is not very applicable for numerical data, we calculate the similarity

between two numerical candidate answers a_{ix} and a_{iy} as follows:

$$sim(a_{ix}, a_{iy}) = 1 - \frac{\left|a_{ix} - a_{iy}\right|}{\left|max(a_{ix}, a_{iy})\right|}. \tag{3}$$

Definition 7 (Overall Answer Similarity): Given a task t_i and its candidate answer similarity set s^{t_i}, the Overall Answer Similarity s_i^* of task t_i is the median of s^{t_i}, which can be calculated as follows:

$$s_i^* = \begin{cases} s_{ig} & \text{for } b \text{ is odd, } g = \frac{b+1}{2} \\ \frac{s_{ig}+s_{ig'}}{2} & \text{for } b \text{ is even, } g = \frac{b}{2} \textit{ and } g' = \frac{b}{2}+1. \end{cases} \tag{4}$$

A larger s_i^* indicates that the candidate answers of t_i are more indistinguishable.

Step 3: Quantifying Task Difficulty Based on the domain entropy and the overall answer similarity of a task, we can quantify task difficulty. Given domain entropy e_i and candidate the overall answer similarity s_i^* of task t_i, task difficulty d_i can be calculated as follows:

$$d_i = w_1 \cdot s_i^* + w_2 \cdot e_i, \tag{5}$$

where w_1 and w_2 represent the weight of similarity and domain entropy to task difficulty learned by *Entropy weight method* [27] respectively.

The algorithm of quantifying task difficulty is summarized in Algorithm 1. By analyzing the information of task t_i, we first compute its domain entropy e_i based on domain vector v^{t_i} (line 2-3). Then, we generate the similarity set s^{t_i} to obtain the overall answer similarity s_i^* for t_i (line 4-6). At last, we can finally obtain task difficulty d_i (line 7-8). The time complexity of the algorithm is $\mathscr{O}(k+\mathscr{C}_z^2)$.

Algorithm 1: Quantification of Task Difficulty

Input: t_i, v^{t_i}, A^{t_i}
Output: d_i $(1 \le i \le |\mathscr{T}|)$

1 Initialize $e_i = 0, s^{t_i} = \{0, 0, ..0\}$; // the size of s^{t_i} is $\mathscr{C}_z^2$;
2 **while** $|g| \le k$ **do**
3 $\quad e_i += -v_{ig} \cdot \log v_{ig}$;
4 **for** *each pair* $a_x^{t_i}, a_y^{t_i} \in A^{t_i}$ **do**
5 $\quad sim(a_{ix}, a_{iy}) = dist(word(a_{ix}), word(a_{iy})) = \sqrt{\sum_{g=1}^{h}(\omega_{xg} - \omega_{yg})^2}$;
$\quad\quad s^{t_i} \leftarrow sim(a_{ix}, a_{iy})$;
6 s_i^*= medians of sorted s^{t_i};
7 $w_1, w_2 \leftarrow$ *Entropy weight* $\{e_i, s_i^*\}$;
8 $d_i = w_1 \cdot s_i^* + w_2 \cdot e_i$;
9 **return** d_i;

4.2 Truth Inference

Based on answers collected from multiple workers, truth inference works on deriving the final answer for a task. Intuitively, (1) as to a task, if the answers are collected from workers with higher qualities in the domains closely related to the task, then the truth inference by aggregating answers is more reliable; (2) as to a worker, the probability that the worker answers correctly is affected by both the task difficulty and his expertise in different domains. MFICrowd implements truth inference in two steps.

Step 1: Aggregating Answers, Inferring Truths: We compute the probability that each candidate answer is correct, and select the answer with the highest posterior probability as the truth. Different from existing works, we take multiple factors into account for truth inference. In fact, the answer submitted by a worker relies mainly on the difficulty of the task, the expertise of the worker, and the truth. As the difficulty degree of a task increases, even the most competent worker only has a 50% chance of giving the right answer [20, 21].

Suppose each worker answers independently and the difficulty of task t_i is d_i, the probability $\eta_\pi^{w_j}$ that the worker w_j's answer u_{ij} is correct for t_i in the domain π can be calculated as follows:

$$\eta_\pi^{w_j} = \mathrm{P}(u_{ij} = a_{ic} \mid d_i, \theta_i = \pi, r_i^* = a_{ic}) = \frac{1}{2}(1 + (1-d_i)^{\frac{1}{q_{j\pi}}}). \tag{6}$$

Under the model, as the task difficulty d_i increases or worker quality $q_{j\pi}$ decreases, the probability of the worker's answer being correct tends to 0.5, indicating that the worker chooses the answer at random.

Without any prior knowledge, we assume the probability of each incorrect answer submitted by worker w_j equals to $\frac{1-\eta_\pi^{w_j}}{z-1}$. Let $\delta_{\{\cdot\}}$ denote Kronecker delta function[1], whose output value is 1 if the input values are equal, otherwise is 0. And U^{t_i}=$\{u_{ij} \mid w_j \in W\}$ denotes answer set of workers for t_i, where u_{ij} is the worker w_j's answer. Therefore, in domain π, the probability of w_j's answer u_{ij} for t_i is correct or not can be represented as follows:

$$\mathrm{P}(u_{ij} \mid d_i, \theta_i = \pi, r_i^* = a_{ic}) = \left(\eta_\pi^{w_j}\right)^{\delta_{\{u_{ij}=a_{ic}\}}} \cdot \left(\frac{1-\eta_\pi^{w_j}}{z-1}\right)^{\delta_{\{u_{ij}\neq a_{ic}\}}}. \tag{7}$$

Let $\rho^{t_i} = \{\rho_{i1}, \rho_{i2}, ..., \rho_{iz}\}$, $\rho_{ic} \in \rho^{t_i}$ is the probability that a_{ic} is the truth of t_i. In fact, a task may relate to more than one domain. After collecting the answer set U^{t_i} for task t_i from all workers, the posterior probability of each candidate answer a_{ic} being the truth can be calculated as:

$$\rho_{ic} = \mathrm{P}(r_i^* = a_{ic} \mid d_i, U^{t_i}) = \sum_{\pi=1}^{k} v_{i\pi} \cdot \mathrm{P}(r_i^* = a_{ic} \mid d_i, \theta_i = \pi, U^{t_i}). \tag{8}$$

Here, $\mathrm{P}(r_i^* = a_{ic} \mid d_i, \theta_i = \pi, U^{t_i})$ represents the probability that a_{ic} is the truth of t_i in the specific domain π, without any priori knowledge.

$$\mathrm{P}(r_i^* = a_{ic} \mid d_i, \theta_i = \pi, U^{t_i}) = \frac{\mathrm{P}(U^{t_i} \mid d_i, \theta_i = \pi, r_i^* = a_{ic})}{\sum_{a_{ib} \in A^{t_i}} \mathrm{P}(U^{t_i} \mid d_i, \theta_i = \pi, r_i^* = a_{ib})} = \frac{\prod_{u_{ij} \in U^{t_i}} \mathrm{P}(u_{ij} \mid d_i, \theta_i = \pi, r_i^* = a_{ic})}{\sum_{a_{ib} \in A^{t_i}} \prod_{u_{ij} \in U^{t_i}} \mathrm{P}(u_{ij} \mid d_i, \theta_i = \pi, r_i^* = a_{ib})}. \tag{9}$$

[1] https://en.wikipedia.org/wiki/Kronecker_delta

Step 2: Updating Worker Quality To estimate a worker's quality accurately, we take into account domains that relate to the tasks answered by the worker, the difficulty of tasks and the posterior probability of each candidate answer being the truth. Let $\mathscr{T}^{w_j}$ denote all the tasks answered by worker w_j, and the quality of w_j in the domain π is derived as follows:

Besides worker expertise in domains, task difficulty is also an important factor for quality evaluation. Intuitively, a worker should be given a higher quality when he correctly answers a difficult task. Different from previous quality estimation methods, which only consider the amount of answers that workers answer correctly, MFICrowd takes the task difficult and task domain into consideration and get worker quality more accurately.

$$q_{j\pi} = \frac{\sum_{t_i \in \mathscr{T}^{w_j}} \mathrm{P}(\theta_i = \pi) \cdot \mathrm{P}(r_i^* = u_{ij}) \cdot d_i}{\sum_{t_i \in \mathscr{T}^{w_j}} \mathrm{P}(\theta_i = \pi) \cdot d_i}. \tag{10}$$

Algorithm 2: Truth Inference Algorithm

Input: $\mathscr{T}$, v^{t_i} ($t_i \in \mathscr{T}$), U^{t_i} ($t_i \in \mathscr{T}$)
Output: R, Q

1 Initialize q^{w_j} for $w_j \in W$ with qualification test;
2 **for** $t_i \in \mathscr{T}$ **do**
3 **for** $a_{ic} \in \rho^{t_i}$ **do**
4 Initialize $\rho_{ic} = 0$;
5 **while** $|\pi| \leq k$ **do**
6 $\rho_{ic} += v_{i\pi} \cdot \mathrm{P}(r_i^* = a_{ic} \mid d_i, \theta_i = \pi, U^{t_i})$;
7 r_i^*=*argmax* $\{\rho^{t_i}\}$;
8 **for** $u_{ij} \in U^{t_i}$ *and* $w_j \in W$ **do**
9 **while** $|\pi| \leq k$ **do**
10 $q_{j\pi} = \frac{\sum_{t_i \in \mathscr{T}^{w_j}} \mathrm{P}(\theta_i = \pi) \cdot \mathrm{P}\left(r_i^* = u_{ij}\right) \cdot d_i}{\sum_{t_i \in \mathscr{T}^{w_j}} \mathrm{P}(\theta_i = \pi) \cdot d_i}$;
11 $Q \leftarrow \{q^{w_1}, q^{w_2}, ..., q^{w_m}\}$, $R \leftarrow \{r_1, r_2, ..., r_n\}$;
12 **return** *R, Q*

Algorithm 2 gives the solution of truth inference in MFICrowd, which illustrates the process of step 1: *Aggregating Answers, Inferring Truths* (line 3-7) and step 2: *Updating Worker Quality* (line 8-10) respectively. For initializing the quality of workers, qualification test (line 1) is adopted by assigning some golden tasks with ground truths to new workers. For a worker, his performance is changing during the process of doing tasks, and even the most reliable worker may submit wrong answers. We update worker quality once the truth of a task is inferred for capturing worker quality accurately. The time complexity of Algorithm 2 is $\mathscr{O}(|\mathscr{T}| \cdot |W|)$.

5. EXPERIMENT EVALUATION

In order to evaluate the effectiveness and efficiency of our MFICrowd framework for truth inference, we compare our method with other 5 crowdsourcing answer aggregation methods on both stimulated and real datasets. All the experiments are implemented in Python on a 4GB memory PC with an Intel Core i5 processor.

5.1 Experiment Setup

5.1.1 Dataset

(1) *Millionaire-Game Dataset*[2]: a real dataset, which contains 1906 multiple-choice tasks with 18 different domains and 214658 answers from 37332 players. From the dataset, we can get the correct answer, the related domain and the difficulty level (1-12) of each task. Each task is only related to one domain on this dataset.

(2) *Encyclopedia Dataset*[3]: a simulated dataset crawled from the Chinese encyclopedic knowledge base, which contains 288 multiple-choice tasks with 4 different domains (*Sports, Arts, Science and Others*). A task may relate to more than one domain on this dataset. Specifically, we simulate 40 workers whose qualities in different domains are randomly sampled from a uniform distribution. Each task is allocated to 8 workers, and we totally collect 2592 answers.

5.1.2 Comparison algorithms

We compare our **MFICrowd** with other 5 truth inference methods. **MV** [9] is the most straightforward answer aggregation method which treats each worker equally, and the number of votes of each candidate answer is the only criterion for truth inference. **PM** [23] is an improved MV method, which takes into account the weights of workers. A worker's weight is calculated on the ratio of the number of correctly answered tasks by the worker to the total number of answered tasks. **DOCS** [19] is the latest work to propose a metric for quantifying crowdsourcing task domain, and models tasks and workers on the basis of diverse domains. **GLAD** [20] takes task difficulty into truth inference and worker ability estimation as well. However, task difficulty in GLAD is determined all by the performance of workers. **DASM** [17] is an improved GLAD method by considering the similarity of candidate answers. However, DASM does not analyze semantic similarity. Our method MFICrowd is the first work to quantify task difficulty objectively, and integrate task domains, task difficulty and semantic answer similarity into truth inference. Table 1 gives the brief comparison of 6 truth inference methods.

For initialization, we assign 20 tasks with ground truths to each worker, and take the ratio of the number of correctly answered tasks to the total number of answered tasks as his initial quality.

5.2 Task Difficulty Calibration

To verify whether the task difficulty quantified by our MFICrowd agrees with the reality, we conduct experiments on Millionaire-Game dataset with given task difficulty

[2] https://github.com/bahadiri/Millionaire

[3] https://wenku.baidu.com/

Table 1. Comparison of truth inference methods.

Method	Influence Factors Considered				Task Difficulty
	Task Difficulty	Task Domain	Similarity	Worker Quality	
MFICrowd	✓	✓	✓	✓	Quantified objectively by task itself
GLAD	✓	×	×	✓	Changed during inference process
DASM	✓	×	✓	✓	Changed during inference process
DOCS	×	✓	×	✓	No modeled
PM	×	×	×	✓	No modeled
MV	×	×	×	×	No modeled

level (1-12). Learned by *Entropy Weight* method, the weights of similarity and domain entropy calculated by Eq. (5) are 0.4 and 0.6 respectively on Millionaire-Game dataset. We select 103 tasks containing each difficulty level from the dataset. After computing difficulty for each task by task difficulty quantification method proposed in MFICrowd, we project the difficulty values to the range [1,12]. In Fig. 2, each point represents a task. In the ideal case, all points should lie on the line $Y = X$, which means that the task difficulty is estimated the same as the true difficulty. Obviously, all points drawn in Fig. 2 lie close to the line $Y = X$. Namely, task difficulty quantified by MFICrowd is close to its actual state. Experimental results show that our MFICrowd can estimate task difficulty objectively.

5.3 Impact of Factors on Truth Inference

To demonstrate the necessity of multiple factors in truth inference, we implement 3 other strategies based on MFICrowd: MFICrowd-diff leaves out task difficulty, MFICrowd-sim discards the similarity of candidate answers, and MFICrowd-dom removes domain information Entropy. Fig. 3 gives accuracy for 4 truth inference strategies on Encyclopedia dataset. MFICrowd significantly outperforms the other strategies since it tasks into account all the three influencing factors. Furthermore, observing the performance of other 3 strategies, we can see that MFICrowd-sim is the best and MFICrowd-diff is the worst. Obviously, the task difficulty is the most important factor to truth inference.

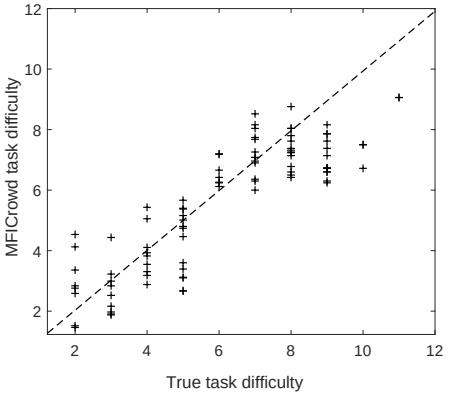

Fig. 2. Task difficulty calibration.

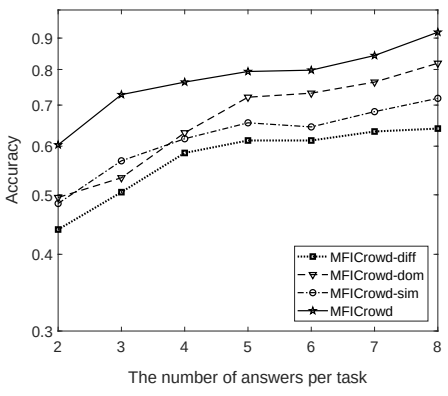

Fig. 3. The impact of factors on truth inference.

5.4 Evaluation of Performance

5.4.1 Accuracy

As an evaluation metric in our experiments, accuracy is defined as the ratio of the number of inferred truths being the ground truth to the number of tasks. Given a task set $\mathscr{T}$ with n tasks, the accuracy of truth inference method is defined as follows:

$$Accuracy = \frac{\sum_{i=1}^{n} \delta\{r_i^* = r_i\}}{n}. \tag{11}$$

At first, we compare the accuracy of truth inference for 6 answer aggregation methods on both Encyclopedia and Millionaire-Game datasets in Fig. 4. The accuracy of our MFICrowd is consistently higher than all other methods, and reaches above 0.92 on both datasets. Experimental results show that our MFICrowd can get more high-quality results, because it quantifies task difficulty objectively and integrates the task difficulty, task domain information, and semantic similarity of candidate answers into truth inference as well as quality control.

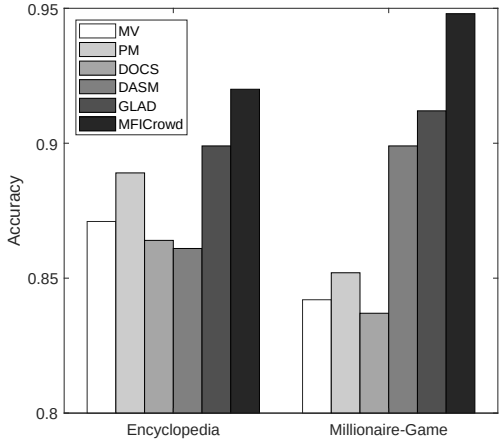

Fig. 4. Accuracy comparison on truth inference.

Furthermore, we conduct 4 experiments for each answer aggregation method to observe the impact of different aspects on answer accuracy. The experiments of *Varying Collected Answers* and *Varying Answered Tasks* are conducted on Encyclopedia dataset for its flexibility and adjustability. Meanwhile, the experiments of *Varying Task Difficulty* and *Varying Worker Expertise* are conducted on Millionaire-Game dataset with large number of workers and given difficulty levels for tasks. The experimental results show that our method performs well on different aspects.

Varying Collected Answers: Here, we vary the number of collected answers per task from 2 to 8. Fig. 5(a) shows the change of accuracy for different methods. The accuracy of each method gets higher when more answers are collected. MFICrowd does the

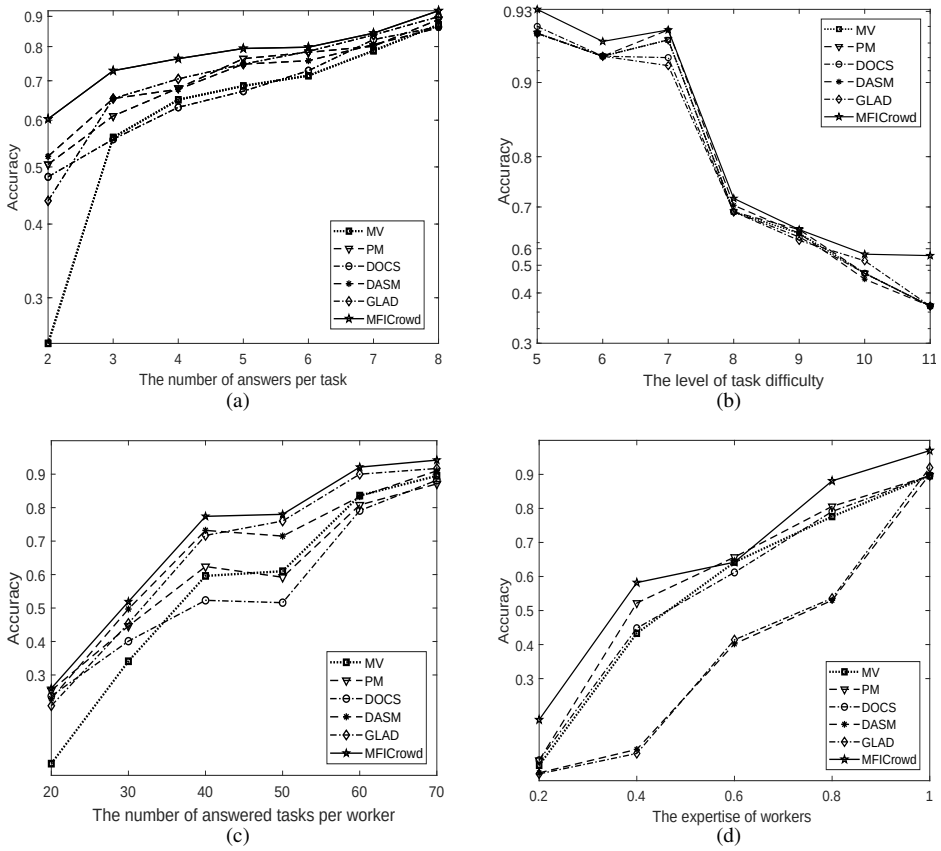

Fig. 5. The impact of different aspects on accuracy.

best with the accuracy always above 0.6, even the number of collected answers is small. Nevertheless, it is obvious that a small number of collected answers lead to the poor performance for other methods.

Varying Task Difficulty: We compare the accuracy of 6 methods on different levels of difficulty, and give results in Fig. 5(b). MFICrowd outperforms all other methods on both easy tasks (Level 1 to Level 6) and difficult tasks (Level 7 to Level 11). We focus the accuracy change on Level 5 to Level 11 in Fig. 5(b). Obviously, the accuracy of MFICrowd keeps more stable than that of other methods when difficulty level of tasks increases, and it gets 20 percentage points higher than other methods on the hardest (Level 11) tasks. The result proves that our task difficult quantification method is reasonable and it is helpful for truth inference and worker quality estimation.

Varying Answered Tasks: We vary the number of answered tasks per worker from 20 to 70, and compare the accuracy of 6 methods on different number of answered tasks. The

result in Fig. 5(c) illustrates that the accuracy of each method gets higher when more tasks are answered by a worker. In fact, the more tasks a worker answers, the more accurate his quality estimated by the truth inference method is. From Fig. 5(c), we can judge that MFICrowd can estimate worker quality more accurately, as it gets a higher accuracy compared with other methods.

Varying Worker Expertise: To observe the influence of worker expertise (quality) on answer accuracy, we select *Music* domain on Millionaire-Game dataset and vary worker expertise from 0 to 1. The accuracy comparison of 6 methods on different worker expertise is shown in Fig. 5(d). Intuitively, the more professional a worker is in a domain, the more tasks he can answer correctly in this domain. The accuracy of each method gets higher when worker quality is higher, and MFICrowd outperforms all the other methods.

5.4.2 Runtime

Table 2 gives the runtime of 6 answer aggregation methods on both Encyclopedia and Millionaire-Game datasets. Obviously, MV is the fastest for its simple truth inference strategy, and so is PM which only takes worker's answer accuracy into account. Other 3 iterative methods GLAD, DASM and DOCS take much time in truth inference. Both GLAD and DASM adopt EM which requires a lot of iterative process, and DOCS keeps iterating until convergence. However, our MFICrowd stops iterating when all truths of tasks are inferred with limited number of iterations while keeping high inference accuracy. In summary, the efficiency of our MFICrowd significantly outperforms other 3 iterative methods although MFICrowd takes more factors into account in truth inference.

Table 2. Runtime comparison on truth inference.

Method	Encyclopedia	Millionaire-Game
MFICrowd	4.31s	10.42s
GLAD	302.63s	415.88s
DASM	793.67s	1311.94s
DOCS	202.65s	374.58s
PM	0.05s	0.46s
MV	0.02s	0.18s

Furthermore, we design an experiment on Millionaire-Game dataset to observe the scalability of MFICrowd as the number of collected answers for each task increases. We set the number of tasks as 60, 80 and 100 respectively, and vary the number of collected answers per task from 50 to 300. The runtime for truth inference by MFICrowd is given in Fig. 6. Obviously, for given number of collected answers per task, the algorithm needs much time to process more tasks. However, for given tasks, the time for truth inference increases nearly linearly with the number of collected answers.

Generally, considering comprehensively both effectiveness and efficiency, our truth inference framework based on multiple factors is the best, compared with existing state-of-the-art approaches.

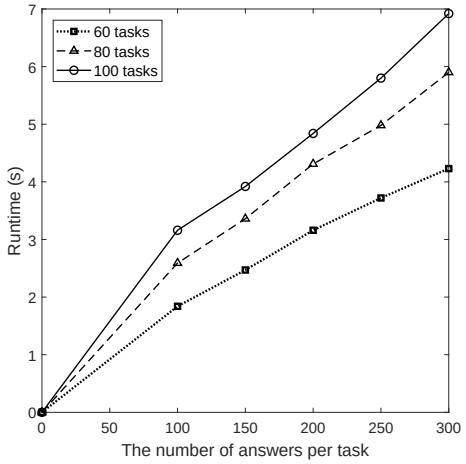

Fig. 6. Scalability of MFICrowd.

6. CONCLUSION

The comprehensive consideration of multiple influencing factors is crucial for truth inference. In this paper, we are the first to leverage information of task domains and semantic similarity of candidate answers to quantify task difficulty. By tasking multiple influencing factors into account, we propose a comprehensive truth inference framework MFICrowd and experimental results show that MFICrowd is more effective and efficient than existing methods.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (2018YFC 0809800), the National Natural Science Foundation of China (61370060) and The Fundamental Research Funds for the Central Universities (2017YJS065).

REFERENCES

1. O. F. Zaidan and C. Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, 2011, pp. 1220-1229.
2. A. Persaud and S. O'Brien, "Quality and acceptance of crowdsourced translation of web content," *Social Entrepreneurship: Concepts, Methodologies, Tools, and Applications*, 2019, pp. 1177-1194.
3. B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, Vol. 5, 2012, pp. 1-167.
4. "Amazon mechanical turk," https://www.mturk.com/.

5. "Crowdflower," http://crowdflower.com/.
6. N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *Proceedings of International Conference on Web Information Systems Engineering*, 2013, pp. 1-15.
7. J. Fan, G. Li, B. C. Ooi, K.-L. Tan, and J. Feng, "iCrowd: An adaptive crowdsourcing framework," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1015-1030.
8. Y. Chung, S. Krishnan, and T. Kraska, "A data quality metric (dqm): how to estimate the number of undetected errors in data sets," *Proceedings of the VLDB Endowment*, Vol. 10, 2017, pp. 1094-1105.
9. M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "Crowddb: answering queries with crowdsourcing," in *Proceedings of ACM SIGMOD International Conference on Management of data*, 2011, pp. 61-72.
10. F. K. Khattak and A. Salleb-Aouissi, "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, 2011, p. 5.
11. D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems*, 2011, pp. 1953-1961.
12. V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, Vol. 13, 2012, pp. 491-518.
13. C. Eickhoff and A. P. de Vries, "Increasing cheat robustness of crowdsourcing tasks," *Information Retrieval*, Vol. 16, 2013, pp. 121-137.
14. X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "Cdas: a crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, Vol. 5, 2012, pp. 1040-1051.
15. G. Li, C. Chai, J. Fan, X. Weng, J. Li, Y. Zheng, Y. Li, X. Yu, X. Zhang, and H. Yuan, "Cdb: optimizing queries with crowd-based selections and joins," in *Proceedings of ACM International Conference on Management of Data*, 2017, pp. 1463-1478.
16. J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," *Proceedings of the VLDB Endowment*, Vol. 5, 2012, pp. 1483-1494.
17. Y.-L. Fang, H.-L. Sun, P.-P. Chen, and T. Deng, "Improving the quality of crowdsourced image labeling via label similarity," *Journal of Computer Science and Technology*, Vol. 32, 2017, pp. 877-889.
18. 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 *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 745-754.
19. Y. Zheng, G. Li, and R. Cheng, "Docs: a domain-aware crowdsourcing system using knowledge bases," *Proceedings of the VLDB Endowment*, Vol. 10, 2016, pp. 361-372.
20. 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 *Advances in Neural Information Processing Systems*, 2009, pp. 2035-2043.
21. P. Dai, D. S. Weld *et al.*, "Decision-theoretic control of crowd-sourced workflows," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
22. A. Kurve, D. J. Miller, and G. Kesidis, "Multicategory crowdsourcing accounting for

variable task difficulty, worker skill, and worker intention," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, 2014, pp. 794-809.
23. B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas, "Crowdsourcing for multiple-choice question answering," in *Proceedings of the 26th IAAI Conference*, 2014, pp. 2946-2953.
24. A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Vol. 28, 1979, pp. 20-28.
25. J. Feng, G. Li, H. Wang, and J. Feng, *Incremental Quality Inference in Crowdsourcing*, 2014, pp. 453-467.
26. "Word2vec," https://en.wikipedia.org/wiki/Word2vec.
27. M. Kelbert, I. Stuhl, and Y. Suhov, "Weighted entropy: basic inequalities," *arXiv Preprint*, 2017, arXiv:1710.10798.

Guangyuan Zhang received her Bachelor degree in Computer Science and Technology from Yunnan University, China. Currently, she is an MS student in School of Computer and Information Technology, Beijing Jiaotong University, China. Her research interests include data quality and crowdsourcing.

Ning Wang received her Ph.D. degree in Computer Science in 1998 from Southeast University in Nanjing, China. She is currently serving as a Professor in School of Computer and Information Technology, Beijing Jiaotong University, China. Her research interests include web data integration, big data management, data quality and crowdsourcing.