Hierarchical Crowdsourcing for Data Labeling with Heterogeneous Crowd

Haodi Zhang[1,4], Wenxi Huang[1], Zhenhan Su[1], Junyang Chen[1], Di Jiang[2,3], Lixin Fan[3], Chen Zhang[4]
Defu Lian[5] and Kaishun Wu[1]
[1]College of Computer Science and Software Engineering, Shenzhen University
[2]WeBank Institute of Financial Technology, Shenzhen University
[3]Webank Co., Ltd.
[4]Department of Computing, Hong Kong Polytechnic University
[5]School of Data Science, University of Science and Technology of China

2023 IEEE 39th International Conference on Data Engineering (ICDE) | 979-8-3503-2227-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDE55515.2023.00099

***Abstract*—With the rapid and continuous development of data-driven technologies such as supervised learning, high-quality labeled data sets are commonly required by many applications. Due to the easiness of crowdsourcing small tasks with low cost, a straightforward solution for label quality improvement is to collect multiple labels from a crowd, and then aggregate the answers. The aggregation strategies include majority voting and its many variants, EM-based approaches, Graph Neural Nets and so on. However, due to the uncertainty information loss and commonly existing task correlations, the aggregated labels usually contain errors and may damnify the downstream model training.**

To address the above problem, we propose a hierarchical crowdsourcing framework[1] for data labeling with noisy answers about correlated data. We make use of the heterogeneity of the labeling crowd and form an initialization-checking-update loop to improve the quality of labeled data. We formalize and successfully solve the core optimization problem, namely, selecting a proper set of checking tasks for each round. We prove that maximizing the expected quality improvement is equivalent to minimizing the *conditional entropy* of the *observations* given the *crowdsourced answer families* for the selected task set, which is NP-hard to solve. Therefore, we design an efficient approximation algorithm and conduct a series of experiments on real data. The experimental results show that the proposed method effectively improves the quality of the labeled data sets as well as the SOTA performance, yet without extra human labor costs.

***Index Terms*—hierarchical crowdsourcing, heterogeneous crowd, label aggregation, truth inference**

I. Introduction

Data-driven technologies such as supervised learning models [1]–[3] have been commonly used in many applications and have shown impressive achievements. Consequentially, high-quality labeled data sets are commonly required for training the models. Crowdsourcing is often used to generate such labeled training data sets. Due to the development of crowdsourcing techniques and platforms, crowdsourcing small tasks with low cost has become easy now. However, in real-world applications, the crowdsourced labeled data sets are usually noisy, thus insufficient for training good models [4].

A straightforward solution is to collect multiple labels from a crowd and then aggregate the answers [4]–[10]. Usually, a set of labeling tasks are given to a heterogeneous crowd of workers. For each task, the answers from the crowd are collected to finalize the label with some strategies. The aggregation strategies include majority voting [11] and its many variants [12], EM-based approaches [13], (Markov CMC) sampling Graph neural nets [14] and so on. The rationality of label aggregation is that the chance that a crowd makes a mistake is relatively lower compared with a single person if each person in the crowd has an acceptable error rate (for instance, lower than 0.5) and gives their answers independently. For example, there are three crowdsourcing workers with the same error rate e. With majority voting strategy, the aggregated label has an error rate $3e^2(1-e)+e^3 < e$ when $e < 0.5$. A number of algorithms have been proposed to collect and aggregate the labels from the crowd. Majority voting is a straightforward and simple strategy, but the uncertainty information of the labels is lost after aggregation. Many variants of majority voting have been proposed, such as MV-Freq, MV-Beta, Paired-MV [15], etc. In [16] a label-cleaning advisor is proposed to provide potentially valuable advice for data scientists when they need to train or test a model with noisy labels.

Another strategy is to sort the data entities by noise rates, and choose the one with the largest noise rate for cleaning. Although the method can efficiently reduce the uncertainty of the data labels, it ignores correlations among the data entities, which commonly exist in real-world data sets. ActiveClean [17] estimates how the cleaning in each instance will change the model based on the ground truth labels predicted for each instance. Then, it selects the instance with the largest change. However, ActiveClean does not leverage the current belief state of the noisy data to predict the ground-truth labels. Besides, it uses stochastic gradient descent to update the model after cleaning each batch of instances, which may not be stable in performance, especially when the noise comes from a small part of the data. Most of the above approaches ignore or at least can not well capture the task correlations, resulting that the aggregated labels usually contain errors and may damnify the following model training.

In this paper, we propose to make use of the heterogeneity of the crowd and build a hierarchical crowdsourcing framework. A crowd of workers is divided into two parts, namely, the

[1]The code and data of this work are publicly available at https://github.com/ResearchGroupHdZhang/HC-code

2375-026X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDE55515.2023.00099

labeling workers and the checking workers, according to a given accuracy threshold. The workers in the latter part have relatively higher accuracy rates, in other words, are experts in labeling. The main idea is to initialize the belief state of the uncertain labels with the crowdsourced answers from labeling workers, and then let the checking workers repeatedly check some labels to improve the data quality. Note that, compared with the label aggregation approaches, the proposed framework requires at least the same human labor cost, or even lower, depending on the budget for label checking. The idea of labor division according to expertise level in a heterogeneous crowd is actually commonly used in data labeling. For example, the X-ray images for training in [18] are labeled by many ordinary crowdsourcing doctors, with uncertain information. An expert group that consists of 5 radiologists decides which are the ground truths. To our knowledge, we are the first to theoretically solve the optimal checking task selection for data labeling with a heterogeneous crowd.

We summarize our contributions in the following aspects.

- First, we propose a hierarchical crowdsourcing framework to improve the quality of labeled data, utilizing noisy answers from a group of heterogeneous workers.
- Second, we prove that the core optimization problem, namely, selecting the optimal set of label checking tasks, is equivalent to minimizing the *conditional entropy* of the observations given the crowdsourced answer families of the selected set. Solving the optimization problem is proved NP-hard.
- Third, an efficient approximation algorithm is proposed. We conduct a series of experiments on real data set to demonstrate the effectiveness of the proposal. The proposed method successfully improves the SOTA performance.

II. Preliminaries

We formally define the data model, and then present the basic crowdsourcing model, and finally state the core problem.

A. Data Model

Let C be a crowd of workers for data labeling. A basic assumption is that each worker in the crowd is not perfect, namely, has the probability of giving wrong answers. In some previous work, the crowdsourced answers are assumed to always be correct [19], [20], which is actually not realistic. We adopt a more general and widely used error model [21]–[23], which ensures that the answer returned by each crowd worker has an acceptable confidence, namely, not less than $1/2$. The confidence of a crowdsourced answer is the same as the accuracy rate of the worker cr who gives the answer, denoted by Pr_{cr}. The accuracy rates of each worker $cr \in C$ can be easily estimated with a set of sample tasks with ground truth. The entire crowd is then divided into two parts according to a threshold of accuracy rate θ.

Definition 1. *For a crowd C and a threshold θ, we call those workers with accuracy rate no less than θ* expert workers*, denoted by CE, and the rest* preliminary workers*, denoted by CP, namely,*

$$\begin{aligned} CE &= \{cr | cr \in C \wedge Pr_{cr} \geq \theta\} \\ CP &= C - CE \end{aligned} \tag{1}$$

TABLE I
Example of observations

	f_1	f_2	f_3	$P(o)$
o_1	*false*	*false*	*false*	0.09
o_2	*true*	*false*	*false*	0.11
o_3	*false*	*true*	*false*	0.10
o_4	*true*	*true*	*false*	0.20
o_5	*false*	*false*	*true*	0.08
o_6	*true*	*false*	*true*	0.09
o_7	*false*	*true*	*true*	0.15
o_8	*true*	*true*	*true*	0.18

A human intelligence *task* for a preliminary worker is to label a data instance, and for an expert worker, the task is to check whether the labeled data is correct. To unify our data model, we formalize both the labeling task and checking task into a Yes-or-No query: "Is data instance e should be labeled as l?". In other words, both the preliminary and expert workers are asked to determine the truth value of a fact that the given instance e should be labeled as l. The crowdsourced answer is either *Yes* or *No*. If the original labeling task is a multi-label classification with m labels, each labeling task can be divided into m queries about m binary facts, as was done in [23], [24]. Of course, the facts are correlated. Some works model the output of label checking by the conditional probability of the relabeling result given the previous label. But in this work, we focus on how to make use of existing labels to improve the data quality, without extra effort or change on the crowdsourcing mechanism. Therefore, in this paper, we assume that each crowdsourcing worker gives answers independently.

For n binary labeling tasks, there are in total 2^n possible truth-value interpretations. Note that the n tasks are usually correlated, and the correlations among them can be represented by their joint distributions. These interpretations are mutually exclusive in a deterministic world, and each of them presents a possible state of the data, namely, the ground truth. In the rest of the paper, we call such an interpretation an *observation*, and denote the set of all observations by O, and the ground truth by $gt(O)$. For a fact set $\mathcal{F} = \{f_1, f_2, f_3\}$, Table I shows the observations of the fact set.

For an observation $o \in O$, the probability that observation o is the ground truth is denoted by $P(gt(O) = o)$, indicating the chance that o is the real state of the data. In the following, we write the probability $P(gt(O) = o)$ as $P(o)$ for abbreviation. If a fact f is interpreted to be true in an observation o, we say o is a positive model of f, denoted by $o \models f$. Otherwise, o is a negative model for f, $o \not\models f$. An observation o assigns truth values for all facts in $\mathcal{F}$, so $o \not\models f$ is equivalent to $o \models \neg f$ for any $f \in \mathcal{F}$. Moreover, the following assertions hold for

each $f \in \mathcal{F}$.

$$P(f) = \sum_{o \in O \wedge o \models f} P(o)$$
$$P(\neg f) = 1 - \sum_{o \in O \wedge o \models f} P(o) = \sum_{o \in O \wedge o \not\models f} P(o) \quad (2)$$

But the following equation is not necessarily true for $o \in O$,

$$P(o) = \prod_{f_i \in \mathcal{F}} P(l_i) \quad (3)$$

where

$$l_i = \begin{cases} f_i & \text{if } o \models f_i, \\ \neg f_i & \text{otherwise.} \end{cases}$$

Equation (3) above is not necessarily true, since the facts in $\mathcal{F}$ are not necessarily independent of each other. For the example in Table I,

$$P(f_1) = P(o_2) + P(o_4) + P(o_6) + P(o_8) = 0.58$$
$$P(f_2) = P(o_3) + P(o_4) + P(o_7) + P(o_8) = 0.63 \quad (4)$$
$$P(f_3) = P(o_5) + P(o_6) + P(o_7) + P(o_8) = 0.50$$

But $\prod_{i=1}^{3} P(\neg f_i) = 0.78 \neq P(o_1)$.

Intuitively, a distribution on the observation set O can be viewed as a belief state of the data. Our framework aims to make use of the crowdsourced answers to update the belief state, and improve the quality of the labeled data. We use *Shannon Entropy* to measure the quality of a given set of facts.

Definition 2 (Data quality)**.** *Given a set $\mathcal{F}$ and its corresponding observation set O, the estimation of quality of $\mathcal{F}$, denoted by $Q(\mathcal{F})$ as a quality function, is the sum value of Shannon Entropy,*

$$Q(\mathcal{F}) = -H(O) = \sum_{o \in O} P(o) \log P(o),$$

where $\sum_{o \in O} P(o) = 1$.

The above Q value is used as a quality function to estimate the quality of an uncertain label data set.

B. Basic Crowdsourcing Model

Each crowdsourcing worker is required to answer the given query independently for relatively high crowd reliability. By a query, we mean the task to judge whether a fact f is true or not. In the rest of the paper, we call a fact f a query if f is selected to give the workers. Each worker usually receives multiple queries to answer, instead of one by one. We assume that each worker's accuracy rate is no less than 1/2, otherwise, the collected answer from the worker is useless for the quality improvement of the data.

Informally, we need to select a proper subset of facts $\mathcal{T} \subseteq \mathcal{F}$ as queries, and assign them to the crowdsourcing workers, and finally collect their answers for each query. We call the set of answers for all the queries in $\mathcal{T}$ from a single worker the *crowdsourced answer set*,

Definition 3 (Crowdsourced answer set)**.** *For a fact set $\mathcal{F} = \{f_1, \ldots, f_n\}$ and a query set $\mathcal{T} \subseteq \mathcal{F}$, the* crowdsourced answer set *from a worker $cr \in C$ is a set $A_{cr}^{\mathcal{T}} = \{a_{f_i} | f_i \in \mathcal{T}\}$, where the value for a_{f_i} is either* true *or* false *for $f_i \in \mathcal{T}$.*

We call the set of the answer sets from all workers in C, denoted by $A_C^{\mathcal{T}}$, the *crowdsourced answer family*, $A_C^{\mathcal{T}} = \{A_{cr}^{\mathcal{T}} | cr \in C\}$. For convenience, for a fact $f \in \mathcal{T}$, the crowdsourced answer for f from worker cr is denoted by $A_{cr}^{\mathcal{T}}(f)$ (either *Yes* or *No*, indicating respectively that f is *true* or *false*), and $A_C^{\mathcal{T}}(f) = \{A_{cr}^{\mathcal{T}}(f) | cr \in C\}$. In label aggregation, for a fact $f \in \mathcal{F}$, the label is usually finalized by majority voting,

$$label(f) = \begin{cases} true & \frac{|\{a=\text{'Yes'} | a \in A_{CE}^{\mathcal{T}}(f)\}|}{|A_{CE}^{\mathcal{T}}(f)|} \geq 1/2 \\ false & otherwise \end{cases} \quad (5)$$

However, the majority labeling policy ignores the uncertainty information in the crowdsourced answers.

Before we formally introduce our hierarchical crowdsourcing framework, we consider the probability of the crowdsourced answer sets.

Lemma 1 (Computation of answer set probability)**.** *For a fact set $\mathcal{F}$ a query set $\mathcal{T} \subseteq \mathcal{F}$, a crowdsourcing worker cr, and an observation o, the conditional probability of receiving $A_{cr}^{\mathcal{T}}$ from worker cr given o is*

$$P(A_{cr}^{\mathcal{T}} | o) = \prod_{f \in T^+(o, A_{cr}^{\mathcal{T}})} Pr_{cr} \cdot \prod_{f \in T^-(o, A_{cr}^{\mathcal{T}})} (1 - Pr_{cr})$$
$$= Pr_{cr}^{|T^+(o, A_{cr}^{\mathcal{T}})|} \cdot (1 - Pr_{cr})^{|T^-(o, A_{cr}^{\mathcal{T}})|} \quad (6)$$

where $T^+(o, A_{cr}^{\mathcal{T}})$ and $T^-(o, A_{cr}^{\mathcal{T}})$ are the consistent set *and* inconsistent set *of o and $A_{cr}^{\mathcal{T}}$, respectively,*

$$T^+(o, A_{cr}^{\mathcal{T}}) = \{f | f \in \mathcal{T} \wedge (o \models f \Leftrightarrow A_{cr}^{\mathcal{T}} \models f)\}$$
$$T^-(o, A_{cr}^{\mathcal{T}}) = \{f | f \in \mathcal{T} \wedge (o \models f \Leftrightarrow A_{cr}^{\mathcal{T}} \models \neg f)\} \quad (7)$$

and the probability of $A_{cr}^{\mathcal{T}}$ is

$$P(A_{cr}^{\mathcal{T}}) = \sum_{o \in O} \big(P(o) \cdot P(A_{cr}^{\mathcal{T}} | o)\big)$$
$$= \sum_{o \in O} \big(P(o) \cdot \prod_{f \in T^+(o, A_{cr}^{\mathcal{T}})} Pr_{cr} \cdot \prod_{f \in T^-(o, A_{cr}^{\mathcal{T}})} (1 - Pr_{cr})\big)$$
$$= \sum_{o \in O} P(o) \cdot Pr_{cr}^{|T^+(o, A_{cr}^{\mathcal{T}})|} \cdot (1 - Pr_{cr})^{|T^-(o, A_{cr}^{\mathcal{T}})|} \quad (8)$$

The above sets $T^+(o, A_{cr}^{\mathcal{T}})$ and $T^-(o, A_{cr}^{\mathcal{T}})$ respectively give the consistent and inconsistent information between observation o and the crowdsourced answer set $A_{cr}^{\mathcal{T}}$. For each fact $f \in \mathcal{T}$, if its truth value in o is consistent with the corresponding answer a_f in $A_{cr}^{\mathcal{T}}$, the fact f is in $T^+(o, A_{cr}^{\mathcal{T}})$. Otherwise, if they are not consistent, for instance, $o \models f$ and $a_f = False$, the fact f is in $T^-(o, A_{cr}^{\mathcal{T}})$. Note that for the facts that are not in $\mathcal{T}$, $A_{cr}^{\mathcal{T}}$ does not give any information about them. Hence, they are not in either $T^+(o, A_{cr}^{\mathcal{T}})$ or $T^-(o, A_{cr}^{\mathcal{T}})$. For an observation o and a crowdsourced answer set $A_{cr}^{\mathcal{T}}$, we have,

$$T^+(o, A_{cr}^{\mathcal{T}}) \cap T^-(o, A_{cr}^{\mathcal{T}}) = \emptyset$$
$$T^+(o, A_{cr}^{\mathcal{T}}) \cup T^-(o, A_{cr}^{\mathcal{T}}) = \mathcal{T} \quad (9)$$

The above Lemma 1 provides the computation of the probability that some answer set $A_{cr}^{\mathcal{T}}$ is crowdsourced from worker cr. In particular, for a single fact $f \in \mathcal{T}$ and a single worker cr,

$$P(A_{cr}^{\mathcal{T}}(f) = \text{'Yes'}) = \begin{cases} \sum_{o \in O} P(o) Pr_{cr} = Pr_{cr} & \text{if } o \models f_i \\ \sum_{o \in O} P(o)(1 - Pr_{cr}) = 1 - Pr_{cr} & \text{if } o \models \neg f_i \end{cases} \tag{10}$$

With the probability of the crowdsourced answer set from a single worker, we can calculate the probability of a given answer family, since the workers answer the queries independently.

Lemma 2 (Computation of answer family probability)**.** *For a fact set $\mathcal{F}$, a query set $\mathcal{T}$ and a crowd C, the probability of crowdsourced answer family is*

$$\begin{aligned} P(A_C^{\mathcal{T}}) &= \sum_{o \in O} \big(P(o) \prod_{cr \in C} P(A_{cr}^{\mathcal{T}} | o)\big) \\ &= \sum_{o \in O} P(o) \prod_{cr \in C} \big(\prod_{f \in T^+(o, A_{cr}^{\mathcal{T}})} Pr_{cr} \prod_{f \in T^-(o, A_{cr}^{\mathcal{T}})} (1 - Pr_{cr})\big) \\ &= \sum_{o \in O} \big(P(o) \prod_{cr \in C} Pr_{cr}^{|T^+(o, A_{cr}^{\mathcal{T}})|} (1 - Pr_{cr})^{|T^-(o, A_{cr}^{\mathcal{T}})|}\big) \end{aligned} \tag{11}$$

The answers from the workers are used to update the distribution and improve the data quality. Note that for a fact $f \notin \mathcal{T}$, there is no answer for it, as it is not sent to the crowd as a query. For an answer set $A_{cr}^{\mathcal{T}}$ and a fact $f \in \mathcal{T}$, we denote $A_{cr}^{\mathcal{T}} \models f$ if a_f in $A_{cr}^{\mathcal{T}}$ is 'Yes', and $A_{cr}^{\mathcal{T}} \models \neg f$ if a_f in $A_{cr}^{\mathcal{T}}$ is 'No'. Note that different from an observation, an answer set is not a complete assignment over $\mathcal{F}$. Therefore, $A_{cr}^{\mathcal{T}} \not\models f$ does not implies $A_{cr}^{\mathcal{T}} \models \neg f$.

For a query set $\mathcal{T}$, we denote the set of all possible answer families from a crowd C by $\mathcal{AS}_C^{\mathcal{T}}$.

Definition 4 (Entropy of the answer families)**.** *For a fact set $\mathcal{F}$, a query set $\mathcal{T}$ and a set of crowdsourcing workers C, the entropy of the answer families $H(\mathcal{AS}_C^{\mathcal{T}})$ is*

$$H(\mathcal{AS}_C^{\mathcal{T}}) = -\sum_{A_C^{\mathcal{T}} \in \mathcal{AS}_C^{\mathcal{T}}} P(A_C^{\mathcal{T}}) \log P(A_C^{\mathcal{T}}) \tag{12}$$

The probability that the crowdsourced answer a_f is consistent with the observation o is the same as the accuracy rate of the worker cr who gives the answer a_f, namely, Pr_{cr}, and similarly, the inconsistent probability is $1 - Pr_{cr}$.

As stated above, we divide the crowd C into two parts, namely, the expert worker set CE and the preliminary worker set CP. The data is first labeled by workers in CP. Then we make use of the expertise from workers in CE and incrementally check the labeled data set. In each round, if there is still checking budget left, a proper set of queries is selected from the labeled data and sent to CE for checking. After sending the selected query set $\mathcal{T}$ to CE, the answer set for $\mathcal{T}$ is a random variable on $\{true, false\}^{\mathcal{T}}$. Before crowdsourcing the answer set from the workers, we can only calculate the expected quality of the data.

TABLE II
SUMMARY OF NOTATIONS

Notation	Meaning
f	fact
$P(f)$	probability that fact f is true
$\mathcal{F}$	fact set (data)
o	observation
$P(o)$	probability of observation o is the ground truth
$o \models f/\neg f$	fact f is *true* / *false* in observation o
$\mathcal{T}$	query (task) set
cr	crowdsourcing worker
Pr_{cr}	accuracy rate of worker cr
C	crowd of workers
CE	expert workers
CP	preliminary workers
$A_{cr}^{\mathcal{T}}$	crowdsourced answer set of $\mathcal{T}$ from worker cr
$A_C^{\mathcal{T}}$	crowdsourced answer family of $\mathcal{T}$ from all workers in crowd C
$\mathcal{AS}_C^{\mathcal{T}}$	set of all possible answer families of $\mathcal{T}$ from crowd C
$T^+(o, A^{\mathcal{T}})$	consistent set of o and $A^{\mathcal{T}}$
$T^-(o, A^{\mathcal{T}})$	inconsistent set of o and $A^{\mathcal{T}}$
$Q(\mathcal{F})$	quality of data $\mathcal{F}$
$H(O)$	observation entropy
$H(\mathcal{AS}_C^{\mathcal{T}})$	answer family entropy of query set $\mathcal{T}$
$\mathbb{Q}(\mathcal{F}\|\mathcal{T})$	expected quality of the data with query set $\mathcal{T}$
$\Delta\mathbb{Q}(\mathcal{F}\|\mathcal{T})$	expected quality improvement of the data by query set $\mathcal{T}$

Definition 5. *For a data set $\mathcal{F}$, a query set $\mathcal{T}$ and the expert crowd CE, the expected quality of the data with respect to $\mathcal{T}$ is*

$$\mathbb{Q}(\mathcal{F}|\mathcal{T}) = \sum_{A_{CE}^{\mathcal{T}} \in \mathcal{AS}_{CE}^{\mathcal{T}}} P(A_C^{\mathcal{T}}) Q(\mathcal{F}|A_{CE}^{\mathcal{T}}) \tag{13}$$

where the $Q(\mathcal{F}|A_{CE}^{\mathcal{T}})$ is the conditional data quality after receiving the crowdsourced answer family $A_{CE}^{\mathcal{T}}$ from CE with confidence.

C. Problem Statement

With the above data model and the basic crowdsourcing model, we can formally define the core problem.

Definition 6 (Checking task selection)**.** *Given a fact set $\mathcal{F}$, possible observations O with probability joint distribution and an expert crowd CE with diverse private accuracy Pr_{cr} for each $cr \in CE$, our goal is to maximize the expected quality $\mathbb{Q}(\mathcal{F}|\mathcal{T})$ by selecting a size-k query set $\mathcal{T}^*$ to ask the crowd,*

$$\mathcal{T}^* = \underset{\mathcal{T} \subseteq \mathcal{F}, |\mathcal{T}| = k}{\arg\max}\ \mathbb{Q}(\mathcal{F}|\mathcal{T}) \tag{14}$$

We utilize crowdsourcing as a powerful tool to improve the labeled data quality. The above goal is to maximize an expected value instead of a concrete value, due to the uncertainty in the crowdsourced answer set $A_{CE}^{\mathcal{T}}$.

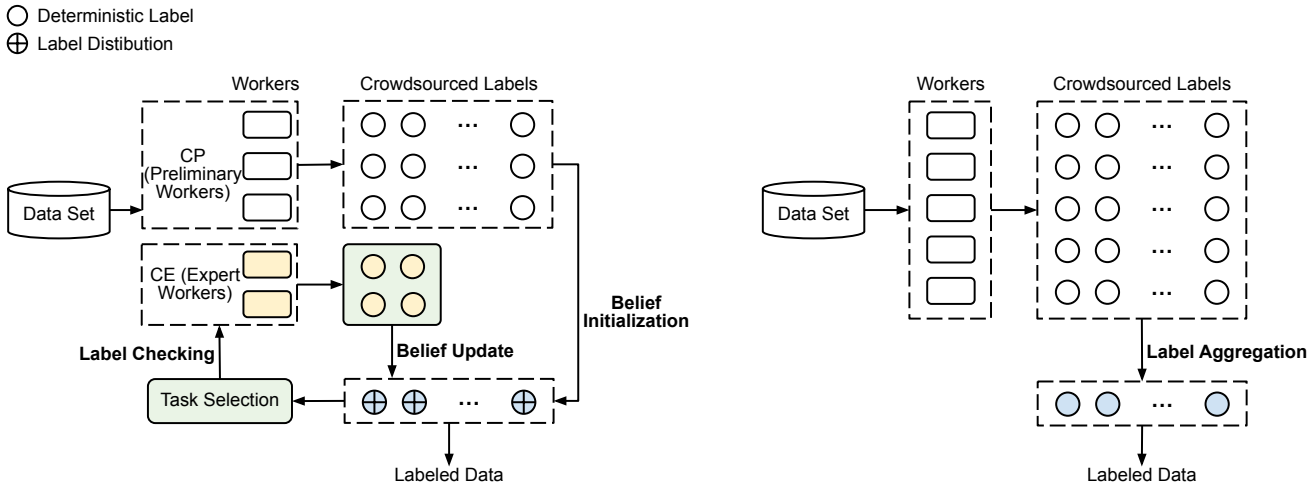


Fig. 1. Hierarchical Crowdsourcing vs. Label Aggregation

III. Hierarchical Crowdsourcing

In this section, we formally present our hierarchical crowdsourcing framework, as shown in Figure 1. As introduced above, the crowd is divided into two parts, preliminary worker set CP and expert worker set CE. The former provides the answers for the initialization of the label belief. The belief state of the labels can be initialized by the existing probability-based label aggregation method. The expert worker set provides answers for the belief update of the labels, driven by a task selector that iteratively selects proper tasks to check. Similar to the typical crowdsourcing methods, the whole process terminates when the budget is used up, and the results are finally generated. The budget in our framework refers to the crowdsourced answers from the expert workers. When the relatively high-quality answers are used up for belief update, the process terminates and returns labels according to current belief state.

In the following, we first discuss how to initialize and update the belief state of observations by crowdsourcing answers. Then we introduce checking task selection, which is the core optimization problem of the proposed method, and prove that it is NP-hard. Finally, we give the approximate solution and algorithm for the above problem.

A. Belief Initialization and Update

We first look into how the belief state of the observation is initialized by CP and updated by CE. The initialization can be done simply by aggregating the labels given by CP. As in the framework, the improvement of the label quality mainly depends on how we check the labels, we can commit to the same strategies with the baseline aggregation for belief initialization, by majority voting or weighted majority voting, or any other aggregation strategy. For instance,

$$P(o) = \prod_{f \in \mathcal{F}} ob(o, f) \tag{15}$$

where

$$ob(o,f) = \begin{cases} \frac{|\{cr | cr \in CP \wedge cr \text{ answers 'Yes' for f}\}|}{|CP|} & o \models f \\ \frac{|\{cr | cr \in CP \wedge cr \text{ answers 'No' for f}\}|}{|CP|} & o \models \neg f \end{cases} \tag{16}$$

As for label checking, due to the uncertainty embedded in the facts and crowdsourced answers, updating the label distribution according to the CE's answers can be viewed as a posterior observation probabilities conditioning on the answers, which can be calculated by Bayesian theorem.

Now we discuss how the crowdsourced answer set from CE could be used to update the belief state and improve the quality of the data. Suppose that we have selected a query set $\mathcal{T}$ for the crowd and have successfully collected the answer set $A_{cr}^{\mathcal{T}}$. For each observation $o \in O$, the probability of o now can be updated to $P(o|A_{cr}^{\mathcal{T}})$,

$$P(o|A_{cr}^{\mathcal{T}}) = \frac{P(o) \cdot P(A_{cr}^{\mathcal{T}}|o)}{P(A_{cr}^{\mathcal{T}})} \tag{17}$$

The probability $P(A_{cr}^{\mathcal{T}})$ can be calculated according to Equation (8) in Lemma 1, and

$$P(A_{cr}^{\mathcal{T}}|o) = \prod_{f \in T^+(o, A_{cr}^{\mathcal{T}})} Pr_{cr} \cdot \prod_{f \in T^-(o, A_{cr}^{\mathcal{T}})} (1 - Pr_{cr}) \tag{18}$$

We formally give the data update by the crowdsourced answers.

Lemma 3 (Belief update)**.** *For a fact set $\mathcal{F}$, a query set $\mathcal{T}$, suppose the answer set crowdsourced from an expert worker cr is $A_{cr}^{\mathcal{T}}$, the belief state of observation $o \in O$ should be updated to*

$$\begin{aligned} P(o|A_{cr}^{\mathcal{T}}) &= \frac{P(o) \cdot P(A_{cr}^{\mathcal{T}}|o)}{P(A_{cr}^{\mathcal{T}})} \\ &= \frac{P(o) \cdot \prod_{f \in T^+(o, A_{cr}^{\mathcal{T}})} Pr_{cr} \cdot \prod_{f \in T^-(o, A_{cr}^{\mathcal{T}})} (1 - Pr_{cr})}{\sum_{o' \in O} \left(P(o') \cdot \prod_{f \in T^+(o', A_{cr}^{\mathcal{T}})} Pr_{cr} \cdot \prod_{f \in T^-(o', A_{cr}^{\mathcal{T}})} (1 - Pr_{cr}) \right)} \end{aligned} \tag{19}$$

Note that when we finish checking the label and updating the belief, we can either return the observation distribution, or simply update the discrete labels according to the distribution,

$$label(f) = \begin{cases} true & o^* \models f \\ false & o^* \models \neg f \end{cases} \tag{20}$$

where $o^* = \arg\max_{o \in O} P(o)$.

To sum up, we have already described 1) how to initialize the belief state of the observations before label checking, and 2) how to update the belief state after each round of label checking. Now we introduce the core problem, namely, how to select an optimal query set for label checking.

B. Optimal Selection of Checking Tasks

In this sub-section, We formally introduce the optimal task selection for label checking.

Definition 7 (Expected quality improvement)**.** *Given a query set $\mathcal{T}$, the quality improvement of the data after updating is:*

$$\Delta\mathbb{Q}(\mathcal{F}|\mathcal{T}) = \mathbb{Q}(\mathcal{F}|\mathcal{T}) - Q(\mathcal{F})$$

where $Q(\mathcal{F})$ is the quality of the data, and $\mathbb{Q}(\mathcal{F}|\mathcal{T})$ is the expected quality of the data after being updated by the answer set $A_C^{\mathcal{T}}$.

Lemma 4 (Optimization objective)**.** *For fact set $\mathcal{F}$, the optimization objective of k-selection in definition 6*

$$\mathcal{T}^* = \underset{\mathcal{T}\subseteq\mathcal{F},|\mathcal{T}|=k}{\arg\max}\ \mathbb{Q}(\mathcal{F}|\mathcal{T}) = \underset{\mathcal{T}\subseteq\mathcal{F},|\mathcal{T}|=k}{\arg\max}\ \Delta\mathbb{Q}(\mathcal{F}|\mathcal{T}) \tag{21}$$

Proof. For a given fact set $\mathcal{F}$, $Q(\mathcal{F})$ is constant. □

With the crowdsourced answer sets $A_{CE}^{\mathcal{T}}$, we can directly update the distribution of the observations $P(o)$ for $o \in O$ with $P(o|A_{CE}^{\mathcal{T}})$,

$$P(o|A_{CE}^{\mathcal{T}}) = \frac{P(A_{CE}^{\mathcal{T}}|o)\cdot P(o)}{P(A_{CE}^{\mathcal{T}})} \tag{22}$$

Note that for two expert workers cr_1 and cr_2 in CE, given an observation o, the two workers give their answer sets independently. Thus,

$$P(o|A_{CE}^{\mathcal{T}}) = \frac{P(A_{CE}^{\mathcal{T}}|o)\cdot P(o)}{P(A_{CE}^{\mathcal{T}})} = \frac{P(o)\cdot\prod_{cr\in CE} P(A_{cr}^{\mathcal{T}}|o)}{\sum_{o'\in O} P(o')\cdot\prod_{cr\in CE} P(A_{cr}^{\mathcal{T}}|o')} \tag{23}$$

In Lemma 4 above, the expected quality

$$\begin{aligned}\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) &= \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}} P(A_{CE}^{\mathcal{T}})Q(\mathcal{F}\mid A_{CE}^{\mathcal{T}}) \\ &= \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}} P(A_{CE}^{\mathcal{T}})\sum_{o\in O} P(o\mid A_{CE}^{\mathcal{T}})\log P(o\mid A_{CE}^{\mathcal{T}})\end{aligned} \tag{24}$$

where $P(o \mid A_{CE}^{\mathcal{T}})$ is computed by equation (23), and the probability $P(A_{CE}^{\mathcal{T}})$ is calculated by Equation (11) in Lemma 2.

Now we introduce the main theoretical result of our work.

Theorem 1 (**Expected quality improvement).** *For fact set $\mathcal{F}$, its observation set O and expert crowd CE, the expected quality improvement by selecting $\mathcal{T}$ as query set from $\mathcal{F}$ is,*

$$\Delta\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) = H(O) - H(O|\mathcal{AS}_{CE}^{\mathcal{T}}) \tag{25}$$

Proof. The expected quality gain by selecting $\mathcal{T}$ from $\mathcal{F}$ is

$$\begin{aligned}\Delta\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) &= \mathbb{Q}(\mathcal{F}\mid\mathcal{T}) - Q(\mathcal{F}) \\ &= \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}} P(A_{CE}^{\mathcal{T}})\sum_{o\in O} P(o\mid A_{CE}^{\mathcal{T}})\log P(o\mid A_{CE}^{\mathcal{T}}) \\ &\quad - \sum_{o\in O} P(o)\log P(o)\end{aligned} \tag{26}$$

where

$$P(o|A_{CE}^{\mathcal{T}}) = \frac{P(A_{CE}^{\mathcal{T}}|o)\cdot P(o)}{P(A_{CE}^{\mathcal{T}})} = \frac{P(o)\cdot\prod_{cr\in CE} P(A_{cr}^{\mathcal{T}}|o)}{\sum_{o'\in O} P(o')\cdot\prod_{cr\in CE} P(A_{cr}^{\mathcal{T}}|o')} \tag{27}$$

and

$$\begin{aligned}P(A_{cr}^{\mathcal{T}}) &= \sum_{o\in O}\big(P(o)\cdot P(A_{cr}^{\mathcal{T}}|o)\big) \\ &= \sum_{o\in O}\Big(P(o)\cdot\prod_{f\in T^+(o,A_{cr}^{\mathcal{T}})} Pr_{cr}\cdot\prod_{f\in T^-(o,A_{cr}^{\mathcal{T}})}(1-Pr_{cr})\Big) \\ &= \sum_{o\in O} P(o)\cdot Pr_{cr}^{|T^+(o,A_{cr}^{\mathcal{T}})|}\cdot(1-Pr_{cr})^{|T^-(o,A_{cr}^{\mathcal{T}})|}\end{aligned} \tag{28}$$

where $T^+(o, A_{cr}^{\mathcal{T}})$ and $T^-(o, A_{cr}^{\mathcal{T}})$ are the *consistent set* and *inconsistent set* of o and $A_{cr}^{\mathcal{T}}$, respectively. Therefore,

$$\begin{aligned}\Delta\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) &= \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}} P(A_{CE}^{\mathcal{T}})\sum_{o\in O} P(o\mid A_{CE}^{\mathcal{T}})\log P(o\mid A_{CE}^{\mathcal{T}}) \\ &\quad - \sum_{o\in O} P(o)\log P(o) \\ &= \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}}\sum_{o\in O} P(A_{CE}^{\mathcal{T}}|o)P(o)\log P(A_{CE}^{\mathcal{T}}|o)P(o) \\ &\quad - \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}}\log P(A_{CE}^{\mathcal{T}})\sum_{o\in O} P(A_{CE}^{\mathcal{T}}|o)P(o) - \sum_{o\in O} P(o)\log P(o)\end{aligned} \tag{29}$$

Note that $\sum_{o\in O} P(A_{CE}^{\mathcal{T}}|o)P(o)$ is exactly $P(A_{CE}^{\mathcal{T}})$, and according to Definition 4 of the answer family space entropy, we have

$$\begin{aligned}&\Delta\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) \\ &= H(\mathcal{AS}_{CE}^{\mathcal{T}}) + \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}}\sum_{o\in O} P(A_{CE}^{\mathcal{T}}|o)P(o)\log P(A_{CE}^{\mathcal{T}}|o)\end{aligned} \tag{30}$$

Note that $P(A_{CE}^{\mathcal{T}}|o)$ can be computed by Equation (6) in Lemma 1. Therefore, we have

$$\begin{aligned}&\Delta\mathbb{Q}(\mathcal{F}\mid\mathcal{T}) \\ &= H(\mathcal{AS}_{CE}^{\mathcal{T}}) + \sum_{A_{CE}^{\mathcal{T}}\in\mathcal{AS}_{CE}^{\mathcal{T}}}\sum_{o\in O} P(A_{CE}^{\mathcal{T}}|o)P(o)\log P(A_{CE}^{\mathcal{T}}|o) \\ &= H(\mathcal{AS}_{CE}^{\mathcal{T}}) - H(\mathcal{AS}_{CE}^{\mathcal{T}}|O)\end{aligned} \tag{31}$$

Algorithm 1 Hierarchical Crowdsourcing for Data Labeling

Require: Data set $\mathcal{F}$, crowd C, data labels for $\mathcal{F}$ from C, label checking budget B

1: Divide C into CE and CP according to Equation (1)
2: Initialize $P(O)$ with the labels from CP according to Equation (15)
3: **repeat**
4: Select $\mathcal{T} = \arg\min_{\mathcal{T}\subseteq\mathcal{F}} H(O|\mathcal{AS}^{\mathcal{T}}_{CE})$ according to Equation (34)
5: Send $\mathcal{T}$ to CE and collect answer family $A^{\mathcal{T}}_{CE}$
6: Update $P(O)$ with $A^{\mathcal{T}}_{CE}$ according to Equation (19)
7: Set $B \leftarrow B - |\mathcal{T}| \cdot |CE|$
8: **until** $B < |\mathcal{T}| \cdot |CE|$
9: Update the labels with $P(O)$ according to Equation (20)

Algorithm 2 Approximate Optimal Checking-Task Selection

Require: Fact set $\mathcal{F}$, current observation distribution $P(O)$, query number k per round and current checking budget B

1: Initialize $\mathcal{T} \leftarrow \emptyset$
2: **repeat**
3: Select $\hat{f} = \arg\max_{f\in\mathcal{F}}(-H(O|\mathcal{AS}^{\mathcal{T}\cup\{f\}}_{CE}) + H(O|\mathcal{AS}^{\mathcal{T}}_{CE}))$
4: **if** $-H(O|\mathcal{AS}^{\mathcal{T}}_{CE}) + H(O|\mathcal{AS}^{\mathcal{T}\cup\{f\}}_{CE}) \leq 0$ **then**
5: **break**
6: **else**
7: Set $\mathcal{T} \leftarrow \mathcal{T} \cup \{\hat{f}\}$
8: Set $\mathcal{F} \leftarrow \mathcal{F}\backslash\{f\}$
9: **end if**
10: **until** $|\mathcal{T}| = min(k, B)$
11: **return** $\mathcal{T}$

According to the chain rule of Shannon entropy, we have

$$\begin{aligned}\Delta\mathbb{Q}(\mathcal{F} \mid \mathcal{T}) &= H(\mathcal{AS}^{\mathcal{T}}_{CE}) + H(O) - H(\mathcal{AS}^{\mathcal{T}}_{CE}, O) \\ &= H(O) - \big(H(\mathcal{AS}^{\mathcal{T}}_{CE}, O) - H(\mathcal{AS}^{\mathcal{T}}_{CE})\big) \\ &= H(O) - H(O|\mathcal{AS}^{\mathcal{T}}_{CE})\end{aligned} \tag{32}$$

□

As $H(O)$ is constant for a given distribution $P(O)$, our optimization goal is to maximize $-H(O|\mathcal{AS}^{\mathcal{T}}_{CE})$, i.e. to minimize the conditional entropy $H(O|\mathcal{AS}^{\mathcal{T}}_{CE})$.

Theorem 2 (**Optimization objective**). *Given a fact set $\mathcal{F}$, its corresponding observation set O and a set of expert workers CE, our goal is to maximize the expected quality $\mathbb{Q}(\mathcal{F}|\mathcal{T})$ by selecting a k-query set $\mathcal{T}^*$ to ask the crowd,*

$$\mathcal{T}^* = \arg\max_{\mathcal{T}\subseteq\mathcal{F},|\mathcal{T}|=k} \mathbb{Q}(\mathcal{F}|\mathcal{T}) = \arg\min_{\mathcal{T}\subseteq\mathcal{F},|\mathcal{T}|=k} H(O|\mathcal{AS}^{\mathcal{T}}_{CE}) \tag{33}$$

Thus, the objective is to minimize the following conditional entropy of the observations given the answer family from CE,

$$H(O|\mathcal{AS}^{\mathcal{T}}_{CE}) = \sum_{A^{\mathcal{T}}_{CE}\in\mathcal{AS}^{\mathcal{T}}_{CE}} \sum_{o\in O} P(o|A^{\mathcal{T}}_{CE}) \log P(o|A^{\mathcal{T}}_{CE}) \tag{34}$$

Proof. The observation entropy $H(O)$ is constant for a given belief state $P(O)$. □

Note that $P(o|A^{\mathcal{T}}_{CE})$ can be calculated by Equation (23). Now we can give the entire procedure of the hierarchical crowdsourcing for data labeling in Algorithm 1. The Algorithm consists of the following steps, 1) divide crowd C into CE and CP, and initialize $P(O)$ using the labels from CP (Lines 1-2), 2) repeatedly select and distribute a k-query set $\mathcal{T}$ according to the optimization objective (Lines 4-5), and then update $P(O)$ with the collected answers to $\mathcal{T}$ from CE (Lines 6-7), until budget B is used up (Line 8). Solving the above optimization problem in Algorithm 1 is NP-hard.

Theorem 3 (**Computation hardness**). *Solving the optimization problem in Theorem 2 is NP-hard.*

Proof. Consider a special case of the optimization that there is only one expert worker in CE, say $CE = \{cr\}$. The optimization problem is equivalent to selection the optimal query set $\mathcal{T} \in \mathcal{F}$ with respect to the expected quality improvement. The optimal selection in this special case can be formalized to the optimization problem in [24], by mapping cr to the crowdsourcing workers with the same accuracy rates in their work. The latter optimization problem is proved to be NP-hard by their Theorem 5.1 in [24]. Therefore, finding an optimal query set in Theorem 2 is NP-hard. □

C. Approximate Solution

Since the checking-task selection is NP-hard, selecting the exact optimal checking tasks for each round is quite expensive. But due to the submodularity of the conditional entropy function, the problem of choosing a subset of checking tasks to maximize the conditional entropy can be approximated by iteratively choosing the most uncertain task each time. The greedy strategy guarantees the performance difference lower than $1 - 1/e$. Namely, we can select the checking task subset iteratively with a modified greedy algorithm to get a $(1-1/e)$-approximate solution. We define the *quality gain* of adding f into query set $\mathcal{T}$ as

$$gain^{\mathcal{T}}(f) = -H(O|\mathcal{AS}^{\mathcal{T}\cup\{f\}}_{CE}) + H(O|\mathcal{AS}^{\mathcal{T}}_{CE}) \tag{35}$$

Algorithm 2 calculates the approximate optimal k queries, given currently checking budget. Since within each round of query selection, the queries are incrementally selected and added into $\mathcal{T}$, the time complexity of the algorithm is $O(Nk)$, where N is the size of query space. The complete approximate hierarchical crowdsourcing for data labeling is shown in Algorithm 3. With the above approximate k-query selection algorithm, the hierarchical crowdsourcing is no longer NP-hard but reduced to $O(N^2)$.

D. Discussion

There are two important parameters in the above algorithms. In Algorithms 1 and 3, the crowd is divided into two groups according to the parameter, accuracy threshold θ, and in

Algorithm 3 Approximate Hierarchical Crowdsourcing

Require: Data set $\mathcal{F}$, crowd C, data labels for $\mathcal{F}$ from C, label checking budget B

Divide C into CE and CP according to Equation (1)
Initialize $P(O)$ with the labels from CP according to Equation (15)
repeat
  Select approximate optimal query set $\mathcal{T}$ (call Algorithm 2)
  Send $\mathcal{T}$ to CE and collect answer family $A_{CE}^{\mathcal{T}}$
  Update $P(O)$ with $A_{CE}^{\mathcal{T}}$ according to Equation (19)
  Set $B \leftarrow B - |\mathcal{T}| \cdot |CE|$
until $B < |\mathcal{T}| \cdot |CE|$
Update the labels with $P(O)$ according to Equation (20)

Algorithm 2, the parameter k decides how many queries would be selected to check. Intuitively, the selection of the query size k is a trade-off between data quality improvement and time cost. The smaller the k is, the more precise the crowdsourced answers are, meanwhile the more time-consuming the crowdsourcing process is. The selection of the accuracy threshold θ is a bit more complicated. With a higher threshold, the answers from the CE group are more precise, but simultaneously, the CE group is smaller and more high-accuracy workers only affect the result by initialization and have no chance to check the labels. A thorough theoretical study on the selection of the parameters would be interesting.

Another interesting question is whether the crowd can be divided into more groups than just two. There could be multiple choices for the framework design, and some of them would make the optimization quite complicated. To our knowledge, there has been no existing theoretical research on the topic. The only known fact is that for a very special case where there is only one expert worker in each CE group, a concatenation design (the labels are initialized by a single CP group, and then sequentially checked by multiple CE group) is equivalent to combining all the CE groups into one and assigning each task independently to the experts, no matter in what order the experts are arranged (a special case in [24]).

Some existing data labeling methods, for instance in medical image labeling practice, set some labeling workers as oracles [25]–[27], namely, their answers are always correct. The rationale of the setting is that there are always knowledgeable and trustful experts for data labeling, but it is impossible to let them complete the entire labeling work due to the high human resource costs. Our framework can be extended to consider the cost of each crowdsourcing worker, and the cost is related to his/her accuracy rate. With the extension of the crowdsourcing model, the optimization and approximation algorithms need to be re-designed. We leave the interesting topic for future work.

IV. EXPERIMENTAL EVALUATION

We conduct a series of experiments on a real dataset [28] to evaluate our proposal. The experiments are designed to study the following three aspects. First, we compare the performance of our proposal with 8 baseline algorithms in accurately inferring the ground-truth labels of decision-making tasks. Second, we study the robustness of the proposed approach with varying parameters, including k, θ, selection methods for label checking. Last, we discuss the efficiency of the approximation algorithm.

A. Experiment Setup

A single sentiment task is denoted as st and the set of sentiment tasks is sTs where $st \in sTs$. A st contains a tweet related to a company. For example, "The service of this company is too rude". Workers need to determine whether the tweet of each task has positive sentiment to the mentioned company and return "yes" or "no" to each task. This kind of task that only needs to answer "yes" and "no" is also called decision-making task. A worker w_1's answer could be "yes" or "no" for each sentiment task st. For instance, a twit "The recent products are amazing!" is sentimentally positive, and the ground truth for the fact that it should be labeled as "positive" is true.

In order to evaluate the effectiveness of our method better, for the dataset mentioned above, we aggregate 5 tasks of the same dataset to form a new task. Then, each task has 5 facts. For example, if there are 1000 tasks in the sentiment datasets, we can form 200 new tasks and each task has 5 facts to be labeled. In the experiment, we set the threshold $\theta = 0.9$ for dividing the preliminary workers and expert workers. Note that for those datasets with complete labels from all workers, the label checking is done offline and does not involve human interaction. The repeated task selection and answer collection can be regarded as a simulated online crowdsourcing framework. In our experiment, we select from the raw data 8 workers (including preliminary ones and expert ones) for each task. The **EBCC** algorithm [29] is used to initialize the answers of the preliminary workers.

The experiments are run on a Linux server with 2×24-core Intel® Xeon® Platinum 8260 CPU @ 2.40GHz processors and 251 GB Physical memory, and the Linux distribution Debian 10, x86_64 edition.

B. Accuracy Improvement

In this section, we will compare the accuracy of the proposed hierarchical crowdsourcing algorithm and the baseline methods in inferring the ground truth labels of decision-making tasks. Especially, we consider the following baseline methods:

- **Majority vote (MV)**: The final aggregated label result is the label that most people choose.
- **DS** [30]: **DS** will model the reliability of each worker by confusion matrix when aggregating labels.
- **ZenCrowd (ZC)** [31]: **ZC** model develops a probabilistic framework. It iteratively estimates the reliability of workers, identifies unreliable workers and infers ground truth labels.
- **GLAD** [32]: **GLAD** extends **ZC** model in task model and its task model considers the difficulty of tasks.

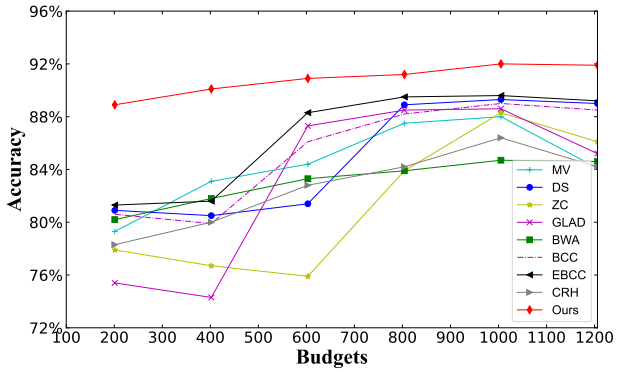


Fig. 2. Comparison with baseline algorithms

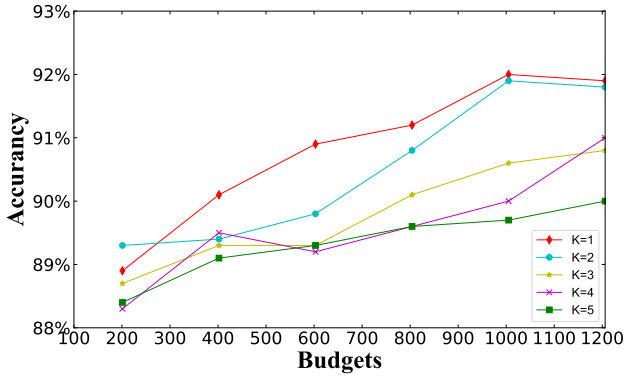
- **CRH** [33]:In **CRH** framework, there are two sets of unknown variables - truths and source reliability. The goal is to minimize the bias between truth and multi-source observations.
- **BWA** [34]: **BWA** is a method based on Bayesian graph model, which has conjugate prior and simple iterative expectation maximization reasoning.
- **BCC** [35]: **BCC** is an extension of **DS** and its optimization goal is to maximize the posterior joint probability.
- **EBCC** [29]: **EBCC** is an enhanced Bayesian classifier combination model which considers the correlation among workers.

We use the open-sourced code repository provided by Li et al. [34] and Zheng et al. [28] to implement the above baseline method.

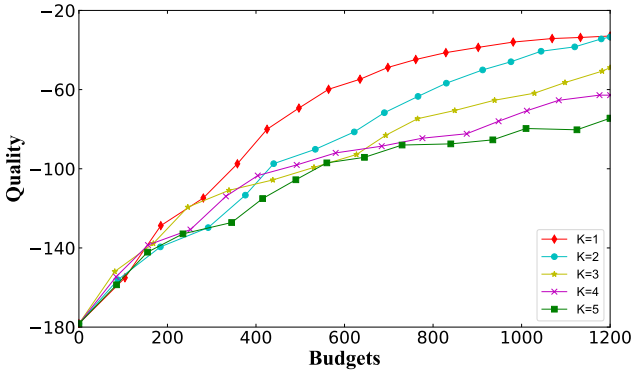
As shown in Figure 2, the performance of our hierarchical crowdsourcing (**HC**) outperforms other baselines. No matter what the budget is, the accuracy of **HC** is consistently higher than those of other algorithms. With budget 1000, a higher accuracy rate of 92.0% can be achieved. It's worth noticing that **HC** can still achieve a high accuracy rate (88.9%) even at a relatively low budget. To sum up, the proposed **HC** method shows good performance with high accuracy at low budget. **ZC**, **GLAD**, **BWA** and **CRH** algorithms do not perform well in our experiments. The accuracy of **ZC** and **GLAD** reasoning is affected by the accuracy of workers. Workers with higher accuracy answer fewer questions when budget is limited. The accuracy of **ZC** and **GLAD** is not high with less budget at the beginning. **CRH** and **BWA** require highly redundant data. High redundancy means more workers and more budgets. The **HC** proposed algorithm can achieve superior performance than most of the algorithms in the limited situation.

C. Quality Improvement

Besides comparing different methods on output label accuracy, we also use the data quality defined above to evaluate them. The data quality comparison provides an internal perspective and shows how the proposed algorithm approximates the optimal solution. The quality values of the data instances are simply summarized for evaluation.



(a) Accuracy improvement

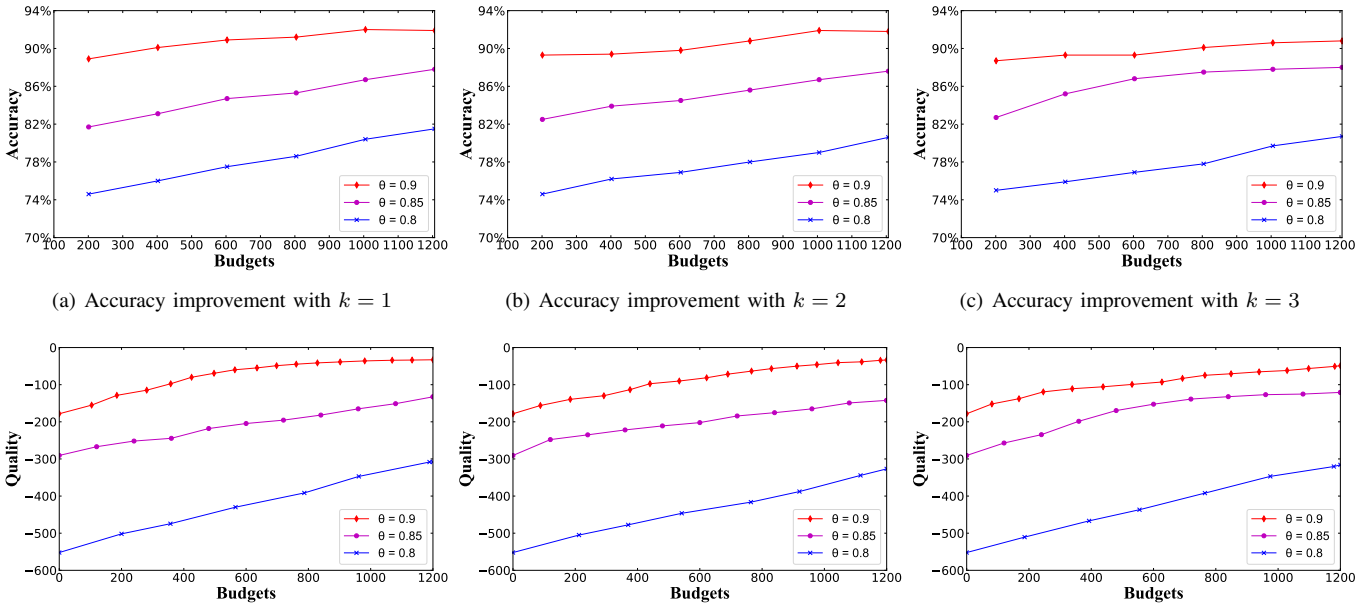


(b) Quality improvement

Fig. 3. Varying k

1) Varying k: As mentioned above, with complete labels from all workers, the checking process is offline and there is no interaction cost with human workers. However, if the answers from the expert workers are collected online, the interaction for answer collection will become expensive. Therefore, it is worth considering selecting and distributing multiple queries each round. As shown in Figure 3, the curves with smaller k values are of better quality and accuracy. The larger k is, the more tasks are answered at the same time. Obviously, as the budget increases, the growth rates of both accuracy and quality decrease. However, smaller k will cause more rounds with a limited budget. In each round, larger task set does not noticeably increase the waiting time to complete answer collection. Of course, we can accomplish our tasks faster accompanied by sacrificing accuracy if we take a larger k. In the whole iteration process, the quality of different k varies up to 45 and the maximum difference in accuracy is 3.7%.

2) Varying θ: To further explore how different thresholds affect the quality and accuracy of our experiments, we re-classified workers by taking the thresholds $\theta = 0.8, 0.85$ and 0.9. As budget increases, the accuracy and quality of the data after the different threshold classifications improve. Figure 4



(a) Accuracy improvement with $k = 1$ (b) Accuracy improvement with $k = 2$ (c) Accuracy improvement with $k = 3$

(d) Quality improvement with $k = 1$ (e) Quality improvement with $k = 2$ (f) Quality improvement with $k = 3$

Fig. 4. Varying θ

shows the larger θ is, the higher the probability of selecting expert workers, and the higher the accuracy and quality can be achieved with a small budget. When $\theta = 0.9$, the quality can reach -32.92 when $k = 1$. It is clear that the smaller the θ, the faster the increase in accuracy and quality, and the larger the θ, the slower the increase in accuracy and quality. When the budget is about 800, the slope of $\theta = 0.9$ quality curve and accuracy curve become significantly smaller, indicating that the marginal benefit of the budget decreases above 800. When the budget exceeds a certain amount, the performance curve with $\theta = 0.9$ decreases slightly. A possible reason is that most labels have been already checked and a few queries with wrong answers from the experts are repeatedly selected for updates.

3) Varying selection methods: In this subsection, the results of the following competitive algorithms are demonstrated: (1) **OPT**: selecting exact optimal algorithm by brute-force method, (2) **Approx**: the approximation algorithm, and (3) **Random**: the random algorithm. Please note that if k equals 1 for a subset, there is no difference between the **OPT** method and the **Approx** method. So we choose to discuss the difference between the three methods when $k = 2$ or $k = 3$. In Figure 5, the quality curves of **OPT** and **Approx** are almost identical and far higher than those of the random algorithm whether $k = 2$ or $k = 3$, which shows that our approximate algorithm is effective in reaching optimization target. The choice between OPT and Approx makes little difference when $k = 2$ and $k = 3$. The quality of Approx is slightly lower than that of the OPT algorithm with a margin of less than 0.1 when $k = 4$.

4) Varying Initialization: Obviously, the quality of **EBCC**, **DS** and **BCC** is always higher than that of **MV**, **ZC**, **GLAD**, **BWA** and **CRH** in Figure 6. The performance of the first three initialization methods is far better than that of the other five and gradually reaches a plateau after the budget is over 800. Among them, the data initialized by **EBCC** performs best in our framework. However, the quality gap between 8 curves narrows as the budget grows. After the iteration of our proposed algorithm, the accuracy of different initialization algorithms can reach more than 89.3%.

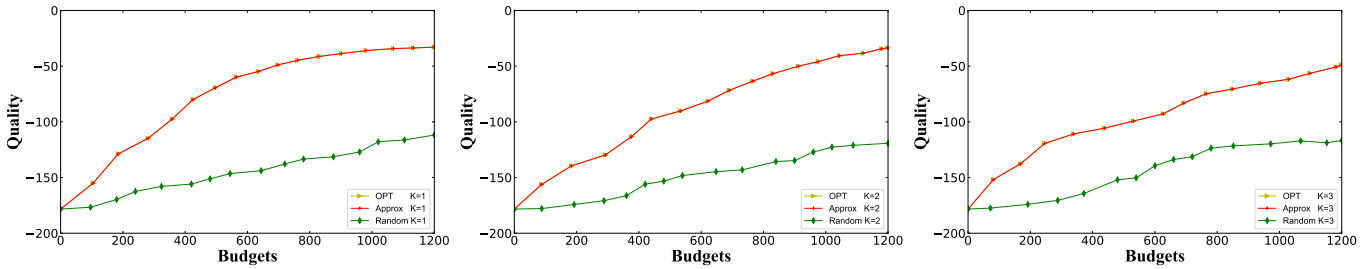
TABLE III
AVERAGE RUNNING TIME PER ROUND (SECONDS)

k	OPT	Approx
1	15.99	14.86
2	271.74	41.33
3	2840.04	84.20
4	timeout	144.58
5	timeout	240.73
6	timeout	398.73
7	timeout	708.98
8	timeout	1309.57
9	timeout	2489.51
10	timeout	4946.08

5) HC vs NO HC: We also compare our proposed hierarchical framework (**HC**) with brute-force label checking, namely, all workers serve as checking workers and the distribution is initialized as a uniform distribution (**NO HC**). As shown in Figure 7, for the same budget, the hierarchical design improves the data quality much faster.

D. Efficiency Evaluation

We compare the time cost of the proposed approximation algorithm (Approx) with optimal selection (OPT). To clearly show the performance difference between them in efficiency,



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Fig. 5. Varying selection methods for checking tasks

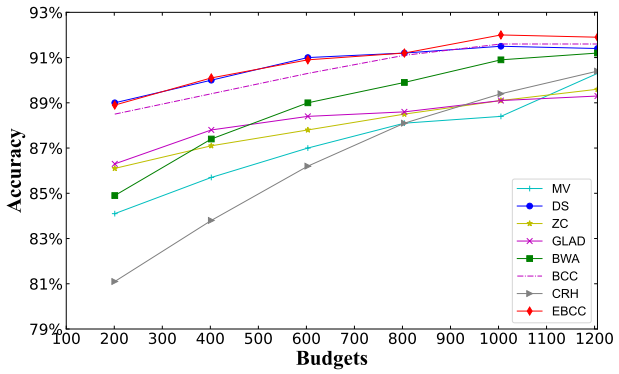


Fig. 6. Varying belief initialization

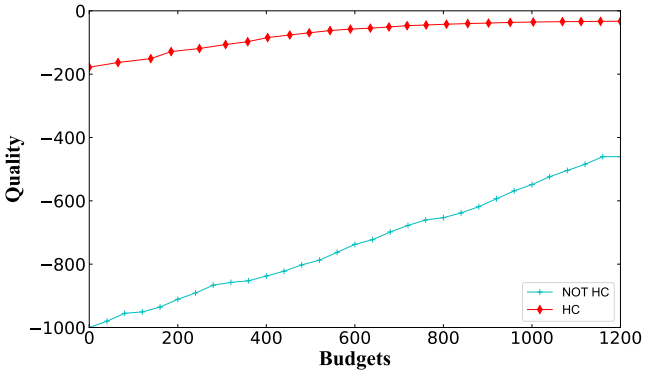


Fig. 7. HC vs NO HC

the algorithms are repeatedly tested on tasks that contain more than 20 facts. Table III shows the experimental results with different k values. The optimal selection requires much more computation time, for instance, it does not terminate after 6 hours when $k = 4$. The time cost of the optimal selection shows an exponential increase. Hence, it cannot finish the checking task selection within the limited time when k is relatively large. From the results in Section IV-C, it is obvious that the value of k allows to balance time and quality. The parameter k should be set to a small value or a large one depending on the bottleneck of the computation resources, crowdsourcing budget or calculation time. We should set k to a small value if we want better results with limited budget and to a large one if the computation time is limited and the budget is relatively sufficient.

V. Related Work

Label aggregation approaches are closely related to our work in this paper. A straightforward solution is to collect multiple labels from a crowd and then aggregate the answers [4]–[10]. The aggregation strategies include majority voting [11] and its many variants such as MV-Freq, MV-Beta, Paired-MV [12], [15], EM-based approaches [13], Markov CMC sampling, Graph neural nets [14] and so on. A major application of Label aggregation is truth inference. A classical EM algorithm [30] for inferring truth value iterates over two steps (estimating the result of each question and the accuracy of each worker's answer) until the algorithm converges. A probabilistic model (ZC) was proposed, which was based on factor graphs and got the result by integrating the factors such as worker's answer and worker's answer accuracy [31]. In [32], Whitehill et al. designs a probabilistic graph model (GLAD) for truth inference, simulates task difficulty to infer task annotation, error rate per crowdsourcing participant and task difficulty. The iterative process included in CRH [33] has high convergence and accuracy, in which an optimization problem is presented and aims to minimize the total distance between observations and truths provided by multiple data sources. Before the classification problem is solved by Li et al. [34], a conjugate Bayesian model (BWA) was proposed. Based on discrete binary classification tasks, it extends it to multi-class classification and uses expectation maximization (EM) for direct inference. BWA relies on Bayesian models to judge highly redundant annotations. BCC is proposed for unsupervised integration of discrete outputs from multiple black box classifiers [35]. It has been adopted for crowdsourcing with success. By simulating that workers are black box classifiers, their discrete outputs are the labels. BCC focuses on truth inference using worker relevance. EBCC focuses on the discrete problem of unsupervised crowdsourcing and extends BCC by considering potential worker relevance [29]. In EBCC model, Li et al. simulate worker dependencies by considering that the real class is a mixture of subtypes.

In [15], two strategies for label aggregation are proposed. Majority voting or its variants are used when the certainty level is high, and a paring scheme that generates weighted pairwise examples is used when the certainty level is low. The certainty level is measured simply by the portion of a certain label among all crowdsourced labels, or by a Bayesian estimation. However, both strategies require the assumption that all crowdsourced labels are equally important. In [16] a label-cleaning advisor is proposed which provides data scientists with two practical suggestions when noisy labels are required for training and testing a model. A mixture model is employed for worker annotations in [5], which learns a prediction model directly from samples to labels for efficient out-of-sample testing. In [6], a convex optimization formulation is proposed for learning from crowds without estimating the true labels by introducing personal models of the individual crowd workers. They also propose an iterative algorithm to solve the convex optimization problems by exploiting conditional independence structures in multiple classifiers. two approaches for crowd aggregation on multicategory answer spaces are studied in [7], which are respectively stochastic modeling-based and deterministic objective function-based crowd aggregation. In order to improve crowd aggregation, the skill and intention of individual workers are explicitly modeled in both approaches. A unified statistical potential model is proposed in [8] in which the differences between popular methods in this field correspond to different choices of model parameters. In [9], the authors discuss the problem of knowledge discovery in image databases when absolute ground truth is not available. The work concludes that, in the absence of calibrated ground truth, knowledge discovery methodologies can be modified to handle the situation provided the sources of uncertainty in the data are carefully handled. In [36], the labels from different workers are collected sequentially, and a stopping rule is proposed to define the total crowdsourced labels for a given task,

$$|V_{Y,t} - V_{N,t}| > C\sqrt{t} - \epsilon t \tag{36}$$

where the parameters C and t need to be chosen in advance. The label for the task is then finalized to the most frequent one, namely in majority rule. The difference between the two top probabilities of the possible labels, which is called bias, is used to measure the difficulty of the labeling tasks. In [4], a framework that calibrates the reliability and bias of expert labelers is proposed. The approach is used for detecting small volcanoes in Magellan SAR images of Venus. Their results suggest that it may be quite important to consider subjective noise when quantifying human and algorithmic detection performance. To further improve the quality of labels after ground truth inference in crowdsourcing, another framework [10] proposes an adaptive voting noise correction algorithm. The algorithm identifies and corrects the most likely noise in ground truth inference provided with the help of the estimated quality of the labelers. Label-cleaning advisors such as the one in [16] provides two pieces of valuable advice for data scientists when they need to train or test a model using noisy labels.

Active learning with crowd in [17], [37], [38] is another series of related works. They typically label the data instances by interactively selecting the most important ones based on some given criteria. The approach aims to improve the performance of classifiers by selecting a limited amount of data for labeling, which is different from our selectively relabeling work. Typical active learning methods include the active learning methods specially designed for crowd-sourced databases [37], [38]. ActiveClean [17] predicts the ground-truth label of each instance. Then, based on predicated ground-truth labels, estimates how cleaning each instance would change the model, and finally choose the instance that would lead to the greatest change. However, ActiveClean does not leverage current belief state of the noisy data to predict the ground-truth labels. Besides, it uses stochastic gradient descent to update the model when each batch of instances has been cleaned up, which may not be stable in performance, especially when the noise comes from a small part of the data. Most above approaches ignore or at least can not well capture the task correlations, resulting that the aggregated labels usually contain errors and may damnify the following model training.

Another closely related work [39] presents a crowdsourcing framework to reduce the uncertainty of data tuples. In their work, a proper set of human intelligence tasks (HIT) are selected and assigned to a crowd, and each task is done by one single worker in the crowd. However, the method assumes that for a query, only one answer is crowdsourced, and label aggregation is not considered. Therefore, the main theoretical result of the work shares a common special case with ours, namely, only one task is selected each round and assigned to a single worker. The special case has a trivial solution, namely, selecting the query with the maximum entropy.

VI. Conclusions

In this paper, we present a novel hierarchical crowdsourcing framework to improve the quality of labeled data, utilizing noisy answers from a group of heterogeneous workers. We divide the labeling crowd into preliminary and expert workers, and build an initialization-checking-update loop that effectively improves the data quality. The core optimization problem of checking task selection is proved to be NP-hard, so we propose an efficient algorithm to approximate the optimal solution. Our experiments on the real data set show that our proposed method can effectively improve the quality of labeled data for the downstream model training.

Acknowledgment

The work is partially supported by the Guangdong Basic and Applied Basic Research Foundation (Project 2022A1515010675), Swift Fund Fintech Funding, Shenzhen Talents Special Project - Guangdong Provincial Innovation and Entrepreneurship Team Supporting (Project 2021344612), Guangdong Pearl River Talent Recruitment Program (Project 2019ZT08X603) and Guangdong Pearl River Talent Plan (Project 2019JC01X235). The corresponding author is Chen Zhang.

REFERENCES

[1] K. Atarashi, S. Oyama, and M. Kurihara, "Semi-supervised learning from crowds using deep generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018, pp. 1555–1562.

[2] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, M. Carbin, and Z. Wang, "The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 306–16 316.

[3] A. G. Kupcsik, M. Spies, A. Klein, M. Todescato, N. Waniek, P. Schillinger, and M. Bürger, "Supervised training of dense object nets using optimal descriptors for industrial robotic applications," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 6093–6100.

[4] W. Bi, L. Wang, J. T. Kwok, and Z. Tu, "Learning to predict from crowdsourced data," in *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2014, pp. 82–91.

[5] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of venus images," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 7, pp. 1085–1092, 1994.

[6] P. Smyth, M. C. Burl, U. M. Fayyad, and P. Perona, "Knowledge discovery in large image databases: Dealing with uncertainties in ground truth," in *Proceedings of KDD workshop*, 1994, pp. 109–120.

[7] H. Kajino, Y. Tsuboi, and H. Kashima, "A convex formulation for learning from crowds," in *Proceedings of Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, 2012, pp. 73–79.

[8] 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, no. 3, pp. 794–809, 2014.

[9] J. Muhammadi, H. R. Rabiee, and A. Hosseini, "A unified statistical framework for crowd labeling," *Knowledge and Information Systems*, vol. 45, no. 2, pp. 271–294, 2015.

[10] J. Zhang, V. S. Sheng, J. Wu, X. Fu, and X. Wu, "Improving label quality in crowdsourcing using noise correction," in *Proceedings of the 24th ACM international on conference on information and knowledge management (CIKM)*, 2015, pp. 1931–1934.

[11] S. Nitzan and J. Paroush, "Optimal decision rules in uncertain dichotomous choice situations," *International Economic Review*, vol. 23, no. 2, pp. 289–297, 1982.

[12] V. S. Sheng, "Simple multiple noisy label utilization strategies," in *Proceedings of 2011 IEEE 11th International Conference on Data Mining (ICDM)*, 2011, pp. 635–644.

[13] J. Yang, J. Fan, Z. Wei, G. Li, T. Liu, and X. Du, "A game-based framework for crowdsourced data labeling," *The Very Large Data Bases Conference Journal*, vol. 29, no. 6, pp. 1311–1336, 2020.

[14] H. Wu, T. Ma, L. Wu, F. Xu, and S. Ji, "Exploiting heterogeneous graph neural networks with latent worker/task correlation information for label aggregation in crowdsourcing," *ACM Transactions on Knowledge Discovery from Data*, vol. 16, no. 2, pp. 1–18, 2021.

[15] V. S. Sheng, J. Zhang, B. Gu, and X. Wu, "Majority voting and pairing with multiple noisy labeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 7, pp. 1355–1368, 2017.

[16] M. Dolatshah, "Cleaning crowdsourced labels using oracles for statistical classification," Ph.D. dissertation, Applied Sciences: School of Computing Science, 2018.

[17] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "Activeclean: Interactive data cleaning for statistical modeling," *Proceedings of the Very Large Data Bases Conference (VLDB) Endowment*, vol. 9, no. 12, pp. 948–959, 2016.

[18] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on Artificial Intelligence (AAAI)*, 2019, pp. 590–597.

[19] S. E. Whang, P. Lofgren, and H. Garcia-Molina, "Question selection for crowd entity resolution," *Proceedings of the Very Large Data Bases Conference (VLDB) Endowment*, vol. 6, no. 6, pp. 349–360, 2013.

[20] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng, "Leveraging transitive relations for crowdsourced joins," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2013, pp. 229–240.

[21] S. B. Davidson, S. Khanna, T. Milo, and S. Roy, "Using the crowd for top-k and group-by queries," in *Proceedings of the 16th International Conference on Database Theory (ICDT)*, 2013, pp. 225–236.

[22] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom, "Crowdscreen: Algorithms for filtering data with humans," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2012, pp. 361–372.

[23] C. Zhang, H. Zhang, W. Xie, N. Liu, K. Wu, and L. Chen, "Where to: Crowd-aided path selection by selective bayesian network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 1072–1087, 2021.

[24] C. Zhang, H. Zhang, W. Xie, N. Liu, Q. Li, K. Wu, D. Jiang, P. Lin, and L. Chen, "Cleaning uncertain data with crowdsourcing - a general model with diverse accuracy rates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3629–3642, 2022.

[25] M. Bergman, T. Milo, S. Novgorodov, and W.-C. Tan, "Query-oriented data cleaning with oracles," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2015, pp. 1199–1214.

[26] O. Drien, M. Freiman, and Y. Amsterdamer, "Activepdb: active probabilistic databases," *Proceedings of the Very Large Data Bases Conference (VLDB) Endowment*, vol. 15, no. 12, pp. 3638–3641, 2022.

[27] C. Chai, L. Cao, G. Li, J. Li, Y. Luo, and S. Madden, "Human-in-the-loop outlier detection," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2020, pp. 19–33.

[28] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" in *Proceedings of the Very Large Data Bases Conference (VLDB) Endowment*, vol. 10, no. 5. Very Large Data Bases Conference (VLDB) Endowment, 2017, pp. 541–552.

[29] Y. Li, B. Rubinstein, and T. Cohn, "Exploiting worker correlation for label aggregation in crowdsourcing," in *Proceedings of International conference on machine learning (ICML)*, 2019, pp. 3886–3895.

[30] 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, no. 1, pp. 20–28, 1979.

[31] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st international conference on World Wide Web (WWW)*, 2012, pp. 469–478.

[32] J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proceedings of Advances in neural information processing systems (NeurIPS)*, vol. 22, 2009, pp. 2035–2043.

[33] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proceedings of the ACM SIGMOD international conference on Management of data (SIGMOD)*, 2014, pp. 1187–1198.

[34] Y. Li, B. IP Rubinstein, and T. Cohn, "Truth inference at scale: A bayesian model for adjudicating highly redundant crowd annotations," in *Proceedings of the World Wide Web Conference (WWW)*, 2019, pp. 1028–1038.

[35] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2012, pp. 619–627.

[36] I. Abraham, O. Alonso, V. Kandylas, R. Patel, S. Shelford, and A. Slivkins, "How many workers to ask? adaptive exploration for collecting high quality labels," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*, 2016, pp. 473–482.

[37] L. Zhao, G. Sukthankar, and R. Sukthankar, "Robust active learning using crowdsourced annotations for activity recognition," in *Proceedings of Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2011, pp. 74–79.

[38] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden, "Scaling up crowd-sourcing to very large datasets: a case for active learning," *Proceedings of the Very Large Data Bases Conference (VLDB) Endowment*, vol. 8, no. 2, pp. 125–136, 2014.

[39] Y. Chen, L. Chen, and C. J. Zhang, "Crowdfusion: A crowdsourced approach on data fusion refinement," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2017, pp. 127–130.