Harnessing Confidence for Report Aggregation in Crowdsourcing Environments

Hadeel Alhosaini
School of Computer Science
University of Technology Sydney
Sydney, Australia
Hadeel.Alhosaini@student.uts.edu.au

Xianzhi Wang
School of Computer Science
University of Technology Sydney
Sydney, Australia
Xianzhi.Wang@uts.edu.au

Lina Yao
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
Lina.Yao@unsw.edu.au

Zhong Yang
School of Computer Science
University of Technology Sydney
Sydney, Australia
Zhong.Yang@student.uts.edu.au

Farookh Hussain
School of Computer Science
University of Technology Sydney
Sydney, Australia
Farookh.Hussain@uts.edu.au

Ee-Peng Lim
School of Computing and Information Systems
Singapore Management University
Singapore
eplim@smu.edu.sg

***Abstract*—Crowdsourcing is an effective means of accomplishing human intelligence tasks by leveraging the collective wisdom of crowds. Given reports of various accuracy degrees from workers, it is important to make wise use of these reports to derive accurate task results. Intuitively, a task result derived from a sufficient number of reports bears lower uncertainty, and higher uncertainty otherwise. Existing report aggregation research, however, has largely neglected the above uncertainty issue. In this regard, we propose a novel report aggregation framework that defines and incorporates a new confidence measure to quantify the uncertainty associated with tasks and workers, thereby enhancing result accuracy. In particular, we employ a link analysis approach to propagate confidence information, subgraph extraction techniques to prioritize workers, and a progressive approach to gradually explore and consolidate workers' reports associated with less confident workers and tasks. The framework is generic enough to be combined with existing report aggregation methods. Experiments on four real-world datasets show it improves the accuracy of several competitive state-of-the-art methods.**

***Index Terms*—crowdsourcing, report aggregation, confidence propagation, experimental evaluation**

I. Introduction

Crowdsourcing is based on the notion of the collaborative work created by public participants. Its platforms have drawn increasing attention in recent years as a means of accomplishing human intelligence tasks effectively and economically on a large scale [1], [2]. It holds the great potential of data collecting, maximizing human resources, the ability to tackle problems on a large scale of the crowd rather than individually, supporting decisions making, and allowing access to diverse knowledge and experience [3]. It has been leveraged in various applications to exchange information for benefits or rewards. In general, the key elements of a typical crowdsourcing system include the outsourced task, requester (or crowdsourcer), workers, and platform for managing the within processes, Fig. 1 shows the overall workflow.

As these platforms offer an extensive pool of workers with variant expertise to undertake distinct tasks, they result in a large number of accumulated output tasks reports that need to be collected for further analysis. However, since workers can be of differed reliability [6], they may provide inaccurate, out-of-date, erroneous, or even false reports. Such uncertainty poses challenges to the distillation of accurate results from these reports [7]. In fact, data veracity has become one of the most critical challenges for data crowdsourcing [8]. It refers to the accuracy and quality of collected data. As the collection of trustworthy data can enhance the analysis of crowdsourcing platforms' data, there is an emphasis on implementing techniques to handle the data veracity challenge.

Currently, most crowdsourcing platforms address the data veracity challenge by letting multiple workers contribute to the same tasks and then aggregating those workers' reports to derive the final results. We define this **report aggregation problem** as follows:

Definition 1: Given a set of tasks (each representing a question), T, a pool of workers, W, and a collection of workers' reports for the tasks (i.e., workers' answers to the questions), V, the objective of report aggregation is to predict an answer for every task $t \in T$, where each answer is expected to be as close to the true answer as possible.

According to a US study [9], data aggregation is ranked as one of the top threats that can impact the reputation of organizations. As the quality of their techniques may be restricted by several challenges such as crowd workers' expertise or malicious attacks, it is crucial to control the crowdsourcing aggregation quality [10]. Therefore, an excellent report aggregation method should deliver accurate results given the collected reports.

Intuitively, more reports help reduce the uncertainty with the estimation of both worker reliability and task results—workers who contribute more reports give us more evidence to estimate their reliability, tasks that received more reports give

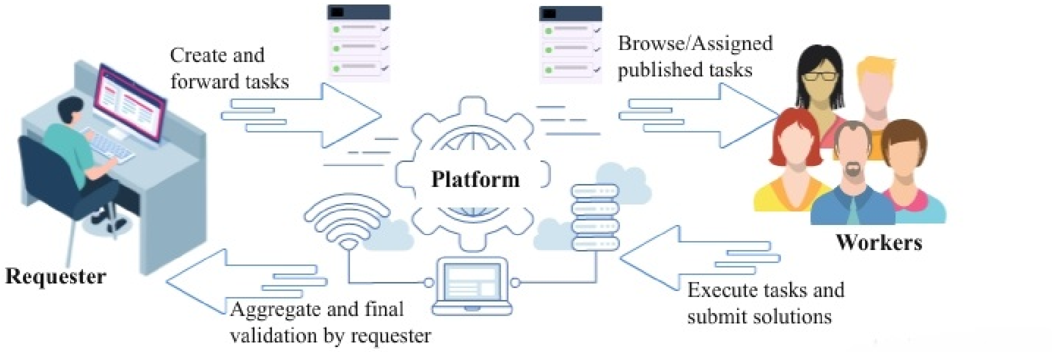

Create and forward tasks
Browse/Assigned published tasks
Platform
Requester
Workers
Execute tasks and submit solutions
Aggregate and final validation by requester

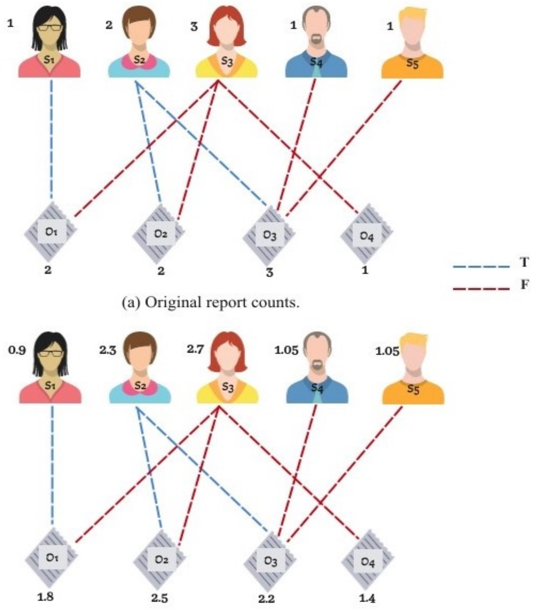
Fig. 1: A typical crowdsourcing framework

us more evidence to predict their results. So far, most previous research efforts have not considered such uncertainty in report aggregation thus compromising the accuracy of results [11], [12]. In fact, reducing this uncertainty is important from a practical point of view, as many real-world crowdsourcing scenarios witness uneven distributions of reports, demonstrated by the long-tail phenomenon [11], [13]. This phenomenon leads to drastically different uncertainty associated with workers and tasks. Based on this insight, we introduce the notion of confidence to handle such uncertainty. Specifically, in this paper, we propose a new concept, **confidence**, which measures the impact of report count on the uncertainty with predicted results, to improve the quality of crowdsourcing results:

Definition 2: Given an object (either a worker or a task), $o \in W \cup T$, and its associated reports $V(o)$, the confidence of o measures the degree to which we trust the evaluation result[1] of o and is a monotone increasing function of report size $|V(o)|$.

Confidence has two important properties: *monotonicity by report count* and *mutual dependency*. The first property suggests that more reports indicate stronger evidence and more reliable results; conversely, a smaller number of reports carry weaker evidence and lead to less reliable results. The second property suggests that *i*) high-confidence workers are likely to provide more confident reports and *ii*) workers who provide reports on high-confidence tasks are likely to be more confident. Therefore, it is necessary to propagate confidence among workers and tasks to incorporate this mutual dependency. Here, we use an example to illustrate the idea of confidence and demonstrate how it affects the aggregation results.

Example 1: Consider the scenario where a requester wants to determine the correctness of a textual statement via crowdsourcing. Five workers submit T or F labels to four statements (i.e., four tasks), denoted by labeled connections in Fig. 2. If we treat all workers equally and use majority voting to predict task labels, we can get the results of {?,?,F,F}, where '?' denotes an undetermined result. By using report count as *confidence*, we trust the workers and tasks with more reports, resulting in the confidence scores shown next to workers and tasks in Fig. 2a and leads us to the new prediction results of {F, F,?, F}. Further, by propagating the confidence among workers and tasks using our method (to be introduced later), we are able to obtain updated confidence scores shown in Fig. 2b and obtain the prediction results of {F,F,T,F}. We will later extract some real examples from our experimental results to validate the advantages of confidence propagation.



(a) Original report counts.

(b) Propagated report counts.

Fig. 2: An example of report counts before and after confidence propagation in a report aggregation problem.

Given the significance of considering confidence to the quality of crowdsourcing results, we propose a `General Report Aggregation` (`GRA`) framework that uses *confidence* as a heuristic to derive more accurate crowdsourcing results. We make the following contributions in this paper:

- We introduce a new concept of *confidence* to the crowdsourced data or report aggregation problem. We define

[1] The result is either the reliability of a worker or the predicted result for a task

base confidence as the numbers of associated reports of workers or tasks and propagate confidence among workers and tasks to take into account their interactions to derive the final confidence.

- We propose a progressive report aggregation approach that gives priority to evaluating high-confidence workers and tasks and then leverages high-confidence results to reduce the uncertainty associated with lower-confidence workers and tasks to improve the overall accuracy of results for all tasks.
- We incorporate confidence with performance measures of workers and tasks to derive crowdsourcing results. Extensive experiments on real-world datasets demonstrate that GRA improves the accuracy of existing methods. The results also support our analysis that GRA does not increase the time complexity of existing methods.

II. Related Work

Data aggregation methods, also known as truth discovery techniques, have broad applications in the integration of multi-source data for various purposes such as fact checking, truth inference, and spam detection [14]. Existing truth discovery research mostly focuses on unsupervised methods due to the lack of ground truth in many real-world scenarios [15], [16].

Until now, the related techniques have rarely been applied to the crowdsourcing domain [14]. They generally fall into three categories: *i*) iterative methods, which use heuristic functions to alternately evaluate workers' reliability and reports' veracity to predict true answers [17]–[20]; *ii*) optimization methods, which predict true answers by explicitly solving optimization models [11], [21]–[24]; *iii*) probabilistic methods, which employ Bayesian models, typically generative models, to predict true answers via point estimate [25], maximum likelihood [26]–[28], or maximum a posteriori probability estimation [29]–[31]. Some methods further incorporate problem-specific clues such as value similarity [20], item difficulty [19], and source dependence [27], [32] to achieve better performance.

Although most of the above truth discovery techniques can be applied to crowdsourcing scenarios, they share two common deficiencies. First, they do not consider the uncertainty associated with the prediction results derived from differed numbers of reports received by different tasks. Consequently, they may not perform well on many real-world crowdsourcing scenarios that exhibit long-tailed distributions of report counts over workers or tasks. Second, previous studies [6] have shown that it reduces nearly half of the mistakes made by a general truth discovery technique to know the reliability of information sources (i.e., workers in the crowdsourcing context) a priori. However, most existing methods use random or default values to initialize parameters (e.g., the reliability of information sources and the veracity of sources' reports), which may not result in satisfactory performance.

The most relevant work to ours is the confidence-aware truth discovery method [11]. This method considers the positive correlation between information sources' *confidence* and the numbers of its provided reports and shows the positive effect of considering such confidence in predicting values of tasks from multi-sourced data. Our framework distinguishes from the above work in employing a progressive approach (rather than a one-off approach) that incorporates propagated confidence (rather than merely the confidence of information sources) to derive the crowdsourcing results. While the progressive approach enables better settings of parameters to accelerate the aggregation process, confidence propagation considers the mutual reinforcement of confidence among workers and tasks to deliver more reasonable confidence computation. Besides, instead of being a proprietary method, our framework is general and can be applied to multiple existing methods to life their performance.

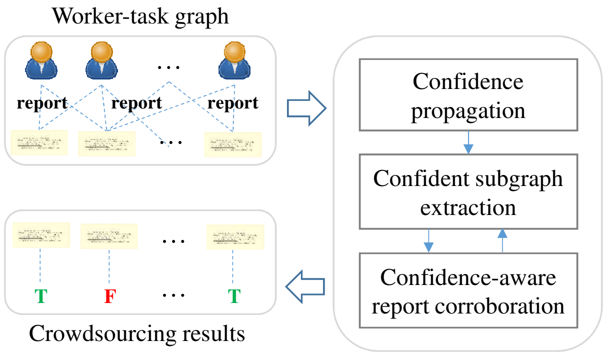


Fig. 3: General report aggregation framework

III. Our Proposed Framework

Our GRA framework (see Fig. 3) involves three components: confidence propagation, confident subgraph extraction, and confidence-aware report aggregation. The *confidence propagation* component aims to compute confidence scores from report counts of workers and tasks. It considers not only the confidence of workers and tasks but also the interaction between workers and tasks in the computation. The *confident subgraph extraction* and *confidence-aware report aggregation* components together form a progressive report aggregation approach. We hypothesize that some subsets of workers and tasks can provide a better estimation of the optimal parameters than default or random settings. Therefore, the approach does not require parameter tuning for every individual dataset but has to potential to yield higher accuracy.

In particular, we use *confidence* as the criterion to differentiate workers and tasks and to derive subsets of different confidence levels. On this basis, the progressive approach learns the parameter settings from high-confident subsets of workers and tasks and apply them to lower-confident subsets to boost the overall accuracy, where *confident subgraph extraction* is responsible for progressively inducing better prior parameters for report aggregation methods, while *confidence-aware report aggregation* incorporates confidence scores to improve the accuracy of crowdsourcing results. By using confidence as a heuristic to induce higher-quality results for low-confidence

TABLE I: Notations used in the paper.

Notation	Explanation
W	Set of workers
T	Set of tasks
E	Set of edges/interactions among workers and tasks
$T(w)$	Set of tasks on which a worker w provides reports
$W(t)$	Set of workers who provide reports on a task t
$V(w)$	Set of reports provided by worker w
$V(t)$	Set of reports received by task t
C	Confidence of workers and tasks
R	Reliability of workers
F	Veracity of answers for tasks
$c(w)$, $c(t)$	Confidence of worker w and task t
$r(w)$	Reliability of worker w
$f(v_t)$	Veracity of answer v_t for task t

Algorithm 1 The GRA Framework

```
Input: a bipartite graph $G(W, T; E)$;
       a predefined number of iterations, $k$.
Output: the answer predicted for every task.
1: $R \leftarrow \{r(w) | r(w) = 0.8, w \in W\}$
2: $F \leftarrow \{f(v_t) | f(v_t) = \frac{1}{|V(t)|}, v_t \in V(t), t \in T\}$
3: $\hat{G} \leftarrow G$
4: $C \leftarrow$ Algorithm 2 ($G$)
5: while fewer than $k$ iterations do
6:     $G \leftarrow$ Algorithm 3 ($G$, $C$, $k$)
7:     $R$, $F \leftarrow$ Algorithm 4 ($\hat{G}/G$, $C$, $R$, $F$)
8: end while
9: return $\{\arg\max_{v_t \in V(t)} f(v_t) | t \in T\}$
```

workers and tasks from those results for high-confidence workers and tasks, a report aggregation method to continuously adapt its parameters and achieves higher accuracy.

We list the main notations used in this paper in Table I. As shown in Algorithm 1, our report aggregation framework starts by initializing the parameters, including the reliability of workers (line 1) and the veracity of answers for every task (line 2). We initialize every answer with the equal veracity because we know nothing about answers at the beginning. The framework then computes the propagated confidence of workers and tasks based on the full worker-task bipartite graph (line 4), followed by an iterative subgraph extraction (line 6) and report corroboration (line 7) procedure. In each iteration, Algorithm 3 extracts a most confident subgraph from the worker-task graph. The subgraph size is determined by the predefined number of iterations k. All the extracted subgraphs together (denoted by $\hat{G}/G$) represent a subproblem of the original report aggregation problem and form the input of Algorithm 4 for report corroboration. The corroboration results will be used as priors of future corroborations. In this way, the results from previous iterations serve as seeds to boost the result accuracy of future iterations. The final result of a task is the answer with the highest veracity score among the scores of all answers provided by workers for the task (line 9).

The framework ensures confident predictions always precede less confident predictions during the corroboration process. But it never prevents methods from using low-confidence subsets. The rationale lies in the predicted results can be passed down to future iterations and provide better seedings for result predictions on less confident workers and tasks. Note that the original input, G, should be a connected graph; otherwise, each independent subgraph represents a different problem and should be solved separately.

A. *Confidence Propagation*

Our confidence computation is based on two ideas. First, workers and tasks associated with different numbers of reports should bring about varying degrees of uncertainty. Second, all clues including the workers and tasks with sparse reports should be considered to improve the result accuracy. On the above basis, we define *confidence score* to measure the strength at which the multi-source data support the evaluation/prediction regarding different workers and tasks.

The straightforward way to compute confidence is to equate confidence with report count as done in Example 2 (a). However, this method considers only the direct report counts of workers and tasks but neglects their interactions. As discussed, we recognize the mutual dependency of confidence between workers and tasks, i.e., the confidence of a worker or task is determined not only by its report count but also by the confidence and report counts of other workers and tasks both directly and indirectly associated with it.

Therefore, we propose a propagation-based confidence computation method based on a variant of Hyperlink-Induced Topic Search (HITS) [17] and describe it in Algorithm 2. The principle is that a worker or task associated with more workers and tasks of higher confidence also has higher confidence. In particular, the hub scores and authority scores in HITS correspond to the confidence of workers and the confidence of tasks, respectively. In each iteration, the algorithm updates the confidence of both workers and tasks following the edges of the bipartite worker-task graph (lines 4-9). The sum of all propagated scores are bounded by the original total report count ($\hat{N}$) throughout iterations (lines 10-12); this ensures the algorithm can always converge to the eigenvector (lines 3-13). Since $\hat{N}$ represents the total number of reports over all workers and tasks, the propagation process eventually reaches a stationary redistribution of this number among all workers and tasks. The final step of confidence propagation is to transform the propagated scores into a reasonable range to ease future use (lines 14-16). The final output is a confidence score for every worker and task (line 17).

Unlike in HITS, where the hub scores and authority scores are normalized separately, it is crucial to normalize the confidence of workers and tasks simultaneously due to their mutual dependency. Since confidence scores might be distributed arbitrarily over workers and tasks, we consider them as univariate independent and identically distributed samples drawn from some distribution with an unknown density g and conduct

Algorithm 2 Confidence Propagation

```
Input: a bipartite graph $G(W, T; E)$.
Output: confidence scores of every worker and task.
$c(w), c(t) \leftarrow 1, \forall w \in W, \forall t \in T$.
$\hat{N} \leftarrow \sum_{w \in W} |T(w)| + \sum_{t \in T} |W(t)|$
while convergence not reached do
    for $w \in W$ do
        $c(w) \leftarrow \sum_{t \in T(w)} \sum_{w' \in W(t)} c(w')$
    end for
    for $t \in T$ do
        $c(t) \leftarrow \sum_{w \in W(t)} \sum_{t' \in T(w)} c(t')$
    end for
    for $o \in W \cup T$ do
        $c(o) \leftarrow \frac{\hat{N}}{\sum_{w \in W} c(w) + \sum_{t \in T} c(t)} \cdot c(o)$
    end for
end while
for $o \in W \cup T$ do
    $c(o) \leftarrow$ update $c(o)$ by Eq. (2)
end for
return $\{c(o) | o \in W \cup T\}$
```

kernel density estimation [33] to transform confidence into probabilistic representations:

$$g_h(c) = \frac{1}{(|W| + |T|)h} \sum_{o \in W \cup T} \mathcal{K}(\frac{c - c(o)}{h}) \tag{1}$$

where $\mathcal{K}$ is the kernel, a non-negative function that integrates to one, and $h > 0$ is the bandwidth. Since the density estimation technique is not the focus in this work, we just choose the standard normal kernel function due to its convenient mathematical properties, and use the rule-of-thumb bandwidth estimator for estimating Gaussian density to select the optimal bandwidth, i.e., $h = 1.06\sigma \cdot (|W| + |T|)^{-\frac{1}{5}}$, where σ is the standard variance of all confidence scores. On the above basis, we calculate the transformed confidence score for each worker or task o, $c(o)$, as follows:

$$c(o) = \int_{-\infty}^{c(o)} g_h(c) \, \mathrm{d}c \tag{2}$$

This method essentially assigns confidence scores to workers and tasks based on their standings in the list of raw confidence scores of all workers and tasks. In this way, this method enforces a more even and coherent distribution that is insensitive to the differences among the original confidence scores, thereby avoiding the dominance of those extremely large or tiny raw confidence scores in subgraphs (to be extracted later).

B. Confident Subgraph Extraction

In this section, we aim to extract some confident subgraphs from a worker-tasks bipartite graph to facilitate more accurate report aggregation on the original graph. As discussed, workers and tasks of higher confidence are associated with less uncertainty and therefore liable to derive more accurate results.

Algorithm 3 Confident Subgraph Extraction

```
Input: a bipartite graph, $G(W, T; E)$;
       confidence scores, $C = \{c(o) | o \in W \cup T\}$;
       the number of iterations, $k$.
Output: updated $G$.
1:  for $(w, t) \in E$ do
2:      $\omega(w, t) \leftarrow c(w) + c(t)$
3:  end for
4:  Initialize an empty graph $G'$
5:  while fewer than $\lceil |E|/k \rceil$ iterations and $E \neq \emptyset$ do
6:      $e \leftarrow \underset{(w,t) \in E \wedge (w \in W' \vee t \in T')}{\arg\max} \omega(w, t)$
7:      Add $e$ to $G'$
8:      Remove $e$ from $G$
9:  end while
10: return $G$
```

Based on this insight, given a worker-task bipartite graph, we hypothesize that a confident subgraph can better approximate a graph than a random or lower-confidence subgraph. Therefore, it can derive a better estimation of workers' reliability as well as task results.

We define subgraph extraction as the problem of maximizing the total weights of edges, subject to a maximum number of edges of the resulting subgraph:

$$\underset{E' \subseteq E \wedge |E'| = \max(\lceil |E|/k \rceil, |E|)}{\arg\max} \sum_{(w,t) \in E'} \omega(w, t) \tag{3}$$
$$\text{s.t., } G'(W', T'; E') \text{ is a connected graph}$$

where the subgraph is formed by the selected edges and their vertexes in the original graph; the weight of an edge is calculated as the sum of confidence scores of both vertexes of the edge.

This is a typical dense subgraph extraction problem and can be solved by different state-of-the-art techniques such as dynamic programming and evolutionary algorithms [34]. To maximally preserve the efficiency of the `GRA` framework, we design a heuristic algorithm to find an approximate solution. Specifically, Algorithm 3 shows our proposed subgraph extraction procedure. First, each edge is assigned a weight equaling the sum of confidence scores of both its vertexes (lines 1-3). Then, a subgraph is initialized (line 4) and updated through iterations until reaching a predefined size limit (lines 5-9). In particular, each iteration adds a new edge to the subgraph. The new edge meets two criteria: i) it connects with the subgraph but is not a duplicate of an edge in the subgraph; ii) it has the biggest edge weight in the input graph. After the required number of edges are selected, they together form a connected subgraph to be returned as the output of the subgraph extraction algorithm (line 10).

C. Confidence-Aware Report Corroboration

Algorithm 4 shows the report corroboration procedure, which takes the results from previous iterations of the progressive approach as inputs and uses them as priors in the

Algorithm 4 Report Corroboration

```
Input: a bipartite graph $G^\star(W^\star, T^\star; E^\star)$;
       a set of reports from workers $V$;
       prior reliability of workers $R$;
       prior answer veracity of tasks $F$.
Output: estimated reliability of every worker and predicted
    true answer for every task.
 1: while convergence not reached do
 2:   for $t \in T^\star$ do
 3:     for $v_t \in V(t)$ do
 4:       $f(v_t) \leftarrow$ estimate the veracity of $v_t$ based on $R$
          smoothed by $C$
 5:     end for
 6:   end for
 7:   for $w \in W^\star$ do
 8:     $r(w) \leftarrow$ estimate the reliability of $w$ based on $F$
        smoothed by $C$
 9:   end for
10: end while
11: return $R$, $F$.
```

subsequent iterations of Algorithm 1. It first initializes worker reliability and answer veracity using the input variables R and F, which contains either *i*) default values if the workers and tasks are not involved in previous iterations of Algorithm 1), or *ii*) the estimation results from the last previous iteration otherwise (lines 3-5). Since all existing truth discovery methods follow a similar principle, i.e., inferring worker reliability and answer veracity alternately from each other, here, we show a general report aggregation procedure (lines 1-10) while different methods may employ various techniques to conduct the specific calculation (lines 4 & 8). Once new results are obtained, they would be updated into F (line 4) and R (line 8) for subsequent iterations or deriving the final results. In particular, each $f(v_t) \in F$ and $r(w) \in R$ is smoothed as follows:

$$f(v_t) = c(v_t) \cdot f(v_t) + \big(1 - c(v_t)\big) \cdot \frac{\sum_{v'_t \in V(t)} f(v'_t)}{|V(t)|} \tag{4}$$

$$r(w) = c(w) \cdot r(w) + \big(1 - c(w)\big) \cdot \frac{\sum_{w' \in W} r(w')}{|W|} \tag{5}$$

The above equations incorporate the influence of confidence by giving more credit the the results related to higher-confidence workers and tasks and tuning the results on lower-confidence more towards the average results.

D. Time Complexity

Given the number of workers, $|W|$, the number of tasks, $|T|$, and workers' reports for tasks, $|V|$, suppose the iteration times is M, then the time complexity of a traditional report aggregation method is $O(|W||T||V|M)$.

The time complexity of the GRA framework consists of two parts: confidence computation and progressive iteration. Suppose Algorithm 2 converges after N iterations, then the time complexity of confidence computation is $O(|W||T|N)$. Given the iteration number k, the second part has the complexity of $O\big(|W||T||V|M\sum_{i=1}^{k}(\frac{i}{k})^3\big) \leq O\big(|W||T||V|Mk(\frac{k}{k})^3\big) = O(k|W||T||V|M)$. Since k is manually defined and cannot be excessively large, it can usually be regarded as a constant. That reduces the time complexity to $O(|W||T||V|M)$.

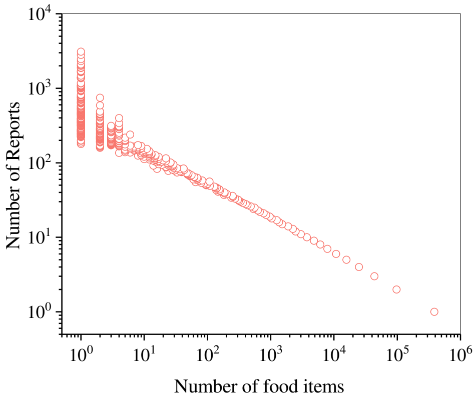


Fig. 4: Reports distribution over food items

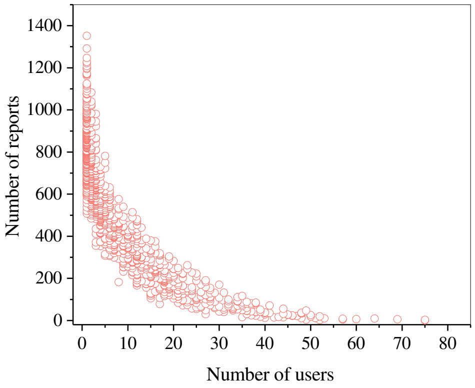


Fig. 5: Reports distribution over users

IV. Experiments

In this section, we present our experiments to evaluate the GRA framework against several state-of-the-art methods as well as under different configurations.

A. Experimental Setup

1) Datasets: We employed four real-world crowdsourcing datasets in our experiments.

- *Bird recognition* [35]. This is also a real crowdsourcing dataset from Amazon Mechanical Turk. It contains 4,212 judgments of 39 workers on 108 images to distinguish between two types of birds.
- *Weather sentiment* [36]. This is a real crowdsourcing dataset from Amazon Mechanical Turk. It contains 6,000 classification judgments by 110 workers on weather-related sentiment in 300 tweets. Each tweet belongs to one of five classes: 'negative', 'neutral', 'positive', 'irrelevant to weather', and 'can't tell'.
- *City population* [18], [37]. This dataset contains people's editing records on the population sizes of cities as of

the year 2010 in Wikipedia. After discarding duplicate and ambiguous reports, we obtained 51,761 reports from 4,264 users regarding the population sizes of 41,197 cities. We used the latest editing records as the ground truth.

- *Food nutrition*. We prepared the fourth dataset by crawling 2,085,121 reports about 17 nutrition aspects (e.g., calories, dietary fiber, and total fat) of 630,567 foods contributed by 9,896 users (i.e., workers) from myfitnesspal.com. Each nutrition aspect measures the amount of a type of nutrient contained in per serving or unit mass of food items.

To gain a better understanding of the new dataset (food nutrition), we investigated the distributions of reports over users and tasks, respectively. Both distributions show an evident long-tail phenomenon—over 70% of the food items are covered by only one user, and 98% are covered by no more than ten users (Fig. 4); similarly, over 33% users provide reports on no more than ten food items (Fig. 5). These observations imply that we may not be able to make confident evaluations of many workers and tasks that make up the long tail and further suggest the necessity of incorporating confidence in report aggregation. Similar distributions were also observed in the other datasets except the bird recognition dataset; we particularly select this dataset as a contrary example to explore the influence of considering confidence on datasets with different report distributions in our experiments.

Besides, we pre-standardized the values in the numerical dataset into z-scores to avoid bias caused by the different value ranges of tasks. Besides, we executed all methods 20 runs under each configuration setting and used their average performance for the evaluation.

2) Evaluation Metrics: For categorical datasets (the first two datasets), we evaluate methods' performance using *accuracy*, i.e., the proportion of correct answers among all the answers predicted by each method for all tasks. For numerical datasets (the last two datasets), use *Mean Absolute Mean Error (MAE)* and *Root Mean Square Error (RMSE)* to measure the discrepancy of task results against ground truth. For accuracy, a larger value indicates better performance; while for any of the other two metrics, a smaller value indicates better performance.

B. Evaluation of Methods

We evaluated the effectiveness of our GRA framework before and after it is applied to four categories of methods:

- Primitive approach, i.e., Majority Voting (**mVoting**). For each item, it predicts the answer provided by most workers, or a random answer in case of a tie, as the truth.
- Iterative approach, including **Sums** [17], **Avg-Log** [18], **Invest** [18], **PooledInvest** [18], **Cosine** [19], **2-Estimates** [19], and **TruthFinder** [20]. They estimate worker reliability and answer veracity simultaneously using different calculation methods.
- Optimization approach, i.e., the Conflict Resolution on Heterogeneous Data (**CRH**) framework [21], which minimizes the difference between workers' inputs and the predicted answer to derive the true answer.
- Probabilistic approach, including **AccuPr** [27] and Gaussian Truth Model (**GTM**) [29]. These methods use point estimate and generative models, respectively, to infer values' a posterior truth probability.
- Confidence-Aware Truth Discovery (**CATD**) [11]. This method gives workers with fewer reports smaller weights but considers only worker confidence as a one-off, non-generic approach.

Note that, GTM is only applicable to numerical datasets; our GRA framework was not applied to CATD as it already considered the notion of confidence in a different way. The convergence of these methods has been either proved or empirically evaluated before; therefore, we re-implemented all the methods in Python and configured them with the optimal parameter values suggested by their original authors. Meanwhile, we set the number of iterations $k = 4$ for our GRA framework. All experiments were conducted on a PC with Intel® Core™ i7-4790 processors (3.6GH×8) and 16GB RAM. We avoided incorporating more features that are available to some of these methods such as input transformation [30] and source dependence [27] to ensure a fair comparison.

Table II shows the performance of methods before and after applying our framework on the four datasets, where ¬GRA and GRA represent the performance (in terms of either accuracy or errors) before and after applying our framework, respectively. The results show that our framework improves the accuracy or reduces the errors of all methods significantly, in many cases by half, on all datasets excluding the bird recognition dataset. The reason is that we have specially chosen this dataset, where the reports are distributed evenly among all workers and all tasks. Under this circumstance, our confidence-aware approach degrades into a simple iterative approach that extracts random subgraphs each time. The results show the random method does not have a significant impact on the result accuracy. Another observation is that the improvement is more evident on the larger dataset (e.g., food nutrition) than on smaller datasets (the others). This observation implies that our framework favors large datasets from the accuracy standpoint.

C. Evaluation of Configuration Settings

In this section, we report the performance of the GRA framework under different confidence computation methods, progressive strategy, and parameter settings. To ease illustration, we use TruthFinder as an example to demonstrate the evaluation results; all the other methods draw similar conclusions. In all the tables showing the evaluation results, we show the best performing methods/strategies with a gray background and the best performance values in boldface.

1) Confidence Computation Method: To evaluate the impact of confidence computation to our GRA framework, in this experiment, we kept all the other configuration aspects unchanged while evaluating the performance under different confidence computation methods as follows:

TABLE II: Accuracy of methods before and after applying our framework, indicated by ¬GRA and GRA, respectively. The best performance values are in boldface

Method	Bird recognition		Weather sentiment		City population				Food nutrition			
	Accuracy		Accuracy		MAE		RMSE		MAE		RMSE	
	¬GRA	GRA	¬GRA	GRA	¬GRA	GRA	¬GRA	GRA	¬GRA	GRA	¬GRA	GRA
mVoting	0.69	0.69	0.77	0.85	10,327	9,072	126,217	123,426	356	208	469	276
Sums	0.72	0.72	0.79	0.82	3,026	2,645	17,023	15,738	316	164	412	227
Avg-Log	0.74	0.74	0.79	0.82	2,923	2,623	16,923	15,624	316	164	413	228
Invest	0.74	0.74	0.74	0.82	1,787	1,622	9,797	9,501	320	165	417	229
PooledInvest	0.74	0.74	0.74	0.85	1,792	1,653	9,901	9,589	327	167	421	230
Cosine	0.72	0.72	0.72	0.77	1,695	1,609	9,329	8,623	316	163	413	228
2-Estimates	**0.76**	**0.76**	0.79	**0.87**	1,652	1,542	9,058	8,342	316	163	412	227
TruthFinder	0.74	0.74	**0.82**	**0.87**	1,633	1,489	8,823	8,018	315	163	412	227
AccuPr	0.74	0.74	**0.82**	0.85	1,639	1,505	8,898	8,190	306	**157**	398	223
CRH	0.72	0.74	0.79	0.85	1,636	1,502	8,864	8,123	**304**	**157**	**394**	**222**
GTM	—	—	—	—	41,623	**1,463**	8,582	**7,636**	306	164	412	227
CATD	0.74	0.74	**0.82**	—	**1,594**	—	**7,740**	—	306	—	412	—

- Source popularity-based method. This method measures each source's confidence by its *popularity*, i.e., the number of tasks on which the source provides reports.
- Item popularity-based method. This method measures each worker's confidence by its *popularity*, i.e., the number of workers that provide reports on this task.
- Two-sided popularity-based method. This method considers the confidence of both workers and tasks in terms of their *popularity*.
- HITS-based method. This is our method, which propagates confidence among workers and tasks based on a variant of HITS (Section III-A).

The results (Table III) show that our propagation-based method always yields highest accuracy and smallest errors among all methods, which demonstrates the advantages of propagated *confidence* over the alternative measure of popularity (i.e., direct report count).

2) Seeding Strategy: Table IV shows the performance of our approach following different seeding strategies to extract subgraphs. The extracted subgraphs will be used as seeds to improve the accuracy of methods on the unexplored workers and tasks.

- Non-progressive strategy. This strategy considers confidence but performs one-off report aggregation using predefined values to initialize the methods.
- Random seeding. This strategy randomly selects workers and tasks from the unexplored workers and tasks to participate in the next iteration of the progressive procedure.
- Low-confidence seeding. This strategy always selects the source and tasks with the lowest confidence among the unexplored workers and tasks for the next iteration.
- High-confidence seeding. This is the strategy adopted in the proposed approach, which always selects the workers and tasks with the highest confidence among the unexplored workers and tasks for the next iteration.

The results (Table IV) show that high-confidence seeding outperforms the other strategies, demonstrated by the higher accuracy on categorical datasets and smaller MAE and RMSE on numerical datasets. In contrast, inappropriate strategies, such as random and low-confidence seeding, could produce worse results than the traditional non-progressive approach.

To further investigate the optimization process of the progressive approach under different strategies, Fig. 6 shows the average MAE over all the explored tasks calculated for each of the ten iterations of our approach. For high-confidence seeding, we observed that earlier subgraphs (formed by higher-confidence workers and tasks) generally yielded smaller errors. Although the overall error increased when more low-confidence workers/items were involved, the previous computation seemed to prevent it from growing excessively. In contrast, both random seeding and low-confidence seeding found more subgraphs useful for reducing their average errors. This observation aligns with our intuitions that *more evidence generally yields better predictions when we have no prior knowledge about the dataset* (as in random seeding) and *better evidence leads to better predictions* (as in low-confidence seeding). Since a traditional method does not employ the progressive approach, its error remains unchanged in Fig. 6.

3) Parameter Setting: In this section, we study the impact of subgraph number, k, on the framework's performance, where we used *TruthFinder* and the food nutrition dataset as examples to discuss the results. In particular, when k=1, the approach degrades to the non-progressive approach. The results (Fig. 7) show that the computation time grows approximately linearly with k, indicating the predictable time cost of our approach. A small k significantly reduced the MAE, but it started to increase beyond some point (e.g., 7 in this experiment). This reveals the importance of configuring a proper value of k. Specifically, we set $k = 4$ throughout our experiments for the best performance, and automatic selection of k should be a topic of future research.

D. Scalability Studies

Further, Fig. 8 and Fig. 9 shows the comparison of different methods concerning computation time on the food nutrition dataset and weather sentiment dataset, respectively. We omit the results on the biography dataset as they are similar to those on the food nutrition dataset. The results show that our

TABLE III: Performance of our framework using different confidence computation methods.

Strategy	Bird recognition	Weather sentiment	City population		Food nutrition	
	Accuracy	Accuracy	MAE	RMSE	MAE	RMSE
Worker popularity-based	0.74	0.82	1,621	8,748	307	401
Task popularity-based	0.74	0.82	1,610	8,699	265	352
Two-sided popularity-based	0.74	0.85	1,548	8,336	252	308
HITS-based method	**0.74**	**0.87**	**1,489**	**8,018**	**163**	**227**

TABLE IV: Performance of our framework under different progressive strategies.

Strategy	Bird recognition	Weather sentiment	City population		Food nutrition	
	Accuracy	Accuracy	MAE	RMSE	MAE	RMSE
Non-progressive (traditional)	0.74	0.82	1,633	8,823	315	412
Random seeding	0.74	0.82	16,766	14,682	423	928
Low-confidence seeding	0.74	0.77	16,473	14,398	437	940
High-confidence seeding	**0.74**	**0.87**	**1,489**	**8,018**	**163**	**227**

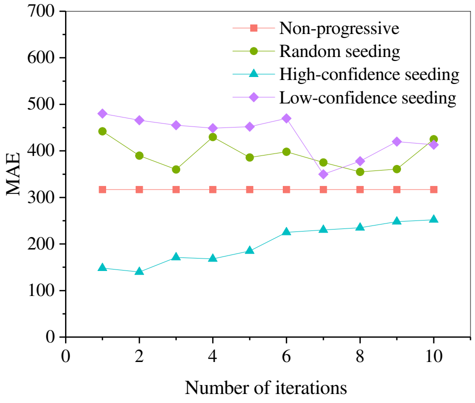


Fig. 6: Performance under different numbers of iterations

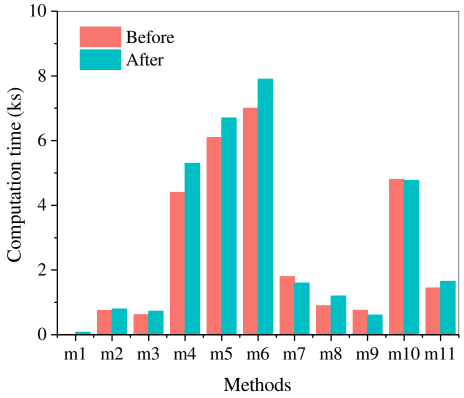


Fig. 8: Computation time on food nutrition dataset

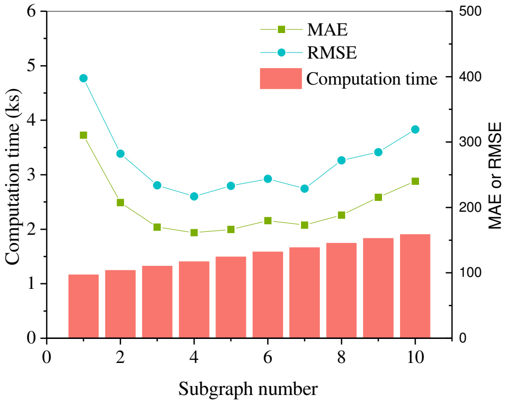
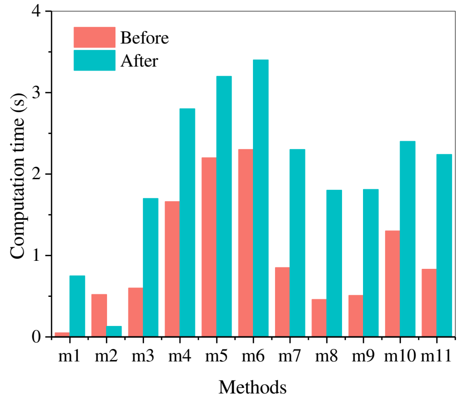


Fig. 7: Performance under different subgraph number k



0.4

Fig. 9: Computation time on weather sentiment dataset

approach may take additional time to complete, but the effect is specific to the datasets. Another observation is that the time increase is less evident for larger datasets. The reason may lie in that the amount of extra time caused by our approach is relatively stable. Thus the additional time cost immediately becomes marginal when the original methods (i.e., the methods without adopting our framework) themselves require a considerable amount of time by themselves. Indeed, for both the biography and food nutrition datasets, all progressive versions of the methods incurred similar or only slightly longer time than their original implementations. Some even incurred less time than their original versions on both datasets. This may imply that our proposed approach not only improves the accuracy but also, in many cases, preserves the efficiency of methods if configured properly.

V. Conclusion

In this paper, we have proposed a general report aggregation framework, which employs confidence propagation and a progressive approach to derive more accurate results. To the best of our knowledge, this is the first work that recognizes the interaction among the confidence of workers and tasks in crowdsourcing scenarios; it is also the first to progressively take into account more confident workers and tasks to improve the accuracy for those less confident workers and tasks. The proposed approach can mitigate the uncertainty brought by data sparsity and continuously improve the priors of existing methods. It is generic enough to be applied to various existing methods. Experiments on four real-world datasets demonstrate the effectiveness of our approach. Our future work would involve developing more sophisticated implementations of our approach and evaluating the framework with more datasets.

References

[1] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proc. of the 16th ACM Conference on Computer Supported Cooperative work*, 2013, pp. 1301–1318.

[2] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[3] R. Lenart-Gansiniec, "The benefits of crowdsourcing in science: Systematic literature review," in *European Conference on Research Methodology for Business and Management Studies*. Academic Conferences International Limited, 2021, pp. 130–X.

[4] S. S. Bhatti, X. Gao, and G. Chen, "General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey," *Journal of Systems and Software*, vol. 167, p. 110611, 2020.

[5] D. Hettiachchi, N. Van Berkel, V. Kostakos, and J. Goncalves, "Crowdcog: A cognitive skill based system for heterogeneous task assignment and recommendation in crowdsourcing," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW2, pp. 1–22, 2020.

[6] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: is the problem solved?" *Proc. of the VLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2012.

[7] L. Berti-Equille and M. L. Ba, "Veracity of big data: challenges of cross-modal truth discovery," *Journal of Data and Information Quality*, vol. 7, no. 3, p. Paper No. 12, 2016.

[8] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios, "Challenges in data crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 901–911, 2016.

[9] J. Johnson, "U.s. common insider threat types 2020," Jan 2021. [Online]. Available: https://www.statista.com/statistics/1155585/most-common-insider-threat-types-united-states/

[10] X. Zhang, X. Chen, H. Yan, and Y. Xiang, "Privacy-preserving and verifiable online crowdsourcing with worker updates," *Information Sciences*, vol. 548, pp. 212–232, 2021.

[11] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proc. of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.

[12] H. Xiao, J. Gao, Q. Li, F. Ma, L. Su, Y. Feng, and A. Zhang, "Towards confidence in the truth: a bootstrapping based truth discovery approach," in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1935–1944.

[13] X. Wang, Q. Z. Sheng, L. Yao, X. Li, X. S. Fang, X. Xu, and B. Benatallah, "Empowering truth discovery with multi-truth prediction," in *Proc. of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 881–890.

[14] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han, "Faitcrowd: fine grained truth discovery for crowdsourced data aggregation," in *Proc. of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 745–754.

[15] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 576–587.

[16] C. Castillo, M. Mendoza, and B. Poblete, "Predicting information credibility in time-sensitive social media," *Internet Research*, vol. 23, no. 5, pp. 560–588, 2013.

[17] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[18] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *Proc. of the 23rd International Conference on Computational Linguistics*, 2010, pp. 877–885.

[19] A. Galland, S. Abiteboul, A. Marian, and P. Senellart, "Corroborating information from disagreeing views," in *Proc. of the 3rd ACM International Conference on Web Search and Data Mining*, 2010, pp. 131–140.

[20] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[21] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2014, pp. 1187–1198.

[22] N. Sabetpour, A. Kulkarni, and Q. Li, "Optsla: an optimization-based approach for sequential label aggregation," *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[23] F. Yang, K. Lu, M. Li, S. Chen, Y. Chen, M. Guizani, and W. Hu, "Text data truth discovery using self-confidence of sources," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 131–136.

[24] N. Sabetpour, A. Kulkarni, S. Xie, and Q. Li, "Truth discovery in sequence labels from crowds," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 539–548.

[25] J. Pasternack and D. Roth, "Latent credibility analysis," in *Proc. of the 22nd International World Wide Web Conference*, 2013, pp. 1009–1020.

[26] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. of the 11nd ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2012, pp. 233–244.

[27] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence," *Proc. of the VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009.

[28] A. R. Kurup, G. Sajeev, and J. Swaminathan, "Aggregating reliable submissions in crowdsourcing systems," *IEEE Access*, vol. 9, pp. 153 058–153 071, 2021.

[29] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," in *Proc. of the 10th International Workshop on Quality in DataBases, coheld with VLDB*, 2012.

[30] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proc. of the VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2012.

[31] T. Han, H. Sun, Y. Song, Y. Fang, and X. Liu, "Find truth in the hands of the few: acquiring specific knowledge with crowdsourcing," *Frontiers of Computer Science*, vol. 15, no. 4, pp. 1–12, 2021.

[32] H. Zhang, Q. Li, F. Ma, H. Xiao, Y. Li, J. Gao, and L. Su, "Influence-aware truth discovery," in *Proc. of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 851–860.

[33] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 3, pp. 683–690, 1991.

[34] D. J. Cook and L. B. Holder, *Mining graph data*. Bridgewater Township, USA: John Wiley & Sons, 2006.

[35] W. Peter, "Bluebirds recognition dataset," https://github.com/welinder/cubam/tree/public/demo/bluebirds/, last accessed: 30 September 2021, 2 2011.

[36] M. Venanzi, W. Teacy, A. Rogers, and N. Jennings, "Weather sentiment—amazon mechanical turk dataset," https://eprints.soton.ac.uk/376543/, last accessed: 29 September 2021., 4 2015.

[37] J. Pasternack and D. Roth, "City population dataset," https://github.com/daz45/population---truth-discovery, last accessed: 11 August 2021, 4 2019.