Incremental Quality Inference in Crowdsourcing

Jianhong Feng, Guoliang Li, Henan Wang, and Jianhua Feng

Department of Computer Science, Tsinghua University, Beijing 100084, China
{fengjh11,whn13}@mails.thu.edu.cn, {liguoliang,fengjh}@tsinghua.edu.cn

Abstract. Crowdsourcing has attracted significant attention from the database community in recent years and several crowdsourced databases have been proposed to incorporate human power into traditional database systems. One big issue in crowdsourcing is to achieve high quality because workers may return incorrect answers. A typical solution to address this problem is to assign each question to multiple workers and combine workers' answers to generate the final result. One big challenge arising in this strategy is to infer worker's quality. Existing methods usually assume each worker has a fixed quality and compute the quality using qualification tests or historical performance. However these methods cannot accurately estimate a worker's quality. To address this problem, we propose a worker model and devise an incremental inference strategy to accurately compute the workers' quality. We also propose a question model and develop two efficient strategies to combine the worker's model to compute the question's result. We implement our method and compare with existing inference approaches on real crowdsourcing platforms using real-world datasets, and the experiments indicate that our method achieves high accuracy and outperforms existing approaches.

1 Introduction

Crowdsourcing has attracted widespread attention from many communities such as database and machine learning. The primary idea of Crowdsourcing is to take advantage of human intelligence to solve problems which are still difficult for computers, such as language translation, image recognition [6,17]. Several Crowdsourcing-based database systems have been proposed recently, e.g., CrowdDB [5,4], Qurk [10,11] and DECO [13]. These systems embedded in traditional relational database implement complicated crowdsourcing-based operations. Crowdsourcing platforms, such as Amazon Mechanical Turk (AMT) [2] and CrowdFlower [1] , provide APIs to facilitate these systems to accomplish crowdsourcing tasks. Task publishers (called requesters) can easily publish a large number of tasks on these crowdsourcing platforms, and obtain the answers completed by many human labors (called workers). Workers receive the pre-set financial rewards if their answers are accepted by the requester.

Workers on Crowdsourcing platforms typically have different backgrounds (e.g., age and education), coming from different countries or regions [12], and thus the answers may be affected by the various subjective experiences. Besides, spam workers provide answers randomly to get financial rewards. Therefore the

S.S. Bhowmick et al. (Eds.): DASFAA 2014, Part II, LNCS 8422, pp. 453–467, 2014.

answers collected from crowdsourcing platforms are usually not accurate. In order to achieve high quality of final results, a typical solution is to assign each task to multiple workers and infer the final results from the received answers[7,8].

To infer the final results, Majority Vote (MV) is the most popular inference method which has already been employed by CrowdDB [5,4] and DECO [13]. In MV, workers are assumed to have the same quality, and the answer provided by majority workers is taken as the result. Obviously MV ignores the fact that workers with different background and experience may have different quality, thus it leads to a low-quality inference result. In order to address the problem, the inference methods in [9,7,14,16] consider different qualities for each worker. The strategies used to reflect the quality of each worker can be categorized into two types. The first one is a fixed strategy adopted by CDAS [9], with the quality of each worker estimated by the worker's historical performance or qualification tests. This strategy is simple but not precise enough. For example, a worker's quality may increase as she learns more about questions through the answering procedure, or her quality may decrease when she is a little bit tired of answering questions. Therefore, modeling the quality for each worker is necessary for inferring the final results. The second inference method [7,14,16] is an iterative strategy. This strategy is based on the Expectation-Maximization [3] algorithm which can improve the results' accuracy by modeling each worker's quality dynamically. However, this inference method is rather expensive, because whenever it receives a new answer submitted by workers, it uses all received answers to re-estimate every worker's quality.

In summary, CDAS [9] can rapidly return the inference results, at expense of low quality. EM obtains results with higher quality while involving large inference time. To overcome these limitations, we propose an incremental quality inference framework, called Inquire, which aims to make a better tradeoff between the inference time and result quality. We devise a novel worker model and a question model to quantify the worker's quality and infer the question's result, respectively. When a worker submits her answer, Inquire can incrementally update the worker model and the question model, and return the inference results instantly. We compare Inquire with existing inference methods on real crowdsourcing platforms using real-world datasets, and the experiments indicate that our method achieves high accuracy and outperforms existing approaches.

This paper makes the following contributions:

- We formulate the incremental quality inference problem, and propose the Inquire framework to solve this problem.
- We devise a novel worker model to quantify the worker's quality, and a question model to infer the question's result instantly.
- We propose two incremental strategies to effectively update the question model and an incremental strategy to update the worker model.
- We compare Inquire with MV, CDAS and EM on real crowdsourcing platforms using real-world datasets. Our experimental results illustrate that Inquire can achieve a better tradeoff between the inference's time and accuracy.

This paper is organized as follows. We formulate the problem in Section 2 and introduce the INQUIRE's framework in Section 3. Question model and worker model are discussed in Section 4 and we discuss how to update the two models incrementally in Section 5. In Section 6, we show our experiment results and provide result analysis. Section 7 concludes the paper.

2 Problem Formulation

Since workers do not want to answer complicated questions, the tasks on crowdsourcing platforms are usually very simple and most of them are binary choices questions. For example, in entity resolution, each question contains two entities and asks workers to decide whether the two entities refer to the same entity [15]. In this paper, we also focus on these binary questions with only two possible choices. For ease of presentation, we assume there is only one correct choice for each question. It is worth noting that our method can be easily extended to support the questions with multiple choices.

Formally, a requester has a set of n binary questions $Q = \{Q_1, Q_2, \ldots, Q_n\}$ where each question asks workers to select the answer from two given choices. To achieve high quality, each question will be assigned to m workers. The true result for each question Q_i is denoted as R_i. R_i is 1 (or 0) indicating the returned result is the first choice (the second choice) for question Q_i. For example, if each question has two pictures, and workers are required to decide whether the people in two pictures are the same person. The two possible choices for this question is "same" (first choice) or "different" (second choice). If they are the same person, then R_i=1, otherwise R_i=0. After the requester published questions on the crowdsourcing platform, workers' answers are returned in a streaming manner. We use $\langle Q_i, W_k, L_{ik}\rangle$ to denote the result received from worker W_k for question Q_i with answer L_{ik} where $L_{ik} \in \{0,1\}$. Every time an answer $\langle Q_i, W_k, L_{ik}\rangle$ receives, we infer the result of question Q_i based on the current answer L_{ik}, the accuracy of worker W_k, and previous results of Q_i.

3 INQUIRE Framework

The goal of INQUIRE is to accurately and efficiently infer the final results of each question. To achieve this goal, we design two models, question model and worker model. The framework of INQUIRE is illustrated in Figure 1.

INQUIRE publishes all the questions to a crowdsourcing platform. Interested workers answer the questions. Each time a worker W_k completes a question Q_i, INQUIRE gets the corresponding answer $\langle Q_i, W_k, L_{ik}\rangle$.

We build a question model for each question, denoted by QM_i, which is designed to decide the inference result. INQUIRE updates QM_i based on both the worker's accuracy and the newly received answers. Section 5.1 gives how to incrementally update question model.

We construct a worker model for each worker, denoted by WM_k, to capture the quality of each worker. The accuracy of worker W_k can be directly derived

from WM_k and INQUIRE updates WM_k based on the worker model QM_i and the answer of the worker $\langle Q_i, W_k, L_{ik}\rangle$. We present the strategy of incrementally updating the worker model in section 5.2.

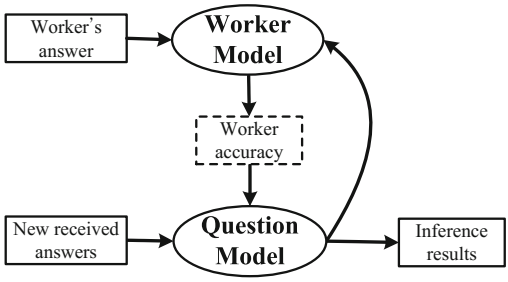


Fig. 1. INQUIRE Framework

The process of INQUIRE includes two steps:

In the first step, when W_k submits an answer of Q_i, QM_i is updated according to the worker model WM_k and the new answer L_{ik}. Then INQUIRE returns the inference result.

In the second step, if Q_i has already been answered m times, INQUIRE respectively updates the workers' model for those workers who have answered Q_i.

Following the process above, Algorithm 1 illustrates the pseudo code of our algorithm. The triple $\langle Q_i,W_k,L_{ik}\rangle^j$ is the j-th received answer of Q_i answered by W_k with answer L_{ik}. The inference result given by INQUIRE in the j-th round is denoted as $\langle Q_i\_Result\rangle^j$.

Example 1. Assume that a requester publishes a set of questions Q and asks three workers to answer each question. Take Q_1 as an example. Q_1 has received two answers $\langle Q_1, W_1,L_{11}\rangle^1$, $\langle Q_1, W_9, L_{19}\rangle^2$ and has a result $\langle Q_1\_Result\rangle^2$. For the arrival of $\langle Q_1, W_4, L_{14}\rangle^3$, the QM_1 is updated with the current accuracy of W_4 and QM_1. Then INQUIRE returns the new result $\langle Q_1\_Result\rangle^3$ depending on QM_1 (line 3 to line 5). After that, three answers of Q_1 have been completely received and then INQUIRE updates WM_1, WM_4, and WM_9 (line 7).

Compared INQUIRE to CDAS, the main difference between them is that the workers' accuracy never changes in CDAS, while the variation of workers' accuracy is expressed by updating worker model in INQUIRE. Differing from INQUIRE, every time EM receives a new answer, it re-estimates every worker's quality and infers new results relying on all received answers. For example, in example 1, when EM receives $\langle Q_1, W_4, L_{14}\rangle^3$, in addition to three answers of Q_1, EM applies other questions' answers $\langle Q_i, W_k, L_{ik}\rangle^j$ collected so far to obtain every question's result and every worker's accuracy.

4 Question Model and Worker Model in INQUIRE

In this section, we introduce the question model and worker model. The question model is utilized to infer the result of questions and the worker model is used to evaluate the quality of workers.

Algorithm 1. INQUIRE

Input: $\langle Q_i, W_k, L_{ik}\rangle^j$
Output: $\langle Q_i\_Result\rangle^j$

```
1 begin
2     for arriving ⟨Q_i, W_k, L_ik⟩^j do
3         W_k_accuracy ← WM_k;
4         QM_i ← (QM_i, W_k_accuracy);
5         ⟨Q_i_Result⟩^j ← QM_i;
6         for W_k has answered Q_i do
7             WM_k ← (QM_i, WM_k, L_ik);
```

4.1 Question Model

INQUIRE builds question model QM_i: $(p_i, 1-p_i)$ for question Q_i where p_i is the probability that question Q_i's true result is the first choice and $1-p_i$ is the probability that question Q_i's true result is the second choice. For each question Q_i, INQUIRE compares the value of p_i with 1-p_i and chooses the choice with larger probability as the inference result. That is, if $p_i > 1-p_i$, INQUIRE takes the first choice as the result, otherwise INQUIRE returns the second choice. The initial value of p_i is 0.5.

4.2 Worker Model

The key part of achieving high-quality inference result is to estimate workers' quality in time. The fixed-quality strategy in CDAS would make the inference results not very accurate since setting a fixed value as each worker's quality neglects the change of each worker's quality with time. To address this problem, some algorithms [7,14,16] propose to use confusion matrix to calculate workers' quality. Confusion matrix is builded by comparing worker's answers to inference results that EM returns. However, because EM probably infers results with low accuracy when it receives a small number of answers, the values of confusion matrix may be inaccurate.

Different from previous work, this paper proposes a more accurate worker model to compute worker's quality. INQUIRE builds a worker model (WM_k) for each worker

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$$

The row subscript (the first 0/1 number) means the answer that worker gives and the column subscript (the second 0/1 number) means the true result of the question. Let c_{ij} denote the total contribution of the worker's answers to questions. To a question, the worker's contribution is represented by the value of QM_i based on the worker's answer. For example, suppose that QM_1 is (0.6, 0.4) and L_{11} is 1. Then the contribution of W_1 to that Q_1's true result is first

choice is 0.6. Our worker model is different from confusion matrix. In confusion matrix, the value is the number of times a question which inference result is j was answered as i. For example, if p_2=0.54 and L_{23}=0, that is Q_2's inference result is 1. Then c_{01} in W_3's confusion matrix plus one. Because p_2 is just 0.54, the inference result is incorrect with high probability. That means confusion matrix can not precisely represent actual worker's performance sometimes.

We can easily use the worker model to calculate the accuracy of a worker. There are two methods we can compute the worker's accuracy. The first one is that the worker's accuracy is computed separately when the worker gives different choice. If the answer L_{ik} is 1, the worker W_k's accuracy is denoted by α_k. Let β_k denote the W_k's accuracy if L_{ik} is 0. α_k and β_k can be computed with Formulas 1 and 2, respectively.

$$\alpha_k = p(R_i = 1|L_{ik} = 1) = \frac{c_{11}}{c_{11} + c_{10}} \tag{1}$$

$$\beta_k = p(R_i = 0|L_{ik} = 0) = \frac{c_{00}}{c_{00} + c_{01}} \tag{2}$$

Some workers show biases for certain types of questions and their answers tend to one choice [7], so α_k and β_k are not accurate for bias workers. In this paper, if the difference between α_k and β_k is more than 50%, we consider this worker as a bias one.

The second method uses a general accuracy, which is that no matter what answers the worker returns, the W_k's accuracy is calculated as (called γ_k) :

$$\gamma_k = \frac{c_{11} + c_{00}}{c_{11} + c_{10} + c_{00} + c_{01}} \tag{3}$$

We can initialize each worker model by qualification test or the worker's historical records. If there is not any pre-information of workers, then c_{ij} is 0, and α_k, β_k, γ_k are all set to 50%.

Example 2. Assume that, in example 1, the worker models for three workers who answered Q_1 are the following.

$$WM_1{:}\begin{bmatrix} 11 & 6 \\ 7 & 12 \end{bmatrix}, WM_4{:}\begin{bmatrix} 3 & 15 \\ 2 & 9 \end{bmatrix}, WM_9{:}\begin{bmatrix} 11 & 6 \\ 3 & 10 \end{bmatrix}$$

We calculate these workers' accuracy respectively by Formulas 1, 2 and 3. The results are shown in Table 1. From Table 1, we observed that W_4 tends to choose first choice as the answer. The difference between α_4 and β_4 is more than 50%, so W_4 is a biased worker.

Table 1. Workers' accuracy

WorkerID	α	β	γ
W_1	0.632	0.647	0.639
W_4	0.818	0.167	0.414
W_9	0.769	0.647	0.7

5 Updating Question Model and Worker Model

In this section, we discuss how to incrementally update the two models in INQUIRE.

5.1 Updating Question Model

Whenever a new answer returns, QM_i is updated. We propose two different updating strategies for QM_i: Weighted Strategy and Probability Strategy. From the voting perspective, we design the first strategy and from the probabilistic perspective, we design the second strategy.

Weighted Strategy. Weighted Strategy (called *WS*) is a weighted voting method. The current inference result is gotten by weighted voting, and QM_i is the weight of this vote. The new answer is another vote, and its weight is the accuracy of the worker. Since the accuracy of a worker can be expressed in two patterns $\alpha\beta$ and γ (as discussed in Section 4.2), updating QM_i can be calculated by Formulas 4 or 5. Formula 4 is called *WS*-$\alpha\beta$ for short and Formula 5 is called *WS*-γ:

$$p_i = \frac{p_i + (\alpha_k \cdot L_{ik} + (1-\beta_k)\cdot(1-L_{ik}))}{2} \tag{4}$$

$$p_i = \frac{p_i + (\gamma_k \cdot L_{ik} + (1-\gamma_k)\cdot(1-L_{ik}))}{2} \tag{5}$$

The example 3 illustrates how the *WS* works.

Example 3. Assume that, in example 1, the received answers are set as $\langle Q_1,W_1, 1\rangle^1$,$\langle Q_1,W_9,0\rangle^2$ and $\langle Q_1,W_4,1\rangle^3$. We take these three answers and the workers' accuracy in Table 1 as an example. Figure 2 and Figure 3 describe *WS* more specifically. Figure 2 is the process of updating QM_1 by *WS*-$\alpha\beta$. Since $\langle Q_1,W_1,1\rangle^1$ is the first received answer of Q_1, based on Formula 1 which is expressed "(1)" in the figure, the accuracy of W_1 is α_1=0.632. Taking α_1=0.632 and p_1=0.5 of current QM_1 into Formula 4, new p_1 is calculated as 0.656. Then QM_1 is (0.656,0.434). At this time, p_1>1-p_1, so the first inference result of Q_1 is $\langle Q_1\_1\rangle^1$. The second answer of Q_1 is $\langle Q_1,W_9,0\rangle^2$ and the accuracy of W_9 is calculated by Formula 2, so β_9=0.647. Formula 4 calculates new p_1 using β_9=0.647 and p_1=0.656. Then QM_1 is (0.459,0.541). Since p_i<1-p_i, the second inference result of Q_1 is $\langle Q_1\_0\rangle^2$. Received the third answer $\langle Q_1,W_4,1\rangle^3$, based on α_4=0.818 and p_1=0.459, Formula 4 calculates new p_1=0.639. QM_1 is (0.639,0.361). p_1>1-p_1, so the third inference result of Q_1 is $\langle Q_1\_1\rangle^3$.

Figure 3 illustrates the process of *WS*-γ. The updating process of *WS*-γ is similar to that of *WS*-$\alpha\beta$. There are only two differences between them: the first is that the worker's accuracy is calculated by Formula 3 and the second is that Formula 5 updates p_i, in *WS*-γ.

As seen in Figure 2 and Figure 3, the third inference result of *WS*-$\alpha\beta$ is different from the result of *WS*-γ. According to these workers' accuracy in Table 1, because W_4 is bias as mentioned in Section 4.2, it is reasonable that the third

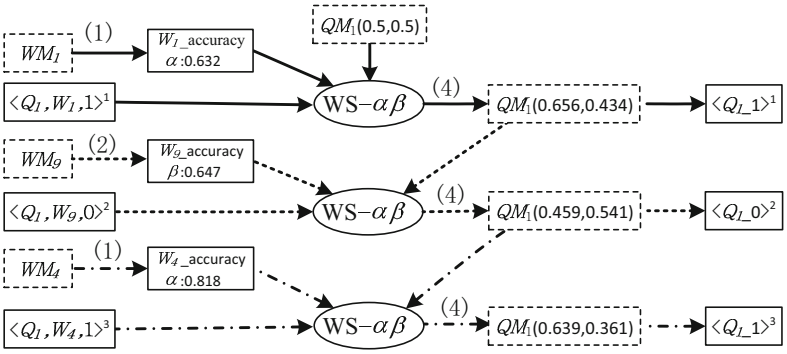


Fig. 2. Updating process of *WS*-$\alpha\beta$

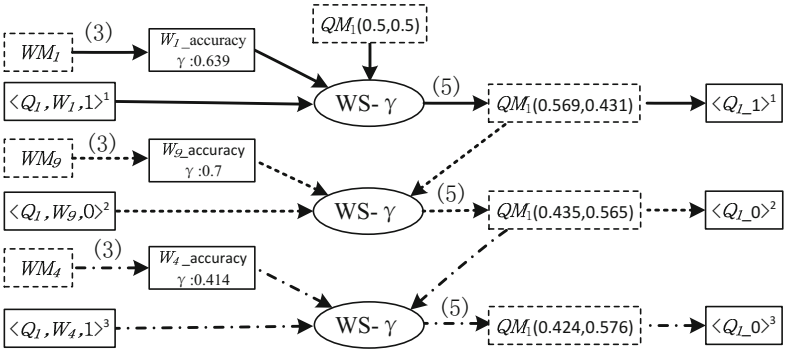


Fig. 3. Updating process of *WS*-γ

inference result is 0. That is, when some specific workers are biased, the inference results are more accurate using *WS*-γ to update question model. We compare these two strategies through experiments in Section 6.

Probability Strategy. Probability strategy (called *PS*) applies Bayesian methods to update the question model. Let A_1 denote the new arriving answer which is L_{ik}. And A_2 represents the current inference result. Given $A = \{A_1, A_2\}$, we can compute the probability that the first choice is the true result (i.e. p_i) by Bayesian formula, following Formula 6.

$$p(R_i = 1|A) = \frac{p(A|1) \cdot p(1)}{p(A)} = \frac{p(A|1) \cdot p(1)}{p(A|1) \cdot p(1) + p(A|0) \cdot p(0)} \tag{6}$$

The prior probability of first choice is equal with the prior probability of second choice, so Formula 6 can be simplified as:

$$p(R_i = 1|A) = \frac{p(A|1)}{p(A|1) + p(A|0)} \tag{7}$$

We know that $p(A|R_i) = p(A_1|R_i) \cdot p(A_2|R_i)$ $(R_i$=0,1). Formulas 8 and 9 calculate $p(A_1|R_i)$ as $\alpha\beta$ expresses the workers' accuracy. Formula 10 calculates $p(A_1|R_i)$ as γ expresses the worker's accuracy. We denote $I(cond)$ as the decision function. That is, the result is 1 if the *cond* is true, otherwise the result is 0.

$$p(A_1|R_i = 1) = \alpha^{I(A_1=R_i)} \cdot (1-\alpha)^{I(A_1 \neq R_i)} \tag{8}$$

$$p(A_1|R_i = 0) = \beta^{I(A_1=R_i)} \cdot (1-\beta)^{I(A_1 \neq R_i)} \tag{9}$$

$$p(A_1|R_i) = \gamma^{I(A_1=R_i)} \cdot (1-\gamma)^{I(A_1 \neq R_i)} \tag{10}$$

$P(A_2|R_i)=\frac{p(R_i|A_2)\cdot p(A_2)}{p(R_i)}$. The prior probabilities of A_2 and R_i are equal. Then $P(A_2|R_i)$ is calculated as:

$$p(A_2|R_i = 1) = \frac{P(R_i = 1|A_2) \cdot p(A_2)}{p(R_i = 1)} = p(R_i = 1|A_2) = p_i \tag{11}$$

$$p(A_2|R_i = 0) = \frac{P(R_i = 0|A_2) \cdot p(A_2)}{p(R_i = 0)} = p(R_i = 0|A_2) = 1 - p_i \tag{12}$$

As $\alpha\beta$ expresses the worker's accuracy, Formulas 8, 9, 11 and 12 are substituted into Formula 7. We have Formulas 13 which is called *PS*-$\alpha\beta$.

$$p(R_i = 1|A) = \frac{\alpha^{I(A_1=1)} \cdot (1-\alpha)^{I(A_1 \neq 1)} \cdot p_i}{\alpha^{I(A_1=1)} \cdot (1-\alpha)^{I(A_1 \neq 1)} \cdot p_i + \beta^{I(A_1=0)} \cdot (1-\beta)^{I(A_1 \neq 0)} \cdot (1-p_i)} \tag{13}$$

As γ expresses the worker's accuracy, combining Formulas 10, 11 and 12, we have Formulas 14 (called *PS*-γ).

$$p(R_i = 1|A) = \frac{\gamma^{I(A_1=1)} \cdot (1-\gamma)^{I(A_1 \neq 1)} \cdot p_i}{\gamma^{I(A_1=1)} \cdot (1-\gamma)^{I(A_1 \neq 1)} \cdot p_i + \gamma^{I(A_1=0)} \cdot (1-\gamma)^{I(A_1 \neq 0)} \cdot (1-p_i)} \tag{14}$$

Example 4 shows how *PS* works.

Example 4. We use the answers and workers in Example 3 as an example. Figure 4 illustrates the process of *PS*-$\alpha\beta$, and Figure 5 shows the process of *PS*-γ. The process of *PS* is similar to that of *WS* in Example 3. The only difference between *PS* and *WS* is that they apply different formulas to update p_i.

The results in Figures 4 and 5 are similar to the results in Figure 2 and 3. The reason that Figure 4 and 5 return different third inference result is similar to the analysis of Figure 2 and 3. In Section 6, we compare *WS* with *PS* through experiments.

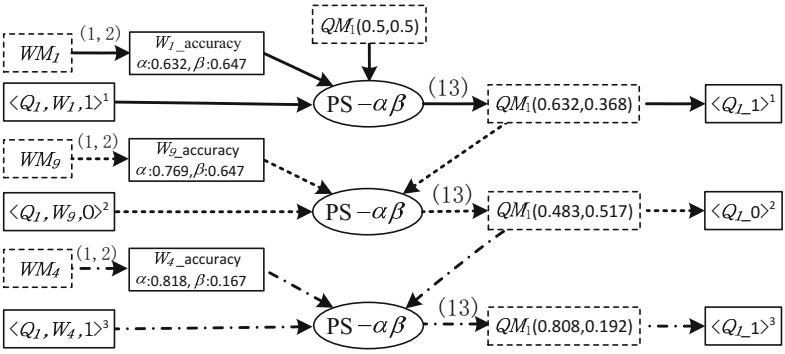


Fig. 4. Updating process of *PS*-$\alpha\beta$

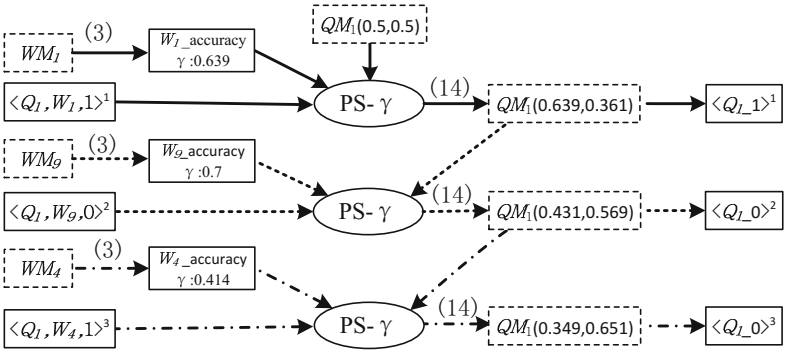


Fig. 5. Updating process of *PS*-γ

5.2 Updating Worker Model

As discussed in Section 3, INQUIRE has to compute the change of workers' accuracy in time, and update m workers' model when a question has received m answers. We use QM_i to calculate c_{ij} as mentioned in Section 4.2. According to L_{ik}, WM_k is updated as:

$$c_{00} = c_{00} + (1 - p_i), c_{01} = c_{01} + p_i \qquad \textbf{if } L_{ik} = 0 \tag{15}$$

$$c_{10} = c_{10} + (1 - p_i), c_{11} = c_{11} + p_i \qquad \textbf{if } L_{ik} = 1 \tag{16}$$

The example 5 shows the process of updating worker model.

Example 5. We take the results in Figure 5 as an example. Since L_{11}=1 and L_{14}=1, we can apply Formula 16 to update WM_1 and WM_4. WM_9 is updated by Formula 15 since the answer returned by W_9 is 0. The new workers' models are as follows.

WM_1: $\begin{bmatrix} 11 & 6 \\ 7+0.651 & 12+0.349 \end{bmatrix}$, WM_4: $\begin{bmatrix} 3 & 15 \\ 2+0.651 & 9+0.349 \end{bmatrix}$, WM_9: $\begin{bmatrix} 11+0.651 & 6+0.349 \\ 3 & 10 \end{bmatrix}$

6 Experiments

In this section, we conduct experiments to evaluate our method. Our experimental goal is to test the efficiency and quality of different inference methods.

6.1 Experiment Setting

Platform. We conducted our experiments on real crowdsourcing platform Amazon Mechanical Turk. We implemented our method using python. All the experiments were run on a PC with Intel core i5 duo 2.6GHz CPU and 4GB RAM.

Datasets. We used two sets of binary choices tasks: Filmpair and Animal. (a) Filmpair is a dataset of movie poster. Each question contains two movie posters and workers decide which movie is released earlier. The ground truth of Filmpair is the actual released time of movies. We choose the most well-known movie posters in 1996-2006 from IMDB. There are 2000 questions and each question is answered three times. A total number of 146 workers answered these questions. (b) `Animal` is a dataset of animal pictures. Each question contains two animal pictures and workers decide which animal's size is larger. The ground truth of Animal is based on Animal-Size-Comparison-Chart[1]. Animal contains 300 questions and each question is completed by five workers. 36 workers participated in Animal. We use these datasets in all experiments because these are different types of questions and different number of questions.

Comparison Methods. Firstly, we choose 600 answers and 150 answers respectively from Filmpair and Animal. These answers are used as historical information to initialize workers' model. The remaining answers are the validation dataset. Then we evaluate Inquire on the validation dataset from the following three aspects: (1) Section 6.2 evaluates the effectiveness of worker model; (2) Section 6.3 compares the inference results' accuracy between Inquire and other inference methods (MV,CDAS and EM); (3) Section 6.4 compares the runtime between Inquire and other three existing methods mentioned above. To be general, all these methods run three times and the final experimental results are the average of the three results.

6.2 Worker Model Analysis

We verify the effectiveness of the worker model by comparing the similarity between worker model and real confusion matrix (called CM_k). This paper applies the cosine distance to measure the similarity. The value of cosine distance is between 0 to 1. The bigger value means that WM_k and CM_k are more similar. Let define CM_k:

$$\begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$

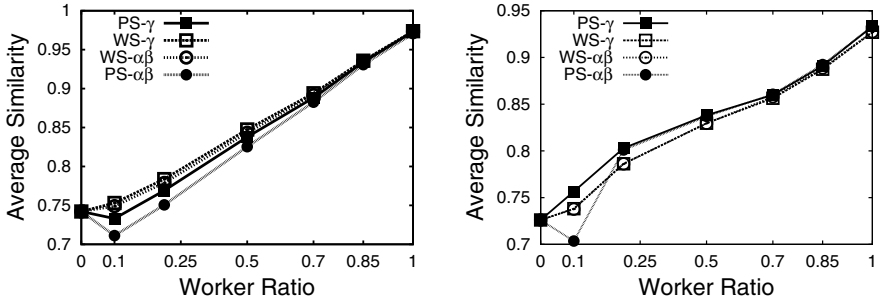
[1] `http://myuui.deviantart.com/art/Animal-Size-Comparison-Chart-109707959`

The similarity between WM_k and CM_k is calculated as follows:

$$Sim(WM_k, CM_k) = \frac{(c_{00} \cdot b_{00} + c_{01} \cdot b_{01} + c_{10} \cdot b_{10} + c_{11} \cdot b_{11})}{\sqrt{c_{00}^2 + c_{01}^2 + c_{10}^2 + c_{11}^2} \cdot \sqrt{b_{00}^2 + b_{01}^2 + b_{10}^2 + b_{11}^2}}$$

As mentioned in Section 4.2, we need worker's historical performance to initialize the value of worker model. We define the percentage of workers with historical-information in total workers as `worker ratio`. Because we cannot guarantee that every worker has historical performance, we study the effect of `worker ratio` on similarity between WM_k and CM_k. We vary `worker ratio` from 0%, 10%, 25%, 50%, 70%, 85% to 100% and plot the average similarity of four strategies (*WS*-$\alpha\beta$, *WS*-γ, *PS*-$\alpha\beta$, *PS*-γ) as addressed in Section 5.1.

From Figure 6, we can see that the similarity grows with `worker ratio` increase regardless of the *PS* or *WS*. The worker model can achieve relatively similar results to the real confusion matrix when `worker ratio` is higher than 50%. Especially, when the `worker ratio` is 100%, i.e., all the workers has initial information, the similarity is above 90%. This indicates that our worker model is more reasonable for the workers' quality with a higher `worker ratio`.



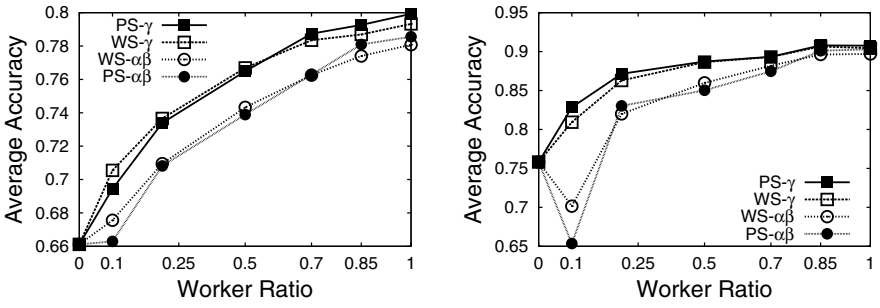
(a) Filmpair (b) Animal

Fig. 6. Average similarity between worker model and real confusion matrix

6.3 Accuracy Analysis

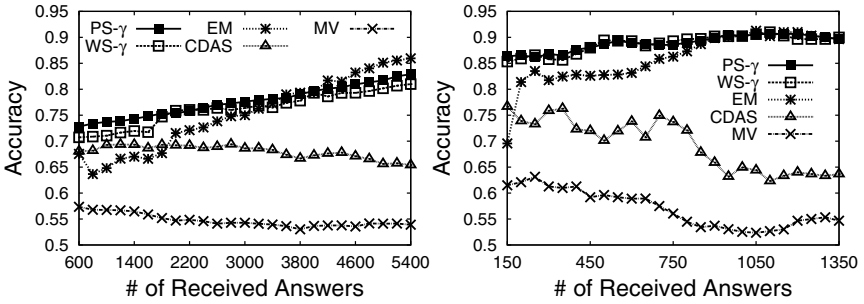
We first compare the results' accuracy of four strategies (*WS*-$\alpha\beta$, *WS*-γ, *PS*-$\alpha\beta$, *PS*-γ). The results' accuracy is calculated by comparing the inference results with the ground truth. Figure 7 shows the effect of `worker ratio` on the average accuracy of four strategies. We can see that the results of *PS*-γ and *WS*-γ are better than that of *PS*-$\alpha\beta$ and *WS*-$\alpha\beta$ no matter how `worker ratio` varies. The main reason is that some workers subjectively want to choose certain choice as an answer as mentioned in Section 4.2. This results indicate that, to reflect workers' accuracy, γ method is better than $\alpha\beta$. Thus in the following experiments, we only evaluate γ method. In addition, the accuracy with 70% `worker ratio` is close to that with 100% `worker ratio`, so we use 70% `worker ratio` in the following experiments.

Next we evaluate the online results of MV, CDAS, EM and Inquire. The accuracy of the results are shown in Figure 8. We can observe that, on both



(a) Filmpair (b) Animal

Fig. 7. Average accuracy by varying `worker ratio`



(a) Filmpair (b) Animal

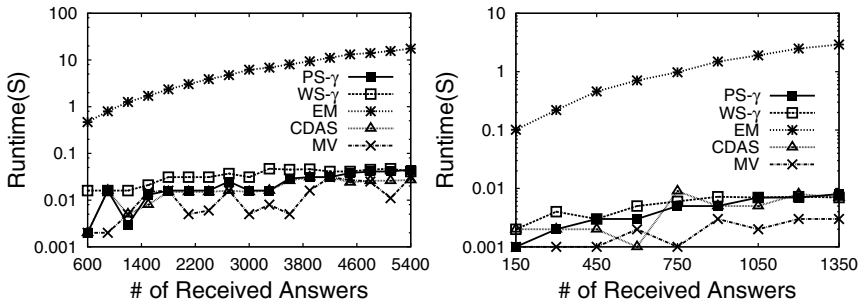
Fig. 8. Accuracy on the number of received answers

datasets, the accuracy of Inquire with *WS*-γ or *PS*-γ is always higher than that of MV and CDAS, because Inquire can reflect the change of workers' accuracy in time. Since the workers' quality are equal in MV, just like the analysis in section 1, the results of MV is worse than those of CDAS. We also observed that the results of EM changes a lot. For example, on Filmpair dataset, the accuracy is only about 0.65 when receiving 600 answers. Once received all answers, the accuracy can achieve 0.85 which is 0.03 higher than those of Inquire with *PS*-γ. However, the accuracy of Inquire increased gradually with increased answers. When receiving less than 70% of answers Inquire performs better than EM. On Animal dataset, EM and Inquire works similarly when receiving 100% answers, since the amount of answers is not large. As the results in Filmpair, EM also performs poorly in the beginning. We addressed in section 4.2 that the values of confusion matrix maybe not precise when the number of answers is small. And then the EM's results are possible not accurate. That is the main reason why EM has worse performance than Inquire with fewer answers.

6.4 Runtime Analysis

In this section, we compare the runtime of our method to other three methods mentioned in Section 6.3. Figure 9 shows the effect of the number of received answers on runtime. Compared the runtime of Inquire with that of MV and

CDAS, we can see that these three methods always run in milliseconds. The number of received answers has slight influence on these methods' runtime. However, we can observe that EM's runtime rises sharply when received answers grows. For example, on Filmpair dataset, the running time of EM costs almost 20 seconds with 100% received answers, while its processing takes less than a second in the beginning.



(a) Filmpair (b) Animal

Fig. 9. Runtime on the number of received answers

Next we study time complexity of EM and INQUIRE. EM algorithm is iteratively calculated by two steps. In the E-step, it calculates the results' probability of every question by known workers' accuracy. In the M-step, it calculates every worker's accuracy according to the results obtained in the E-step. The time complexity of the E-step is $O(2n+2a)$, n is the number of questions and a is the number of received answers. The time complexity of the M-step is $O(max(2n, 2a))$. Thus the running time of EM increases linearly with the increase of n and a. So when the number of answers is large EM does not perform well. According to Algorithm 1 in Section 3, time complexity of INQUIRE is $O(1)$, that is, neither the number of questions nor received answers influence the runtime of INQUIRE.

In summary, when the accuracy and runtime are considered together, INQUIRE is obviously superior to MV, CDAS and EM. Comparing the results of *PS*-γ with *WS*-γ in Figure 7 and 8, we can see that the accuracy of *PS*-γ is better than that of *WS*-γ with more answers. So we propose to adopt *PS*-γ in INQUIRE.

7 Conclusion

In this paper, we studied the problem of incremental quality inference in Crowdsourcing. We presented an incremental inference method, INQUIRE, which contains the question model and worker model. For the question model, we proposed two different updating strategies to efficiently infer results. For the worker model, it can dynamically represent workers' quality, and we proposed an incremental strategy to update worker model. We evaluated our method on real-world datasets. Compared to MV, CDAS and EM, INQUIRE achieved a good trade-off between accuracy and time, and thus INQUIRE is more effective and efficient.

Acknowledgement. This work was partly supported by the National Natural Science Foundation of China under Grant No. 61373024, National Grand Fundamental Research 973 Program of China under Grant No. 2011CB302206, Beijing Higher Education Young Elite Teacher Project under grant No. YETP0105, a project of Tsinghua University under Grant No. 20111081073, Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, the "NExT Research Center" funded by MDA, Singapore, under Grant No. WBS:R-252-300-001-490, and the FDCT/106/2012/A3.

References

1. http://crowdflower.com/
2. http://www.mturk.com
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J.R.Statist.Soc.B 30(1), 1–38 (1977)
4. Feng, A., Franklin, M., Kossmann, D., Kraska, T., Madden, S., Ramesh, S., Wang, A., Xin, R.: Crowddb: Query processing with the vldb crowd. Proceedings of the VLDB Endowment 4(12) (2011)
5. Franklin, M.J., Kossmann, D., Kraska, T., Ramesh, S., Xin, R.: Crowddb: Answering queries with crowdsourcing. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 61–72. ACM (2011)
6. Howe, J.: Crowdsourcing: How the power of the crowd is driving the future of business. Random House (2008)
7. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation, pp. 64–67. ACM (2010)
8. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in Neural Information Processing Systems, pp. 1953–1961 (2011)
9. Liu, X., Lu, M., Ooi, B.C., Shen, Y., Wu, S., Zhang, M.: Cdas: A crowdsourcing data analytics system. Proceedings of the VLDB Endowment 5(10), 1040–1051 (2012)
10. Marcus, A., Wu, E., Karger, D.R., Madden, S., Miller, R.C.: Demonstration of qurk: a query processor for humanoperators. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 1315–1318. ACM (2011)
11. Marcus, A., Wu, E., Karger, D.R., Madden, S.R., Miller, R.C.: Crowdsourced databases: Query processing with people. In: CIDR (2011)
12. Mason, W., Suri, S.: Conducting behavioral research on amazon mechanical turk. Behavior Research Methods 44(1), 1–23 (2012)
13. Park, H., Garcia-Molina, H., Pang, R., Polyzotis, N., Parameswaran, A., Widom, J.: Deco: A system for declarative crowdsourcing. Proceedings of the VLDB Endowment 5(12), 1990–1993 (2012)
14. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. The Journal of Machine Learning Research 99, 1297–1322 (2010)
15. Wang, J., Kraska, T., Franklin, M.J., Feng, J.: Crowder: Crowdsourcing entity resolution. Proceedings of the VLDB Endowment 5(11), 1483–1494 (2012)
16. Whitehill, J., Wu, T.-F., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in Neural Information Processing Systems, pp. 2035–2043 (2009)
17. Yuen, M.-C., King, I., Leung, K.-S.: A survey of crowdsourcing systems. In: 2011 IEEE Third International Conference on Social Computing (socialcom), pp. 766–773. IEEE (2011)