A Dynamic Difficulty-Sensitive Worker Distribution Model for Crowdsourcing Quality Management

Miao Zheng(✉), Lizhen Cui, Wei He, Wei Guo, and Xudong Lu

School of Software, Shandong University, Jinan, China
zheng_miao@outlook.com, {clz,hewei,guowei,dongxul}@sdu.edu.cn

Abstract. Crowdsourcing utilizes the intelligence of people to solve problems that are difficult for machines such as entity resolution, sentiment analysis and image recognition. In crowdsourcing systems, requesters publish tasks that are answered by workers. However, the responses collected from the crowd are ambiguous as the workers on internet with unknown and very diverse abilities, skills, interests and knowledge background. In order to ensure the quality of crowdsourcing results, it is important to characterize worker quality accurately. Many previous works model the worker quality by a fixed value (such as probability value or confusion matrix). But even when workers complete the same type of tasks, the quality is affected by some factors (task difficulty) to varying degrees. Here we propose a dynamic difficulty-sensitive worker quality distribution model. In our model, the worker's ability is affected by task difficulty and fits a functional distribution. This model reflects the relationship between worker reliability and task difficulty. In addition, we utilize Expectation-Maximization approach (EM) to obtain maximum likelihood estimates of the parameters of worker quality distribution model and the true answers to the tasks. We conduct extensive experiments with synthetic data and real-world data. The experimental results show that our method significantly outperforms other state-of-the-art approaches.

Keywords: Crowdsourcing quality management · Worker quality distribution model · Maximum likelihood estimation

1 Introduction

Crowdsourcing can utilize the intelligence of the internet workers to solve some problems which are difficult for machine. At present, there are many successful crowdsourcing platforms, such as AMT [1], Upwork [3] and Crowdflow [2]. When requester publish tasks in the crowdsourcing platforms, workers can accept and complete tasks to obtain the corresponding reward. Moreover, many types of tasks can be solved by crowdsourcing, including annotation and tagging of

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 12–27, 2019.
https://doi.org/10.1007/978-3-030-30146-0_2

images and documents, writing and reviewing software codes, product design and financing. Although corwdsourcing use wisdom of workers enables to solve machine-hard tasks more quickly and accurately, the quality of workers is also uneven due to the different age, education and knowledge background. It is very important to model the quality of workers accurately in crowdsourcing. Worker quality also plays an important role in answer aggregation. When estimating the answer to a task, we mainly rely on the answers from different workers. The quality of workers will have a great impact on estimating the true answer to the task. In addition, quality of workers also be taken into account in the task assignment.

Consider the following example. A machine learning system needs a lot of labeled data. It publishes some Twitter messages on crowdsourcing platforms as emotional analysis tasks. Workers need to judge whether twitter messages express negative or positive emotions, then workers complete tasks and submit answers. Each task may receive multiple answers. Each worker may also answer multiple tasks and the difficulty of each task varies. If a task receives 10 responses from different workers, one of them is "negative" and nine responses are "positive", then we can see that this task is less controversial and simpler. Generally, workers may be better at completing tasks which are easier to identify the answer. On the contrary, their reliability will decrease when answering difficult tasks, and each worker's reliability will be affected differently. In previous works on crowdsourcing quality management, the quality of workers is mostly expressed by the accuracy of task completion or a confusion matrix. However, these worker models can not accurately describe the quality characteristics of workers.

In this paper, we focus on labeling tasks, such as emotional analysis, pattern recognition, image recognition and so on. We propose a worker quality distribution model, in which the worker's quality is a distribution that changes with the task difficulty. It can describe the worker's quality characteristics more accurately, and we also propose a fitting algorithm to form each worker's quality model. In addition, we propose an inference algorithm based on maximum likelihood estimation to estimate the parameters of the worker quality model and the true answer of tasks. Our aim is to estimate the correct answers to the tasks and the quality of the workers more accurately. This paper mainly makes contributions including the following three points.

(1). We propose a dynamic difficulty-sensitive worker quality distribution model, which can describe the worker quality characteristics more accurately. In our worker quality distribution model, we model the task difficulty according to the response received by the task and design a worker quality distribution function.

(2). We propose an inference algorithm based on maximum likelihood estimation. We maximize the likelihood to estimate the parameters of the worker quality model and the true answers to the tasks by EM method. Since there are many unobserved variables in the likelihood function, we also propose a fitting algorithm to get the worker quality distribution model of each worker.

(3). We compare our method with other crowdsourcing quality management algorithms through experiments with synthetic data and real-world data, and results show that our method has higher estimation accuracy than other algorithms.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes our conceptual definitions and the problem we study. Section 4 presents the proposed worker quality distribution. We describe our inference algorithms in Sect. 5. Section 6 discusses our experimental evaluation and Sect. 7 concludes this work.

2 Related Work

In order to address the quality management problem in crowdsourcing, various techniques have been proposed to estimate the true answers of tasks based on the worker quality.

The general methods for quality management are majority voting [10,19], The majority voting strategy returns the result with the most votings. In addition, Wang et al. [5] propose a worker quality-aware model is similar to Hidden Markov Models (HMM). Ma et al. [7] propose a fine grained truth discover model to estimate both worker topical expertise and true answers. Before the task is answered, the workers can be evaluated whether they know the relevant knowledge of the task by adding a Qualification test. This can not only eliminate some fraudsters, but also eliminate some workers who do not know the task. It can also make the workers more familiar with the task and improve the quality of the results. Randomly add one in the task. Questions that have the right answers are tested for worker quality [8]. [4] obtained the accuracy of workers'answers by increasing the test questions, and used Bayesian theory to combine the accuracy of workers' answers with the answers given by workers to get the final results.

The Expectation Maximization (EM)-based methods [6,14,21,25,27–29] are the state-of-the-art approaches to estimate task true answers and worker quality. The EM algorithm is primarily used for making up for the lack of data by iteration calculation of maximum likelihood estimate from incomplete data [13]. Each iteration of algorithm includes an expectation step and a maximization step. In Crowdsourcing, the incomplete data is workers responses for tasks and the unobserved latent variables is the task true answer and worker quality. Worker quality can be characterized by worker model, include worker probability [15,20] and confusion matrix [16,24]. EM-based methods iteratively update the parameters of worker model and the true answers of tasks until convergence. Dawid and Skene [12] used EM algorithm in a scene where was similar to Crowdsourcing, for the problem about errors existing in the collected patient records. Later, Ipeirotis et al. [16] Proposed the EM algorithm for crowdsourcing with data from AMT platform, not only estimate the correct answer of tasks, but also get the workers quality represented by the error rate matrix. Moreover, there are many factors that influence the quality of workers. Movellan et al. [30] proposed the method which took task difficulty as parameters into EM algorithm. Hence the iterative process contains three unknown variables, the true answer of tasks, workers' expertise and tasks' difficulties. Afterwards, Kurve et al. [22] proposed to utilize EM algorithm to calculate the task answers and workers quality. They take four latent variables into consideration, the true answer of task, skills of workers, workers intention (i.e. honest worker or dishonest worker) and the task difficulty.

Some workers focus on theoretical guarantees, they provide probabilistic bounds [11,18,26,31] for task answers and worker quality estimates. For example, Dalvi et al. [11]show that the error in their estimates of worker quality is lower by θ under certain assumptions about the graph structure. Das Sarma et al. [26] proposed a technique for globally optimal quality management, finding the maximum likelihood item ratings and worker quality estimates. But its assumptions are not true in reality and the amount of computation increases dramatically as the number of tasks increases.

3 Problem Statement

In this chapter, we first introduce the notation (Table 1) involved in our method, and then describe the problems we study in this paper, some of which will be described in detail in later chapters.

Table 1. Notation table.

Symbol	Explanation
t	Task
w	Worker
r_t^w	Responses from worker w to task t
z_t	The true answer of task
M_t	The task response set
d_t	The task difficulty
$Q(d)$	The worker reliability function
L	Overall likelihood

Task Question and Option. Consider a group of tasks $\{t\}^n$, whose total number is n, these tasks are completed by a group of workers $\{w\}^m$, whose total number is m. Worker w completes task t through k options $\{1, 2, 3..., k\}$ as his response r_t^w. Each worker may accomplish many different tasks, and each task may be accomplished by many different workers. And each task has a true answer z (that is, one of the k options is the correct answer).

Example 1. For example, there is a set of emotional analysis tasks $\{t_1, t_2\}$ with three options: option 1 is positive, option 2 is natural and option 3 is negative. A group of workers $\{w_1, w_2\}$ give responses to the emotional tasks. The real answer to task t_1 is positive, and the real answer to task t_2 is natural.

Task Response Set. After the task receives the worker's responses, we can know the response set of the task $M_t = (v_1, v_2, v_3...v_k)$ refers to the number of responses received by each option of the task.

Example 2. For example, there is a task with three options $\{1, 2, 3\}$, one worker chooses option 1, three workers choose option 2, and seven workers choose option 3. So the response set of this task is (1, 3, 7).

Task Difficulty. We get the task difficulty d_t according to the state of the task's response set. The more difficult a task is, the more difficult it is for workers to distinguish the correct answer for task. Then we can judge the difficulty of a task by the response set. The closer each option of a task receives, the more difficult the task is. The specific method for expressing task difficulty is introduced in Sect. 4.1.

Worker Reliability. Workers have different abilities to accomplish tasks with different difficulties, and the reliability of the answers given by workers is different. We use function $Q(d)$ to express the relationship between worker reliability and task difficulty. We have a detailed introduction in Sect. 4.2.

When a group of tasks $\{t\}^n$ receive the responses r_t^w from a group of workers $\{w\}^m$. We model the task difficulty d_t for each task and build the worker quality distribution model for each worker. Our goal is to estimate the true answer to the task accurately based on the worker quality distribution model.

4 Worker Quality Distribution Model

In this paper, we propose a worker quality distribution model, in which a functional distribution is used to represent the relationship between worker quality and task difficulty. In general, the quality of workers will be affected by the difficulty of the task. The more difficult the task is, the lower probability that workers can correctly answer the task. Here, we consider that the difficulty of tasks has a great relationship with the state of tasks (response set). We propose a method to reflect the difficulty of task according to the state of tasks(response set). In addition, the reliability of workers is affected by the task difficulty. Some workers still have high accuracy even when they complete the difficult tasks, and some workers have poor quality even if they complete simple tasks. We use a functional distribution model to visualize worker's reliability and how worker's reliability is affected by task difficulty.

4.1 Modeling Task Difficulty

We know that the more difficult a task is, the more difficult it is to distinguish the correct answer of task, that is, the more similar the number of answers it receives in different options. Consider an example, the response set of an emotional analysis task is (19, 17, 15). That is to say, 19 workers answered "positive", 17 workers answered "natural" and 15 workers answered "negative". The three options of the task received a similar number of responses. It shows that the task caused great controversy among the workers. It indicates that the task is difficult to estimate the correct answer. We divide tasks into binary task and multi-task, and express the difficulty of tasks according to response set of task, respectively.

In the binary task (tasks with two options), the response set of the task is (v_1, v_2). We use the ratio of the number of answers received by the two options to express the task difficulty. That is, the ratio of a small number to a large number of responses received with two options.

$$d_t = \frac{min(v_1, v_2)}{max(v_1, v_2)} \tag{1}$$

In this way, the task difficulty is controlled between 0 and 1. When the value of task difficulty is 0, it is the easiest to distinguish the answer for the representative task. When the value of task difficulty is 1, the task is the most controversial and the most difficult to estimate the true answer.

Example 3. For example, a task receives 10 responses, in which 2 workers answered 0/no and 8 workers answered 1/yes. Then the value of task difficulty is 0.25.

In multi-task (i.e. the number of task options is greater than two), the task has K options $\{1, 2, 3...k\}$, then the task receives a response set $(v_1, v_2, v_3...v_k)$. We know that variance is a measure of the degree of confusion in a set of numbers. However. if we simply use the variance of the task response set to indicate the difficulty of the task, sometimes there will be some errors. Let's see the following example: a group of tasks $\{t_1, t_2\}$ with three options. The response set of task t1 is (1, 2, 6) and the response set of task t_2 is (1, 5, 6). According to variance formula, the difficulty values of both tasks are 14, and we can easily see that task t_2 is more difficult than task t_1.

Therefore, we propose a difficulty representation method suitable for crowdsourcing multi-task. In the task response set, the higher degree of the most supported option is, the easier task can identify the true answer.

$$d_t = \sqrt{\frac{(\frac{v_1}{max\{v\}})^2 + (\frac{v_2}{max\{v\}})^2 + ... + (\frac{v_k}{max\{v\}})^2}{k}} \tag{2}$$

We represent task difficulty by the average sum of squares of the ratio of the number of responses received by each option to the highest number of responses. Similarly, task difficulty ranges from 0 to 1. The greater the value of d, the more difficult the task is. When $d = 1$, the task is the most controversial and the most difficult to estimate the true answer.

4.2 Worker Quality Distribution Function

There is a negative correlation between the quality of workers and the difficulty of tasks, that is, the more difficult the task is, the lower the reliability of workers. Here, we assume that the relationship between worker's reliability and task difficulty is bell-shaped distribution. The distribution function is as follows:

$$Q(d) = \mu + e^{-\delta d^2} \tag{3}$$

The worker quality distribution model can describe worker's quality characteristics more vividly than other worker model. For example, a worker w_1 completes a binary task t_1 with response set $(4, 1)$. If we use a fixed value such as accuracy to represent worker quality. No matter how difficult the task is, the reliability of worker is 0.7. In fact, the worker's reliability may be 0.8 when answering simple tasks. Therefore, the worker quality distribution model can describe the worker's characteristics in more details.

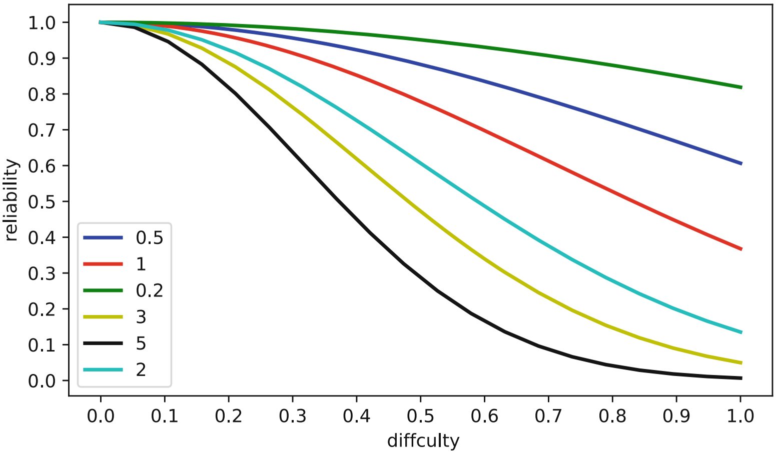


Fig. 1. Worker model

In the Fig. 1, the ordinates indicate the quality of the worker (that is, the ability to answer questions correctly). Where $\mu = 0$, the value of δ is constantly changing. From the figure, we can see that the smaller the value of δ, the smaller the impact of task difficulty on the quality of workers. And the parameters of each worker's quality distribution model are different.

5 Inference Algorithm

After we get the response r_t^w of workers $\{w\}$ to tasks $\{t\}$. We use maximum likelihood estimation to estimate the parameters of the worker model and the true answer to the task. We assume that all worker's responses are independent, then our goal is to maximize the likelihood function.

$$argmax_{z_t, d_t, q_w} \prod_{t,w} p(r_w^t | z_t) \tag{4}$$

5.1 Parameter Estimation Method

Here we use EM method to obtain maximum likelihood estimates of the parameters. EM method iteratively estimates the parameters of worker quality model and true answers of tasks through E-step and M-step. We know that EM algorithm needs initial parameters. Here, the initial parameter input of EM is task answer or worker model parameter. The initial answers of tasks can be obtained by the rule of majority voting; the initial parameters of the worker quality model can be set to the worker of good quality.

E-step. We get the response r_t^w of workers $\{w\}$ to tasks $\{t\}$. We estimate the probability of the true answer z_t of the task according to the values of the parameters of the worker quality model derived by M-step.

$$\begin{aligned} P(z^t | r, d, \mu, \delta) &= P(z^t | r_t^w, d_t, \mu_w, \delta_w) \\ &\propto p(z_t) \prod_w p(r_t^w | z^t, d_t, \mu_w, \delta_w) \end{aligned} \tag{5}$$

M-step. We maximize the expectation of likelihood function L log-Likelihood to estimate the parameters of worker quality model δ and μ based on the estimation of the answers to tasks derived by E-step and the workers' responses to tasks.

$$\begin{aligned} \mathbb{E}(\ln L) &= \mathbb{E}\left[\ln p(z, r, d | \mu, \delta)\right] \\ &= \mathbb{E}\left[\ln \prod_t \left(p(z_t) \prod_w p(r_t^w | z_t, d_t, \mu_w, \delta_w)\right)\right] \end{aligned} \tag{6}$$

Since there are many unobserved variables (z, δ and μ) in the expectation of likelihood function. We estimate the parameters of the worker quality model δ and μ by a fitting algorithm(in Sect. 5.2) based on our worker quality distribution function and the estimation of the answers of tasks derived by E-step.

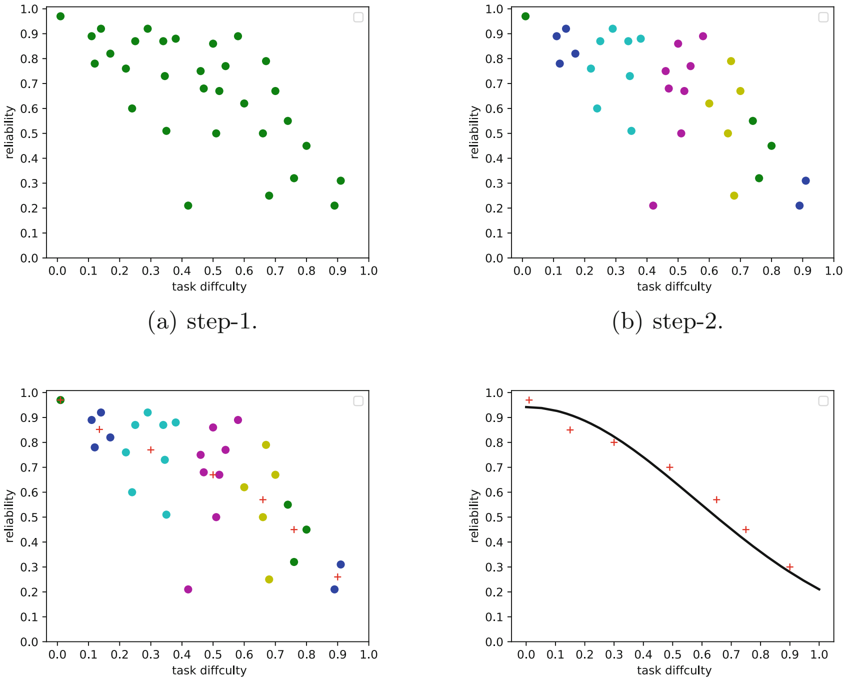
5.2 Fitting Algorithm for Worker Distribution Model

In our worker quality distribution model, each worker's reliability varies when he or she completes tasks with different difficulty. And reliability of workers will be affected differently by task difficulty. We propose a fitting algorithm (show in Algorithm 1) to derive the parameters of worker quality distribution model. Next, we describe the process of fitting the worker quality distribution function in detail.

Step 1: Finding the discrete data points of each worker model. In the previous chapter, we describe the representation method of task difficulty (i.e. independent variables) in the model. We use the response set of each task to indirectly represent the difficulty of the task. In addition, the reliability of the worker's response to the task represents the worker's ability (i.e. dependent variable).

Step 2: Clustering the discrete points in each worker's quality model. The purpose is to take some outliers into account as well. Because each worker completes many tasks with different difficulties, the data points in the model may be too scattered and there are many abnormal points. As a result, it is difficult to fit the worker model that best fits the characteristics of workers. So, we cluster the discrete data points to find some centroids that best reflect the characteristics of workers. When clustering, we will limit the corresponding task difficulty of data points in the same cluster to a centroid.

Step 3: We use the least square method to fit a curve closest to the worker's quality distribution function according to the centroids obtained by clustering, and get the parameters of each worker quality distribution model.



(a) step-1.

(b) step-2.

(c) step-2.

(d) step-3.

Fig. 2. An example of fitting algorithm (Color figure online)

The Fig. 2 shows an example of fitting worker quality model. In the Fig. 2(a), green points represent discrete data points. Figure 2(b) shows the results of our clustering, in which different color points represent different clusters. The red points in Fig. 2(c) represent centroid points obtained by clustering, and Fig. 2(d) shows the worker quality model fitted according to centroid points.

Algorithm 1. Fitting Algorithm.

```
Input: response set of task, $M_t$; reliability of worker, $Q_w$;
Output: Parameters of worker quality model $\mu, \delta$;
Model task difficulty $d_t$ based on $M_t$;
discrete data points $\{(d_t, Q_w)\}$
Initialize $k$ centroids $\{c\}$;
if cluster of any point changes then
    for each point$(d_t, Q_w)$ do
        for each centroid $c$ do
            Calculate the Distance between centroid $c$ and point $(d_t, Q_w)$;
            Put point into the nearest cluster;
            for each cluster do
                new centroid $c \leftarrow$ mean of points;
            end for
        end for
    end for
end if
$\mu, \delta \leftarrow$ least squares method with centroids $\{c\}$;
return $\mu, \delta$;
```

In the fitting algorithm, We first compute all discrete data points in the worker model. Then we cluster the discrete data points to find some centroids that best reflect the characteristics of workers. The time complexity of clustering is related to the number of discrete data points, $O(n)$. At last, we use the least square method to estimate the parameters of each worker quality distribution model.

6 Experiment

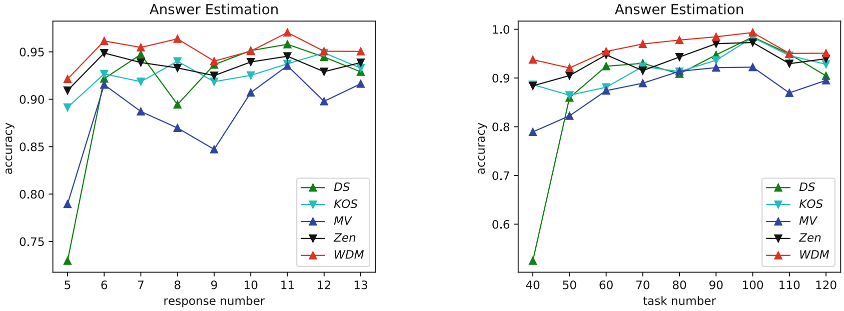
In the experimental part, we evaluate our method (WDM) with a set of real-world data and synthetic data, and we compare our method with other algorithms (MV, DS [12], KOS [33], Zen [32]) that are also estimating the true answer of tasks and worker quality.

6.1 Synthetic Data Experiments

In the synthetic experiment, we use a set of data generated by the model itself to explore the performance of our method. In this case, we can know the real values of the parameters of the worker model and the true answers of the tasks. We compare our method with other four algorithms on accuracy of task true answer estimation. In addition, we also evaluate the similarity between estimated and actual parameters of the worker quality model.

Data Generation. In the process of generating synthetic data, we first generate a set of ture answers to tasks. Given a fixed probability value u, we assign a real answer to each task. The probability of task answer 1 is u, and vice versa, the probability is $(1 - u)$. For each task, we set a difficulty value between 0–1. Then, we generate a different worker quality curve for each worker, i. e. setting the parameters of each worker model, the constraints generated are that most workers (more than 90%) are better than random ones. Then, we generate the corresponding workers'responses to the tasks according to the quality of these workers and the difficulty of the task. In the synthetic data, we set the total number of tasks $n = 1000$ and total number of workers $m = 100$, then vary the number of responses received by each task and the number of tasks completed by each worker.

Experimental Process and Results. We evaluate our method on estimating task true answers and parameters of worker quality model. We know that EM algorithm needs to be inputted a set of initial parameters. Here, we set the worker quality $\mu = 0$ and $\delta = 0.5$ as the initial parameters to carry out experiments. We conducted experiments with multiple combinations of data: setting 1, each task receives s response, the total number of workers $m > s$; setting 2, each worker completes h tasks, and each task receives a different number of responses from workers.

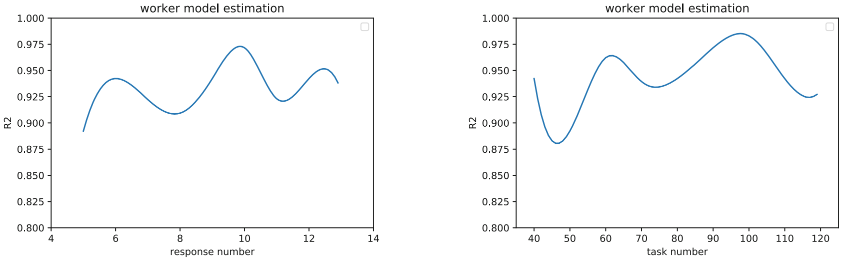


(a) setting 1.

(b) setting 2.

Fig. 3. True answer of task estimation

True Answer of Task Estimation. In Fig. 3(a), (b), we plot the accuracy of task truth estimation, and each algorithm correctly estimates the score of task answer (the higher the score, the better). Here, our strategy estimates the true answer of the task with higher accuracy than other algorithms.



(a) setting 1.

(b) setting 2.

Fig. 4. Worker quality model estimation

Worker Quality Model Estimation. Figure 4(a) and (b) show the gap between estimated and actual worker quality. We calculated the coefficient of determination R^2(the higher the score, the better) between the estimated worker quality model and the actual worker quality. We observe that the similarity between our estimated worker model and the real worker model is generally high, and with the increase of data, the accuracy of worker quality estimation is on the rise.

Summary. In the synthetic data experiments, our method is superior to other algorithms on estimation for task true answer. In addition, our method can accurately estimate the parameters of the worker quality model.

6.2 Real-World Data Experiments

In our real-world data experiments, data is collecting by publishing a large number of movie reviews as emotional analysis tasks on AMT. The workers on the platform will complete these emotional analysis tasks, that is, to judge whether the movie reviews express positive or negative emotions, and finally collect the responses of workers to the tasks. The data contains a total of 5,000 tasks, which were responsed by about 200 workers.

Experimental Process and Results. In real data experiments, we know the real answers of tasks, but we can not know the real quality of workers in real data. And we randomly extract data from real data sets in three different ways. For each data setting, we compare our method with four algorithms in the accuracy of task true answer estimation.

Data setting 1: We randomly select a certain number of workers from all workers in the data, then select all the response data of these workers, and we constantly change the number of workers selected. In the Fig. 5, we plot the accuracy of task true answer estimation (the higher the score, the better) returned by our method and four other algorithms on varying the number of worker. We observe that our method obtains more accurate estimation results than other methods.

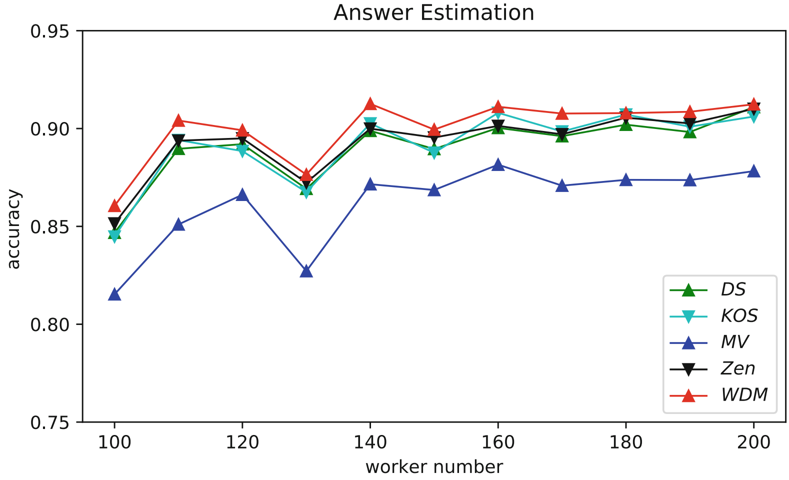


Fig. 5. Data setting 1: true answer of task estimation

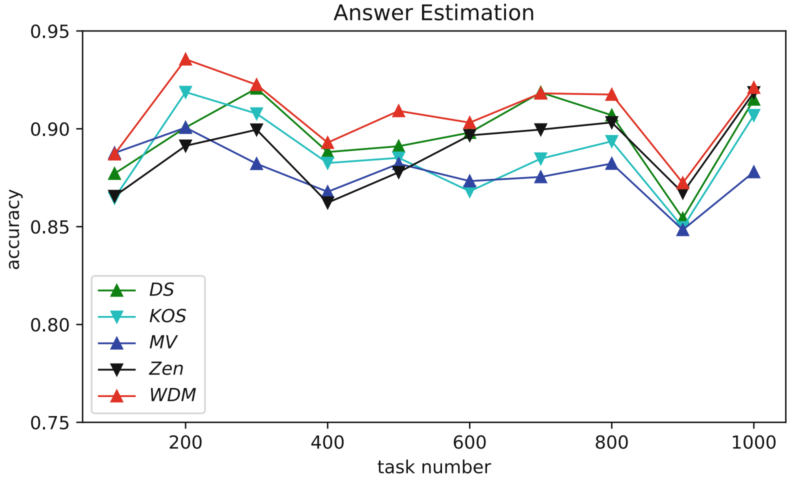


Fig. 6. Data setting 2: true answer of task estimation

Data setting 2: We randomly select a certain number of tasks from all tasks in the data, then extract all the responses received by these tasks, we constantly change the number of tasks extracted. The Fig. 6 shows the accuracy of task truth estimation returned by our method and four other algorithms on varying the number of task. Here, again, our method returns more accurate estimation results in the case of different numbers of tasks.

Data setting 3: We randomly sample a certain number of labels from all data, then we constantly change the number of labels extracted. The Fig. 7 shows the results of comparing our method with four other answer estimation algorithms on

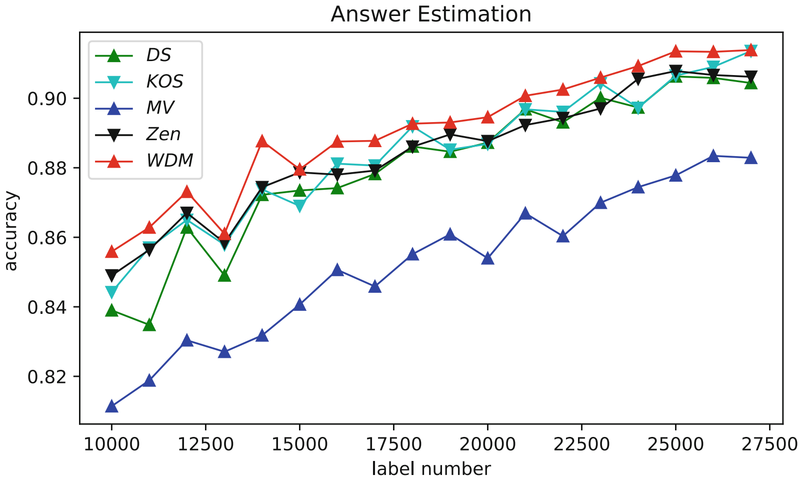


Fig. 7. Data setting 3: true answer of task estimation

varying the number of label. We observe that our method obtain more accurate estimation results than other methods.

Summary. From the results of experiments, we observe that our method always returns more accurate results on the three data setting. And the richer the worker's response data is, the better the estimation result of our method is. Although we can not know the real quality of workers, we believe that our method with worker quality distribution model can estimate the characteristics of workers more accurately.

7 Conclusion

In crowdsourcing, quality management is an important problem because of the uneven ability of workers on internet. The quality of workers can influence the results of crowdsourcing. And the quality of workers will be affected by the difficulty of the task. The more difficult the task is, the lower probability that workers can correctly answer the task. We propose a dynamic difficulty-sensitive worker quality distribution model to improve the quality of results from crowdsourcing. Our model more accurately describes the relationship between worker reliability and task difficulty. We conduct extensive experiments with synthetic data and real-world data. The results show that our method has higher accuracy of task answer estimation than other algorithms. Moreover, our worker quality distribution model can not only describe the characteristics of workers more accurately, but also predict the reliability of workers in tasks they have not done. In the future work, our worker quality distribution model can be applied to other studies of crowdsourcing, such as task allocation. We can assign tasks to the suitable workers based on our worker quality distribution model as the task states change dynamically.

Acknowledgements. The research work was supported by the National Key R&D Program ($No.\,2017YFB1400100$), the Innovation Method Fund of China ($No.\,2018IM020200$), the SDNFSC ($No.\,ZR2017ZB0420, No.\,ZR2018MF014$) and the Science and Technology Development Plan Project of Shandong Province ($No.\,2018YFJH0506$).

References

1. https://www.mturk.com/
2. http://www.crowdflower.com
3. https://www.upwork.com
4. Heymann, P., Garcia-Molina, H.: Turkalytics: analytics for human computation. In: International Conference on World Wide Web DBLP (2011)
5. Wang, H., Guo, S., Cao, J., et al.: MeLoDy: a long-term dynamic quality-aware incentive mechanism for crowdsourcing. IEEE Trans. Parallel Distrib. Syst. **PP**(99), 1 (2018)
6. Hu, H., Zheng, Y., Bao, Z., et al.: Crowdsourced POI labelling: location-aware result inference and task assignment. In: IEEE International Conference on Data Engineering, pp. 61–72. IEEE (2016)
7. Ma, F., Li, Y., Li, Q., et al.: FaitCrowd: fine grained truth discovery for crowdsourced data aggregation. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 745–754. ACM (2015)
8. Liu, X., Lu, M., Ooi, B.C., Shen, Y., Wu, S., Zhang, M.: CDAS: a crowdsourcing data analytics system. PVLDB **5**(10), 1040–1051 (2012)
9. Bo, P., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, pp. 115–124 (2005)
10. Cao, C.C., She, J., Tong, Y., et al.: Whom to ask?: jury selection for decision making tasks on micro-blog services. Proc. VLDB Endowment **5**(11), 1495–1506 (2012)
11. Dalvi, N.N., Dasgupta, A., Kumar, R., et al.: Aggregating crowdsourced binary ratings, pp. 285–294 (2013)
12. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. J. Roy. Stat. Soc. **28**(1), 20–28 (1979)
13. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. **39**(1), 1–38 (1977)
14. Feng, J., Feng, J., Feng, J., et al.: QASCA: a quality-aware task assignment system for crowdsourcing applications. In: ACM SIGMOD International Conference on Management of Data, pp. 1031–1046. ACM (2015)
15. Guo, S., Parameswaran, A., Garcia-Molina, H.: So who won?: dynamic max discovery with the crowd. In: ACM SIGMOD International Conference on Management of Data, pp. 385–396. ACM (2012)
16. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on Amazon mechanical turk. In: ACM SIGKDD Workshop on Human Computation, pp. 64–67. ACM (2010)
17. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: International Conference on Neural Information Processing Systems, pp. 1953–1961. Curran Associates Inc. (2011)
18. Karger, D.R., Oh, S., Shah, D.: Efficient crowdsourcing for multi-class labeling. ACM Sigmetrics Perform. Eval. Rev. **41**(1), 81–92 (2013)

19. Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., et al.: Limits on the majority vote accuracy in classifier fusion. Pattern Anal. Appl. **6**(1), 22–31 (2003)
20. Liu, X., Lu, M., Ooi, B.C., et al.: CDAS: a crowdsourcing data analytics system. Proc. VLDB Endowment **5**(10), 1040–1051 (2012)
21. Marcus, A., Wu, E., Karger, D., et al.: Human-powered sorts and joins. Proc. VLDB Endowment **5**(1), 13–24 (2011)
22. Kurve, A., Miller, D.J., Kesidis, G.: Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. IEEE Trans. Knowl. Data Eng. **27**(3), 794–809 (2015)
23. Parameswaran, A.G., Garciamolina, H., Park, H., et al.: CrowdScreen: algorithms for filtering data with humans, pp. 361–372 (2012)
24. Raykar, V.C., Yu, S., Zhao, L.H., et al.: Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, pp. 889–896. DBLP, June 2009
25. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: Proceedings ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 269–278 (2002)
26. Das Sarma, A., Parameswaran, A., Widom, J.: Towards globally optimal crowdsourcing quality management: the uniform worker setting, pp. 47–62 (2016)
27. Smyth, P., Fayyad, U., Burl, M., et al.: Inferring ground truth from subjective labelling of venus images. In: International Conference on Neural Information Processing Systems, pp. 1085–1092. MIT Press (1994)
28. Wang, J., Kraska, T., Franklin, M.J., et al.: CrowdER: crowdsourcing entity resolution. Proc. VLDB Endowment **5**(11), 1483–1494 (2012)
29. Wang, J., Li, G., Kraska, T., et al.: Leveraging transitive relations for crowdsourced joins. In: ACM SIGMOD International Conference on Management of Data, pp. 229–240. ACM (2013)
30. Whitehill, J., Ruvolo, P., Wu, T., et al.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: International Conference on Neural Information Processing Systems, pp. 2035–2043. Curran Associates Inc. (2009)
31. Zhang, Y., Chen, X., Zhou, D., et al.: Spectral methods meet EM: a provably optimal algorithm for crowdsourcing. Adv. Neural Inf. Process. Syst. **2**, 1260–1268 (2014)
32. Demartini, G., Difallah, D.E., CudréMauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: International Conference on World Wide Web. ACM (2012)
33. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowd-sourcing systems. In: NIPS, pp. 1953–1961 (2011)