A Lightweight, Effective and Efficient Model for Label Aggregation in Crowdsourcing

YI YANG, School of Computer Science and Information Engineering, Hefei University of Technology, China
ZHONG-QIU ZHAO*, School of Computer Science and Information Engineering, Hefei University of Technology, China
GONGQING WU, School of Computer Science and Information Engineering, Hefei University of Technology, China
XINGRUI ZHUO, School of Computer Science and Information Engineering, Hefei University of Technology, China
QING LIU, Data61, Commonwealth Scientific and Industrial Research Organisation, Australia
QUAN BAI, School of Technology, Environments and Design, University of Tasmania, Australia
WEIHUA LI, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, New Zealand

Due to the presence of noise in crowdsourced labels, label aggregation (LA) has become a standard procedure for post-processing these labels. LA methods estimate true labels from crowdsourced labels by modeling worker quality. However, most existing LA methods are iterative in nature. They require multiple passes through all crowdsourced labels, jointly and iteratively updating true labels and worker qualities until a termination condition is met. As a result, these methods are burdened with high space and time complexities, which restrict their applicability in scenarios where scalability and online aggregation are essential. Furthermore, defining a suitable termination condition for iterative algorithms can be challenging. In this paper, we view LA as a dynamic system and represent it as a Dynamic Bayesian Network. From this dynamic model, we derive two lightweight and scalable algorithms: $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. These algorithms can efficiently and effectively estimate worker qualities and true labels by traversing all labels at most twice, thereby eliminating the need for explicit termination conditions and multiple traversals over the crowdsourced labels. Due to their dynamic nature, the proposed algorithms are also capable of performing label aggregation online. We provide theoretical proof of the convergence property of the proposed algorithms and bound the error of the estimated worker qualities. Furthermore, we analyze the space and time complexities of our proposed algorithms, demonstrating their equivalence to those of majority voting. Through experiments conducted on 20 real-world datasets, we demonstrate that our proposed algorithms can effectively and efficiently

*Corresponding author

Authors' addresses: Yi Yang, yyang@hfut.edu.cn, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China; Zhong-qiu Zhao, z.zhao@hfut.edu.cn, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China; Gongqing Wu, wugq@hfut.edu.cn, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China; Xingrui Zhuo, zxr@mail.hfut.edu.cn, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China; Qing Liu, q.liu@data61.csiro.au, Data61, Commonwealth Scientific and Industrial Research Organisation, Hobart, Australia; Quan Bai, Quan.Bai@utas.edu.au, School of Technology, Environments and Design, University of Tasmania, Hobart, Australia; Weihua Li, weihua.li@aut.ac.nz, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand.

1556-4681/2023/10-ART $15.00
https://doi.org/10.1145/3630102

aggregate labels in both offline and online settings, even though they traverse all labels at most twice. The code is on https://github.com/yyang318/LA_onepass.

CCS Concepts: • **Information systems** → **Crowdsourcing**; • **Computing methodologies** → *Unsupervised learning*; • **Mathematics of computing** → Bayesian networks.

Additional Key Words and Phrases: crowdsourcing, truth inference, online truth inference, Bayesian networks

1 INTRODUCTION

Machine learning tasks frequently necessitate training labels. Traditional methods of label collection, such as seeking assistance from domain experts or data vendors, are often costly and time-intensive, thus failing to meet the burgeoning demand for labels. In contrast, crowdsourcing emerges as a cost-effective and efficient method for label acquisition [1, 3, 5, 20]. Crowdsourcing platforms, such as Amazon's Mechanical Turk[1] [21] and FigureEight [10], provide a platform for label requesters to delegate labeling tasks to crowd workers. These crowd workers, in return for a monetary reward, undertake the task of labeling. Nevertheless, despite the low-cost of crowdsourcing, the labels acquired may be prone to errors as crowd workers, unlike experts, may label tasks incorrectly [25]. This results in crowdsourced labels typically being less accurate than those sourced from experts. To mitigate these inaccuracies, it is common to gather multiple labels for each task from various workers, and subsequently aggregate these labels [23].

Crowdsourced label aggregation (LA) is a process that aggregates labels from a crowd of workers and estimates the true label for each task. LA is also referred to as truth inference in crowdsourcing [62] or truth discovery in the database community [33]. LA is typically unsupervised due to the unavailability of ground-truth labels for supervision. The simplest form of LA, Majority Voting (MV) [15], assumes that the label supported by the most workers is the true label for each task. While this approach is straightforward and comes with low space and time complexities — $O(M+T)$ and $O(TM)$, respectively, where M is the number of workers and T is the number of tasks, it fails to consider the varying reliability of workers in a realistic crowdsourcing scenario [23].

In order to address the limitation of MV, recent LA methods model worker qualities during the label aggregation process. The underlying principle is that labels from high-quality workers are more likely to be accurate, thus these labels should carry more weight in determining the true label for each task. According to this principle, these LA algorithms jointly and iteratively estimate true labels and worker qualities until a certain convergence condition is satisfied [9, 27, 37, 57]. Empirical results have demonstrated the superiority of such LA methods over MV in terms of accuracy [62].

However, despite their advantages, the iterative nature of most LA methods introduces two primary limitations. Firstly, these methods need to load the entire dataset of crowdsourced labels into memory, resulting in a space complexity of at least $O(TM)$. Secondly, they require multiple iterations over the entire dataset, leading to a time complexity of at least $O(ITM)$, where I represents the number of iterations necessary for the algorithm to converge. These limitations highlight the necessity for more efficient and scalable LA methods.

The limitations of iterative LA methods pose significant challenges for their practical applications:

Scalability. Crowdsourced datasets can be considerably large. For instance, the ImageNet project employed crowdsourced workers from Amazon Mechanical Turk to label and verify over 11 million images [29]. In our experiments, we found that a recently developed iterative LA method EBCC [37], which has the second-best overall mean accuracy, took about 4 hours to train a dataset (*senti*) with over 500K labels, while MV completed in less than a second. Additionally, iterative LA methods like LAA [40] and TiReMGE [54] need to load all the crowdsourced labels in memory, leading to memory exhaustion for large datasets. Hence, the development of LA methods with low space and time complexities is highly desirable for scalability.

[1] https://www.mturk.com/

Online Aggregation. Many crowdsourcing projects are continuous and can last for years. For example, the eBird project[2], started in 2002, crowdsources bird populations and species globally. With around 225 million observations reported in 2022 alone [13], the accumulation of data challenges not only the scalability of LA methods but also their ability to perform online label aggregation, i.e., LA can be executed continuously on the latest data subset without loading the complete dataset, and aggregated labels can be discarded due to limited storage or privacy concerns [8].

Termination Condition. Iterative algorithms require a termination condition to stop. However, setting this condition is non-trivial. A loose condition may lead to non-convergence, hurting LA performance, whereas a strict condition may result in unnecessary computations without improving LA's performance.

Several attempts have been made to address the aforementioned limitations. In order to improve scalability, methods such as IWMV [30] and BWA [34] employ simplified, computationally efficient procedures for the estimation of worker qualities and true labels. These methods have managed to enhance the efficiency of LA algorithms to some degree. However, being iterative in nature, these methods still necessitate the loading of all crowdsourced labels into memory to perform label aggregation. More recent developments include SBIC [42] and BiLA [19], which were created specifically for online label aggregation. SBIC, however, is restricted to decision-making tasks with only two classes. On the other hand, BiLA, being a neural network-based method, demands substantial resources for label aggregation, thereby diminishing its scalability and efficiency. While a small number of online aggregation methods, such as SBIC [42] and iCRH [36] do not require explicit termination conditions, SBIC is constrained by its task limitations, and iCRH lacks theoretical assurances for worker quality convergence. Thus, there is an immediate need for LA algorithms that are scalable, facilitate online aggregation, do not require explicit termination conditions, and provide robust theoretical guarantees.

The identified need inspires us to design an effective and efficient LA algorithm, named $\text{LA}^{\text{onepass}}$. This algorithm is scalable, capable of online label aggregation, and does not require a termination condition. This algorithm addresses the three aforementioned limitations concurrently. Specifically, we assign each label a time-slice, representing the index of the label's task, giving both labels and tasks temporal attributes. We then view LA as a dynamic system, with worker qualities evolving over time following the estimation of true labels. To model this dynamic system, we utilize the Dynamic Bayesian network [28]. Worker qualities are treated as (unknown) temporal variables evolving over time, while the (observed) crowdsourced labels and (unknown) true labels of each task are modeled as non-temporal variables instantiated within one time-slice. During the estimation of unknown variables at each time-slice, worker qualities can be efficiently estimated by Maximum A Posterior (MAP), and the true label can be estimated by analytically solving a straightforward optimization problem. This allows crowdsourced labels to be traversed only once, reducing both the space and time complexities to $O(M+T)$ and $O(TM)$, respectively, matching the complexities of MV. We also provide proof of convergence for the estimated worker quality and a rate of convergence. Importantly, even when the crowdsourced labels are traversed only once, the error of the estimated worker quality can be bounded with high probability. Unlike iterative algorithms, $\text{LA}^{\text{onepass}}$ terminates after all crowdsourced labels have been traversed once, removing the need to set an explicit termination condition.

However, a drawback of the single traversal is that the true labels estimated earlier might not be accurate since the worker quality estimates have not yet converged. To mitigate this issue, we develop $\text{LA}^{\text{twopass}}$, an extension of $\text{LA}^{\text{onepass}}$. It uses the (converged) worker quality estimates from $\text{LA}^{\text{onepass}}$ to re-estimate the true labels by performing a weighted majority vote. Though $\text{LA}^{\text{twopass}}$ traverses the crowdsourced labels twice, it can generally improve aggregation accuracy. The overhead for $\text{LA}^{\text{twopass}}$ is minimal compared to $\text{LA}^{\text{onepass}}$, as it does not re-estimate worker qualities during the second pass through the labels. Owing to their low space and time complexities and the dynamic system's nature, both $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ can be configured for

[2]www.ebird.org/

online label aggregation without any algorithmic modifications. Benefiting from the advantages conferred by the dynamic system, $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ are distinguished as lightweight label aggregation algorithms, because they exhibit the following shared characteristics:

- Efficiency: $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ require only one or two passes over the entire crowdsourced labels, respectively, ensuring efficient label aggregation. They eliminate the need for multiple repetitive iterations, often required by other iterative LA algorithms, thereby simplifying the overall computational process.
- Online Label Aggregation: The ability to handle online label aggregation without requiring algorithmic modifications further emphasizes the lightweight characteristics of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. This allows these algorithms to adapt to real-time changes and accommodate new crowdsourced labels seamlessly.
- Scalability: The design of the proposed dynamic system enables $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ to handle large datasets. Their efficiency does not degrade significantly with the increase in data volume, demonstrating scalability - a critical feature of lightweight and online algorithms.
- Lower Time and Space Complexities: The time and space complexities of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ are on par with Majority Voting, one of the simplest and lightest label aggregation methods. This further solidifies the lightweight property of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$.

To summarize, we make the following contributions in this paper:

(1) We propose viewing crowdsourced label aggregation as a dynamic system, with task identifiers serving as time-slices. Using a Dynamic Bayesian Network to model this system, we develop two label aggregation algorithms, $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. These algorithms traverse all the labels once and twice, respectively. Importantly, both $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ aggregate labels without the need for explicit termination conditions.
(2) We prove that the estimated worker qualities in $\text{LA}^{\text{onepass}}$ converge at a rate of $1/\sqrt{t}$, where t is the number of tasks for which true labels have been estimated. Furthermore, we show that the error in the estimated worker quality can be bounded with a higher probability as t increases.
(3) We perform an analysis of the space and time complexities of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. Our findings reveal that these complexities are equal to those of the Majority Voting (MV) method and considerably lower than those of iterative techniques.
(4) Owing to the dynamic nature of our system model, we show that $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ can perform label aggregation online without any additional algorithmic modifications.
(5) Extensive experiments are conducted on 20 real-world datasets to demonstrate the efficiency and effectiveness of our proposed algorithms. When compared with state-of-the-art label aggregation (LA) methods in both offline and online scenarios, our methods not only achieve comparable accuracy but also exhibit superior efficiency. This confirms their practical applicability and effectiveness.

2 PROBLEM STATEMENT & RELATED WORK

In this section, we begin by formally defining the problem of Label Aggregation (LA). Subsequently, we review related works relevant to this topic.

2.1 Problem Statement of Label Aggregation

In this paper, we consider a scenario where there are T tasks and M workers. Each task t has K mutually exclusive classes, indexed from 1 to K, and its unknown true label y_t is drawn from the set $[K]$, which represents the integers $1, \ldots, K$. Each task is labeled by the workers, and the label from worker $i \in [M]$ for task $t \in [T]$ is denoted as $x_{i,t}$, where $x_{i,t} \in [K]$. Each worker i is associated with an unknown quality variable w_i, representing the reliability of the worker's labels. For ease of reference, we denote the complete set of crowdsourced labels as $\mathcal{X} = \{x_{i,t} | i \in [M], t \in [T]\}$.

The primary objective of this study is to aggregate $\mathcal{X}$ to estimate the true labels $\mathcal{Y} = \{\hat{y}_t | t \in [T]\}$ for each task as well as the worker qualities $\mathcal{W} = \{\hat{w}_i | i \in [M]\}$. For the sake of simplicity, we assume that each worker labels all tasks. However, our proposed algorithms can also accommodate situations where each worker labels only a subset of tasks, effectively dealing with label sparsity.

It is important to highlight that this paper specifically concentrates on the single-truth LA problem, where each task has only one true label. This stands in contrast to multi-truth LA methods, such as MCMLD [60]. Furthermore, our method is universal and solely relies on crowdsourced labels as input. This distinguishes our method from other LA methods that incorporate features of workers or tasks into their inputs. For instance, ART [57] integrates workers' social networks into their aggregation model, and CLA [39] utilizes side information to group tasks. However, obtaining such features can be challenging in crowdsourcing scenarios and may not always be available in every crowdsourcing application. Additionally, we adhere to the traditional label aggregation setting, where labels are aggregated locally on a single machine, distinguishing our approach from the setting where labels can be aggregated in a decentralized manner [16]. The setting of the label aggregation problem studied in this paper is summarized in comparison with other methods in Table 1.

Table 1. Label aggregation methods' seetings and assumptions comparison

Method	Number of true labels	Input to the method	Working environment
MCMLD [60]	Multiple	Labels	Local
ART [57]	Single	Labels & social network structure	Local
CLA [39]	Single	Labels & task features	Local
Decentralized [16]	Single	Labels & decentralized source network	Decentralized
Ours	Single	Labels	Local

2.2 Related Work

Crowdsourcing, a problem-solving approach leveraging collective intelligence [12], has demonstrated significant advantages in addressing many challenges that are intuitively understood by humans yet difficult for computers to solve. These challenges span fields such as transportation [46], task assignment [48], recommender systems [47], pandemic prevention [7], among others. In this paper, our focus is on scenarios wherein cost-effective crowdsourced labels have been collected, necessitating the label aggregation algorithms to estimate the true labels for each crowdsourced task. Therefore, in this section, we review prior work related to our study in the field of label aggregation.

2.2.1 Worker modeling. Active research works of LA focus on how to model worker qualities. The first LA method considering worker quality can be dated back to 1979 when Dawid and Skene proposed an algorithm, commonly known as DS, to aggregate clinical diagnoses of doctors [9]. DS is classified as the "confusion matrix" model in the literature, because it uses a $K \times K$ confusion matrix to capture the probabilities that a worker's label is generated conditioned on the task's true label. A large number of LA methods are descendants of DS [19, 27, 61]. For example, LFC [44] extends DS by adding priors to confusion matrices. EBCC [37] clusters the tasks when estimating the confusion matrices.

Another commonly adopted worker quality model is "one-coin" model. One-coin model treats the quality of each worker as a single parameter reflecting the quality of worker's labels. For example, ZC [11] models worker quality as a value between 0 and 1, representing the probability of a worker's label being correct. IWMV [30] transforms such probability for estimating true labels, having a provable theoretical guarantee on the error rate. There are also some one-coin models treating worker quality as a real number, where a higher value means the worker's labels are more likely to be true [32, 34].

Beyond modeling worker qualities and true labels, some works explore other characteristics of label aggregation. For instance, Li et al. [38] propose three fairness-enhancing methods: Pre-TD, Post-TD, and FairTD. These methods estimate worker qualities and biases iteratively and selectively remove bias from the labels during label aggregation, striking an optimal balance between fairness and accuracy. Jiang et al. [22] propose a two-stage incentive mechanism for crowdsourcing to incentivize workers to discover the true labels while addressing the issue of copied labels, and introduce a label aggregation algorithm and a reverse auction mechanism to ensure high accuracy and maximize social welfare.

2.2.2 Modeling techniques and solution framework. The Probabilistic Graphical Model (PGM) [28] is the most prevalent technique employed to solve LA problems [18, 23, 27, 34, 51]. A PGM illustrates the conditional dependencies between random variables. Most PGMs utilized in LA are generative, modeling the conditional probability of a worker's label given the unknown worker quality and true label.

Techniques other than PGMs are also in use. The optimization-based method [32, 45] directly constructs an objective function that encapsulates the relationship between worker qualities and true labels. Ma et al. [41] model the crowdsourced labels as a matrix and estimate worker qualities and true labels via matrix completion. Methods employing neural networks, such as LAA [40], are also used in LA, modeling the non-linear relationship among crowd labels, worker qualities, and true labels.

Recently, several methods have emerged that learn features to aid label aggregation. TiReMGE [54] learns the features of workers and tasks from the task-worker interaction graph using a graph neural network and uses the learned features to guide the estimation of worker qualities. TIRA [55] utilizes a hierarchical graph auto-encoder and a vector initialization method based on worker reliabilities for label aggregation in crowdsourcing. TILCC [53] uses CRH [32] to extract task features and subsequently applies these features to aggregate labels based on clustering.

Regardless of the modeling techniques used, most LA algorithms inherently follow an iterative process [62]. Xiao and Wang also argue that most existing label aggregation algorithms can be unified under one iterative framework. They provide evidence that the unknown variables - worker qualities, and true labels converge if these variables are estimated iteratively [56]. It is important to note that the iterative LA algorithms estimate true labels and worker qualities by traversing the entire dataset multiple times, until a specific termination condition is met. This iterative nature constitutes the primary bottleneck in enhancing the time and space complexities of iterative LA algorithms.

Online LA. Online LA has been explored in the database community, specifically in scenarios where labels are continuous and passively collected from data sources such as the web. For instance, Li et al. delve into the temporal relationships among true labels and web reliability, proposing an incremental truth inference framework capable of dynamically updating true labels and web reliabilities [35]. Yang et al. develop a PGM-based truth inference mechanism capable of incrementally estimating true labels, taking into account object correlations [58]. Yang et al. propose a method to dynamically update web reliability during the estimation of true labels over data streams, enhancing the efficiency of truth inference algorithms [58]. More recently, Pang et al. [43] and Wang et al. [50] independently develop privacy-preserving truth inference methods for protecting individual privacy while estimating true labels over data streams.

The above-mentioned online LA methods primarily focus on the aggregation of numerical labels. However, in the context of crowdsourcing applications, labels are typically categorical. Feng et al. [14] propose the INQUIRE framework that incrementally updates its internal probabilistic worker and task models. However, their heuristic combination of worker and task models does not guarantee the convergence of estimated worker qualities, thereby impacting its overall performance. Li et al. [36] develop iCRH, an incremental version of CRH [32], for incremental label aggregation. Nevertheless, iCRH also does not guarantee the convergence of its estimated worker qualities. More recent works include SBIC [42] and BiLA [19], both developed for online label aggregation. SBIC leverages

streaming variational inference [4] to update worker qualities and true labels and, like our proposed method, it can aggregate labels online by traversing labels only once. However, SBIC is limited to the aggregation of labels of tasks with only two classes. BiLA utilizes neural networks to represent its internal probability distributions but requires multiple traversals of the present labels to update worker qualities, which is proven to be inefficient in our experiments. In contrast, the algorithms we propose in this paper can aggregate labels of tasks with any number of classes and require at most two traversals of all the crowdsourced labels, providing an accurate and efficient means to aggregate labels online.

3 METHOD

In this section, we describe the proposed Dynamic Bayesian network (DBN) model for LA, and present LA$^{\text{onepass}}$ algorithm derived from the proposed dynamic model.

3.1 The Dynamic Model

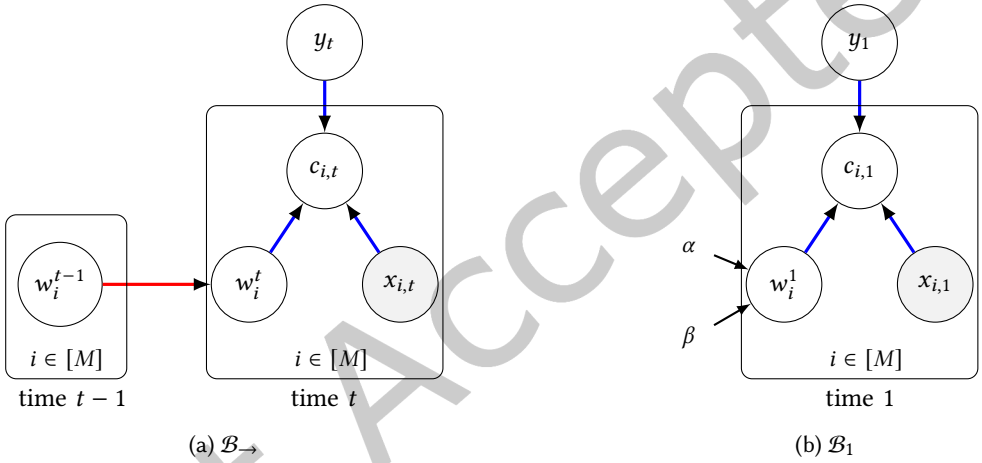


Fig. 1. Proposed DBN model. The intra-time-slice edge is in red; the inter-time-slice edge is in blue.

We consider LA as a dynamic system that evolves over T time-slices, corresponding to the total number of tasks. Here, the index t serves a dual purpose: it signifies the index of the task t and also represents the time-slice or the state of the dynamic system. Consequently, T refers to the total number of tasks as well as the system's life span. At each time-slice $t \in [T]$, the system estimates the true label for task t. Each worker i has an associated quality score $w_i \in [0, 1]$, representing the likelihood of their label being true. This score evolves over time, getting updated after the estimation of y_t until the system concludes at time-slice T.

This dynamic system can be modeled using a Dynamic Bayesian Network (DBN), where w_i is considered as a temporal variable evolving over time. For the sake of convenience, we introduce a superscript to the worker quality w_i^t to denote its state at time-slice t. Meanwhile, $x_{i,t}$ and y_t are modeled as non-temporal variables, instantiated within their own time-slices. The DBN can be effectively described using two Bayesian networks: $\mathcal{B}_{\rightarrow}$ and $\mathcal{B}_1$ as in Fig. 1. $\mathcal{B}_{\rightarrow}$ is a 2-time-slice Bayesian network (2TBN) illustrating the relationship between variables within a single time-slice and the evolution of variables across two consecutive time-slices. On the other hand, $\mathcal{B}_1$ depicts the initial state of the system.

3.1.1 2TBN $\mathcal{B}_{\rightarrow}$. In the 2TBN $\mathcal{B}_{\rightarrow}$, there are two kinds of edges, namely, inter-time-slice and intra-time-slice edges connecting the variables. The inter-time-slice edges connect variables w_i^t, $x_{i,t}$, and y_t within time-slice t, expressing their relationship through an auxiliary deterministic variable $c_{i,t}$ that indicates whether worker i labeled task t correctly. Hence, $c_{i,t} = \mathbb{1}(x_{i,t} = y_t)$, where $\mathbb{1}(\cdot)$ is the indicator function. As the correctness of worker i's label is determined by $w_i^t \in [0, 1]$, we model $c_{i,t}$ as a Bernoulli random variable with parameter w_i^t:

$$c_{i,t} \sim Ber(w_i^t), \quad p(c_{i,t}|w_i^t) = (w_i^t)^{c_{i,t}}(1 - w_i^t)^{1-c_{i,t}}. \tag{1}$$

The intra-time-slice edge connects w_i^{t-1} and w_i^t, depicting the evolution of a worker's quality over time by specifying the transition probability $p(w_i^t|w_i^{t-1})$. We treat $p(w_i^t|w_i^{t-1})$ as the posterior distribution of w_i^{t-1} after observing the labels of task $t-1$ and estimating its true label. In other words, w_i^{t-1} is the prior of w_i^t.

3.1.2 Initial state $\mathcal{B}_1$. $\mathcal{B}_1$ expresses the initial state of the dynamic system. The relation of variables is the same as that of time-slice t in $\mathcal{B}_{\rightarrow}$, except that it needs to specify the initial state of temporal variable w_i^1. Given that w_i is the probability of worker i labeling tasks correctly, we model w_i as a Beta random variable, and its initial state w_i^1 corresponds to a Beta distribution with hyperparameters α and β:

$$w_i^1 \sim Beta(\alpha, \beta), \quad p(w_i^1) \propto (w_i^1)^{\alpha-1}(1 - w_i^1)^{\beta-1}. \tag{2}$$

3.2 Estimation

Given the model architecture, the joint probability of $\mathcal{W}_t = \{w_i^t | i \in [M]\}$, $C_t = \{c_{i,t} | i \in [M]\}$ and $\mathcal{X}_t = \{x_{i,t} | i \in [M]\}$ within time-slice t can be factorized as

$$p(\mathcal{W}_t, C_t, \mathcal{X}_t, y_t) = \prod_i p(c_{i,t}|w_i^t, x_{i,t}, y_t) = \prod_i (w_i^t)^{c_{i,t}}(1 - w_i^t)^{1-c_{i,t}}, \tag{3}$$

and its log-likelihood function $l_t = \log p(\mathcal{W}_t, C_t, \mathcal{X}_t, y_t)$ is

$$\begin{aligned} l_t &= \sum_{i=1}^{M} c_{i,t} \log w_i^t + (1 - c_{i,t}) \log(1 - w_i^t) \\ &= \sum_{i=1}^{M} \mathbb{1}(x_{i,t} = y_t) \log w_i^t + (1 - \mathbb{1}(x_{i,t} = y_t)) \log(1 - w_i^t). \end{aligned} \tag{4}$$

We estimate the true label y_t by maximizing l_t, which can be easily solved via:

$$\hat{y}_t = \arg\max_k \{\sum_{i=1}^{M} w_i^t \mathbb{1}(x_{i,t} = k) | k \in [K]\}. \tag{5}$$

Provided that w_i^{t-1} is the prior of w_i^t, and the DBN in Fig. 1 is a first-order Markov model, the posterior probability of w_i^t after observing $x_{i,t}$ and estimating y_t is

$$p(w_i^t|c_{i,t}, x_{i,t}, y_t, w_i^{t-1}) \propto p(c_{i,t}|w_i^t, x_{i,t}, y_t) p(w_i^t|w_i^{t-1}). \tag{6}$$

By the chain rule of probability, the above posterior probability can be expanded as:

$$\begin{aligned} p(w_i^t|c_{i,t}, w_i^{t-1}) &\propto \prod_{t'=1}^{t} p(c_{i,t'}|w_i^{t'}) p(w_i^1) \\ &= (w_i^t)^{C_{i,t}+\alpha-1}(1 - w_i^t)^{t-C_{i,t}+\beta-1}, \end{aligned} \tag{7}$$

where $C_{i,t} = \sum_{t'=1}^{t} \mathbb{1}(x_{i,t'} = \hat{y}_{t'})$ is the number of tasks worker i has labeled correctly up to time-slice t, and $t - C_{i,t}$ is the number of tasks worker i has labeled incorrectly up to time-slice t. The variables $C_{i,t}$ and t encapsulate

Algorithm 1 $\text{LA}^{\text{onepass}}$

Input: Crowdsourced labels $\mathcal{X}$, hyperparameters α and β
Output: Estimated true labels $\mathcal{Y}$ and estimated worker qualities $\mathcal{W}$

1: Initialize qualities $\mathcal{W} = \{w_i | i \in [M]\}$ by α and β.
2: **for** $t \in [T]$ **do**
3: Estimate true label $\hat{y}_t$ by Equation (5);
4: **for** $i \in [M]$ **do**
5: Update worker quality $\hat{w}_i^t$ by Equation (8);
6: **end for**
7: **end for**
8: **return** $\mathcal{Y} = \{\hat{y}_t | t \in [T]\}, \mathcal{W} = \{\hat{w}_i^T | i \in [M]\}$.

all the necessary information about the labeling history of worker i, and they are sufficient statistics for the worker quality w_i^t. Therefore, to estimate worker qualities at time t and the true label of task t, it's not necessary to store all the historical crowdsourced labels before time t. Instead, we just need to keep track of $C_{i,t}$ and t, which significantly reduces the memory requirements of the algorithm. This efficient use of information is a key advantage of the $\text{LA}^{\text{onepass}}$ approach.

It can be observed that the posterior $p(w_i^t | c_{i,t}, w_i^{t-1})$ can be compactly written as $p(w_i^t | C_{i,t})$. $p(w_i^t | C_{i,t}) \sim Beta(C_{i,t} + \alpha, t - C_{i,t} + \beta)$ is again a Beta distribution. Therefore, we can estimate w_i^t by Maximum a Posteriori (MAP):

$$\hat{w}_i^t = \frac{C_{i,t} + \alpha - 1}{t + \alpha + \beta - 2}. \tag{8}$$

The form of Equation (8) aligns with our expectations, encapsulating the estimated likelihood of worker i accurately labeling tasks up to the time-slice t. This estimation is informed by prior beliefs denoted by α and β.

3.3 Algorithm Summary

The LA algorithm derived from the proposed Dynamic Bayesian Network (DBN) is summarized in Algorithm 1. The algorithm commences with the initialization of worker qualities using the hyperparameters α and β (Line 1). Further details regarding the initialization process will be discussed in Section 3.4. Once the worker qualities are initialized, the algorithm proceeds to sequentially aggregate the crowdsourced labels (Lines 2-7). For each task t, given the available crowdsourced labels, it estimates the true label $\hat{y}_t$ (Line 3). Subsequently, based on the estimated true label, the algorithm updates the qualities of the workers who provided labels for task t (Lines 4-6). The algorithm terminates when the true labels of all tasks are estimated, i.e., the execution of the for loop in between Lines 2-7 is completed. In the end, the estimated true labels for each task and the estimated worker qualities are returned. As the algorithm processes all the crowdsourced labels only once, we refer to it as $\text{LA}^{\text{onepass}}$.

3.4 Hyperparameter Settings

Iterative methods like EBCC [37] and BWA [34] traditionally determine their hyperparameters by referencing the results derived from Majority Voting (MV). This implies that their hyperparameters depend on the specific dataset under consideration. However, because $\text{LA}^{\text{onepass}}$ processes the entire dataset in one pass, using MV to initialize the hyperparameters α and β is unfeasible. In this section, we will propose strategies for setting the hyperparameters of our method, based on the commonly observed long-tail phenomenon in crowdsourcing.

The long-tail phenomenon in crowdsourcing refers to a situation where a significant proportion of workers only contribute labels for a few tasks [31]. In such circumstances, it is beneficial to put more faith in workers

who have labeled a larger number of tasks, as their worker quality estimates have a higher level of statistical confidence [31]. However, it is important to remember that the worker quality estimate $\hat{w}_i^t$ given in Equation (8) is a point estimate and does not reflect the confidence associated with the estimate. Moreover, since $\text{LA}^{\text{onepass}}$ processes all labels in one pass, it does not have information about the total number of tasks each worker has labeled at the start of the algorithm.

To circumvent this issue, we propose a pessimistic approach in setting the hyperparameters α and β with low values. By choosing relatively low values for α and β, we ensure that the resultant worker quality estimate $\hat{w}_i^t$ is small at the initialization of the algorithm. Consequently, the true label estimate in Equation (5) behaves more like the Majority Voting (MV) method. As the algorithm processes more tasks, the worker quality estimate becomes more accurate. When a worker labels only a few tasks, the low value of $\hat{w}_i^t$ creates a bias in the prior. Conversely, if a worker labels a considerable number of tasks, the terms $C_{i,t}$ and t in Equation (8) dominate over the hyperparameters α and β in the worker quality estimate $\hat{w}_i^t$. This mechanism allows the algorithm to prioritize worker reliability based on the volume of tasks they have labeled effectively.

4 ANALYSES

In this section, we present theoretical proof to establish that (a) the estimated worker quality provided in Equation (8) converges, and (b) the error in the estimated worker quality can be bounded. Additionally, we conduct an analysis of the space and time complexities of $\text{LA}^{\text{onepass}}$ and compare them with those of iterative algorithms and MV. Lastly, we discuss the termination conditions for our algorithm.

4.1 Convergence of Estimated Worker Quality

We conduct an analysis on the convergence of the worker quality estimation provided in Equation (8), assuming that the majority of workers are honest and do not deliberately mislabel tasks. This assumption has been empirically verified in previous studies [17, 59] and is further supported by our experimental results, which indicate that the mean accuracy of the Majority Voting (MV) method exceeds 80%. By making this assumption, we ensure that Equation (5) accurately estimates the true labels with high confidence [26]. With this premise, we can establish the convergence of w_i through the following theorem.

Theorem 1. *Let $f_t(\mathcal{W})$ be the joint posterior probability of worker qualities at time-slice t, and $L_t(\mathcal{W}) \equiv \log f_t(\mathcal{W})$:*

$$L_t(\mathcal{W}) = \sum_{i=1}^{M} (C_{i,t} + \alpha - 1) \log w_i^t + (t - C_{i,t} + \beta - 1) \log(1 - w_i^t), \tag{9}$$

then $\mathcal{W} = \{\hat{w}_i^t | i \in [M]\}$ in Equation (8) converges to the minimizer $\mathcal{W}_t^* = \arg\min_{\mathcal{W}} L_t(\mathcal{W})$ at rate of $o(1/\sqrt{t})$.

Proof. We use Lemma 1 to prove the theorem.

Lemma 1. *Let $\{f_t(\mathcal{W}), t = 1, 2, \dots\}$ be a sequence of posterior probability density functions $p(\mathcal{W}|C_t)$ of random vectors defined on $[0, 1]^M$. Define $L_t(\mathcal{W}) \equiv \log f_t(\mathcal{W})$ as in Equation (9). Suppose for each t, there exists a strict local maximum, $\mathcal{W}_t^*$, of $L_t(\mathcal{W})$. Then the posterior distribution $p(\mathcal{W}|C_t)$ satisfies asymptotic normality:*

$$(-\nabla^2 L_t(\mathcal{W}_t^*))^{1/2}(\mathcal{W} - \mathcal{W}_t^*) \xrightarrow{d} N(0, 1) \text{ as } t \to \infty, \tag{10}$$

where $C_t = \sum_{i=1}^{M} C_{i,t}$ is the number of correctly labeled tasks by all workers up to time-slice t.

Proof. We use Theorem 2.1 in [6] to prove this lemma. Theorem 2.1 in [6] states if the following conditions (P1-2 and C1-3) are satisfied, the asymptotic normality property in Equation (10) holds.

P1. $\nabla \log p(\mathcal{W}_t^*|C_t) = 0$.

P2. $\Sigma_t \equiv \{-\nabla^2 \log p(\mathcal{W}_t^*|C_t)\}^{-1}$ is positive definite.

C1. "Steepness": as $t \to \infty$, $\sigma_t^2 \to 0$ where σ_t^2 is the largest eigenvalue of Σ_t.

C2. "Smoothness": for any $\epsilon > 0$, there exists an integer N and $\delta > 0$ such that, for any $t > N$, and $\mathcal{W}' \in H(\mathcal{W}_t^*; \delta) = \{|\mathcal{W}' - \mathcal{W}_t^*| < \delta\}$, $\nabla^2 \log p(\mathcal{W}'|C_t)|$ satisfies

$$I - A(\epsilon) \leq \nabla^2 \log p(\mathcal{W}'|C_t)|\{\nabla^2 \log p(\mathcal{W}_t^*|C_t)|\}^{-1} \leq I + A(\epsilon), \tag{11}$$

where I denotes the identity matrix with an appropriate size and $A(\epsilon)$ is the positive semi-definite symmetric matrix with the largest eigenvalue going to 0 as $\epsilon \to 0$.

C3. "Concentration": for any $\delta > 0$, $\int_{H(\mathcal{W};\delta)} p(\mathcal{W}|C_t)d\mathcal{W} \to 1$ as $t \to \infty$.

We will show the satisfaction of these conditions.

Proof of P1 and P2. Since $\mathcal{W}_t^*$ is a local maximum of L_t, the satisfaction of P1 is straightforward. The Hessian of L_t is

$$\nabla^2 L_t(\mathcal{W}_t^*) = diag\Big(- \frac{C_{i,t} + \alpha - 1}{(w_i^*)^2} - \frac{t - C_{i,t} + \beta - 1}{(1 - w_i^*)^2} \Big)_{i,} \tag{12}$$

where $C_{i,t}$ is the number of correctly labeled tasks by worker i up to t. It can be observed that $\nabla^2 L_t(\mathcal{W}_t^*)$ is negative definite because $\nabla^2 L_t(\mathcal{W}_t^*)$ is a diagonal matrix whose diagonal entries are negative given reasonable and small hyperparameters α and β. Therefore Σ_t is positive definite, and P2 satisfies.

Proof of C1. As $t \to \infty$, the diagonal entries of $\nabla^2 L_t(\mathcal{W}_t^*)$ approach $-\infty$. Hence the diagonal entries of Σ_t also approach 0. It implies all the eigenvalues of Σ_t go to 0 as $t \to \infty$. Therefore, C1 is satisfied.

Proof of C2. C2 is straightforward because all the entries in $\nabla^2 L_t(\mathcal{W})$ are continuous with respect to each w_i in its domain.

Proof of C3. By setting $\nabla L_t(\mathcal{W}) = 0$, we can easily find $(\mathcal{W}_t^*)_i$ has the form as given in Equation (8), which is the mode of a posterior distribution $Beta(C_{i,t} + \alpha, t - C_{i,t} + \beta)$. The variance of the posterior distribution is $\frac{(C_{i,t}+\alpha)(t-C_{i,t}+\beta)}{(t+\alpha+\beta)^2(t+\alpha+\beta+1)}$. Because $C_{i,t} \leq t$, the denominator of the variance dominates the numerator. Therefore the variance approaches 0 as $t \to \infty$. This means $E_{p(\mathcal{W}|C_t)}[\mathcal{W} - \mathcal{W}_t^*] \to 0$. Therefore C3 satisfies. □

The lemma shows the posterior distribution of worker quality converges as $t \to \infty$, and it converges to the minimizer $\mathcal{W}_t^*$ of $L_t(\mathcal{W})$.

From Lemma 1, we can take the expectation on the asymptotic distribution in Equation (10) and get

$$E[(-\nabla^2 L_t(\mathcal{W}_t^*))^{1/2}(\mathcal{W} - \mathcal{W}_t^*)] \to 0. \tag{13}$$

It implies

$$\left|E_{p(\mathcal{W}|C_t)}(\mathcal{W}) - \mathcal{W}_t^*\right| = o(1)\left|(-\nabla^2 L_t(\mathcal{W}_t^*))^{-1/2}\right|, \tag{14}$$

where $E_{p(\mathcal{W}|C_t)}(\mathcal{W})$ is the posterior mean of $\mathcal{W}$ at time-slice t. From Equation (12), we can see $-\nabla^2 L_t(\mathcal{W}_t^*) = \Theta(t)$. Therefore, $\left|E_{p(\mathcal{W}|C_t)}(\mathcal{W}) - \mathcal{W}_t^*\right| = o(1/\sqrt{t})$.

Moreover, the $\hat{w}_i^t$ given by Equation (8) is the mode of posterior distribution $p(w_i^t|C_t)$. Therefore we have

$$(\left|\hat{\mathcal{W}} - E_{p(\mathcal{W}|C_t)}(\mathcal{W})\right|)_i = \left|\hat{w}_i^t - E_{p(w_i|C_{i,t})}(w_i)\right| \tag{15}$$

Denote the two parameters of the posterior distribution $p(w_i|C_{i,t})$ as $\alpha_i = C_{i,t} + \alpha$ and $\beta_i = t - C_{i,t} + \beta$. The mode and mean of the posterior can be written as $\frac{\alpha_i - 1}{\alpha_i + \beta_i - 2}$ and $\frac{\alpha_i}{\alpha_i + \beta_i}$, respectively. Therefore, we can write out Equation

(15) as

$$\begin{aligned}\left(\left|\hat{\mathcal{W}} - E_{p(\mathcal{W}|C_t)}(\mathcal{W})\right|\right)_i &= \left|\frac{\alpha_i - 1}{\alpha_i + \beta_i - 2} - \frac{\alpha_i}{\alpha_i + \beta_i}\right| \\ &= \left|\frac{\alpha_i - \beta_i}{(\alpha_i + \beta_i - 2)(\alpha_i + \beta_i)}\right| \\ &\leq \left|\frac{1}{(\alpha_i + \beta_i - 2)}\right| = \left|\frac{1}{t + \alpha + \beta - 2}\right| = \Theta(1/t) = o(1/\sqrt{t}).\end{aligned} \tag{16}$$

Hence, $\left|\hat{\mathcal{W}} - E_{p(\mathcal{W}|C_t)}(\mathcal{W})\right| = o(1/\sqrt{t})$. By triangle inequality:

$$\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \left|\hat{\mathcal{W}} - E_{p(\mathcal{W}|C_t)}(\mathcal{W})\right| + \left|E_{p(\mathcal{W}|C_t)}(\mathcal{W}) - \mathcal{W}_t^*\right|. \tag{17}$$

We have shown that $\left|E_{p(\mathcal{W}|C_t)}(\mathcal{W}) - \mathcal{W}_t^*\right| = o(1/\sqrt{t})$ and $\left|\hat{\mathcal{W}} - E_{p(\mathcal{W}|C_t)}(\mathcal{W})\right| = o(1/\sqrt{t})$, so $\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| = o(1/\sqrt{t})$, which proves the theorem. □

This theorem demonstrates that the worker quality estimated by Equation (8) converges at a rate of $o(1/\sqrt{t})$, even when traversing all the labels only once. Additionally, the corollary below provides an upper bound on the error in the worker quality estimation.

Corollary 1. *$p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon/\sqrt{t}) \geq \Phi(\epsilon) - \Phi(-\epsilon)$ where ϵ is a positive real value, and $\Phi(\cdot)$ is the CDF of standard Normal distribution.*

Proof. From Equation (10) and given the fact that $\mathcal{W}_t^*$ is a vector of scalars, we can derive

$$\hat{\mathcal{W}} \xrightarrow{d} N(\mathcal{W}_t^*, \Sigma_t), \tag{18}$$

where $\Sigma_t \equiv \{-\nabla^2 \log p(\mathcal{W}_t^*|C_t)\}^{-1} = \{-\nabla^2 L_t(\mathcal{W}_t^*)\}^{-1}$ as defined in Lemma 1. By transformation of Normal distribution, we have

$$p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon\Sigma_t^{1/2}) = \Phi(\epsilon) - \Phi(-\epsilon), \tag{19}$$

where ϵ is a positive real number and $\Phi(\cdot)$ is the CDF of standard Normal distribution. Since $\Sigma_t^{1/2}$ is the standard deviation of the Normal distribution in Equation (18), $p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon\Sigma_t^{1/2})$ can be interpreted as *the probability that $\hat{\mathcal{W}}$ falls within ϵ standard deviations away from $\mathcal{W}_t^*$*.

From Equation (12), we have $\left(\nabla^2 L_t(\mathcal{W}_t^*)\right)_i \leq -t$, which implies that $(\Sigma_t)_i^{1/2} \leq 1/\sqrt{t}$. Therefore,

$$p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon\Sigma_t^{1/2}) \leq p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon/\sqrt{t}), \tag{20}$$

which implies $p(\left|\hat{\mathcal{W}} - \mathcal{W}_t^*\right| \leq \epsilon/\sqrt{t}) \geq \Phi(\epsilon) - \Phi(-\epsilon)$. □

Corollary 1 establishes that the error in the estimated worker quality can be more tightly bounded with a higher probability as the value of t increases. We will also empirically verify it in the experiment.

4.2 Space and Time Complexities

We analyze and compare the space and time complexities of $\text{LA}^{\text{onepass}}$, the iterative LA algorithms and Majority Voting (MV). The results of this comparison are summarized in Table 2.

Table 2. Space and time complexities comparison

	MV	$\text{LA}^{\text{onepass}}$	Iterative LA algorithms
Space Complexity	$O(M+T)$	$O(M+T)$	$O(TM)$
Time Complexity	$O(TM)$	$O(TM)$	$O(ITM)$

4.2.1 Space complexity (SC). In Algorithm 1, $\text{LA}^{\text{onepass}}$ requires initializing $\mathcal{W}$ for all the workers, which takes up $O(M)$ space. It also reserves $O(T)$ space for storing the estimated true labels. The space complexity (SC) of caching hyperparameters is $O(1)$. Additionally, the algorithm needs to maintain $C_{i,t}$ and t for each worker, requiring $O(M)$ space. At each time t, it needs to load the labels of task t, which is at most M. After the true label of task t is estimated, the labels of task t can be discarded. Therefore, the SC of loading/storing labels is $O(M)$. The overall space complexity of $\text{LA}^{\text{onepass}}$ is $O(M+T)$.

Regarding the iterative LA algorithms [62], the space complexity (SC) of caching worker qualities and true labels is the same as that of $\text{LA}^{\text{onepass}}$. However, the iterative LA algorithms need to load the entire dataset, requiring $O(TM)$ space. Therefore, the overall space complexity of the iterative LA algorithms is $O(TM)$.

As for MV, it requires reserving space for storing the estimated true labels, which amounts to $O(T)$. Additionally, for each task, it needs to load the corresponding labels, requiring $O(M)$ space. Thus, the overall space complexity of MV is $O(M+T)$.

4.2.2 Time complexity (TC) . As demonstrated in Algorithm 1, $\text{LA}^{\text{onepass}}$ performs the estimation of one true label and updates all worker qualities at each time-slice. Estimating one true label involves aggregating a maximum of M labels, resulting in a time complexity of $O(M)$. Updating one worker quality is done in $O(1)$ time, as shown in Equation (8). Overall, it takes $O(M)$ to update all worker qualities in one time-slice. Considering a total of T tasks, the overall time complexity of $\text{LA}^{\text{onepass}}$ is $O(TM)$.

Regarding the iterative LA algorithms [62], all the estimated true labels and worker qualities need to be updated in each iteration, with a time complexity of $O(TM)$ per iteration, which is equivalent to the time complexity of $\text{LA}^{\text{onepass}}$. Assuming the algorithms take I iterations to converge, the overall time complexity of the iterative LA algorithms is $O(ITM)$ [36].

For MV, it estimates one true label by aggregating a maximum of M labels, taking $O(M)$ time. With a total of T tasks, the overall time complexity is $O(TM)$.

In summary, both the space complexity (SC) and time complexity (TC) of $\text{LA}^{\text{onepass}}$ are equivalent to those of MV. The lower SC and TC of $\text{LA}^{\text{onepass}}$ make it more scalable and practical for aggregating labels in very large-scale datasets.

Remark 1. *The time complexity (TC) analysis in Section 4.2.2 considers the worst-case scenario where each worker labels all the tasks. However, in practice, it is more realistic to assume that each worker labels only a subset of tasks. In such cases, we can replace T with $\bar{T}$ in the "Time" row of Table 2, where $\bar{T}$ represents the average number of tasks labeled by each worker. Similarly, the space complexity (SC) of the iterative LA algorithms in Table 2 is $O(\bar{T}M)$ if each worker only labels a subset of tasks. This adjustment reflects the actual resource requirements when each worker only labels a subset of tasks.*

Remark 2. *The analyzed space complexity (SC) and time complexity (TC) presented in Table 2 are based on the general iterative LA algorithms' framework [62]. They serve as lower bounds for the SC and TC of iterative methods. However, it is important to note that depending on the specific models and implementations of iterative methods, the actual SC and TC can be higher than the values provided in the table.*

For instance, consider the LAA model [40], which is a neural network-based label aggregation approach. LAA takes the "one-hot code" of crowdsourced labels as input, resulting in a higher SC of up to $O(KTM)$. Another example is the

BWA model [34], which can only aggregate labels when $K = 2$. In its multi-class extension where $K > 2$, BWA adopts an "one-versus-all" classifier approach, which requires running its base model K times. Consequently, the TC of BWA can reach up to $O(KITM)$.

Therefore, it is important to consider the specific characteristics and models of iterative methods when assessing their SC and TC, as they can deviate from the general framework and exhibit different resource requirements.

4.3 Analysis of Termination Condition

As outlined in Algorithm 1, $\text{LA}^{\text{onepass}}$ traverses all the labels exactly once and terminates when each label has been processed. Therefore, the termination condition of $\text{LA}^{\text{onepass}}$ is implicit, ending the process once all the labels have been processed once. This contrasts the iterative label aggregation algorithms , which require an explicit termination condition to be set.

Compared to the explicit termination condition in iterative LA methods, the implicit termination condition of $\text{LA}^{\text{onepass}}$ offers several advantages. Firstly, explicitly setting the termination condition to be neither too loose nor too strict can be detrimental to the performance of iterative algorithms. A loose termination condition may cause premature termination before accurate estimation of worker qualities and true labels, thus compromising the performance of the LA algorithm. Conversely, a strict termination condition may prolong the iterative process unnecessarily. The additional iterations resulting from a strict termination condition may not improve the LA's performance but instead waste computational power and delay the completion of label aggregation. In contrast, the implicit termination condition of $\text{LA}^{\text{onepass}}$ eliminates the need for explicitly defining the termination condition.

Secondly, using the implicit termination condition is an inherent property of the dynamic model and estimation procedures described in Section 3.1 and Section 3.2, respectively. This enables $\text{LA}^{\text{onepass}}$ to aggregate labels more efficiently than the iterative algorithm. The update procedures in Algorithm 1 can be viewed as stochastic operations, where each iteration estimates the true label for one task and updates the qualities of workers who provided labels for that task. Consequently, as the algorithm estimates more true labels, the estimated worker qualities become more accurate. Moreover, $\text{LA}^{\text{onepass}}$ terminates when each task's true label is estimated only once, further enhancing its efficiency compared to the iterative LA algorithms.

However, there is a downside to traversing all the labels only once and relying on the implicit termination condition. $\text{LA}^{\text{onepass}}$ estimates the true labels one by one as it traverses all the labels only once. When the algorithm aggregates more labels, the estimated worker qualities converge, and the estimated true labels become more accurate. However, this poses a challenge: the true labels estimated early in the process may not be accurate because the estimated worker qualities are yet to converge. To address this issue, we develop $\text{LA}^{\text{twopass}}$, which will be discussed in detail in Section 5.1.

5 EXTENSIONS

In this section, we introduce two extensions. The first extension can improve the accuracy of $\text{LA}^{\text{onepass}}$ by traversing the crowdsourced labels again. The second extension describes how the proposed algorithms aggregate labels online.

5.1 Two Pass Algorithm

As discussed in Section 4.3, the issue of inaccurate early estimates of true labels arises due to the convergence status of worker qualities. To solve this problem, we develop $\text{LA}^{\text{twopass}}$, a straightforward extension of $\text{LA}^{\text{onepass}}$ that involves a second traversal of the labels. In $\text{LA}^{\text{twopass}}$, the converged worker qualities obtained from $\text{LA}^{\text{onepass}}$ are utilized to perform weighted majority voting (WMV) during the second traversal, resulting in the re-estimation of

true labels. We have chosen to employ WMV, as depicted in Equation (21), due to its proven theoretical guarantee [30].

$$v_i = K\hat{w}_i - 1, \qquad \hat{y}_t = \arg\max_k \left\{ \sum_{i=1}^{M} v_i \mathbb{1}(x_{i,t} = k) | k \in [K] \right\}. \tag{21}$$

The re-estimation of true labels in $\text{LA}^{\text{twopass}}$ incurs little overhead compared to $\text{LA}^{\text{onepass}}$, as it does not involve the estimation of worker qualities during the second traversal. Consequently, the second pass of $\text{LA}^{\text{twopass}}$ can be executed as efficiently as MV, while maintaining the same space complexity (SC) and time complexity (TC) as $\text{LA}^{\text{onepass}}$. By leveraging the converged worker qualities from the first pass, $\text{LA}^{\text{twopass}}$ enhances the accuracy of true label estimation without introducing significant computational burden.

5.2 Online Aggregation

Since we model LA as a dynamic system, both $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ can naturally be configured to aggregate labels online. This is particularly useful when labels are received sequentially in chunks, where each chunk contains labels for a few tasks. In fact, the proposed algorithms can handle the extreme case where each chunk only contains labels for a single task.

In the online aggregation setting, $\text{LA}^{\text{onepass}}$ estimates the worker qualities and true labels for the tasks in the current chunk. On the other hand, $\text{LA}^{\text{twopass}}$ utilizes the worker qualities estimated by $\text{LA}^{\text{onepass}}$ to re-estimate the true labels within the same chunk. Once the true labels in the current chunk are estimated, the labels within that chunk can be discarded.

It's worth noting that the information regarding worker qualities is retained in the posterior worker quality distributions, which are then used to estimate the true labels in the subsequent chunk. As a result, both $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ can effectively aggregate labels online without the need to revisit historical labels.

Remark 3. *If labels are aggregated online, there could be an infinite number of tasks arriving over time. However, if we assume there are a finite number of workers, the space complexity (Space) of both MV and $LA^{onepass}$ (as shown in Table 2) becomes $O(\infty)$ due to the analyses conducted in Section 4.2, which were based on an offline scenario where all the estimated true labels were stored. Notably, neither MV nor $LA^{onepass}$ requires the estimated true label and crowdsourced labels of a task at time-slice t to estimate the true label of a task at time-slice $t+1$. Thus, if we can discard the estimated true label after it has been computed, the space complexity of both MV and $LA^{onepass}$ can be reduced to $O(M)$ even when there is an infinite number of tasks. Similarly, the time complexity (TC) of both MV and $LA^{onepass}$ for estimating one true label at time-slice t is $O(M)$. The same analyses of time and space complexity also apply to $LA^{twopass}$.*

6 EXPERIMENTS

In this section, we provide experimental results to evaluate the performance of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. We conduct evaluations in both offline and online settings. The results from the offline and online experiments are reported and analyzed in Sections 6.2 and 6.3, respectively. In the offline setting, all labels for each dataset are fed into the algorithm in one pass. On the contrary, in the online setting, the labels from each dataset are divided into chunks, and these chunks are fed sequentially into the algorithms. The code is on https://github.com/yyang318/LA_onepass.

Table 3. Datasets statistics.

	Dataset	#Tasks	#Workers	#Classes	#Label	#Tasks with true labels
Single Choice Tasks	senti	98980	1960	5	569282	1000
	fact	42624	57	3	214960	576
	CF	300	461	5	1720	300
	CF_amt	300	110	5	6025	300
	MS	700	44	10	2945	700
	dog	807	109	4	8070	807
	face	584	27	4	5242	584
	adult	11040	825	4	89948	333
	mill	1891	37332	4	214658	1891
	web	2665	177	5	15567	2653
Decision Making Tasks	SP	4999	203	2	27746	4999
	SP_amt	500	143	2	10000	500
	ZC_all	2040	78	2	20372	2040
	ZC_in	2040	25	2	10626	2040
	ZC_us	2040	74	2	11271	2040
	prod	8315	176	2	24945	8315
	tweet	1000	85	2	20000	1000
	bird	108	39	2	4212	108
	trec	19033	762	2	88385	2275
	rte	800	164	2	8000	800

6.1 Experiment Setup

6.1.1 Methods. In the offline setting, the methods for comparison include MV, DS [9], LFC [44], IWMV [30], EBCC [37], BWA [34], LAA [40], ZC [11], TiReMGE [54] and TILCC [53]. All the methods for comparison except MV are iterative methods, which require loading all the crowdsourced labels to perform LA.

In the online setting, we use MV, iCRH [36] and two recent methods BiLA [19] and SBIC [42] for comparison. They all have the ability to perform LA online without revisiting historical data. Descriptions of these methods have been discussed in Section 2.2.

6.1.2 Datasets. We evaluate our methods using 20 publicly available real-world datasets. These datasets were sourced from five different collections [2, 24, 49, 61, 62], and encompass a broad spectrum of tasks such as sentiment analysis, entity resolution, face recognition, and quiz answering. The dataset sizes range from 1,720 to 569,282 entries. According to the taxonomy described in [62], the tasks can be classified as either single-choice tasks ($K > 2$) or decision-making tasks ($K = 2$). The dataset statistics are summarized in Table 3. These datasets are frequently used to evaluate label aggregation methods. For instance, 19 of these datasets were used in BWA and 17 in EBCC. Please note that the column labeled *#Tasks with true labels* in Table 3 denotes the number of ground-truth labels available in each dataset. These ground-truth labels are used solely for evaluation purposes, and not as input for label aggregation algorithms.

6.1.3 Termination condition. In the offline setting, all comparison methods, except MV, are iterative and require a termination condition to stop iterations. We adopt the following termination conditions:

- For DS, LFC, ZC, and IWMV, we utilize the condition used in [62], which halts the algorithms after 20 iterations.

- For EBCC and BWA, we use the condition provided in the authors' code. It terminates the algorithms when the probabilities of each task's classes being true, between two successive iterations, are less than a given threshold. Empirically, we find that EBCC and BWA halt after an average of 113.1 and 36.8 iterations, respectively.
- For LAA and TiReMGE, they cease after 150 epochs according to the authors' code.
- For TILCC, the algorithm terminates when the learned clusters between two consecutive iterations remain the same. Empirically, we observe that TILCC stops after an average of 15.5 iterations.
- For BiLA, it runs 20 epochs at each online chunk according to the authors' code.

6.1.4 Metrics. We employ *accuracy* as a metric to assess the performance of an algorithm. Accuracy is defined as the ratio of correctly estimated true labels to the total number of tasks. The efficiency of a method is evaluated by its runtime, measured in seconds.

6.1.5 Hyperparameters. As per the recommendations outlined in Section 3.4, we set $\alpha = 2$ and $\beta = 2$ for all datasets. The effects of these hyperparameter selections will be empirically validated in Section 6.2.4.

6.1.6 Randomness. The outcomes of the algorithms can be influenced by the inherent randomness present in their executions. The sources of such randomness can be categorized as follows:

Draws: All the methods estimate true labels by comparing the trustworthiness score of each task's classes, as depicted in the in-bracket term of Equation (5). In cases where the trustworthiness scores of some classes are tied, a class is randomly chosen from those with the highest trustworthiness scores to assign the true label.

Task Order: $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ estimate true labels sequentially, and in the online setting, all the algorithms estimate true labels in sequential chunks. Consequently, the task order can affect the algorithms' performance.

To mitigate the impact of this randomness, we run all the methods 20 times for each dataset and report the average accuracies in both offline and online settings. For the online evaluation, we shuffle the order of tasks in each run. When we evaluate the performance of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ offline, we also shuffle the task order because they estimate true labels sequentially. In the case of EBCC, we run the algorithm 40 times with random initialization and report the highest accuracy with the best ELBO (Evidence Lower BOund), in line with the code implemented by the authors of EBCC.

6.1.7 Implementation and experimental environment. The experiments are executed on an AMD 5900 CPU with 32GB RAM. To ensure a fair comparison, we adopt the implementation style in [62] and use pure Python (standard packages) to implement MV, $\text{LA}^{\text{onepass}}$, $\text{LA}^{\text{twopass}}$, MV, DS, LFC, ZC, EBCC, BWA, TILCC, iCRH, and SBIC. As BiLA, LAA, and TiReMGE are neural network-based models and rely on backpropagation for training, we use the codes provided by their authors, which are implemented in Tensorflow and Python, for these experiments.

6.2 Offline Experimental Results

6.2.1 Accuracy results. The average accuracies of each method across 20 datasets are summarized in Table 4. We found that the memory requirements for LAA for the senti and mill datasets, as well as TiReMGE for the senti dataset, exceeded the memory capacity of our machine (32GB). Therefore, we omitted the experiments for these models.

From Table 4, firstly, it can be observed that $\text{LA}^{\text{twopass}}$ and $\text{LA}^{\text{onepass}}$ rank first and fourth respectively, out of all the methods in terms of overall mean accuracy. This indicates the efficacy of our proposed algorithms in estimating true labels, even though they only traverse all labels at most twice.

Secondly, $\text{LA}^{\text{twopass}}$ and $\text{LA}^{\text{onepass}}$ perform exceptionally well for single-choice tasks, ranking within the top three of all methods. However, they do not perform as well in decision-making tasks. Upon analysis, we find that DS, LFC, and EBCC, which belong to the confusion matrix based LA methods, achieve high accuracies on

Table 4. Results for offline accuracy. The final three rows provide a summary of mean accuracy across 10 single-choice tasks, 10 decision-making tasks, and all 20 datasets respectively.

	MV	DS	LFC	IWMV	EBCC	BWA	LAA	ZC	TiReMGE	TILCC	$LA^{onepass}$	$LA^{twopass}$
senti	0.8832	0.8240	0.8180	0.8905	0.8400	0.8900	-	0.8890	0.8925	0.8104	0.8916	0.8921
fact	0.9016	0.8507	0.8611	0.9010	0.8915	0.8872	0.8960	0.9010	0.9025	0.8594	0.9010	0.9010
CF	0.8832	0.7967	0.8167	0.8805	0.8833	0.8933	0.8548	0.8800	0.8808	0.8787	0.8827	0.8830
CF_amt	0.8535	0.8567	0.8367	0.8567	0.8633	0.8600	0.8483	0.8533	0.8447	0.8038	0.8570	0.8563
MS	0.7023	0.7643	0.7743	0.7986	0.7871	0.7857	0.6903	0.7971	0.7148	0.7951	0.7936	0.7960
face	0.6381	0.6404	0.6404	0.6301	0.6336	0.6182	0.6509	0.6284	0.6330	0.5993	0.6315	0.6300
adult	0.7581	0.7447	0.7628	0.7658	0.7477	0.7417	0.6727	0.7219	0.7511	0.7940	0.7622	0.7655
dog	0.8229	0.8426	0.8426	0.8297	0.8401	0.8315	0.8364	0.8302	0.8107	0.8204	0.8317	0.8310
web	0.7298	0.8255	0.8326	0.8450	0.7441	0.8225	0.8420	0.8398	0.5872	0.4069	0.8138	0.8370
mill	0.9050	0.9062	0.9214	0.9060	0.7478	0.9318	-	0.9032	-	0.9040	0.9135	0.9135
SP	0.8865	0.9148	0.9148	0.9049	0.9152	0.9170	0.8794	0.9166	0.8857	0.8967	0.8948	0.9032
SP_amt	0.9425	0.9440	0.9440	0.9448	0.9440	0.9460	0.9440	0.9460	0.9433	0.9436	0.9444	0.9444
ZC_all	0.8312	0.7926	0.7922	0.8342	0.8642	0.8353	0.7754	0.8299	0.8328	0.7832	0.8358	0.8432
ZC_in	0.7406	0.7608	0.7598	0.7490	0.7755	0.7652	0.6703	0.7725	0.7779	0.7182	0.7442	0.7488
ZC_us	0.8613	0.8211	0.8211	0.8706	0.9123	0.8868	0.8103	0.8578	0.8679	0.7784	0.8627	0.8687
product	0.8966	0.9366	0.9373	0.9274	0.9349	0.9194	0.8449	0.9280	0.8966	0.8809	0.9078	0.9257
tweet	0.9321	0.9600	0.9600	0.9476	0.9610	0.9560	0.9569	0.9510	0.9353	0.9550	0.9510	0.9486
bird	0.7593	0.8796	0.8981	0.7222	0.8611	0.7593	0.8847	0.7222	0.5556	0.8426	0.7611	0.7519
rte	0.8966	0.9275	0.9275	0.9283	0.9313	0.9275	0.9172	0.9250	0.9162	0.9209	0.9215	0.9279
trec	0.6524	0.7046	0.7024	0.5912	0.7037	0.6044	0.5832	0.5697	0.6455	0.7016	0.6433	0.6323
mean single choice	0.8078	0.8052	0.8107	0.8304	0.7979	0.8262	0.7864	0.8244	0.7797	0.7672	0.8279	**0.8305**
mean decision making	0.8399	0.8642	0.8657	0.8420	**0.8803**	0.8517	0.8266	0.8419	0.8257	0.8421	0.8467	0.8495
mean overall	0.8238	0.8347	0.8382	0.8362	0.8391	0.8389	0.8088	0.8331	0.8039	0.8047	0.8373	**0.8400**

the bird and trec datasets because the workers in these datasets exhibit significant variability among classes. Consequently, these methods substantially outperform others on these two datasets, which boost their mean accuracies for decision-making tasks. In the case of single-choice tasks, the confusion matrix methods model the quality of each worker using a matrix with at least $K^2 - K$ free parameters. When K is large, there may not be sufficient labels to estimate the parameters accurately. Hence, the confusion matrix methods are less effective than the one-coin methods, including IWMV, BWA, ZC, and our proposed methods, for single-choice tasks.

Thirdly, the mean accuracies of LAA, TiReMGE, and TILCC are even worse than MV. LAA uses a neural network (variational auto-encoder) to learn the non-linear relationship between workers and tasks. However, LAA's objective may be too aggressive to be generalizable. TiReMGE and TILCC are recent methods that learn features from crowdsourced labels while performing LA. Although TiReMGE achieve the best accuracy on senti and fact, and TILCC outperforms other methods on the adult dataset, their feature extraction techniques are not robust, limiting their generalizability to a wide range of crowdsourcing tasks.

Lastly, it can be observed that there is no single method with best performance across all the datasets. As illustrated in Fig. 2, if we count the number of times a method ranks within the top three among all datasets, EBCC emerges as the winner. However, EBCC is unstable as it performs significantly worse on the mill dataset compared to other methods. The mill dataset is a quiz dataset where each task's class represents a choice, and the meanings of the choices vary across tasks. In this case, EBCC incorrectly clusters tasks based on workers' labels. In contrast, the accuracies of $LA^{twopass}$ and $LA^{onepass}$ never rank in the bottom three among all the methods, demonstrating the stability and robustness of our methods.

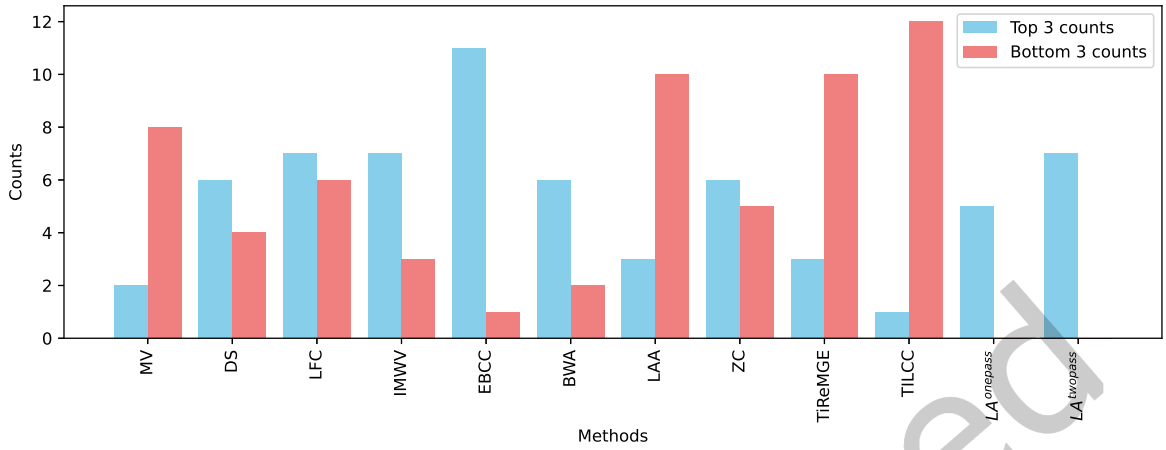


Fig. 2. Number of times each method ranks among the top three or bottom three for each dataset in terms of accuracy.

Table 5. One-sided Wilcoxon signed rank test results

	Overall			Decision Making			Single Choice		
Method	z-statistics	sig. level	p-value	z-statistics	sig. level	p-value	z-statistics	sig. level	p-value
DS	133		0.1559	41		0.0967	29		0.4609
LFC	136		0.1305	42		0.0801	30		0.4229
IMWV	163	*	0.0200	36		0.2158	45		0.0654
EBCC	162	*	0.0181	55	**	0.0010	26		0.5771
BWA	164	*	0.0133	44		0.0577	37		0.1875
LAA	66		0.8036	18		0.8389	17		0.5781
ZC	127.5		0.2152	32.5		0.3477	30		0.4229
TiReMGE	88		0.6160	34		0.2783	15		0.8203
TILCC	85		0.7738	32		0.3477	13		0.9346
$LA^{onepass}$	182	**	0.0014	48	*	0.0186	47	*	0.0244
$LA^{twopass}$	177	**	0.0028	45	*	0.0420	46.5	*	0.0322

6.2.2 Comparison to MV. The Majority Vote (MV) is a simple method for LA and is often used as a baseline to evaluate the performance of more complex LA methods. In this section, we conduct a one-sided Wilcoxon signed-rank test [52] on each method versus MV based on each method's accuracies on the datasets. The results from the Wilcoxon test determine whether each method is significantly more accurate than MV and to what extent. We use two significance levels: p-value thresholds of 0.01(**) and 0.05(*). Z-statistics are computed during the test to derive the p-values.

The results are summarized in Table 5. From Table 5, we observe that both $LA^{twopass}$ and $LA^{onepass}$ reach the ** significance level when the test is performed over all datasets. Furthermore, these two methods are the only ones reaching the * significance level when tests are performed over all datasets, decision-making datasets, and single-choice datasets. This provides statistical evidence that $LA^{twopass}$ and $LA^{onepass}$ perform better than MV in terms of accuracy.

Fig. 3. Offline Runtime Results. The methods in the legend are sorted descendingly based on their mean runtime.

It should be noted that due to the small sample size (the number of datasets) in the test, the results of the Wilcoxon test may not be entirely accurate. However, we consider them complementary to the accuracy results in Table 5, illustrating the improvements of our methods compared to MV.

6.2.3 Runtime results. Figure 3 depicts the average runtime results of the methods performed across all datasets. Given that MV does not estimate worker qualities, it can be regarded as the lower bound for the runtime of LA methods. From the figure, it is evident that $\text{LA}^{\text{onepass}}$ ranks second in terms of runtime among all methods, being very close to the most efficient method, MV. $\text{LA}^{\text{twopass}}$ ranks third, adding only a minimal computational overhead to $\text{LA}^{\text{onepass}}$. Compared to the most efficient iterative method, $\text{LA}^{\text{onepass}}$ is still more than 10 times faster. Notably, even though EBCC has the second highest overall mean accuracy among all methods and significantly outperforms others in decision making tasks, it is extraordinarily inefficient, being several orders of magnitude slower than $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$. The high efficiency of $\text{LA}^{\text{onepass}}$ and $\text{LA}^{\text{twopass}}$ makes them scalable and practical for aggregating very large-scale datasets.

6.2.4 Varying hyperparameters. Fig. 4 presents the accuracies of the proposed methods initialized with different hyperparameters. Both α and β are varied from $(2, 2)$ to $(6, 6)$, resulting in 25 unique initial settings. From Fig. 4, it can be observed that the gaps between the best and worst accuracies of $\text{LA}^{\text{twopass}}$ and $\text{LA}^{\text{onepass}}$ initialized by different hyperparameters are approximately 0.002 and 0.003, respectively. This reveals that the proposed methods are not overly sensitive to hyperparameters, which is in stark contrast to many iterative algorithms that demand careful hyperparameter tuning. For instance, EBCC requires careful adjustment of 6 hyperparameters to achieve the reported accuracies.

Fig. 4 also demonstrates that our methods perform marginally better when $\alpha \leq \beta$. This aligns with our analysis in Section 3.4 that suggests a pessimistic hyperparameter setting is beneficial. It's noteworthy that the accuracies exhibit a downward trend as α and β increase. This is due to the fact that larger α and β values correspond to stronger priors, leading the estimated worker quality in Equation (8) to be largely influenced by the prior rather than the empirical statistics $C_{i,t}$ and t.

6.2.5 Worker Quality Convergence Study. Two simulations are conducted to verify the convergence claims made in Section 4.1. In the first simulation, 20 workers are generated, each with the same true worker quality of 0.6. These workers label 50 tasks, each with 3 classes, and their labels are produced based on their worker qualities. The generated labels are then fed into $\text{LA}^{\text{onepass}}$ to estimate worker qualities. The results are shown in Fig. 5. In Fig. 5, the red line signifies the true worker quality. The green traces represent the evolution of the estimated

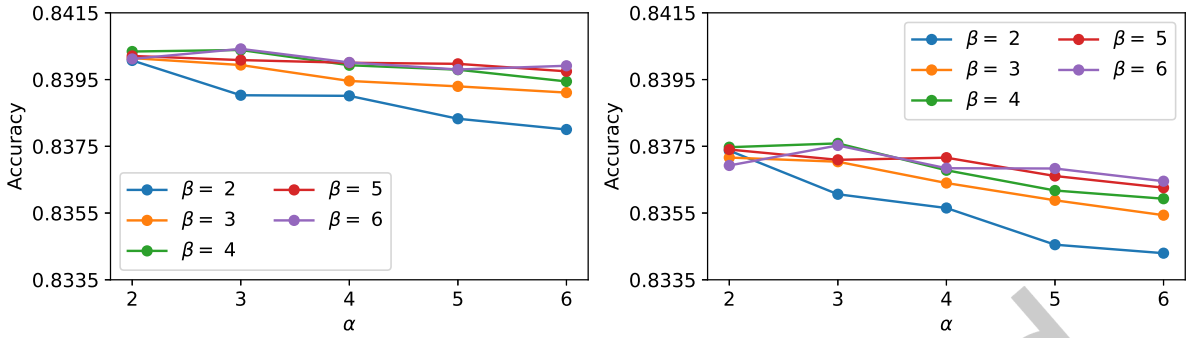


Fig. 4. Accuracies of the proposed methods under different hyperparameters. (Left) $LA^{twopass}$. (Right) $LA^{onepass}$.

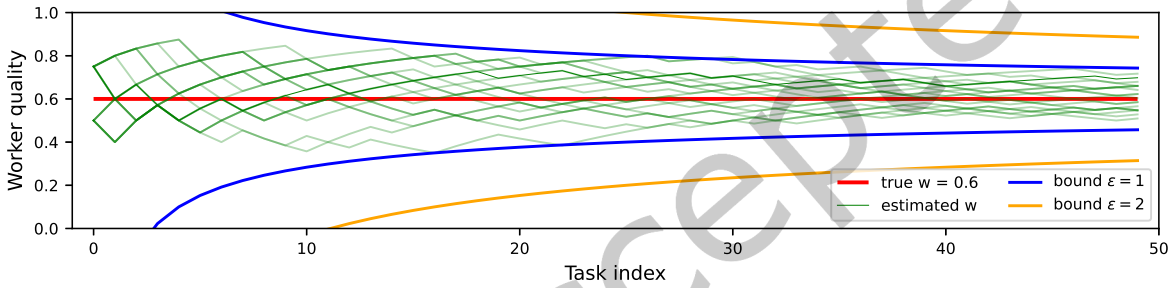


Fig. 5. First worker quality convergence study experimental result. All the true worker qualities are set to 0.6.

worker qualities. The blue and orange curves, computed by setting $\epsilon = 1, 2$ in Corollary (1), denote the error bounds, which restrict the error with at least 67% and 95% probabilities, respectively.

In the second simulation, 20 workers are generated again, but this time their worker qualities are sampled from the range $[0.4, 0.7]$. The other settings are consistent with the first simulation. The results of the second simulation are depicted in Fig. 6. From Fig. 5 and Fig. 6, it can be seen that the estimated worker qualities converge to their true values, irrespective of whether all the workers share the same quality or not. Also, the estimation errors can be bounded with a high probability. These results provide empirical validation for the assertions made in Section 4.1.

6.3 Online Experimental Results

In this section, we discuss the results of online label aggregation experiments. Adhering to the experimental approach outlined in [19], we partition the tasks from each dataset into ten equal parts (chunks) and report the accuracy up to each chunk for every method. It is worth noting that only partial ground truth labels are available for evaluation in certain datasets. If we randomly divided the tasks into ten chunks without accounting for this restriction, some tasks in specific chunks might lack ground truth labels for evaluation. To circumvent this issue, we distribute tasks with and without ground truth labels randomly across the ten chunks, ensuring each chunk contains a subset of tasks with ground truth labels for evaluation.

Fig. 7 displays the accuracies averaged over 20 experimental runs for each dataset. The average accuracies and runtimes for single-choice tasks and decision-making tasks are depicted in Fig. 8 (a) and (b), respectively.

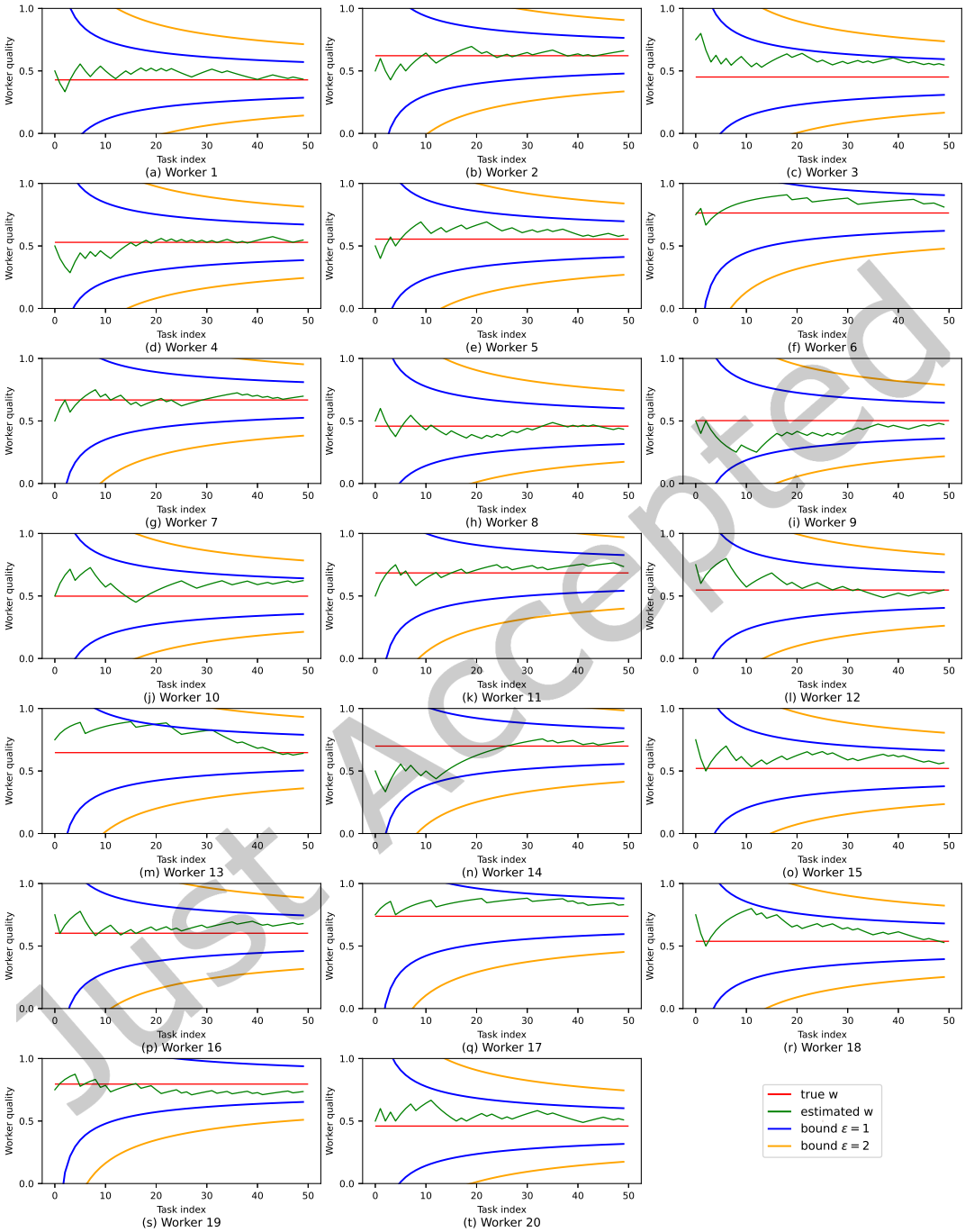


Fig. 6. Second worker quality convergence study experimental result. The true worker qualities are sampled in the interval $[0.4, 0.7]$.

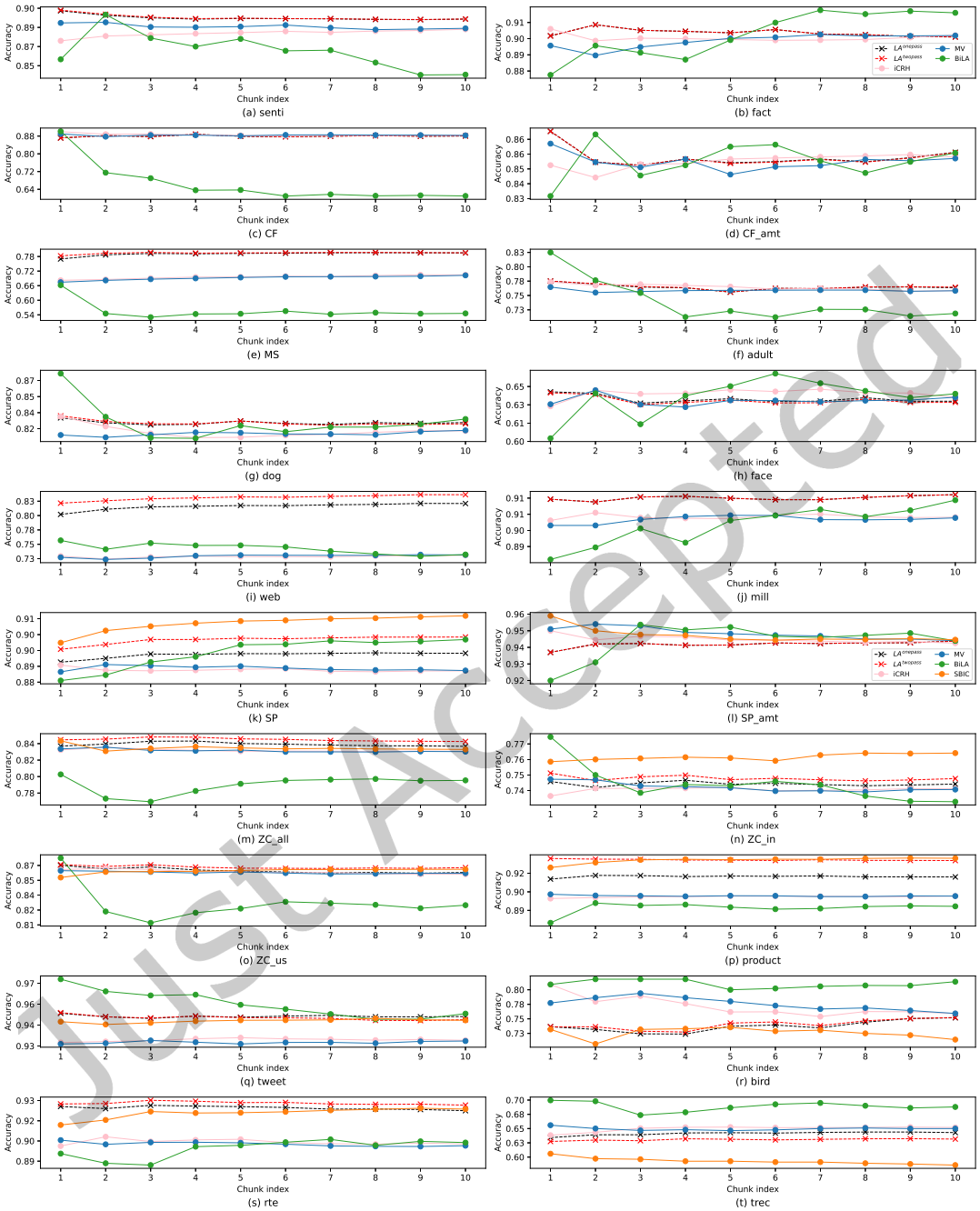
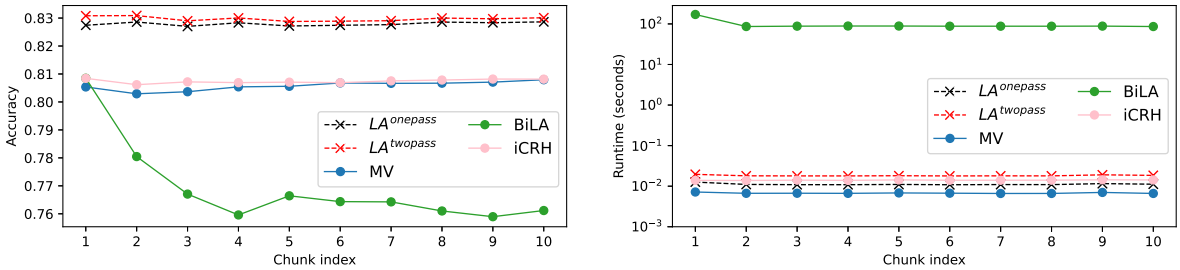
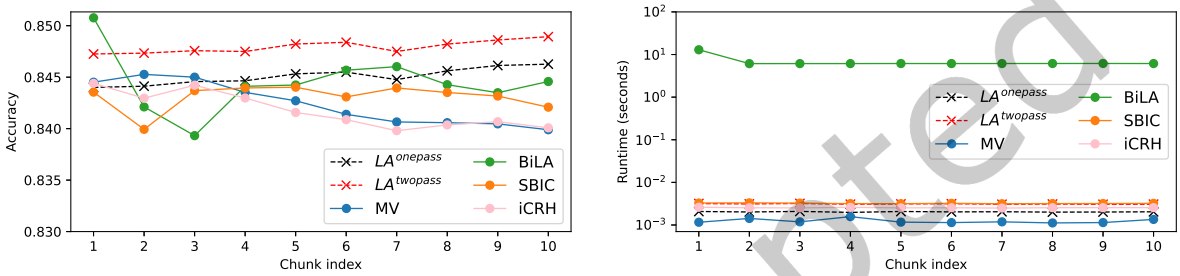


Fig. 7. Online accuracy results for the 20 real-world datasets.



(a) Single choice tasks



(b) Decision making tasks

Fig. 8. Online average accuracy and runtime results for single choice and decision making tasks.

6.3.1 Accuracy results. Based on the experimental findings, we can observe that $LA^{twopass}$ surpasses all other methods in terms of accuracy, excluding the first chunk. $LA^{onepass}$ also showcases strong performance, ranking second for single-choice tasks and second for decision-making tasks with the first 5 chunks. These results highlight the effectiveness of the proposed methods in online label aggregation, without necessitating revisiting historical labels. iCRH [36], an incremental version of the iterative label aggregation method CRH [32], does not guarantee convergence of estimated worker qualities, leading to a lower accuracy compared to our method. SBIC, designed for streaming label aggregation with single pass traversal of labels, displays performance comparable to $LA^{onepass}$. However, its applicability is restricted to decision-making tasks, which limits its usefulness. BiLA, a member of the "confusion matrix" family of methods, demonstrates relatively strong performance on decision-making tasks, particularly on the bird and trec datasets. Yet, it performs poorly in single-choice tasks. Notably, BiLA's accuracies on CF and MS lag behind other methods by more than 10%. This can be ascribed to the larger number of classes and the scarcity of labels in these two datasets, as reflected in Table 3, making it challenging for BiLA to estimate accurately the confusion matrices that represent worker qualities. Additionally, BiLA exhibits considerable fluctuations in accuracy across chunks in multiple datasets, making it unstable for online label aggregation.

6.3.2 Runtime results. According to Fig. 8, we can observe that MV demonstrates the highest efficiency. $LA^{onepass}$ and $LA^{twopass}$ are ranked second and fourth, respectively. iCRH, which processes labels in a single pass similar to $LA^{onepass}$, necessitates additional computational steps to monitor the maximum accumulated errors made by workers to normalize worker qualities. Consequently, iCRH is slower compared to $LA^{onepass}$. Considering that $LA^{twopass}$ necessitates performing weighted majority voting on the crowdsourced labels following the $LA^{onepass}$, it exhibits a marginally slower performance in comparison to iCRH. Even though SBIC processes labels in a single

pass, its average runtime is comparable to that of $\text{LA}^{\text{twopass}}$, which processes labels twice. This can be ascribed to the complexity of computations in SBIC, such as the sigmoid functions employed when updating its model parameters. Conversely, the proposed methods rely on basic arithmetic functions to estimate worker qualities and true labels. BiLA showcases the least efficient performance due to its reliance on neural networks. Although BiLA can aggregate labels online without revisiting historical data, it requires multiple scans of the labels in the current chunk for training its model through backpropagation. Notably, the runtime of BiLA for the first chunk is significantly longer than that of other methods, as it requires additional time to initialize its neural model.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel lightweight model for aggregating crowdsourced labels. We frame label aggregation as a dynamic system and represent it with a Dynamic Bayesian network. Based on this model, we derive two algorithms: $\text{LA}^{\text{twopass}}$ and $\text{LA}^{\text{onepass}}$. These algorithms proficiently aggregate labels by traversing the dataset no more than twice. We prove that the worker quality estimated by $\text{LA}^{\text{onepass}}$ converges at a rate of $o(1/\sqrt{t})$, with bounded error. Moreover, we demonstrate that our proposed algorithms possess space and time complexities comparable to those of MV.

We conduct extensive experiments on 20 real-world datasets to evaluate our methods. The results from offline experiments show that our methods deliver competitive accuracy in comparison to state-of-the-art iterative approaches, and they display high efficiency, with runtime akin to that of MV. Our methods' scalability and practicality are underscored by their low space and time complexities, rendering them suitable for aggregating labels in large-scale datasets. In addition, our methods can perform online label aggregation without the necessity for extra configurations. The results of online experiments demonstrate that our proposed methods outperform state-of-the-art online LA methods in terms of accuracy and real-time label aggregation capability.

Our methods belong to the "one-coin" methods category, which represents worker quality with a single parameter. The experimental results suggest that "confusion matrix" methods outperform "one-coin" methods in decision-making tasks. In our future work, we aim to enhance our method by incorporating the use of a confusion matrix to model worker quality. We anticipate that this approach will yield superior performance, specifically for decision-making tasks.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61976079, the Anhui High-level Talents Program under Grant T000642, and the Central University Basic Research Fund of China under Grant JZ2022HGQA0159.

REFERENCES

[1] Angelos-Christos Anadiotis, Oana Balalau, Théo Bouganim, Francesco Chimienti, Helena Galhardas, Mhd Yamen Haddad, Stéphane Horel, Ioana Manolescu, and Youssr Youssef. 2021. Discovering conflicts of interest across heterogeneous data sources with connectionlens. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4670–4674.

[2] Bahadir Ismail Aydin, Yavuz Selim Yilmaz, and Murat Demirbas. 2017. A crowdsourced "Who wants to be a millionaire?" player. *Concurrency and Computation: Practice and Experience* (2017), e4168.

[3] Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, and Alessandro Bozzon. 2021. What do you mean? Interpreting image classification with crowdsourced concept extraction and analysis. In *Proceedings of the Web Conference 2021*. 1937–1948.

[4] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. 2013. Streaming variational bayes. *Advances in neural information processing systems* 26 (2013).

[5] Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk*. 1–12.

[6] Chan-Fu Chen. 1985. On asymptotic normality of limiting density functions with Bayesian implications. *Journal of the Royal Statistical Society: Series B (Methodological)* 47, 3 (1985), 540–546.

[7] Ana Colovic, Annalisa Caloffi, and Federica Rossi. 2022. Crowdsourcing and COVID-19: How Public Administrations Mobilize Crowds to Find Solutions to Problems Posed by the Pandemic. *Public Administration Review* 82, 4 (2022), 756–763.

[8] Anup Kumar Das et al. 2018. European Union's General Data ProtectionRegulation, 2018: A brief overview. *Annals of Library and Information Studies (ALIS)* 65, 2 (2018), 139–140.

[9] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 20–28.

[10] Joost CF de Winter, Miltos Kyriakidis, Dimitra Dodou, and Riender Happee. 2015. Using CrowdFlower to study the relationship between self-reported violations and traffic accidents. *Procedia Manufacturing* 3 (2015), 2518–2525.

[11] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. 469–478.

[12] Jonathan Dortheimer. 2022. Collective intelligence in design crowdsourcing. *Mathematics* 10, 4 (2022), 539.

[13] Team eBird. 2022. *2022 Year in Review: eBird, Merlin, Macaulay Library, and Birds of the World.* https://ebird.org/news/2022-year-in-review

[14] Jianhong Feng, Guoliang Li, Henan Wang, and Jianhua Feng. 2014. Incremental quality inference in crowdsourcing. In *International Conference on Database Systems for Advanced Applications*. Springer, 453–467.

[15] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 61–72.

[16] Luoyi Fu, Jiasheng Xu, Shan Qu, Zhiying Xu, Xinbing Wang, and Guihai Chen. 2021. Seeking the truth in a decentralized manner. *IEEE/ACM Transactions on Networking* 29, 5 (2021), 2296–2312.

[17] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 2015. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 1631–1640.

[18] Meric Altug Gemalmaz and Ming Yin. 2021. Accounting for Confirmation Bias in Crowdsourced Label Aggregation.. In *IJCAI*. 1729–1735.

[19] Chi Hong, Amirmasoud Ghiassi, Yichi Zhou, Robert Birke, and Lydia Y Chen. 2021. Online label aggregation: A variational Bayesian approach. In *Proceedings of the Web Conference 2021*. 1904–1915.

[20] Jeff Howe. 2008. *Crowdsourcing: How the power of the crowd is driving the future of business.* Random House.

[21] Panagiotis G Ipeirotis. 2010. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM magazine for students* 17, 2 (2010), 16–21.

[22] Lingyun Jiang, Xiaofu Niu, Jia Xu, Dejun Yang, and Lijie Xu. 2021. Incentive mechanism design for truth discovery in crowdsourcing with copiers. *IEEE Transactions on Services Computing* 15, 5 (2021), 2838–2853.

[23] Yuan Jin, Mark Carman, Ye Zhu, and Yong Xiang. 2020. A technical survey on statistical modelling and design methods for crowdsourcing quality control. *Artificial Intelligence* 287 (2020), 103351.

[24] Tatiana Josephy, Matt Lease, Praveen Paritosh, Markus Krause, Mihai Georgescu, Michael Tjalve, and Daniela Braga. 2014. Workshops held at the first aaai conference on human computation and crowdsourcing: A report. *AI Magazine* 35, 2 (2014), 75–78.

[25] Xiangping Kang, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Wei Guo, Yazhou Ren, and Lizhen Cui. 2021. Crowdsourcing with Self-paced Workers. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 280–289.

[26] David R Karger, Sewoong Oh, and Devavrat Shah. 2014. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research* 62, 1 (2014), 1–24.

[27] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*. 619–627.

[28] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

[29] F-F ImageNet Li. 2010. Crowdsourcing, benchmarking & other cool things. *CMU VASC Semin* 16 (2010), 18–25.

[30] Hongwei Li and Bin Yu. 2014. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086* (2014).

[31] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment* 8, 4 (2014), 425–436.

[32] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1187–1198.

[33] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter* 17, 2 (2016), 1–16.

[34] Yuan Li, Benjamin IP Rubinstein, and Trevor Cohn. 2019. Truth inference at scale: A Bayesian model for adjudicating highly redundant crowd annotations. In *The World Wide Web Conference*. 1028–1038.

[35] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2015. On the discovery of evolving truth. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 675–684.

[36] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. *IEEE Transactions on Knowledge and Data Engineering* 28, 8 (2016), 1986–1999.

[37] Yuan Li, Benjamin Rubinstein, and Trevor Cohn. 2019. Exploiting worker correlation for label aggregation in crowdsourcing. In *International conference on machine learning*. PMLR, 3886–3895.
[38] Yanying Li, Haipei Sun, and Wendy Hui Wang. 2020. Towards fair truth discovery from biased crowdsourced answers. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 599–607.
[39] Yunfei Liu, Weinan Zhang, and Yong Yu. 2021. Aggregating crowd wisdom with side information via a clustering-based label-aware autoencoder. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 1542–1548.
[40] Jianhua Han Li'ang Yin, Weinan Zhang, and Yong Yu. 2017. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press*. 1325–1331.
[41] Yao Ma, Alex Olshevsky, Venkatesh Saligrama, and Csaba Szepesvari. 2020. Gradient descent for sparse rank-one matrix completion for crowd-sourced aggregation of sparsely interacting workers. *The Journal of Machine Learning Research* 21, 1 (2020), 5245–5280.
[42] Edoardo Manino, Long Tran-Thanh, and Nicholas Jennings. 2019. Streaming Bayesian inference for crowdsourced classification. *Advances in Neural Information Processing Systems* 32 (2019).
[43] Xiaoyi Pang, Zhibo Wang, Defang Liu, John CS Lui, Qian Wang, and Ju Ren. 2021. Towards personalized privacy-preserving truth discovery over crowdsourced data streams. *IEEE/ACM Transactions on Networking* 30, 1 (2021), 327–340.
[44] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research* 11, 4 (2010), 1297–1322.
[45] Nasim Sabetpour, Adithya Kulkarni, Sihong Xie, and Qi Li. 2021. Truth discovery in sequence labels from crowds. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 539–548.
[46] Hazem Sallouha, Alessandro Chiumento, and Sofie Pollin. 2021. Aerial vehicles tracking using noncoherent crowdsourced wireless networks. *IEEE Transactions on Vehicular Technology* 70, 10 (2021), 10780–10791.
[47] Zheyuan Ryan Shi, Leah Lizarondo, and Fei Fang. 2021. A recommender system for crowdsourcing food rescue platforms. In *Proceedings of the Web Conference 2021*. 857–865.
[48] Jiayang Tu, Peng Cheng, and Lei Chen. 2019. Quality-assured synchronized task assignment in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering* 33, 3 (2019), 1156–1168.
[49] Matteo Venanzi, Oliver Parson, Alex Rogers, and Nick Jennings. 2015. The activecrowdtoolkit: An open-source tool for benchmarking active learning algorithms for crowdsourcing research. In *Third AAAI Conference on Human Computation and Crowdsourcing*.
[50] Dan Wang, Ju Ren, Zhibo Wang, Xiaoyi Pang, Yaoxue Zhang, and Xuemin Shen. 2021. Privacy-preserving streaming truth discovery in crowdsourcing with differential privacy. *IEEE Transactions on Mobile Computing* 21, 10 (2021), 3757–3772.
[51] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems* 22 (2009).
[52] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics*. Springer, 196–202.
[53] Gongqing Wu, Liangzhu Zhou, Jiazhu Xia, Lei Li, Xianyu Bao, and Xindong Wu. 2022. Crowdsourcing Truth Inference Based on Label Confidence Clustering. *ACM Transactions on Knowledge Discovery from Data* (2022), 1–20.
[54] Gongqing Wu, Xingrui Zhuo, Xianyu Bao, Xuegang Hu, Richang Hong, and Xindong Wu. 2022. Crowdsourcing Truth Inference via Reliability-driven Multi-view Graph Embedding. *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2022), 1–26.
[55] Gongqing Wu, Xingrui Zhuo, Liangzhu Zhou, Xianyu Bao, Richang Hong, and Xindong Wu. 2022. TIRA: Truth Inference Via Reliability Aggregation on Object-Source Graph. *IEEE Transactions on Knowledge and Data Engineering* (2022). https://doi.org/10.1109/TKDE.2022.3225308
[56] Houping Xiao and Shiyu Wang. 2022. A Joint Maximum Likelihood Estimation Framework for Truth Discovery: A Unified Perspective. *IEEE Transactions on Knowledge and Data Engineering* (2022), 5521–5533.
[57] Jielong Yang and Wee Peng Tay. 2021. An unsupervised Bayesian neural network for truth discovery in social networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5182–5195.
[58] Yi Yang, Quan Bai, and Qing Liu. 2019. A probabilistic model for truth discovery with object correlations. *Knowledge-Based Systems* 165 (2019), 360–373.
[59] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, 766–773.
[60] Jing Zhang and Xindong Wu. 2019. Multi-label truth inference for crowdsourcing using mixture models. *IEEE Transactions on Knowledge and Data Engineering* 33, 5 (2019), 2083–2095.
[61] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. 2014. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Advances in neural information processing systems* 27 (2014).
[62] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.