INSTANCE DIFFICULTY-AWARE LABEL DISTRIBUTION LEARNING FROM CROWDS

Anonymous authors
Paper under double-blind review

ABSTRACT

In crowdsourcing scenarios, we can often collect a multiple noisy label set of each instance from different crowd workers and then infer its unknown true label by label integration. Therefore, the performance of label integration is mainly influenced by the noise contained in each instance's multiple noisy label set, which primarily stems from the worker quality and the instance difficulty. However, most existing label integration algorithms mainly focus on modeling the worker quality while ignoring the instance difficulty. To fill this gap, we propose a novel label integration algorithm called instance difficulty-aware label distribution learning (IDLDL). Specifically, we first design a dual label distribution enhancement method to obtain the initial label distribution of each instance. Then, we introduce an instance correction matrix to model the instance difficulty and a label distribution learning model to map the relationship between instances and their label distributions. Thirdly, we optimize this matrix and model to learn the true label distribution of each instance from its initial label distribution under the EM framework. Finally, we use the learned true label distribution of each instance to infer its integrated label. Extensive experiments show that IDLDL significantly outperforms all its competitors.

1 INTRODUCTION

Supervised learning has emerged as a mainstream paradigm in the field of machine learning and has achieved significant success in many areas, such as face recognition and natural language processing (Jiang et al., 2019; Zhang et al., 2023a). However, the success of supervised learning heavily relies on a vast amount of high-quality labeled data (Wang & Wu, 2021; Chen et al., 2022b; Hu et al., 2023). Traditionally, the labeling process is performed by domain experts or well-trained workers. Although this ensures the high reliability of the collected labels, it is prohibitively expensive and time-consuming. With the rapid development of artificial intelligence, the demand for large-scale labeled data has increased correspondingly. Consequently, there is an urgent need to develop more economical and time-saving methods to collect labeled data.

Fortunately, the emergence of crowdsourcing has provided an economical and time-saving method to collect large-scale labeled data (Li et al., 2019; He et al., 2022). Employers can distribute labeling tasks through crowdsourcing platforms, such as CrowdFlower, Clickworker, and Amazon Mechanical Turk (AMT), and employ non-professional crowd workers to label these tasks (Tong et al., 2020; Chen et al., 2022a). However, due to the gap in knowledge and expertise between crowd workers and domain experts, the quality of these labels collected by an individual crowd worker can not be guaranteed (Rodrigues & Pereira, 2018; Chen et al., 2020). To address this issue, each instance is typically assigned to multiple crowd workers to obtain its multiple noisy label set (Sheng et al., 2008). Subsequently, label integration is employed to infer the integrated label of each instance from its multiple noisy label set. In recent years, label integration has attracted significant research interest, leading to the development of numerous algorithms from diverse perspectives.

To the best of our knowledge, the performance of label integration is mainly influenced by the noise contained in each instance's multiple noisy label set, which primarily stems from the worker quality and the instance difficulty. However, most existing label integration algorithms mainly focus on modeling the worker quality while ignoring the instance difficulty. To fill this gap, we propose a novel label integration algorithm called instance difficulty-aware label distribution learning

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(IDLDL). In IDLDL, we first design a dual label distribution enhancement method to obtain an initial label distribution of each instance. Then, we introduce an instance correction matrix to model the instance difficulty and a label distribution learning model to map the relationship between instances and their label distributions. Thirdly, we optimize this matrix and model to learn the true label distribution of each instance from its initial label distribution under the Expectation Maximization (EM) framework. Finally, we use the learned true label distribution of each instance to infer its integrated label. The main contributions of this paper can be highlighted as:

- We reveal the shortcomings of existing label integration algorithms, which often model the worker quality to improve the performance of label integration while ignoring the instance difficulty, which limits the ability to further improve the performance of label integration.
- We introduce label distribution learning into label integration and propose a novel instance difficulty-aware label distribution learning (IDLDL) algorithm, which improves the performance of label integration by jointly modeling the worker quality and the instance difficulty.
- We conduct a series of experiments to validate the effectiveness, rationality and convergence of IDLDL on four widely used real-world datasets. The experimental results show that IDLDL significantly outperforms all the other advanced label integration algorithms.

2 Related Work

Depending on whether they explicitly model worker quality, existing label integration algorithms can be divided into two categories. The first category doesn't explicitly model the worker quality. For example, majority voting (MV) (Li et al., 2024) is the simplest but often effective label integration algorithm, which directly chooses the most frequent label as the integrated label. Ground truth inference using clustering (GTIC) (Zhang et al., 2016) generates attributes from multiple noisy label sets and then clusters instances into different groups utilizing generated attributes. Multiple noisy label distribution propagation (MNLDP) (Jiang et al., 2022) estimates the label distribution of each instance and then propagates the distribution to its nearest neighbors. Label augmented and weighted majority voting (LAWMV) (Chen et al., 2022c) merges the neighbors' multiple noisy label sets to obtain its augmented multiple noisy label set and weights each neighbor. Attribute augmentation-based label integration (AALI) (Zhang et al., 2023b) enriches the original attribute space and then trains multiple classifiers on reliable instances to predict all instances. Instance redistribution-based label integration (IRLI) (Zhang et al., 2024) combines the multiple noisy label distribution vectors and the instance redistribution vector of each instance to infer its integrated label. Three-way decision-based label integration (TDLI) (Pan et al., 2025) first divides the crowdsourced dataset into three disjoint subsets and then infers the integrated labels by applying different strategies.

The second category explicitly models the worker quality. For example, Dawid-Skene (DS) (Dawid & Skene, 1979) models the worker quality through a confusion matrix and class priors under the EM (Singh, 2006) framework. In the E-step, DS calculates the posterior probability of each class. In the M-step, DS calculates error rates in the matrix and the class probability. Iterative weighted majority voting (IWMV) (Li & Yu, 2014) utilizes the multiple noisy label sets and the initial integrated labels obtained by MV to evaluate the worker quality, and then iteratively updates the worker quality and integrated labels until convergence. Differential evolution-based weighted soft majority voting (DEWSMV) (Tao et al., 2021) exploits a differential evolution (Storn & Price, 1997) method to estimate the worker quality by minimizing the error, Gini, and entropy of weighted multiple noisy label sets, and then obtains the integrated labels by weighted soft majority voting. Decision tree-based weighted majority voting (DTWMV) (Yang et al., 2022) treats each worker as an instance attribute and then builds decision trees to calculate the weight (quality) of each attribute (worker). Finally, DTWMV infers the integrated label of each instance by weighted majority voting. Label consistency-based ground truth inference (LCGTI) (Li et al., 2025) evaluates the worker quality by label consistency on both same and similar instances, and then selects the label of the highest-quality worker for each instance as the integrated label. Enhanced label distribution propagation (ELDP) (Zhang et al., 2025) first estimates the worker quality and performs label distribution enhancement, then further enhances instances not covered by the first step based on class membership estimation, and finally propagates the enhanced label distributions from accurately enhanced instances to inaccurately enhanced instances.

3 THE PROPOSED LDLLC

As discussed above, most existing label integration algorithms typically model the worker quality to learn the label distribution of each instance, and then infer its integrated label based on this distribution. However, these algorithms mainly focus on modeling the worker quality while ignoring the instance difficulty. In real-world crowdsourcing scenarios, the reliability of the labels collected from different workers is influenced by the worker quality and the instance difficulty. When the instance difficulty is high, even high-quality workers may provide incorrect labels, which may lead to a deviation between the learned label distribution of the instance and its unknown true label distribution. Therefore, relying solely on modeling the worker quality is insufficient for accurately learning the label distribution of each instance, thereby limiting improvement in the performance of label integration. As far as we know, label distribution learning (LDL) (Geng, 2016; Xu & Zhou, 2017; Chen et al., 2022d) provides a natural way to learn a more accurate label distribution of each instance, which can map the relationship between the attributes of each instance and its label distribution. Therefore, we jointly model the worker quality and the instance difficulty to learn a label distribution learning model, so as to further improve the performance of label integration. To achieve this objective, several issues need to be addressed: 1) How to obtain a reliable initial label distribution for each instance? 2) How to model the instance difficulty? 3) How to accurately learn the true label distribution of each instance from its initial label distribution? Correspondingly, we propose IDLDL that mainly includes three components: label distribution enhancement, instance difficulty modeling, and label distribution learning, which are used to address these issues. The detailed process will be introduced in the following subsections.

3.1 LABEL DISTRIBUTION ENHANCEMENT

In crowdsourcing scenarios, a dataset is commonly denoted by $D = \{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$,where N denotes the number of instances. $\boldsymbol{x}_i$ denotes the i-th instance in D, which can be denoted as $\{x_{im}\}_{m=1}^{M}$. M denotes the attribute dimension and x_{im} denotes the attribute value of $\boldsymbol{x}_i$ on the m-th attribute A_m. $\boldsymbol{L}_i$ denotes the multiple noisy label set of $\boldsymbol{x}_i$, which is denoted as $\{l_{ir}\}_{r=1}^{R}$. R denotes the number of workers, l_{ir} denotes the label of $\boldsymbol{x}_i$ labeled by the r-th worker u_r. l_{ir} takes a value from the fixed set $\{-1, c_1, \ldots, c_q, \ldots, c_Q\}$, where Q denotes the number of classes, c_q denotes the q-th class and -1 denotes that u_r does not label $\boldsymbol{x}_i$. In order to infer the unknown true label y_i of $\boldsymbol{x}_i$, label integration is utilized to obtain the integrated label $\hat{y}_i$, striving for the highest possible consistency between y_i and $\hat{y}_i$.

Given the above preliminaries, the first step of label distribution enhancement is to estimate the worker quality w_r of u_r. Since labels provided by high-quality workers are usually more reliable, the reliability of the initial label distribution largely depends on accurately identifying high-quality workers from the crowdsourced dataset. Inspired by internal indices in clustering (Maulik & Bandyopadhyay, 2002), we argue that instances labeled by high-quality workers tend to have small intra-cluster distances and large inter-cluster distances. Therefore, we use the Calinski-Harabasz (CH) (Caliński & Harabasz, 1974) index to estimate the worker quality. Then, the CH index CH_r of u_r can be calculated as follows:

$$CH_r = \frac{[tr(\boldsymbol{\mathcal{B}}_r)/(Q-1)]}{[tr(\boldsymbol{\mathcal{W}}_r)/(|D_r|-Q)]}, \tag{1}$$

where D_r denotes the set of all instances labeled by u_r and $|D_r|$ denotes the number of instances in D_r, $\boldsymbol{\mathcal{B}}_r$ and $\boldsymbol{\mathcal{W}}_r$ denote inter-cluster and intra-cluster instance matrices, $tr(\boldsymbol{\mathcal{B}}_r)$ and $tr(\boldsymbol{\mathcal{W}}_r)$ denote the trace of $\boldsymbol{\mathcal{B}}_r$ and $\boldsymbol{\mathcal{W}}_r$, which can be calculated as follows:

$$tr(\boldsymbol{\mathcal{B}}_r) = \sum_{q=1}^{Q} |D_{rq}| d(\boldsymbol{z}_{rq}, \boldsymbol{z}_r), \tag{2}$$

$$tr(\boldsymbol{\mathcal{W}}_r) = \sum_{q=1}^{Q} \sum_{i=1}^{|D_{rq}|} d(\boldsymbol{x}_i, \boldsymbol{z}_{rq}), \tag{3}$$

where D_{rq} denotes the set of all instances that are labeled as c_q by u_r, $\boldsymbol{z}_{rq}$ denotes the centroid of D_{rq}, $\boldsymbol{z}_r$ denotes the centroid of D_r, and $d(\cdot, \cdot)$ denotes a distance measure function. Similarly, $\boldsymbol{z}_r$

can be denoted as $\{\boldsymbol{z}_{rm}\}_{m=1}^{M}$, where $\boldsymbol{z}_{rm}$ denotes the attribute value of $\boldsymbol{z}_r$ on the m-th attribute A_m in D_r. When A_m is numerical, $\boldsymbol{z}_{rm}$ is the mean of A_m. When A_m is nominal, $\boldsymbol{z}_{rm}$ is the mode of A_m. $d(\cdot,\cdot)$ denotes the heterogeneous Manhattan-overlap metric (HMOM) (Wilson & Martinez, 1997). HMOM uses normalized Manhattan distance for numerical attributes and overlap distance for nominal attributes. The distance between two instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be is given as $d(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sum_{m=1}^{M} d_m(\boldsymbol{x}_i,\boldsymbol{x}_j)$, where $d_m(\boldsymbol{x}_i,\boldsymbol{x}_j)$ denotes the distance between instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ on A_m, which can be calculated as follows:

$$d_m(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}1, & if\ x_{im}\ or\ x_{jm}\ is\ missing\\ 1-\delta(x_{im},x_{jm}), & if\ A_m\ is\ nominal\\ \dfrac{|x_{im}-x_{jm}|}{max_m-min_m}, & if\ A_m\ is\ numeric\end{cases}, \tag{4}$$

where $\delta(\cdot,\cdot)$ denotes an indicator function that outputs 1 if its two input values are equal, and 0 otherwise. $max_m - min_m$ is used to normalize Manhattan distance, where max_m and min_m denote the maximum and minimum attribute values of A_m, respectively.

According to the above equations, we can calculate the CH index CH_r of each worker. Although CH_r is positively correlated with the quality of u_r, it may fall outside the interval $[0,1]$. Therefore, we normalize CH_r to obtain the worker quality w_r as follows:

$$w_r=\frac{CH_r-Min(\{CH_r\}_{r=1}^{R})}{Max(\{CH_r\}_{r=1}^{R})-Min(\{CH_r\}_{r=1}^{R})}, \tag{5}$$

where $Max(\cdot)$ and $Min(\cdot)$ return the maximum and minimum values of the set, respectively.

After obtaining worker quality w_r, we perform the first label distribution enhancement for $\boldsymbol{x}_i$. Let $\boldsymbol{P}_i^1=\{p_{iq}^1\}_{q=1}^{Q}$ denote the enhanced label distribution of $\boldsymbol{x}_i$ after the first enhancement, where p_{iq}^1 can be calculated as follows:

$$p_{iq}^1=\frac{\sum_{r=1}^{R}\delta(l_{ir},c_q)\,w_r}{\sum_{q=1}^{Q}\sum_{r=1}^{R}\delta(l_{ir},c_q)\,w_r}. \tag{6}$$

In crowdsourcing scenarios, each worker typically labels only a subset of instances in the dataset. That is, we can only enhance the label distributions of instances labeled by high-quality workers. For instances that are not labeled by high-quality workers, the first label distribution enhancement is insufficient. We argue that the closer an instance is to a given cluster constructed based on the labels provided by high-quality workers, the more likely it is to belong to the class corresponding to that cluster. Therefore, we perform the second label distribution enhancement by membership degree estimation. Specifically, based on w_r, the highest-quality worker u_{r^*} is identified as follows:

$$u_{r^*}=u_{\underset{r\in\{1,\ldots,r,\ldots,R\}}{\arg\max}\ w_r}. \tag{7}$$

Then, we partition all instances D_{r^*} labeled by u_{r^*} into Q clusters $\{D_{r^*q}\}_{q=1}^{Q}$, where D_{r^*q} can be defined as follows:

$$D_{r^*q}=\{\boldsymbol{x}_i \mid \boldsymbol{x}_i\in D_{r^*}\wedge\delta(l_{ir^*},c_q)\}. \tag{8}$$

Since the labels provided by high-quality workers are reliable, the clusters partitioned by u_{r^*} are considered to be closest to the true distribution of instances. Therefore, instances within the same class should be closer to each other, while instances from different classes should be farther apart. Given an instance $\boldsymbol{x}$, we estimate the membership degree of $\boldsymbol{x}$ belonging to c_q based on the average distance $\bar{d}_{\boldsymbol{x}}^{q}$ from $\boldsymbol{x}$ to all instances in D_{r^*q}. $\bar{d}_{\boldsymbol{x}}^{q}$ can be calculated as follows:

$$\bar{d}_{\boldsymbol{x}}^{q}=\frac{\sum_{i=1}^{|D_{r^*q}|}d(\boldsymbol{x},\boldsymbol{x}_i)}{|D_{r^*q}|}. \tag{9}$$

The smaller $\bar{d}_{\boldsymbol{x}}^{q}$ is, the more likely $\boldsymbol{x}$ belongs to c_q. Therefore, $\bar{d}_{\boldsymbol{x}}^{q}$ is negatively correlated to the membership degree of $\boldsymbol{x}$ belonging to c_q. Let $Mem(\boldsymbol{x},c_q)$ denote the membership degree of $\boldsymbol{x}$ to c_q. We estimate $Mem(\boldsymbol{x},c_q)$ by normalizing $\bar{d}_{\boldsymbol{x}}^{q}$ as follows:

$$Mem(\boldsymbol{x},c_q)=\frac{Max(\{\bar{d}_{\boldsymbol{x}}^{q}\}_{q=1}^{Q})-\bar{d}_{\boldsymbol{x}}^{q}}{Max(\{\bar{d}_{\boldsymbol{x}}^{q}\}_{q=1}^{Q})-Min(\{\bar{d}_{\boldsymbol{x}}^{q}\}_{q=1}^{Q})}. \tag{10}$$

After obtaining the membership degree $Mem(\boldsymbol{x}, c_q)$, we perform the second label distribution enhancement for $\boldsymbol{x}_i$. Let $\boldsymbol{P}_i^2 = \{p_{iq}^2\}_{q=1}^Q$ denote the enhanced label distribution of $\boldsymbol{x}_i$ after the second enhancement, where p_{iq}^2 can be calculated as follows:

$$p_{iq}^2 = \frac{Mem(\boldsymbol{x}_i, c_q)}{\sum_{q=1}^{Q} Mem(\boldsymbol{x}_i, c_q)}. \tag{11}$$

Finally, we merge the two enhanced distributions $\boldsymbol{P}_i^1$ and $\boldsymbol{P}_i^2$ to obtain the initial label distribution $\boldsymbol{P}_i = \{p_{iq}\}_{q=1}^Q$ of $\boldsymbol{x}_i$, where p_{iq} can be calculated as follows:

$$p_{iq} = \frac{p_{iq}^1 + p_{iq}^2}{\sum_{q=1}^{Q}(p_{iq}^1 + p_{iq}^2)}. \tag{12}$$

3.2 Instance difficulty modeling and label distribution learning

Although the dual label distribution enhancement method can obtain a reliable initial label distribution for each instance, it may still deviate from the unknown true label distribution due to the influence of instance difficulty. Therefore, we introduce an instance correction matrix $\boldsymbol{S}$ to model the instance difficulty. We can learn the true label distribution of each instance using the instance correction matrix and its initial label distribution. However, the initial label distribution, the instance correction matrix, and the true label distribution are mutually dependent. Meanwhile, the LDL model requires a reliable label distribution as supervision to learn the relationship between the attributes of each instance and its label distribution. Therefore, directly learning the true label distribution of each instance is very difficult. We treat $\boldsymbol{S}$ as a latent variable and employ the EM framework to optimize the instance correction matrix and the label distribution learning model, thereby learning the true label distribution of each instance from its initial label distribution.

In the E-step, we aim to learn the true label distribution of each instance from its initial label distribution. According to the theoretical foundations of LDL, the output of the maximum entropy model (Ren et al., 2019) naturally satisfies the constraints of a probability distribution and can be learned from the attributes of an instance. Based on this, we first adopt the maximum entropy model to predict the true label distribution of each instance from its attributes. Let $\hat{\boldsymbol{P}}_i = \{\hat{p}_{iq'}\}_{q'=1}^Q$ denote the predicted label distribution of $\boldsymbol{x}_i$, where $\hat{p}_{iq'}$ can be calculated as follows:

$$\hat{p}_{iq'} = p(c_{q'} \mid \boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\left(\sum_{m=1}^{M} \theta_{q'm} x_{im}\right)}{\sum_{q'=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta_{q'm} x_{im}\right)}, \tag{13}$$

where $p(A \mid B)$ denotes the conditional probability of event A given event B, and $\boldsymbol{\theta}$ is the parameter matrix of the LDL model, $\theta_{q'm}$ is the element of $\boldsymbol{\theta}$. In this paper, $\boldsymbol{\theta}$ is initialized such that its diagonal elements are set to 1 and all remaining elements are set to 0.

Although the above prediction exploits the attributes of each instance to estimate its true label distribution, it does not consider the influence of instance difficulty. Therefore, according to Bayes' theorem, we further combine the predicted label distribution $\hat{\boldsymbol{P}}_i$ with the instance correction matrix $\boldsymbol{S}$ to estimate the conditional probability $\gamma_{iq'|q}$ of the true label class $c_{q'}$ conditioned on the initial label class c_q of $\boldsymbol{x}_i$, which can be calculated as follows:

$$\gamma_{iq'|q} = p(c_{q'} \mid c_q, \boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{p(c_q, c_{q'} \mid \boldsymbol{x}_i; \boldsymbol{\theta})}{p(c_q \mid \boldsymbol{x}_i; \boldsymbol{\theta})} = \frac{S_{qq'}\, \hat{p}_{iq'}}{\sum_{q''=1}^{Q} S_{qq''}\, \hat{p}_{iq''}}, \tag{14}$$

where $S_{qq'}$ is the element of $\boldsymbol{S}$, which represents the probability that the true label class $c_{q'}$ is transformed to the initial label class c_q due to the instance difficulty. In this paper, $\boldsymbol{S}$ is initialized such that each diagonal element is randomly sampled from the interval $[0.7, 0.9]$, and each remaining element in the same column is assigned an equal share of the remaining value.

After obtaining the conditional probability $\gamma_{iq'|q}$, we can learn the true label distribution of each instance from its initial label distribution $\boldsymbol{P}_i$ based on the total probability theorem. Let $\boldsymbol{P}_i^* =$

$\{p^*_{iq'}\}_{q'=1}^{Q}$ denote the true label distribution of $\boldsymbol{x}_i$, where $p^*_{iq'}$ is calculated as follows:

$$p^*_{iq'} = \sum_{q=1}^{Q} p(c_q, c_{q'} \mid \boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{q=1}^{Q} p_{iq}\, \gamma_{iq'|q} = \sum_{q=1}^{Q} p_{iq} \frac{S_{qq'}\, \hat{p}_{iq'}}{\sum_{q''=1}^{Q} S_{qq''}\, \hat{p}_{iq''}}. \tag{15}$$

After obtaining $\boldsymbol{P}^*_i$ and the conditional probability $\gamma_{iq'|q}$ in the E-step, we combine them with the initial label distribution $\boldsymbol{P}_i$ to update the instance correction matrix in the M-step according to the definition of the instance correction matrix, where $S_{qq'}$ is calculated as follows:

$$S_{qq'} = \frac{\sum_{i=1}^{N} p(c_q, c_{q'} \mid \boldsymbol{x}_i)}{\sum_{i=1}^{N} p(c_{q'} \mid \boldsymbol{x}_i)} = \frac{\sum_{i=1}^{N} p_{iq}\, \gamma_{iq'|q}}{\sum_{i=1}^{N} p^*_{iq'}}. \tag{16}$$

Then, each column of the matrix $\boldsymbol{S}$ is normalized to satisfy the following constraints:

$$\sum_{q=1}^{Q} S_{qq'} = 1, \quad S_{qq'} \geq 0. \tag{17}$$

We train an LDL model to learn the mapping from the attributes of $\boldsymbol{x}_i$ to its predicted label distribution using the true label distribution $\boldsymbol{P}^*_i$ of $\boldsymbol{x}_i$ obtained in the E-step. Various measures can be used to measure the distance between the predicted label distribution $\hat{\boldsymbol{P}}_i$ and the possible true label distribution $\boldsymbol{P}^*_i$. In IDLDL, we use the Kullback-Leibler (KL) (Zheng et al., 2018) divergence to calculate the distance between the two distributions. Based on the KL divergence, the objective function of label distribution learning can be formulated as minimizing the distance between $\hat{\boldsymbol{P}}_i$ and $\boldsymbol{P}^*_i$. Let $L(\boldsymbol{\theta})$ denote the objective function, which can be calculated as follows:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \mathrm{KL}(\boldsymbol{P}^*_i \parallel \hat{\boldsymbol{P}}_i) = \sum_{i=1}^{N} \sum_{q'=1}^{Q} p^*_{iq'} \ln \frac{p^*_{iq'}}{\hat{p}_{iq'}} = -\sum_{i=1}^{N} \sum_{q'=1}^{Q} p^*_{iq'} \ln \hat{p}_{iq'}, \tag{18}$$

Since $\boldsymbol{P}^*_i$ is a fixed constant in the M-step, the term $\sum_{i=1}^{N} \sum_{q'=1}^{Q} p^*_{iq'} \ln p^*_{iq'}$ is constant with respect to the parameter matrix $\boldsymbol{\theta}$ and can be directly omitted during the BFGS optimization.

In the above objective function, the predicted label distribution is defined according to the maximum entropy model. By substituting Eq. (13) into Eq. (18), we can obtain the final objective function $L(\boldsymbol{\theta})$, which can be define as follows:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left[\ln \sum_{q'=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta_{q'm} x_{im} \right) - \sum_{q'=1}^{Q} p^*_{iq'} \sum_{m=1}^{M} \theta_{q'm} x_{im} \right]. \tag{19}$$

Then, we use the quasi-Newton method BFGS (Jia et al., 2018) to optimize the final objective function. The BFGS method approximates the inverse Hessian matrix by constructing a matrix $\boldsymbol{B}$, thereby simplifying the optimization process of the Newton method (Nocedal & Wright, 2006). In this paper, the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(0)}$ is initialized as the identity matrix. When optimizing $L(\boldsymbol{\theta})$ using BFGS, the optimization process primarily depends on the gradient $\nabla L(\boldsymbol{\theta}^{(k)})$ and the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(k)}$ at each iteration. For each element $\theta_{q'm}$, we calculate the corresponding gradient component $\frac{\partial L(\boldsymbol{\theta}^{(k)})}{\partial \theta_{q'm}}$ of $\nabla L(\boldsymbol{\theta}^{(k)})$ at the k-th iteration as follows:

$$\frac{\partial L(\boldsymbol{\theta}^{(k)})}{\partial \theta_{q'm}} = \sum_{i=1}^{N} \left[\frac{\exp\left(\sum_{m=1}^{M} \theta^{(k)}_{q'm} x_{im} \right)}{\sum_{q'=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta^{(k)}_{q'm} x_{im} \right)} - p^*_{iq'} \right] x_{im}. \tag{20}$$

At the beginning of each iteration, BFGS first calculates the search direction of the current iteration based on the previous gradient $\nabla L(\boldsymbol{\theta}^{(k-1)})$ and the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(k-1)}$. Let $\boldsymbol{p}^{(k)}$ denote the search direction, which can be calculated as follows:

$$\boldsymbol{p}^{(k)} = -\boldsymbol{B}^{(k-1)} \nabla L(\boldsymbol{\theta}^{(k-1)}). \tag{21}$$

After obtaining the search direction of the current iteration k, a line search procedure is performed to determine the step length $\eta^{(k)}$ that satisfies the strong Wolfe condition, which can be determined by the following inequalities:

$$L(\boldsymbol{\theta}^{(k-1)} + \eta^{(k)}\boldsymbol{p}^{(k)}) \leq L(\boldsymbol{\theta}^{(k-1)}) + c_1\eta^{(k)}\nabla L(\boldsymbol{\theta}^{(k-1)})^T\boldsymbol{p}^{(k)}, \tag{22}$$

$$\left|\nabla L(\boldsymbol{\theta}^{(k-1)} + \eta^{(k)}\boldsymbol{p}^{(k)})^T\boldsymbol{p}^{(k)}\right| \leq c_2\left|\nabla L(\boldsymbol{\theta}^{(k-1)})^T\boldsymbol{p}^{(k)}\right|, \tag{23}$$

where $0 < c_1 < c_2 < 1$, with $c_1 = 10^{-4}$ and $c_2 = 0.9$.

With the step length $\eta^{(k)}$ determined, the parameter matrix is updated along the search direction $\boldsymbol{p}^{(k)}$, where $\boldsymbol{\theta}^{(k)}$ can be updated as follows:

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \eta^{(k)}\boldsymbol{p}^{(k)}. \tag{24}$$

Finally, the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(k)}$ is updated as follows:

$$\boldsymbol{B}^{(k)} = (\boldsymbol{I} - \rho^{(k)}\boldsymbol{\Delta}^{(k)}(\boldsymbol{u}^{(k)})^T)\boldsymbol{B}^{(k-1)}(\boldsymbol{I} - \rho^{(k)}\boldsymbol{u}^{(k)}(\boldsymbol{\Delta}^{(k)})^T) + \rho^{(k)}\boldsymbol{\Delta}^{(k)}(\boldsymbol{\Delta}^{(k)})^T, \tag{25}$$

$$\boldsymbol{\Delta}^{(k)} = \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k-1)}, \tag{26}$$

$$\boldsymbol{u}^{(k)} = \nabla L(\boldsymbol{\theta}^{(k)}) - \nabla L(\boldsymbol{\theta}^{(k-1)}), \tag{27}$$

$$\rho^{(k)} = \frac{1}{(\boldsymbol{\Delta}^{(k)})^T\boldsymbol{u}^{(k)}}. \tag{28}$$

Through the BFGS optimization process described above, the parameter matrix $\boldsymbol{\theta}$ is solved in the M-step. Finally, when the EM framework converges, we infer the integrated label of each instance based on the learned true label distribution $\boldsymbol{P}_i^*$ as follows:

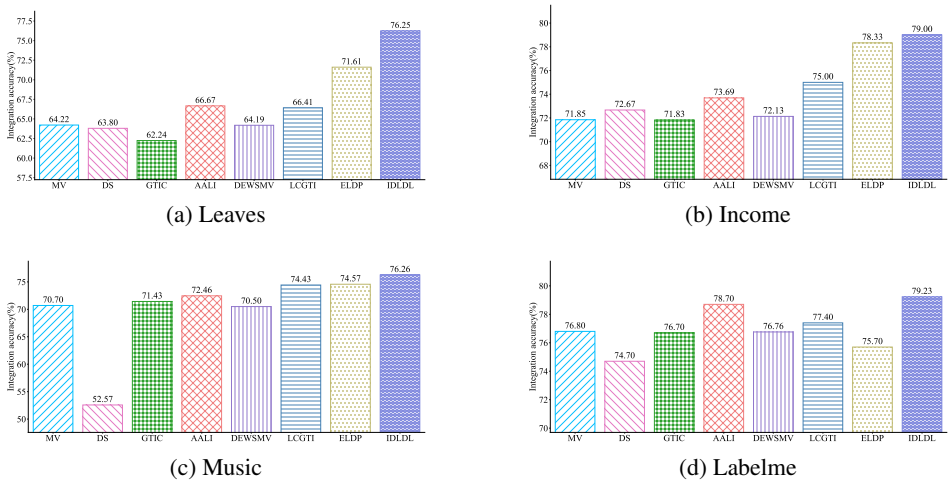
$$\hat{y}_i = \underset{c_{q'} \in \{c_1, c_2, \ldots, c_Q\}}{\operatorname{argmax}} p_{iq'}^*. \tag{29}$$

To this end, the whole learning process of the proposed IDLDL is provided in **Appendix A**, and the time complexity analysis is provided in **Appendix B**.

4 Experiments and Results

4.1 Experimental setup

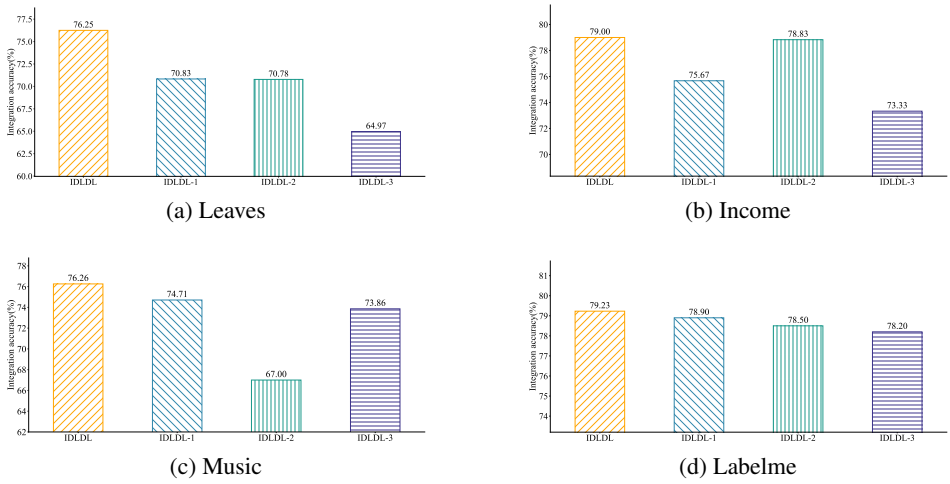
To validate the effectiveness, rationality, and convergence of the proposed LDLLC, we conduct extensive experiments on the four widely used real-world crowdsourced datasets published on the Crowd Environment and its Knowledge Analysis (CEKA) (Zhang et al., 2015) platform. The detailed description of these datasets is provided in **Appendix C**. In order to better train the LDL model, we transform nominal attributes into numerical representations through one-hot encoding and then standardize all attributes. To reduce the impact of randomness, we independently repeat the experiments on each dataset ten times. We compare IDLDL with seven label integration algorithms, including MV (Li et al., 2024), DS (Dawid & Skene, 1979), GTIC (Zhang et al., 2016), AALI (Zhang et al., 2023b), DEWSMV (Tao et al., 2021), LCGTI (Li et al., 2025), and ELDP (Zhang et al., 2025) in terms of integration accuracy. Here, the integration accuracy is defined as the percentage of instances in the dataset whose integrated labels are consistent with their true labels. In our experiments, we implemented IDLDL in MATLAB. Among these comparison algorithms, MV, GTIC, and AALI are algorithms that do not explicitly model the worker quality, while DS, DEWSMV, LCGTI, and ELDP are algorithms that explicitly model the worker quality. We used the existing implementation of DS and GTIC on the CEKA platform, as well as the implementations of AALI, DEWSMV, LCGTI, and ELDP provided by their authors. The parameters of all these comparison algorithms are consistent with those specified in their original papers.



(a) Leaves (b) Income

(c) Music (d) Labelme

Figure 1: Integration accuracy (%) of IDLDL and its competitors on four real-world datasets.



(a) Leaves (b) Income

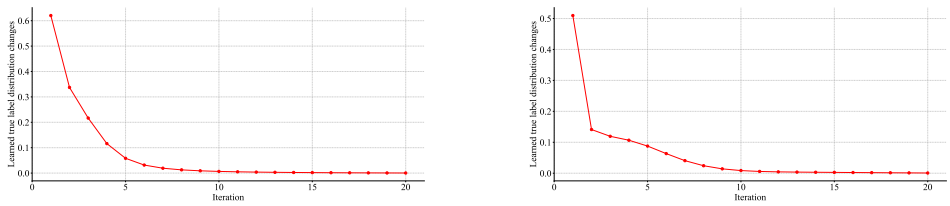
(c) Music (d) Labelme

Figure 2: Integration accuracy (%) of IDLDL and its three variants on four real-world datasets.

4.2 Results and Discussions

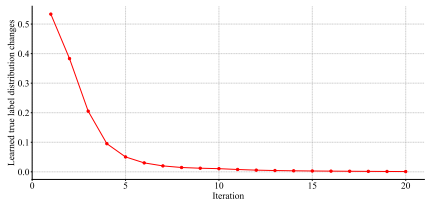
Effectiveness analysis. To validate the effectiveness of the proposed IDLDL, we conduct a contrast experiment on four real-world crowdsourced datasets. The detailed comparison results on these four datasets in terms of integration accuracy are shown in Figure 1. From it, we can see that the integration accuracies of IDLDL on these datasets are 76.25%, 79.00%, 76.26%, and 79.23%, respectively, which are substantially higher than those of other label integration algorithms, including MV, DS, GTIC, AALI, DEWSMV, LCGTI, and ELDP. These results consistently demonstrate that the performance of IDLDL is better than its competitors, which powerfully validates the effectiveness and superiority of IDLDL.

Rationality analysis. In the above experiments, we have only validated the effectiveness of the proposed IDLDL. Now, we construct an ablation experiment on four real-world crowdsourced datasets to validate the rationality of each component in IDLDL. For simplicity, we denote three variants of IDLDL as IDLDL-1, IDLDL-2, and IDLDL-3. Specifically, IDLDL-1 removes the label distribution learning and learns the true label distribution of each instance from its initial label distributions under the EM framework based on the instance correction matrix. IDLDL-2 removes

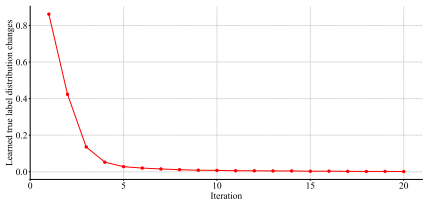


(a) Leaves

(b) Income



(c) Music



(d) Labelme

Figure 3: The sum of learned true label distribution changes across all instances during the EM iterations on four real-world datasets.

the instance difficulty modeling, where the label distribution model is directly trained on the initial label distribution of each instance. IDLDL-3 removes the label distribution enhancement and directly constructs the initial label distribution of each instance from its multiple noisy label set. The integration accuracies of IDLDL and its three variants on four real-world crowdsourced datasets are shown in Figure 2. From it, we can observe that IDLDL consistently achieves the highest integration accuracy compared to its three variants. Specifically, the integration accuracy drops sequentially as components are removed, and the full IDLDL model maintains a superior edge over all variants. These findings further highlight the critical role of label distribution enhancement, instance difficulty modeling, and label distribution learning in improving the performance of IDLDL, thereby validating the rationality of each component in IDLDL.

Convergence analysis. To validate the convergence of the proposed IDLDL on four real-world crowdsourced datasets, we conduct a convergence experiment. We observed that the sum of learned true label distributions changes across all instances during the EM iterations, as shown in Figure 3. From it, we can observe that the learned true label distribution changes are relatively large within the first ten iterations, indicating that IDLDL makes significant corrections to the learned true label distributions. After ten iterations, the changes gradually stabilize and approach zero, indicating that IDLDL has effectively converged to a stable state. These experimental results demonstrate that IDLDL can achieve rapid convergence within a limited number of iterations.

5 Conclusion and Future Work

In this paper, we reveal that most existing label integration algorithms mainly focus on modeling the worker quality while ignoring the instance difficulty. To fill this gap, we propose a novel instance difficulty-aware label distribution learning (IDLDL) algorithm. We introduce an instance correction matrix to model the instance difficulty and a label distribution learning model to map the relationship between instances and their label distributions. Then, we optimize this matrix and model to learn the true label distribution of each instance from its initial label distribution under the EM framework. Extensive experiments show that IDLDL significantly outperforms all its competitors.

The current IDLDL utilizes the information of each instance itself to learn its label distribution by modeling worker quality and instance difficulty. However, it does not exploit the correlations among instances. We believe that the use of techniques such as neighborhood-based learning to model the correlations among instances could improve the performance of the current IDLDL and make its advantage stronger. This is one direction for our future research.

AI USE STATEMENT

In this work, we used generative AI tools only for minor text editing and formatting assistance. We have not used generative AI tools for research ideation, method development, experiment implementation, data analysis, or result interpretation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We submit the code and datasets as supplementary materials, and the details of dataset preprocessing and algorithm implementation are provided in the main text. Once our paper is accepted, we will make the code and datasets publicly available on GitHub.

REFERENCES

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

Chao Chen, Sen Yang, Yasha Wang, Bin Guo, and Daqing Zhang. Crowdexpress: A probabilistic framework for on-time crowdsourced package deliveries. *IEEE Trans. Big Data*, 8(3):827–842, 2022a.

Pengpeng Chen, Hailong Sun, Yili Fang, and Xudong Liu. CONAN: A framework for detecting and handling collusion in crowdsourcing. *Inf. Sci.*, 515:44–63, 2020.

Pengpeng Chen, Hailong Sun, Yongqiang Yang, and Zhijun Chen. Adversarial learning from crowds. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 5304–5312. AAAI Press, 2022b.

Ziqi Chen, Liangxiao Jiang, and Chaoqun Li. Label augmented and weighted majority voting for crowdsourcing. *Inf. Sci.*, 606:397–409, 2022c.

Ziqi Chen, Liangxiao Jiang, and Chaoqun Li. Label distribution-based noise correction for multi-class crowdsourcing. *International Journal of Intelligent Systems*, 37(9):5752–5767, 2022d.

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

Xin Geng. Label distribution learning. *IEEE Trans. Knowl. Data Eng.*, 28(7):1734–1748, 2016.

Guoliang He, Bing Li, Han Wang, and Wenjun Jiang. Cost-effective active semi-supervised learning on multivariate time series data with crowds. *IEEE Trans. Syst. Man Cybern. Syst.*, 52(3):1437–1450, 2022.

Yufei Hu, Liangxiao Jiang, and Chaoqun Li. Instance difficulty-based noise correction for crowdsourcing. *Expert Syst. Appl.*, 212:118794, 2023.

Xiuyi Jia, Weiwei Li, Junyu Liu, and Yu Zhang. Label distribution learning by exploiting label correlations. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3310–3317. AAAI Press, 2018.

Liangxiao Jiang, Lungan Zhang, Chaoqun Li, and Jia Wu. A correlation-based feature weighting filter for naive bayes. *IEEE Trans. Knowl. Data Eng.*, 31(2):201–213, 2019.

Liangxiao Jiang, Hao Zhang, Fangna Tao, and Chaoqun Li. Learning from crowds with multiple noisy label distribution propagation. *IEEE Trans. Neural Networks Learn. Syst.*, 33(11):6558–6568, 2022.

Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. *CoRR*, abs/1411.4086, 2014.

Huiru Li, Liangxiao Jiang, and Chaoqun Li. Certainty weighted voting-based noise correction for crowdsourcing. *Pattern Recognit.*, 150:110325, 2024.

Jiao Li, Liangxiao Jiang, and Wenjun Zhang. Label consistency-based ground truth inference for crowdsourcing. *IEEE Trans. Neural Networks Learn. Syst.*, 36(5):9408–9421, 2025.

Shao-Yuan Li, Yuan Jiang, Nitesh V. Chawla, and Zhi-Hua Zhou. Multi-label learning from crowds. *IEEE Trans. Knowl. Data Eng.*, 31(7):1369–1382, 2019.

Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(12):1650–1654, 2002.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Can Pan, Liangxiao Jiang, and Chaoqun Li. Three-way decision-based label integration for crowdsourcing. *Pattern Recognit.*, 158:111034, 2025.

Tingting Ren, Xiuyi Jia, Weiwei Li, and Shu Zhao. Label distribution learning with label correlations via low-rank approximation. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 3325–3331. ijcai.org, 2019.

Filipe Rodrigues and Francisco C. Pereira. Deep learning from crowds. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1611–1618. AAAI Press, 2018.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In Ying Li, Bing Liu, and Sunita Sarawagi (eds.), *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pp. 614–622. ACM, 2008.

Ashok K. Singh. The EM algorithm and related statistical models. *Technometrics*, 48(1):148, 2006.

Rainer Storn and Kenneth V. Price. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, 11(4):341–359, 1997.

Fangna Tao, Liangxiao Jiang, and Chaoqun Li. Differential evolution-based weighted soft majority voting for crowdsourcing. *Eng. Appl. Artif. Intell.*, 106:104474, 2021.

Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. Spatial crowdsourcing: a survey. *VLDB J.*, 29(1):217–250, 2020.

Ning Wang and Jie Wu. Cost-efficient heterogeneous worker recruitment under coverage requirement in spatial crowdsourcing. *IEEE Trans. Big Data*, 7(2):407–420, 2021.

D. Randall Wilson and Tony R. Martinez. Improved heterogeneous distance functions. *J. Artif. Intell. Res.*, 6:1–34, 1997.

Miao Xu and Zhi-Hua Zhou. Incomplete label distribution learning. In Carles Sierra (ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 3175–3181. ijcai.org, 2017.

Wenjun Yang, Chaoqun Li, and Liangxiao Jiang. Learning from crowds with decision trees. *Knowl. Inf. Syst.*, 64(8):2123–2140, 2022.

Huan Zhang, Liangxiao Jiang, Wenjun Zhang, and Chaoqun Li. Multi-view attribute weighted naive bayes. *IEEE Trans. Knowl. Data Eng.*, 35(7):7291–7302, 2023a.

Jing Zhang, Victor S. Sheng, Bryce Nicholson, and Xindong Wu. CEKA: a tool for mining the wisdom of crowds. *J. Mach. Learn. Res.*, 16:2853–2858, 2015.

Jing Zhang, Victor S. Sheng, Jian Wu, and Xindong Wu. Multi-class ground truth inference in crowdsourcing with clustering. *IEEE Trans. Knowl. Data Eng.*, 28(4):1080–1085, 2016.

Wenjun Zhang, Liangxiao Jiang, and Chaoqun Li. ELDP: enhanced label distribution propagation for crowdsourcing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3):1850–1862, 2025.

Yao Zhang, Liangxiao Jiang, and Chaoqun Li. Attribute augmentation-based label integration for crowdsourcing. *Frontiers Comput. Sci.*, 17(5):175331, 2023b.

Yao Zhang, Liangxiao Jiang, and Chaoqun Li. Instance redistribution-based label integration for crowdsourcing. *Inf. Sci.*, 674:120702, 2024.

Xiang Zheng, Xiuyi Jia, and Weiwei Li. Label distribution learning by exploiting sample correlations locally. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 4556–4563. AAAI Press, 2018.

A THE LEARNING PROCESS OF IDLDL

Algorithm 1 IDLDL

Input: $D = \{(\boldsymbol{x}_i, \{l_{ir}\}_{r=1}^{R})\}_{i=1}^{N}$ — a crowdsourced dataset
Output: $\{\hat{y}_i\}_{i=1}^{N}$ — the integrated labels

1: Initialize the parameter matrix $\boldsymbol{\theta}$
2: Initialize the instance correction matrix $\boldsymbol{S}$
3: Initialize the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(0)}$
4: **for** $r = 1$ to R **do**
5: Estimate the worker quality w_r by Eqs. (1)-(5)
6: **end for**
7: **for** $i = 1$ to N **do**
8: Perform the first label distribution enhancement for $\boldsymbol{x}_i$ by Eq. (6)
9: Perform the second label distribution enhancement for $\boldsymbol{x}_i$ by Eqs. (7)-(11)
10: Merge the two enhanced distributions to obtain the initial label distribution $\boldsymbol{P}_i$ of $\boldsymbol{x}_i$ by Eq. (12)
11: **end for**
12: **repeat**
13: **E-step**:
14: **for** $i = 1$ to N **do**
15: Calculate the predicted label distribution $\hat{\boldsymbol{P}}_i$ of $\boldsymbol{x}_i$ by Eq. (13)
16: Learn the true label distribution $\boldsymbol{P}_i^*$ of $\boldsymbol{x}_i$ by Eqs. (14)-(15)
17: **end for**
18: **M-step**:
19: **for** $i = 1$ to N **do**
20: **for** $q' = 1$ to Q **do**
21: **for** $q = 1$ to Q **do**
22: Update the element $S_{qq'}$ of the instance correction matrix $\boldsymbol{S}$ by Eq. (16)
23: **end for**
24: **end for**
25: **end for**
26: **for** $q = 1$ to Q **do**
27: Normalize each column of the instance correction matrix $\boldsymbol{S}$ by Eq. (17)
28: **end for**
29: Construct the function $L(\boldsymbol{\theta})$ by Eqs. (18)-(19);
30: Calculate the gradient $\nabla L(\boldsymbol{\theta})$ by Eq. (20)
31: $k \leftarrow 1$
32: **repeat**
33: Calculate the search direction $\boldsymbol{p}^{(k)}$ by Eq. (21)
34: Calculate the step size $\eta^{(k)}$ by Eqs. (22)-(23)
35: Calculate the parameter matrix $\boldsymbol{\theta}^{(k)}$ by Eq. (24)
36: Calculate the gradient $\nabla L(\boldsymbol{\theta}^{(k)})$ by Eq. (20)
37: Calculate the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(k)}$ by Eqs. (25)-(28)
38: $k \leftarrow k + 1$
39: **until** $\|\nabla L(\boldsymbol{\theta}^{(k)})\| < 10^{-4}$
40: **until** $\boldsymbol{P}_i^*$ converges
41: **for** $i = 1$ to N **do**
42: Infer the integrated label $\hat{y}_i$ of $\boldsymbol{x}_i$ by Eq. (29)
43: **end for**
44: **return** $\{\hat{y}_i\}_{i=1}^{N}$

B THE TIME COMPLEXITY ANALYSIS OF IDLDL

In **Algorithm 1**. Lines 1-3 initialize the instance correction matrix, the parameter matrix, and the approximation of the inverse Hessian matrix with time complexities of $O(Q^2 + QM + (QM)^2)$. Lines 4-6 estimate the worker quality of all workers with a time complexity of $O(RM|D_r|)$. Lines 4-6 perform the first label distribution enhancement with a time complexity of $O(NR)$. Lines 7-11 obtain the initial label distribution for each instance by the dual label distribution enhancement method with a time complexity of $O(M|D_{r^*}| + NQM + NQ^2)$. Lines 13-17 calculate the predicted label distribution and learn the true label distribution of each instance with a time complexity of $O(NQM + NQ^2)$. Lines 18-28 update the instance correction matrix with a time complexity

of $O(NQ^2 + Q)$. Lines 29-38 update the parameter matrix using BFGS with a time complexity of $O(KNQM)$, where K is the number of internal iterations in the BFGS optimization process. Therefore, the time complexity of a single EM iteration is $O(NQM + NQ^2 + Q + KNQM)$, and the total time complexity of the whole EM framework optimization is $O(T(NQM + NQ^2 + Q + KNQM))$, where T is the number of iterations in the EM framework. Lines 40-42 infer the integrated label of each instance according to the learned true label distribution with a time complexity of $O(NQ)$. Therefore, the overall time complexity of IDLDL is $O(Q^2+QM+(QM)^2+RM|D_r|+NR+M|D_{r^*}|+QNM+NQ^2+T(NQM+NQ^2+Q+KNQM)+NQ)$. In practical scenarios, the number of both instances labeled by worker $|D_r|$ and instances labeled by high-quality worker $|D_{r^*}|$ is smaller than the number of instances N. Therefore, if only the highest order terms are taken, the time complexity of IDLDL simplifies to $O((QM)^2+RMN+TNQ^2+TKNQM)$.

C THE DETAILED DESCRIPTION OF FOUR REAL-WORLD DATASETS

The "Leaves" dataset is a classic multiclass dataset, which can distinguish six types of leaves according to their shape and other attributes. The "Income" dataset is a classic binary crowdsourced dataset, which aims to predict whether a person makes over $50K$ a year. The "Music" dataset is a classic music classification dataset containing ten classes. The "LabelMe" dataset is a multiclass image classification dataset, which consists of all the 256×256 images with the eight classes. The detailed description of four real-world datasets is provided in Table 1. Here, "#Instances" denotes the number of instances, "#Attributes" denotes the number of attributes, "#Classes" denotes the number of classes, "#Attribute type" denotes the type of attributes, including numeric, nominal, and hybrid attributes, "#Workers" denotes the number of crowd workers, and "#Labels" denotes the number of labels provided by crowd workers.

Dataset	#Instances	#Attributes	#Classes	Attribute type	#Workers	#Labels
Leaves	384	64	6	numeric	83	3840
Income	600	10	2	hybrid	67	6000
Music	700	31	10	numeric	44	2946
LabelMe	1000	512	8	numeric	59	2547

Table 1: Descriptions of four real-world datasets