LABEL DISTRIBUTION LEARNING-BASED LABEL COMPLETION FOR CROWDSOURCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Label completion serves as a preprocessing step to handle the sparse crowdsourced label matrix problem, significantly boosting the effectiveness of the downstream label integration. In recent advances, worker modeling has been proven to be a powerful approach to label completion. However, existing algorithms usually treat the labels annotated by workers as single and deterministic labels, thereby failing to capture the class ambiguity and uncertainty in worker annotation and consequently limiting their performance. To address this limitation, we propose a novel label distribution learning-based label completion (LDLLC) algorithm. Specifically, for each worker, we first design a class membership estimation method to obtain the label distribution of each instance annotated by it and develop a worker similarity estimation method to find its similar workers (including itself). Then, we utilize the instances annotated by its similar workers to augment the instances annotated by it and use the worker similarity to weight the augmented instances. Subsequently, we train a label distribution learning model on the augmented and weighted instances to capture the class ambiguity and uncertainty in worker annotation. Finally, we use the trained model to predict the label distribution of each unannotated instance and complete the missing label. Extensive experiments show that LDLLC significantly outperforms all its competitors.

1 INTRODUCTION

Supervised learning has achieved significant success in numerous real-world applications, but its success heavily relies on large-scale and high-quality annotated data (Wang & Wu, 2021; Chen et al., 2022a; Hu et al., 2023). Traditionally, such data is obtained by hiring domain experts or well-trained workers, but this is both time-consuming and expensive (Lu et al., 2023; Dong et al., 2022). Fortunately, the emergence of crowdsourcing offers a time-saving and economical solution to obtain annotated data by employing crowd workers (Li et al., 2019b; Dizaji et al., 2020). However, due to the lack of professional knowledge and specialized training, the labels collected from crowd workers contain a lot of noise (Chen et al., 2020). To address this issue, repeated annotation is commonly employed, where each instance is annotated by different crowd workers to obtain multiple noisy labels (Li et al., 2019a; Pan et al., 2025a). After that, label integration is used to infer the unknown true label of each instance from its multiple noisy labels (Zhang et al., 2025b; Li et al., 2025).

In recent years, label integration has attracted significant research interest, leading to the development of numerous algorithms from diverse perspectives (Zheng et al., 2017; Zhang, 2022). A common consensus among these algorithms is that when the quality of workers' annotations is better than random annotation, the more noisy labels an instance receives, the more accurately its unknown true label can be inferred (Chen et al., 2022b; Zhang et al., 2024). However, in real-world crowdsourcing scenarios, most workers typically only annotate a few instances, and thus most instances do not receive sufficient labels, resulting in a highly sparse crowdsourced label matrix (Jung & Lease, 2012). This sparsity prevents label integration from obtaining sufficient information to learn and infer the unknown true label, thereby severely limiting its performance (Pan et al., 2025b). To break this bottleneck, a small number of researchers have paid attention to the importance of preprocessing the crowdsourced label matrix and proposed a few label completion algorithms.

Among them, Jung & Lease (2012) first introduced the concept of label completion in crowdsourcing scenarios and proposed a label completion algorithm based on probabilistic matrix factorization

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.

(PMF). Zhou & He (2016) proposed a label completion algorithm based on tensor augmentation and completion (TAC). Recent advancements have demonstrated that worker modeling is a powerful approach to improve the performance of label completion. For example, Yang et al. (2024) proposed a PMF-based three-stage label completion (PMF-TLC) algorithm that models workers to estimate label quality and utilizes probabilistic matrix factorization to complete missing labels. Wu et al. (2025) proposed a worker similarity-based label completion (WSLC) algorithm that models workers to measure worker similarity and leverages similar workers to complete missing labels. Zhang et al. (2025a) proposed a transfer learning-based label completion (TLLC) algorithm that enhances worker modeling through pretraining on high-confidence instances and fine-tuning on workers' annotated instances to complete missing labels. Although these label completion algorithms based on worker modeling have shown remarkable effectiveness, they usually treat the labels annotated by workers as single and deterministic labels, thereby failing to capture class ambiguity and uncertainty in worker annotation and consequently limiting their performance. To address this limitation, we propose a novel label distribution learning-based label completion (LDLLC) algorithm. To sum up, our main contributions can be highlighted as:

- We reveal the limitations of existing algorithms, which usually treat the labels annotated by workers as single and deterministic labels, thereby failing to capture the class ambiguity and uncertainty in worker annotation and consequently limiting their performance.
- We design a class membership estimation method to obtain the label distribution of each annotated instance and develop a data augmentation and weighting method to alleviate the insufficiency of annotation information during label distribution learning model training.
- We propose a novel label distribution learning-based label completion (LDLLC) algorithm and conduct extensive experiments on both real-world and simulated crowdsourced datasets to demonstrate that LDLLC significantly outperforms all the other advanced competitors.

2 Related Work

Over the past few years, a large number of label integration algorithms have been proposed from various perspectives. Majority voting (MV) (Sheng et al., 2008) is the simplest but often effective label integration algorithm, which directly chooses the most frequent label as the integrated label for each instance. However, MV assumes that all crowd workers possess identical annotation quality, which is impossible in real-world scenarios and limits the performance of MV. Consequently, many more sophisticated algorithms have been proposed successively. For example, Li & Yu (2014) proposed iterative weighted majority voting (IWMV), which optimizes error rate bounds to find reliable workers. Tao et al. (2021) proposed differential evolution-based weighted soft majority voting (DEWSMV), which uses differential evolution to learn optimal worker weights, thereby improving integration accuracy. Jiang et al. (2022) proposed multiple noisy label distribution propagation (MNLDP), which estimates the label distribution of each instance and then propagates the distribution to its nearest neighbors. Zhang et al. (2023) proposed attribute augmentation-based label integration (AALI), which enriches the original feature space and then trains multiple classifiers on reliable instances to predict all instances. Ying et al. (2024) proposed a label integration algorithm with graph neural networks (LAGNN), which models multiple correlations using graph neural networks and leverages high-quality workers identified through labeling similarity to improve integration accuracy. Hu et al. (2026) proposed a dual-view label integration (DVLI), which iteratively leverages the consistent information and complementary information from two views to update labels, thereby improving integration accuracy.

However, in real-world crowdsourcing scenarios, each worker typically annotates only a few instances, resulting in a highly sparse crowdsourced label matrix, which adversely affects the performance of label integration algorithms. To address this sparsity issue, a small number of researchers have recognized the importance of preprocessing the crowdsourced label matrix and have proposed a few label completion algorithms. Specifically, Jung & Lease (2012) first introduced the concept of label completion in crowdsourcing scenarios and proposed a label completion algorithm based on probabilistic matrix factorization (PMF), which utilizes probabilistic matrix factorization to decompose the crowdsourced label matrix into a worker feature matrix and an instance feature matrix. Zhou & He (2016) proposed a label completion algorithm based on tensor augmentation and completion (TAC), which models the crowdsourced data as a three-dimensional tensor composed of

workers, instances, and classes. Recent advancements have demonstrated that worker modeling is a powerful approach to improve the performance of label completion. For example, Yang et al. (2024) proposed a PMF-based three-stage label completion (PMF-TLC) algorithm, which models workers by training a classifier, combines a confidence-based strategy to flip low-quality original labels, and then filters out low-quality output labels. Wu et al. (2025) proposed a worker similarity-based label completion (WSLC) algorithm, which constructs worker feature vectors to estimate worker similarity, and then uses the label information from similar workers to complete missing labels. Zhang et al. (2025a) proposed a transfer learning-based label completion (TLLC) algorithm, which pretrains a Siamese network on high-confidence instances and then fine-tunes it on each worker's annotated instances to learn the annotation pattern of each worker to complete missing labels.

3 THE PROPOSED LDLLC

As discussed above, existing label completion algorithms usually treat the labels annotated by workers as single and deterministic labels, thereby failing to capture class ambiguity and uncertainty in worker annotation and consequently limiting their performance. In real-world crowdsourcing scenarios, when instances have ambiguous class boundaries or overlapping features, workers' annotation patterns inherently exhibit class ambiguity and uncertainty. To the best of our knowledge, label distribution learning (Geng, 2016; Xu & Zhou, 2017; Chen et al., 2022c) provides a natural and effective way to model such class ambiguity and uncertainty by learning a mapping from instance features to label distributions. Therefore, we incorporate label distribution learning into label completion to improve the performance of label completion. To achieve this objective, several issues need to be addressed: 1) How to obtain a label distribution of each annotated instance? 2) How to alleviate the insufficiency of annotation information? 3) How to effectively train a label distribution learning model? Correspondingly, we propose LDLLC that mainly includes three components: class membership estimation, data augmentation and weighting, and label distribution learning, which are used to address these issues. The overall framework of LDLLC is provided in **Appendix A**, and the detailed process will be introduced in the following subsections.

3.1 CLASS MEMBERSHIP ESTIMATION

In crowdsourcing scenarios, a crowdsourced dataset D is commonly denoted as $\{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$, where N denotes the number of instances. The i-th instance $\boldsymbol{x}_i$ in D can be denoted by a feature vector $\{x_{im}\}_{m=1}^{M}$, where M denotes the number of features and x_{im} denotes the feature value of instance $\boldsymbol{x}_i$ on the m-th feature variable $\boldsymbol{A}_m$. Each instance $\boldsymbol{x}_i$ is associated with the multiple noisy labels $\boldsymbol{L}_i = \{l_{ir}\}_{r=1}^{R}$, where l_{ir} denotes the label provided by the r-th crowd worker u_r on instance $\boldsymbol{x}_i$. l_{ir} takes the value from a specified 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 implies that u_r does not annotate instance $\boldsymbol{x}_i$.

Since a single deterministic label cannot adequately reflect the inherent class ambiguity and uncertainty in worker annotation, we design a class membership estimation method to obtain the label distribution of each instance annotated by each worker. To achieve reliable class membership probability estimation, we first construct a dataset D_r for each worker u_r, which contains all instances annotated by u_r from D along with the corresponding labels provided by u_r, as follows:

$$D_r = \{(\boldsymbol{x}_i, l_{ir}) \mid \text{ for } i = 1, \ldots, N,\ l_{ir} \neq -1\}. \tag{1}$$

Then, we partition all instances D_r annotated by u_r into Q clusters $\{D_{rq}\}_{q=1}^{Q}$ according to their class labels, where D_{rq} can be defined as follows:

$$D_{rq} = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in D_r \wedge \delta(l_{ir}, c_q)\}, \tag{2}$$

where $\delta(\cdot, \cdot)$ is an indicator function that outputs 1 if its two input values are equal, and 0 otherwise.

Intuitively, an instance is more likely to belong to a class if it is closer to the instances of that class in the feature space. That is, instances within the same class should be closer, while instances from different classes should be farther. Therefore, given an instance $\boldsymbol{x}_{ir}$ annotated by worker u_r, we estimate the class membership probability of $\boldsymbol{x}_{ir}$ belonging to c_q based on the average distance $\bar{d}_{irq}$ from $\boldsymbol{x}_{ir}$ to all instances in D_{rq}. $\bar{d}_{irq}$ can be calculated as follows:

$$\bar{d}_{irq} = \frac{\sum_{j=1}^{|D_{rq}|} d(\boldsymbol{x}_{ir}, \boldsymbol{x}_j)}{|D_{rq}|}, \tag{3}$$

where $d(\cdot,\cdot)$ denotes the heterogeneous Manhattan-overlap metric (HMOM) (Wilson & Martinez, 1997). HMOM uses normalized Manhattan distance for numerical features and overlap distance for nominal features. The distance between two instances $\boldsymbol{x}_{ir}$ and $\boldsymbol{x}_j$ is given as $d(\boldsymbol{x}_{ir}, \boldsymbol{x}_j) = \sum_{m=1}^{M} d_m(\boldsymbol{x}_{ir}, \boldsymbol{x}_j)$, where $d_m(\boldsymbol{x}_{ir}, \boldsymbol{x}_j)$ denotes the distance between instances $\boldsymbol{x}_{ir}$ and $\boldsymbol{x}_j$ on A_m, which can be calculated as follows:

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

where $max_m - min_m$ is used to normalize Manhattan distance, max_m and min_m denote the maximum and minimum values of A_m, respectively.

The smaller $\bar{d}_{irq}$ is, the more likely $\boldsymbol{x}_{ir}$ belongs to c_q. Therefore, $\bar{d}_{irq}$ is negatively correlated to the class membership probability of $\boldsymbol{x}_{ir}$ belonging to c_q. Let $\boldsymbol{Mem}_{ir} = \{Mem_{irq}\}_{q=1}^{Q}$ denote the class membership probability vector of $\boldsymbol{x}_{ir}$ belonging to each class. We estimate Mem_{irq} by normalizing $\bar{d}_{irq}$ as follows:

$$Mem_{irq} = \frac{Max(\{\bar{d}_{irq}\}_{q=1}^{Q}) - \bar{d}_{irq}}{Max(\{\bar{d}_{irq}\}_{q=1}^{Q}) - Min(\{\bar{d}_{irq}\}_{q=1}^{Q})}, \tag{5}$$

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

After obtaining $\boldsymbol{Mem}_{ir}$, we can further obtain the label distribution $\boldsymbol{P}_{ir} = \{p_{irq}\}_{q=1}^{Q}$ of x_{ir}, where p_{irq} can be calculated as follows:

$$p_{irq} = \frac{Mem_{irq}}{\sum_{q=1}^{Q} Mem_{irq}}. \tag{6}$$

3.2 Data augmentation and weighting

Since each worker annotates only a few instances in crowdsourcing scenarios, the information provided by an individual worker is often insufficient to effectively model its annotation pattern. To alleviate this issue, we develop a worker similarity estimation method and a data augmentation and weighting method to enrich the available information for each worker. Specifically, we learn a feature vector $\boldsymbol{V}_r$ based on the label variable $\boldsymbol{C}_r$ and each feature variable $\boldsymbol{A}_{rm}$ in D_r. We denote $\boldsymbol{V}_r$ as $\{v_{rm}\}_{m=1}^{M}$, where v_{rm} captures the correlation between the annotation pattern of worker u_r and the m-th feature variable. Here, we adopt the Pearson correlation to learn the feature vector as follows:

$$cor(\boldsymbol{C}_r, \boldsymbol{A}_{rm}) = \frac{\sum_{i=1}^{|D_r|}(l_{ir} - \bar{C}_r)(x_{irm} - \bar{A}_{rm})}{\sqrt{\sum_{i=1}^{|D_r|}(l_{ir} - \bar{C}_r)^2}\sqrt{\sum_{i=1}^{|D_r|}(x_{irm} - \bar{A}_{rm})^2}}, \tag{7}$$

where x_{irm} denotes the m-th feature value of the i-th instance in D_r. $\bar{C}_r$ and $\bar{A}_{rm}$ denote the mean values of the label variable $\boldsymbol{C}_r$ and the feature variable $\boldsymbol{A}_{rm}$, respectively.

Since the label variable is discrete, we adopt a weighted Pearson correlation to measure the relationship between $\boldsymbol{C}_r$ and $\boldsymbol{A}_{rm}$. Specifically, we first decompose $\boldsymbol{C}_r$ into multiple binary label variables, each binary label variable corresponding to a value of $\boldsymbol{C}_r$. Each binary label variable takes the value 1 when the corresponding value of $\boldsymbol{C}_r$ occurs and 0 otherwise. Next, we calculate the Pearson correlation between each binary label variable and $\boldsymbol{A}_{rm}$ and aggregate the results via a weighted summation based on the prior probabilities of the label values. If the original feature variable $\boldsymbol{A}_{rm}$ is also discrete, it is similarly decomposed into multiple binary feature variables. Then, we calculate the Pearson correlation between each binary label variable of $\boldsymbol{C}_r$ and each binary feature variable of $\boldsymbol{A}_{rm}$. Consequently, the feature value v_{rm} of worker u_r is calculated as follows:

$$v_{rm} = \begin{cases} \sum_{l_r} p(\boldsymbol{C}_r = l_r) cor(\boldsymbol{C}_{br}, \boldsymbol{A}_{rm}),\ if\ \boldsymbol{A}_{rm}\ is\ numeric \\ \sum_{l_r}\sum_{x_{rm}} p(\boldsymbol{C}_r = l_r, \boldsymbol{A}_{rm} = x_{rm}) cor(\boldsymbol{C}_{br}, \boldsymbol{A}_{brm}),\ otherwise \end{cases}, \tag{8}$$

where l_r and x_{rm} denote the values of $\boldsymbol{C}_r$ and $\boldsymbol{A}_{rm}$, respectively. $\boldsymbol{C}_{br}$ and $\boldsymbol{A}_{brm}$ denote the corresponding binary label variable of $\boldsymbol{C}_r$ and the binary feature variable of $\boldsymbol{A}_{rm}$, respectively. The prior probabilities $p(\boldsymbol{C}_r = l_r)$ and $p(\boldsymbol{C}_r = l_r, \boldsymbol{A}_{rm} = x_{rm})$ can be calculated as follows:

$$p(\boldsymbol{C}_r = l_r) = \frac{\sum_{i=1}^{|D_r|} \delta(l_{ir}, l_r)}{|D_r|}, \tag{9}$$

$$p(\boldsymbol{C}_r = l_r, \boldsymbol{A}_{rm} = x_{rm}) = \frac{\sum_{i=1}^{|D_r|} \delta(l_{ir}, l_r)\delta(x_{irm}, x_{rm})}{|D_r|}. \tag{10}$$

Based on Eqs.(7)-(10), we obtain the feature vector $\boldsymbol{V}_r$ for each worker u_r. Subsequently, we estimate worker similarity $s(u_r, u_{r'})$ between u_r and $u_{r'}$ using the cosine similarity based on $\boldsymbol{V}_r$ and $\boldsymbol{V}_{r'}$, which is calculated as follows:

$$cos(\boldsymbol{V}_r, \boldsymbol{V}_{r'}) = \frac{\boldsymbol{V}_r \cdot \boldsymbol{V}_{r'}}{|\boldsymbol{V}_r||\boldsymbol{V}_{r'}|} = \frac{\sum_{m=1}^{M} v_{rm} v_{r'm}}{\sqrt{\sum_{m=1}^{M} v_{rm}^2}\sqrt{\sum_{m=1}^{M} v_{r'm}^2}}. \tag{11}$$

Since $cos(\boldsymbol{V}_r, \boldsymbol{V}_{r'})$ takes values in [-1,1], we apply max-min normalization to map these similarities into the interval [0,1], which is calculated as follows:

$$s(u_r, u_{r'}) = \frac{cos(\boldsymbol{V}_r, \boldsymbol{V}_{r'}) - (-1)}{1 - (-1)}. \tag{12}$$

Based on the above estimation of worker similarity, we can obtain a set of K most similar workers for each worker u_r, denoted as $\{u_{rk}\}_{k=1}^{K}$ (including u_r itself). To ensure that the number of similar workers can adaptively match datasets of different scales, we use a hyperparameter α to calculate the number of similar workers K as follows:

$$K = \alpha \cdot R, \tag{13}$$

where the hyperparameter α takes values in $[0, 1]$.

To alleviate the insufficiency of annotation information provided by each worker u_r, we construct an augmented and weighted dataset D_r^* for each worker u_r. Specifically, for each worker u_r, we utilize the instances annotated by its similar workers to augment the instances annotated by it and use the worker similarity to weight the augmented instances as follows:

$$\begin{aligned} D_r^* = \big\{ (\boldsymbol{x}_{irk}, \boldsymbol{P}_{irk}, w_{irk}) \mid w_{irk} = s(u_r, u_{rk}), \\ \text{for } k = 1, 2, \ldots, K, \text{ for } i = 1, 2, \ldots, |D_{rk}| \big\}, \end{aligned} \tag{14}$$

3.3 Label distribution learning

With D_r^* prepared, we aim to train a label distribution learning model on this dataset to capture the class ambiguity and uncertainty in worker annotation. In D_r^*, the label distribution of each instance annotated by worker u_r can be approximately regarded as the true label distribution reflecting the annotation pattern of this worker. According to the theoretical foundation of label distribution learning, the objective function $T(\boldsymbol{\theta}_r)$ is to minimize the Kullback-Leibler (KL) (Zheng et al., 2018) divergence between the predicted label distribution $\widehat{\boldsymbol{P}}_{ir}$ and the label distribution $\boldsymbol{P}_{ir}$ of each instance annotated by worker u_r, where $T(\boldsymbol{\theta}_r)$ can be calculated as follows:

$$T(\boldsymbol{\theta}_r) = \sum_{i=1}^{|D_r^*|} \mathrm{KL}(\boldsymbol{P}_{ir} \parallel \widehat{\boldsymbol{P}}_{ir}) = \sum_{i=1}^{|D_r^*|}\sum_{q=1}^{Q} p_{irq} \ln \frac{p_{irq}}{\widehat{p}_{irq}} = -\sum_{i=1}^{|D_r^*|}\sum_{q=1}^{Q} p_{irq} \ln \widehat{p}_{irq}, \tag{15}$$

where $\boldsymbol{\theta}_r$ is the parameter matrix of the label distribution learning model. In this paper, $\boldsymbol{\theta}_r$ is initialized such that its diagonal elements are set to 1 and all remaining elements are set to 0. Furthermore, since the label distribution $\boldsymbol{P}_{ir}$ is a fixed constant, the term $\sum_{i=1}^{|D_r^*|}\sum_{q=1}^{Q} p_{irq} \ln p_{irq}$ is constant with respect to the parameter matrix $\boldsymbol{\theta}_r$ and can be directly omitted during optimization.

However, this objective function implicitly assumes that all instances annotated by each worker u_r are equally reliable and contribute equally to model training. Since D_r^* contains instances annotated

by both worker u_r and its similar workers $\{u_{rk}\}_{k=1}^{K}$, these instances from different workers usually have different reliability levels. Therefore, we introduce the worker similarity weights into the objective function to design a weighted label distribution learning model as follows:

$$T(\boldsymbol{\theta}_r) = -\sum_{i=1}^{|D_r^*|} w_{ir} \sum_{q=1}^{Q} p_{irq} \ln \widehat{p}_{irq}. \tag{16}$$

In the above objective function, the predicted label distribution $\widehat{\boldsymbol{P}}_{ir}$ is implicitly defined through a parametric model. In label distribution learning, the maximum entropy model (Ren et al., 2019) is adopted to define the predicted label distribution. Therefore, the class probability $\widehat{p}_{irq}$ of class c_q for the instance $\boldsymbol{x}_{ir}$ annotated by worker u_r is defined as follows:

$$\widehat{p}_{irq} = \frac{\exp\left(\sum_{m=1}^{M} \theta_{rqm} x_{irm}\right)}{\sum_{q=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta_{rqm} x_{irm}\right)}, \tag{17}$$

where θ_{rqm} is the element of the parameter matrix $\boldsymbol{\theta}_r$.

Then, we substitute Eq.(17) into Eq.(16) to further obtain the final objective function $T(\boldsymbol{\theta}_r)$, which can be formulated as follows:

$$T(\boldsymbol{\theta}_r) = \sum_{i=1}^{|D_r^*|} w_{ir} \left[\ln \sum_{q=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta_{rqm} x_{irm}\right) - \sum_{q=1}^{Q} p_{irq} \sum_{m=1}^{M} \theta_{rqm} x_{irm} \right]. \tag{18}$$

Subsequently, 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 $T(\boldsymbol{\theta}_r)$ using BFGS, the optimization process primarily depends on the gradient $\nabla T(\boldsymbol{\theta}_r^{(t)})$ and the matrix $\boldsymbol{B}^{(t)}$ at each iteration. For each element θ_{rqm}, we calculate the corresponding gradient component $\frac{\partial T(\boldsymbol{\theta}_r^{(t)})}{\partial \theta_{rqm}}$ of $\nabla T(\boldsymbol{\theta}_r^{(t)})$ at the t-th iteration as follows:

$$\frac{\partial T(\boldsymbol{\theta}_r^{(t)})}{\partial \theta_{rqm}} = \sum_{i=1}^{|D_r^*|} w_{ir} \left[\frac{\exp\left(\sum_{m=1}^{M} \theta_{rqm}^{(t)} x_{irm}\right)}{\sum_{q=1}^{Q} \exp\left(\sum_{m=1}^{M} \theta_{rqm}^{(t)} x_{irm}\right)} - p_{irq} \right] x_{irm}. \tag{19}$$

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

$$\boldsymbol{p}^{(t)} = -\boldsymbol{B}^{(t-1)} \nabla T(\boldsymbol{\theta}_r^{(t-1)}). \tag{20}$$

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

$$T(\boldsymbol{\theta}_r^{(t-1)} + \eta^{(t)} \boldsymbol{p}^{(t)}) \leq T(\boldsymbol{\theta}_r^{(t-1)}) + c_1 \eta^{(t)} \nabla T(\boldsymbol{\theta}_r^{(t-1)})^T \boldsymbol{p}^{(t)}, \tag{21}$$

$$\left| \nabla T(\boldsymbol{\theta}_r^{(t-1)} + \eta^{(t)} \boldsymbol{p}^{(t)})^T \boldsymbol{p}^{(t)} \right| \leq c_2 \left| \nabla T(\boldsymbol{\theta}_r^{(t-1)})^T \boldsymbol{p}^{(t)} \right|, \tag{22}$$

where $0 < c_1 < c_2 < 1$, c_1 and c_2 are set as 10^{-4} and 0.9, respectively.

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

$$\boldsymbol{\theta}_r^{(t)} = \boldsymbol{\theta}_r^{(t-1)} + \eta^{(t)} \boldsymbol{p}^{(t)}. \tag{23}$$

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

$$\boldsymbol{B}^{(t)} = (\boldsymbol{I} - \rho^{(t)}\boldsymbol{\Delta}^{(t)}(\boldsymbol{u}^{(t)})^T)\boldsymbol{B}^{(t-1)}(\boldsymbol{I} - \rho^{(t)}\boldsymbol{u}^{(t)}(\boldsymbol{\Delta}^{(t)})^T) + \rho^{(t)}\boldsymbol{\Delta}^{(t)}(\boldsymbol{\Delta}^{(t)})^T, \tag{24}$$

$$\boldsymbol{\Delta}^{(t)} = \boldsymbol{\theta}_r^{(t)} - \boldsymbol{\theta}_r^{(t-1)}, \tag{25}$$

$$\boldsymbol{u}^{(t)} = \nabla T(\boldsymbol{\theta}_r^{(t)}) - \nabla T(\boldsymbol{\theta}_r^{(t-1)}), \tag{26}$$

$$\rho^{(t)} = \frac{1}{(\boldsymbol{\Delta}^{(t)})^T\boldsymbol{u}^{(t)}}. \tag{27}$$

After the label distribution model is trained on D_r^*, we obtain the parameter matrix $\boldsymbol{\theta}_r$ that characterizes the annotation pattern of worker u_r. For each unannotated instance $\boldsymbol{x}_{ir}$ of each worker u_r, we estimate its predicted label distribution $\boldsymbol{P}_{ir}^* = \{p_{irq}^*\}_{q=1}^Q$ based on instance features, where p_{irq}^* can be calculated as follows:

$$p_{irq}^* = \frac{\exp\left(\sum_{m=1}^M \theta_{rqm}x_{irm}\right)}{\sum_{q=1}^Q \exp\left(\sum_{m=1}^M \theta_{rqm}x_{irm}\right)}. \tag{28}$$

Although the trained model can provide predictions for the unannotated instances of worker u_r, directly using the predicted label distributions for completion may introduce substantial noise. To address this issue, we adopt the class margin $margin_{ir}$ to evaluate the reliability of the predicted label distributions as follows:

$$margin_{ir} = Max(\boldsymbol{P}_{ir}^*) - Sec(\boldsymbol{P}_{ir}^*), \tag{29}$$

where $Sec(\cdot)$ return the second-largest value of the set. If $margin_{ir}$ of instance $\boldsymbol{x}_{ir}$ is less than $1/Q$, it indicates that a definitive judgment cannot be made for this instance based on the historical annotation pattern of worker u_r. Otherwise, it indicates that the label with the highest probability is highly likely to be the one that worker u_r would assign to this instance. Therefore, if $margin_{ir}$ satisfies the condition $margin_{ir} \geq 1/Q$, we directly complete the missing label $\hat{l}_{ir}$ as follows:

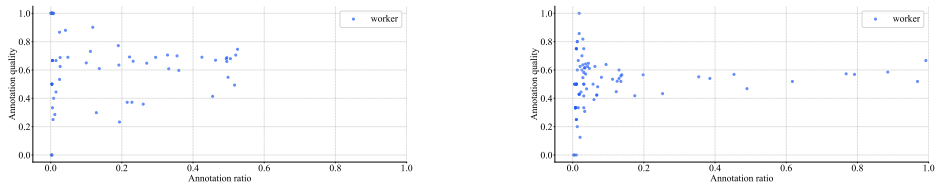
$$\hat{l}_{ir} = \underset{c_q \in \{c_1, c_2, \ldots, c_Q\}}{\operatorname{argmax}} p_{irq}^* . \tag{30}$$

In summary, the whole learning process of LDLLC is provided in **Appendix B**, and the time complexity analysis is provided in **Appendix C**. After the above process, the completed D' can be fed into existing label integration algorithms to enhance their performance.

4 Experiments and Results

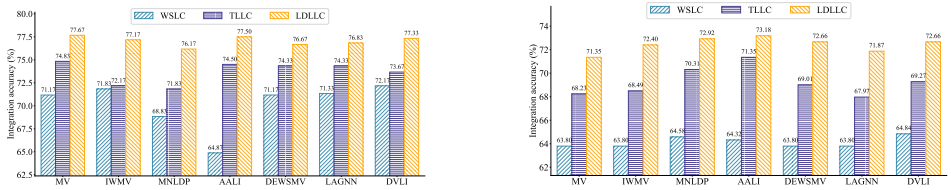
4.1 Experimental setup

In this section, we conduct a series of experiments on both real-world and simulated datasets to validate the effectiveness of LDLLC. We use WSLC (Wu et al., 2025), TLLC (Zhang et al., 2025a), and LDLLC to complete missing labels, respectively, and compare their effectiveness in terms of the integration accuracy of each label integration algorithm, which represents the percentage of instances in the dataset whose integrated labels are consistent with their true labels. In our experiments, the label integration algorithms used include MV (Sheng et al., 2008), IWMV (Li & Yu, 2014), MNLDP (Jiang et al., 2022), AALI (Zhang et al., 2023), DEWSMV (Tao et al., 2021), LAGNN (Ying et al., 2024), and DVLI (Hu et al., 2026). We implement LDLLC in MATLAB. For MV and IWMV, we use the existing implementations on the Crowd Environment and its Knowledge Analysis (CEKA) (Zhang et al., 2015) platform. For MNLDP, AALI, DEWSMV, LAGNN, DVLI, WSLC, and TLLC, we use the implementations provided by their authors. For the parameter settings for all the above comparison algorithms, we set them consistent with those specified in their original papers. For the hyperparameter α of our proposed LDLLC, we set α to 0.5.



(a) Income (b) Leaves

Figure 1: Each worker's annotation ratio and annotation quality distribution on the "Income" and "Leaves" datasets.



(a) Income (b) Leaves

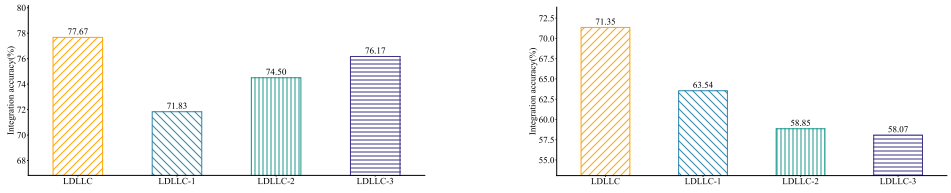
Figure 2: Integration accuracy (%) comparisons of seven label integration algorithms after label completion by WSLC, TLLC, and LDLLC on the "Income" and "Leaves" datasets.

4.2 Experiments on two real-world datasets

To validate the effectiveness of LDLLC, we conduct experiments on the whole two widely used real-world crowdsourced datasets published on the CEKA platform. These two datasets are both collected from the Amazon Mechanical Turk (AMT) platform. The "Income" dataset contains 600 instances and 6000 labels annotated by 67 workers. The "Leaves" dataset contains 384 instances and 3840 labels annotated by 83 workers. Each worker's annotation ratio and annotation quality denote the proportions of the instances it annotated in all instances and the true labels in all its annotated labels, respectively, as shown in Figure 1. We observe that the annotation ratio of most workers falls between 0.1 and 0.4, indicating that each worker annotated only a few instances and reflecting the insufficiency of annotation information. Meanwhile, the annotation quality of most workers falls between 0.4 and 0.7, indicating the presence of class ambiguity and uncertainty in worker annotation. Therefore, we select them as real-world crowdsourced datasets for our experiments.

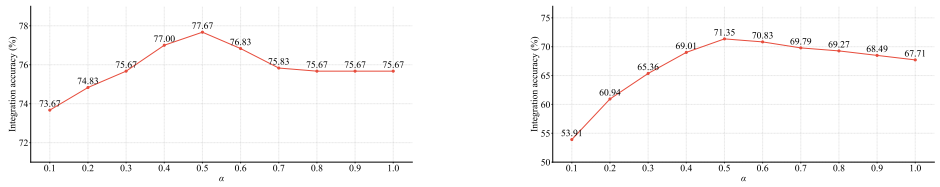
Experimental results. Figure 2 shows the integration accuracies of seven label integration algorithms after label completion by WSLC, TLLC, and LDLLC on the "Income" and "Leaves" datasets, respectively. From it, we can see that the integration accuracies of MV, IWMV, MNLDP, AALI, DEWSMV, LAGNN, and DVLI after label completion by LDLLC are the highest on the "Income" and "Leaves" datasets. These experimental results show that LDLLC improves the performance of label integration algorithms compared with its competitors, confirming its effectiveness.

Ablation analysis. In the above experiments, we have only validated the effectiveness of the full LDLLC. Now, we construct an ablation experiment to validate the effectiveness of each component in LDLLC. For simplicity, we denote three variants of LDLLC as LDLLC-1, LDLLC-2, and LDLLC-3. Specifically, LDLLC-1 removes the class membership estimation and replaces it with one-hot encoding. LDLLC-2 removes the data augmentation and weighting, where each worker is trained only on its own annotated data. LDLLC-3 removes the weighted label distribution learning model and replaces it with a weighted Bayesian classifier. The integration accuracies of MV after label completion by LDLLC and its three variants on the "Income" and "Leaves" datasets are shown in Figure 3. From it, we can see that the integration accuracy of MV improves significantly after label completion by LDLLC compared with its three variants. These experimental results show that all components of LDLLC are working effectively.



(a) Income (b) Leaves

Figure 3: Integration accuracy (%) comparisons of MV after label completion by LDLLC, LDLLC-1, LDLLC-2 and LDLLC-3 on the "Income" and "Leaves" datasets, respectively.



(a) Income (b) Leaves

Figure 4: Integration accuracy (%) of MV after label completion by LDLLC on "Income" and "Leaves" datasets when α varies from 0.1 to 1.

Parameter sensitivity analysis. In LDLLC, we analyze the influence of the hyperparameter α. Theoretically, too small or too large α is detrimental to the performance of LDLLC. When α is too small, the information exploited from similar workers is insufficient, resulting in an inadequate training dataset. Conversely, when α is too large, annotation information from dissimilar workers may be incorporated, resulting in inaccurate worker modeling. Figure 4 shows the integration accuracy of MV after label completion by LDLLC as α varies on the "Income" and "Leaves" datasets. We can see that as α increases from 0.1 to 0.5, the integration accuracy improves. However, as α continues to increase from 0.5 to 1.0, the accuracy slightly declines, demonstrating that the optimal performance is achieved at α = 0.5. Therefore, we set α to 0.5 in our experiments.

4.3 Experiments on simulated datasets

To further validate the effectiveness of the LDLLC, we conduct a series of experiments on the whole 34 simulated datasets. Due to the limited page space, the detailed description of these datasets and the simulation experiment results are provided in **Appendix D**. These experimental results powerfully validate the effectiveness of LDLLC.

5 Conclusion and Future Work

In this paper, we reveal the limitations of existing algorithms, which usually treat the labels annotated by workers as single and deterministic labels, thereby failing to capture the class ambiguity and uncertainty in worker annotation and consequently limiting their performance. To address this issue, we draw inspiration from label distribution learning and propose a novel label distribution learning-based label completion (LDLLC) algorithm. It trains a weighted label distribution model on the augmented and weighted instances to learn the mapping from instance features to label distributions. Extensive experiments show that LDLLC significantly outperforms all its competitors.

In the current version of LDLLC, it is crucial to determine an appropriate number of similar workers. Currently, we use the hyperparameter α to calculate the number of similar workers, and α is empirically fixed at 0.5. We believe that using optimization tools to learn the adaptive value of α based on the characteristics of different datasets can further enhance the performance of LDLLC. This is one of the main research directions for our future work.

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

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, 2022a.

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

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, 2022c.

Kamran Ghasedi Dizaji, Hongchang Gao, Yanhua Yang, Heng Huang, and Cheng Deng. Robust cumulative crowdsourcing framework using new incentive payment function and joint aggregation model. *IEEE Trans. Neural Networks Learn. Syst.*, 31(11):4610–4621, 2020.

Yu Dong, Liangxiao Jiang, and Chaoqun Li. Improving data and model quality in crowdsourcing using co-training-based noise correction. *Inf. Sci.*, 583:174–188, 2022.

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

Jinghan Hu, Liangxiao Jiang, and Wenjun Zhang. Dual-view label integration for crowdsourcing. *Frontiers of Computer Science*, 2026. ISSN 2095-2228. doi: https://doi.org/10.1007/s11704-026-52044-5.

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.

Hyun Joon Jung and Matthew Lease. Improving quality of crowdsourced labels via probabilistic matrix factorization. In Yiling Chen, Panagiotis G. Ipeirotis, Edith Law, Luis von Ahn, and Haoqi Zhang (eds.), *The 4th Human Computation Workshop, HCOMP@AAAI 2012, Toronto, Ontario, Canada, July 23, 2012*, volume WS-12-08 of *AAAI Technical Report*. AAAI Press, 2012.

Chaoqun Li, Liangxiao Jiang, and Wenqiang Xu. Noise correction to improve data and model quality for crowdsourcing. *Eng. Appl. Artif. Intell.*, 82:184–191, 2019a.

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

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, 2019b.

Yunan Lu, Weiwei Li, Huaxiong Li, and Xiuyi Jia. Predicting label distribution from tie-allowed multi-label ranking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12):15364–15379, 2023.

Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Mach. Learn.*, 52(3): 239–281, 2003.

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, 2025a.

Can Pan, Liangxiao Jiang, and Shanshan Si. Cross-worker joint modeling-based label integration for crowdsourcing. *Int. J. Approx. Reason.*, 187:109570, 2025b.

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.

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.

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

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.

Xue Wu, Liangxiao Jiang, Wenjun Zhang, and Chaoqun Li. Worker similarity-based label completion for crowdsourcing. *IEEE Trans. Big Data*, 11(2):710–721, 2025.

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.

Boyi Yang, Liangxiao Jiang, and Wenjun Zhang. Probabilistic matrix factorization-based three-stage label completion for crowdsourcing. In Elena Baralis, Kun Zhang, Ernesto Damiani, Mérouane Debbah, Panos Kalnis, and Xindong Wu (eds.), *IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, United Arab Emirates, December 9-12, 2024*, pp. 540–549. IEEE, 2024.

Zijian Ying, Jing Zhang, Qianmu Li, Ming Wu, and Victor S. Sheng. A little truth injection but a big reward: Label aggregation with graph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(5):3169–3182, 2024.

Jing Zhang. Knowledge learning with crowdsourcing: A brief review and systematic perspective. *IEEE CAA J. Autom. Sinica*, 9(5):749–762, 2022.

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.

Wenjun Zhang, Liangxiao Jiang, Ziqi Chen, and Chaoqun Li. FNNWV: farthest-nearest neighbor-based weighted voting for class-imbalanced crowdsourcing. *Sci. China Inf. Sci.*, 67(10), 2024.

Wenjun Zhang, Liangxiao Jiang, and Chaoqun Li. TLLC: transfer learning-based label completion for crowdsourcing. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025a.

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

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

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.

Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proc. VLDB Endow.*, 10(5):541–552, 2017.

Yao Zhou and Jingrui He. Crowdsourcing via tensor augmentation and completion. In Subbarao Kambhampati (ed.), *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pp. 2435–2441. IJCAI/AAAI Press, 2016.

A OVERALL FRAMEWORK OF LDLLC

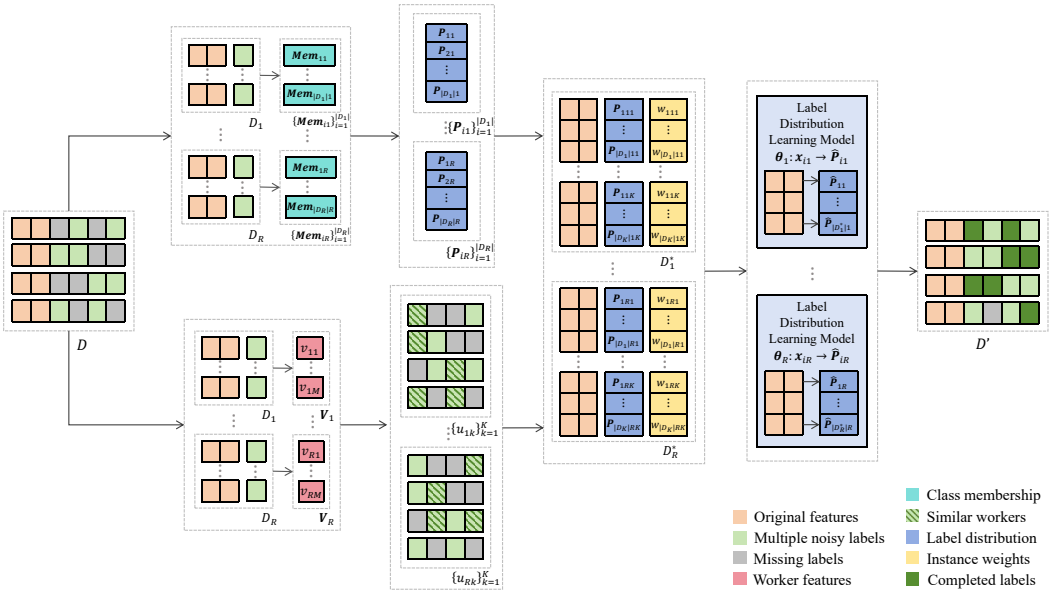


Figure 5: Overall framework of LDLLC. Specifically, for each worker, we first design a class membership estimation method to obtain the label distribution of each instance annotated by it and develop a worker similarity estimation method to find its similar workers (including itself). Then, we utilize the instances annotated by its similar workers to augment the instances annotated by it and use the worker similarity to weight the augmented instances. Subsequently, we train a label distribution learning model on the augmented and weighted instances to effectively capture the class ambiguity and uncertainty in worker annotation. Finally, we use the trained model to predict the label distribution of each unannotated instance and complete the missing label.

B THE LEARNING PROCESS OF LDLLC

Algorithm 1 LDLLC(D, α)

Input: $D = \{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$ - a crowdsourced dataset, α - a hyperparameter.
Output: D' - the completed crowdsourced dataset.

for $r = 1$ to R **do**
  Construct a dataset D_r for u_r by Eq. (1);
  Partition D_r annotated by u_r into Q clusters by Eq. (2);
  for $q = 1$ to Q **do**
    Estimate the class membership probability Mem_{irq} of $\boldsymbol{x}_{ir}$ by Eqs. (3)-(5);
  end for
  for $i = 1$ to $|D_r|$ **do**
    Obtain the label distribution $\boldsymbol{P}_{ir}$ of $\boldsymbol{x}_{ir}$ by Eq. (6);
  end for
end for
for $r = 1$ to R **do**
  for $m = 1$ to M **do**
    Learn the feature value v_{rm} of u_r by Eqs. (7)-(10);
  end for
  for $r' = 1$ to R **do**
    Calculate the worker similarity $s(u_r, u_{r'})$ by Eqs. (11)-(12);
  end for
  Calculate the value of K using α by Eq. (13);
  Find K most similar workers $\{u_{rk}\}_{k=1}^{K}$ of u_r;
  Construct a dataset D_r^* for u_r by Eq. (14);
end for
for $r = 1$ to R **do**
  Initialize the parameter matrix $\boldsymbol{\theta}_r^{(0)}$;
  Initialize the approximation of the inverse Hessian matrix $\boldsymbol{B}^{(0)}$;
  Construct the function $T(\boldsymbol{\theta}_r)$ for u_r by Eqs. (15)-(18);
  Calculate the gradient $\nabla T(\boldsymbol{\theta}_r^{(0)})$ by Eq. (19);
  $t \leftarrow 1$;
  repeat
    Calculate the search direction $\boldsymbol{p}^{(t)}$ by Eq. (20);
    Calculate the step size $\eta^{(t)}$ by Eqs. (21)-(22);
    Calculate the parameter matrix $\boldsymbol{\theta}_r^{(t)}$ by Eq. (23);
    Calculate the gradient $\nabla T(\boldsymbol{\theta}_r^{(t)})$ by Eq. (19);
    Calculate the matrix $\boldsymbol{B}^{(t)}$ by Eqs. (24)-(27);
    $t \leftarrow t + 1$;
  until $\|\nabla T(\boldsymbol{\theta}_r^{(t)})\| < 10^{-4}$
end for
for $i = 1$ to N **do**
  for $r = 1$ to R **do**
    if $l_{ir} == -1$ **then**
      Estimate the predicted label distribution $\boldsymbol{P}_{ir}^*$ of $\boldsymbol{x}_{ir}$ using $\boldsymbol{\theta}_r$ by Eq. (28);
      Calculate the class margin $margin_{ir}$ of $\boldsymbol{x}_{ir}$ by Eq. (29);
      if $margin_{ir} \geq 1/Q$ **then**
        Complete the missing label $\hat{l}_{ir}$ by Eq. (30);
      end if
    end if
  end for
end for
return D'.

C TIME COMPLEXITY ANALYSIS

In **Algorithm 1**, lines 1-10 estimate the class membership probability and obtain the label distribution of each instance annotated by each worker with a time complexity of $O(NR + R|D_r|QM + R|D_r|Q)$. Lines 11-21 estimate the worker similarity and then construct an augmented and weighted dataset with a time complexity of $O(RM|D_r|n_l n_a + R^2M + RNKQ)$, where n_l and n_a are the average number of values for a label variable and a feature variable. Lines 22-24 initialize the parameter matrix and the approximated matrix with a time complexity of $O((RQM) + R(QM)^2)$. Lines 25-26 construct the objective function and calculate the Initial gradient with a time complexity of $O(R|D_r^*|QM)$. Lines 27-36 obtain the parameter matrix with a time complexity of $O(RT|D_r^*|QM)$, where T is the number of BFGS iterations. Finally, Lines 37-47 estimate the predicted label distribution and complete the missing labels with a time complexity of $O(NRQM + NRQ)$. Therefore, the total time complexity of LDLLC is $O(NR + R|D_r|QM + R|D_r|Q + RM|D_r|n_l n_a + R^2M + RNKQ + RQM + R(QM)^2 + R|D_r^*|QM + RT|D_r^*|QM + NRQM + NRQ)$. If only the highest order terms are considered, the total time complexity of LDLLC is $O(R(QM)^2 + RT|D_r^*|QM + NRQM + RNKQ + R^2M)$.

D DESCRIPTIONS OF 34 SIMULATED DATASETS AND EXPERIMENTAL RESULTS

Descriptions of simulated datasets. The descriptions of the whole 34 simulated datasets are listed in Table 1. Here, "#Instances" denotes the number of instances, "#Features" denotes the number of features, "#Classes" denotes the number of classes, "Missing" denotes whether the dataset contains missing values, and "Feature type" denotes the type of features the dataset contains. These datasets are collected from different application scenarios and reflect different crowdsourcing requirements.

Table 1: The description of 34 simulated datasets.

Dataset	#Instances	#Features	#Classes	Missing	Feature type
anneal	898	38	6	yes	hybrid
audiology	226	69	24	yes	nominal
autos	205	25	7	yes	hybrid
balance-scale	625	4	3	no	numeric
biodeg	1055	41	2	no	numeric
breast-cancer	286	9	2	yes	nominal
breast-w	699	9	2	yes	numeric
car	1728	6	4	no	nominal
credit-a	690	15	2	yes	hybrid
credit-g	1000	20	2	no	hybrid
diabetes	768	8	2	no	numeric
heart-c	303	13	5	yes	hybrid
heart-h	294	13	5	yes	hybrid
heart-statlog	270	13	2	no	numeric
hepatitis	155	19	2	yes	hybrid
horse-colic	368	22	2	yes	hybrid
hypothyroid	3772	29	4	yes	hybrid
ionosphere	351	34	2	no	numeric
iris	150	4	3	no	numeric
kr-vs-kp	3196	36	2	no	nominal
labor	57	16	2	yes	hybrid
letter	20000	16	26	no	numeric
lymph	148	18	4	no	hybrid
mushroom	8124	22	2	yes	nominal
segment	2310	19	7	no	numeric
sick	3772	29	2	yes	hybrid
sonar	208	60	2	no	numeric
spambase	4601	57	2	no	numeric
tic-tac-toe	958	9	2	no	nominal
vehicle	846	18	4	no	numeric
vote	435	16	2	yes	nominal
vowel	990	13	11	no	hybrid
waveform	5000	40	3	no	numeric
zoo	101	17	7	no	hybrid

Experimental results on the uniform distribution. In our experiments, since MNLDP cannot handle missing feature values, we first utilize the mean of numeric feature values or the mode of nominal feature values from the available data to replace all missing feature values. Subsequently,

we simulate 40 crowd workers to replicate a real-world crowdsourcing process based on the characteristics of the two real-world crowdsourced datasets mentioned above. Specifically, we simulate the annotation pattern of each worker using two parameters: annotation ratio p_r and annotation quality q_r. Specifically, for each crowd worker, we randomly sample p_r and q_r from the uniform distributions $[0.1, 0.4]$ and $[0.4, 0.7]$, respectively. These experimental settings approximate the characteristics of real-world crowdsourced datasets. To counter the effects of randomness, we repeat each experiment 10 times independently and use the average.

Table 2 and Table 3 show the integration accuracies of seven label integration algorithms after label completion by LDLLC compared with WSLC and TLLC, respectively. Based on these results, we perform pairwise comparisons of label integration algorithms after label completion by LDLLC and its competitors using the corrected paired two-tailed t-test with the significance level $\alpha = 0.05$ (Nadeau & Bengio, 2003; Jiang et al., 2019). In the tables, the symbols • and ◦ indicate that the integration accuracy has a statistically significant improvement or degradation using LDLLC compared with its competitors. The bottom of Tables 2 and 3 represents the average integration accuracy and the $Win/Tie/Lose$ $(W/T/L)$ across the 34 datasets. The average integration accuracy reflects the mean performance of different label integration algorithms after label completion by LDLLC and its competitors. The $W/T/L$ summarizes the comparison results, where LDLLC wins its competitors on W datasets, ties on T datasets, and loses on L datasets in terms of improving the performance of label integration algorithms. These experimental results powerfully validate the effectiveness of LDLLC. The specific conclusions are as follows:

- From Table 2, we can find that the average integration accuracies of MV (73.02%), IWMV (74.84%), MNLDP (77.34%), AALI (75.31%), DEWSMV (73.06%), LAGNN (73.59%), and DVLI (77.22%) after label completion by LDLLC are much higher than those of MV (67.75%), IWMV (67.70%), MNLDP (69.95%), AALI (67.85%), DEWSMV (67.74%), LAGNN (67.72%), and DVLI (69.42%) after label completion by WSLC, respectively. This result powerfully demonstrates that LDLLC achieves superior performance compared to WSLC across all label integration algorithms.
- From Table 2, we can find that LDLLC significantly wins its competitor WSLC on 22, 23, 25, 24, 21, 22, and 23 datasets for MV, IWMV, MNLDP, AALI, DEWSMV, LAGNN, and DVLI, respectively, while significantly losing on 7, 6, 3, 4, 7, 6, and 4 datasets, respectively. The t-test results show that the number of datasets where LDLLC significantly wins (W) its competitor is much higher than those where it loses (L) across all label integration algorithms. This further powerfully demonstrates the effectiveness of LDLLC.
- From Table 3, we can find that the average integration accuracies of MV (73.07%), IWMV (74.88%), MNLDP (77.34%), AALI (75.29%), DEWSMV (73.01%), LAGNN (73.59%), and DVLI (77.21%) after label completion by LDLLC are much higher than those of MV (68.06%), IWMV (68.84%), MNLDP (70.12%), AALI (69.62%), DEWSMV (67.95%), LAGNN (68.16%), and DVLI (71.69%) after label completion by TLLC, respectively. This result powerfully demonstrates that LDLLC achieves superior performance compared to TLLC across all label integration algorithms.
- From Table 3, we can find that LDLLC significantly wins its competitor TLLC on 20, 22, 23, 19, 20, 21, and 20 datasets for MV, IWMV, MNLDP, AALI, DEWSMV, LAGNN, and DVLI, respectively, while significantly losing on 7, 7, 4, 6, 7, 7, and 5 datasets, respectively. The t-test results show that the number of datasets where LDLLC significantly wins (W) its competitor is much higher than those where it loses (L) across all label integration algorithms. This further powerfully demonstrates the effectiveness of LDLLC.

Experimental results on the Gaussian distribution. We also conduct another series of experiments to validate the effectiveness of LDLLC on different annotation quality distributions, in which the annotation quality q of each crowd worker is randomly generated from a Gaussian distribution $N(0.55, 0.15^2)$, while keeping other settings unchanged. The mean and variance of this distribution are determined based on the mean of the above uniform distribution and the three-sigma rule, respectively. The related experimental results are shown in Tables 4 and 5. From these results, we can also conclude that LDLLC significantly improves the overall performance of all label integration algorithms compared with WSLC and TLLC, regardless of whether the simulated annotation quality of each crowd worker follows a uniform or Gaussian distribution.

Table 2: Integration accuracy (%) comparisons of seven label integration algorithms after label completion by WSLC and LDLLC on the uniform distribution, respectively.

	MV		IWMV		MNLDP		AALI		DEWSMV		LAGNN		DVLI	
Dataset	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC
anneal	83.54	77.85 ◦	83.53	78.52 ◦	86.27	80.68 ◦	82.95	79.59 ◦	83.52	77.91 ◦	83.53	78.32 ◦	85.44	73.28 ◦
audiology	54.20	70.05 •	54.07	69.82 •	53.94	69.78 •	53.89	73.23 •	54.16	70.71 •	54.07	69.82 •	54.34	73.05 •
autos	76.92	71.22 ◦	76.88	70.98 ◦	77.85	70.73 ◦	77.37	72.49 ◦	76.88	71.32 ◦	76.93	70.83 ◦	78.44	67.27 ◦
balance-scale	66.91	71.25 •	66.94	72.22 •	70.83	77.50 •	67.06	72.77 •	67.09	71.04 •	66.90	71.49 •	69.73	79.62 •
biodeg	63.00	62.56	62.90	66.87	64.77	65.17	63.40	68.66 •	63.04	62.58	63.02	63.68	64.73	68.10
breast-cancer	58.88	61.71 •	58.85	61.40	59.93	64.19 •	59.12	66.78 •	58.95	60.87	58.92	61.82 •	59.23	64.61 •
breast-w	63.16	72.92 •	63.19	75.82 •	64.55	82.15 •	63.08	80.13 •	63.13	73.06 •	63.18	73.56 •	64.68	92.20 •
car	74.15	69.76 ◦	74.26	71.92 ◦	76.59	80.17 •	74.12	71.08 ◦	74.13	69.78 ◦	74.22	71.18 ◦	75.39	79.20 •
credit-a	64.62	81.20 •	64.64	83.17 •	66.04	83.46 •	65.71	81.46 •	64.58	81.33 •	64.58	81.94 •	66.39	84.26 •
credit-g	62.71	65.70 •	62.65	70.08 •	63.67	67.52 •	63.05	68.20 •	62.69	65.67 •	62.64	66.96 •	63.41	70.82 •
diabetes	61.49	61.00	61.42	63.67 •	62.23	65.40 •	61.56	62.19	61.47	60.66	61.47	61.50	62.38	65.47 •
heart-c	60.14	69.34 •	60.17	72.18 •	61.92	74.39 •	61.62	73.20 •	60.14	69.80 •	60.17	70.79 •	61.62	78.91 •
heart-h	61.73	73.60 •	61.56	76.94 •	63.23	78.06 •	62.35	78.88 •	61.50	74.01 •	61.60	74.59 •	63.13	81.22 •
heart-statlog	61.89	67.19 •	61.59	68.30 •	62.92	74.41 •	61.22	72.15 •	61.96	67.85 •	62.00	68.11 •	63.67	79.19 •
hepatitis	55.03	61.10 •	54.97	60.06 •	56.71	65.35 •	54.58	55.94	55.03	60.52 •	54.84	60.13 •	55.48	65.55 •
horse-colic	60.71	66.52 •	60.57	67.06 •	61.03	68.26 •	60.62	64.18 •	60.71	66.17 •	60.54	66.66 •	61.36	71.55 •
hypothyroid	88.33	86.78 ◦	88.33	88.38	91.75	95.78 •	87.96	88.65	88.35	86.67 ◦	88.33	88.27	90.20	96.97 •
ionosphere	62.16	65.36 •	62.39	68.41 •	64.19	69.26 •	63.50	67.55 •	62.05	65.61 •	62.31	65.81 •	65.84	68.21
iris	71.27	80.13 •	69.47	80.27 •	83.27	90.13 •	72.73	84.93 •	71.27	79.73 •	70.87	80.87 •	79.27	92.00 •
kr-vs-kp	61.62	72.05 •	61.68	77.85 •	62.68	77.01 •	62.00	74.58 •	61.63	72.09 •	61.65	73.37 •	62.30	75.91 •
labor	61.93	64.03	61.93	64.03	64.39	70.70	62.28	66.14	61.75	64.21	61.75	63.68	62.81	67.89
letter	96.23	95.64 ◦	96.23	96.02 ◦	98.28	99.03 •	96.32	96.57 •	96.23	95.61 ◦	96.23	95.72 ◦	97.04	96.53 ◦
lymph	61.62	67.84 •	61.55	69.19 •	62.97	72.30 •	61.35	69.26 •	61.55	68.65 •	61.49	69.12 •	63.17	73.45 •
mushroom	65.20	87.03 •	65.21	88.47 •	66.39	89.09 •	65.06	87.54 •	65.20	87.01 •	65.21	87.44 •	66.37	88.83 •
segment	92.22	93.34 •	92.21	94.22 •	96.00	98.60 •	92.51	95.10 •	92.16	93.44 •	92.20	93.82 •	93.81	93.72
sick	62.34	60.21	62.38	69.12 •	64.42	69.08	62.00	63.65	62.35	60.18	62.35	61.91	60.70	60.50
sonar	61.06	61.88	62.02	61.97	60.72	63.32	56.39	60.67	61.01	62.36	61.01	61.78	64.47	62.60
spambase	63.79	75.97 •	63.79	79.33 •	64.96	80.04 •	63.92	81.65 •	63.79	75.97 •	63.80	77.01 •	64.70	76.03 •
tic-tac-toe	62.59	64.32 •	62.52	68.23 •	63.69	66.78	62.75	67.05 •	62.57	64.62 •	62.56	65.40 •	63.00	67.67 •
vehicle	85.19	77.38 ◦	84.93	78.11 ◦	90.30	79.01 ◦	85.66	79.11 ◦	85.18	77.52 ◦	85.17	77.91 ◦	86.55	68.82 ◦
vote	59.52	74.51 •	59.56	78.07 •	60.58	81.31 •	60.30	78.83 •	59.61	73.93 •	59.54	75.20 •	61.95	85.80 •
vowel	84.11	82.86 ◦	84.13	82.96 ◦	86.62	93.30 •	84.71	87.26 •	84.13	82.68 ◦	84.13	82.72 ◦	84.88	88.75 •
waveform	80.94	89.17 •	80.67	89.56 •	87.90	86.76	81.90	90.12 •	80.94	89.13 •	80.95	89.51 •	87.00	86.48
zoo	54.36	81.29 •	54.46	81.29 •	56.74	80.79 •	53.87	80.99 •	54.56	81.49 •	54.46	81.29 •	56.94	81.09 •
Mean	67.75	73.02	67.70	74.84	69.95	77.34	67.85	75.31	67.74	73.06	67.72	73.59	69.42	77.22
$W/T/L$	22/5/7		23/5/6		25/6/3		24/6/4		21/6/7		22/6/6		23/7/4	

Table 3: Integration accuracy (%) comparisons of seven label integration algorithms after label completion by TLLC and LDLLC on the uniform distribution, respectively.

	MV		IWMV		MNLDP		AALI		DEWSMV		LAGNN		DVLI	
Dataset	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC
anneal	89.16	77.95 ◦	89.69	78.55 ◦	87.84	80.68 ◦	90.08	79.72 ◦	89.13	77.89 ◦	89.30	78.32 ◦	91.89	73.33 ◦
audiology	46.33	70.00 •	45.89	69.91 •	42.70	69.78 •	47.88	73.49 •	46.15	70.22 •	45.71	69.82 •	38.23	72.92 •
autos	71.56	71.17	71.37	71.17	67.81	70.73 •	71.56	72.10	72.10	71.32	71.51	70.83	68.68	67.37
balance-scale	66.00	70.64 •	66.50	72.51 •	66.67	77.50 •	65.73	72.88 •	65.52	70.62 •	66.19	71.49 •	70.91	79.65 •
biodeg	65.78	62.60	67.23	66.71	66.87	65.17	67.36	68.55	65.50	62.79	66.06	63.68	71.45	67.68
breast-cancer	60.28	61.36	60.03	62.76	60.35	64.19	64.02	67.24	59.79	61.29	59.97	61.82	61.96	64.47
breast-w	67.30	72.82 •	71.14	75.84 •	70.41	82.15 •	76.62	80.43	67.14	72.95 •	67.80	73.56 •	91.22	92.23 •
car	76.59	70.05 ◦	78.62	71.93 ◦	80.34	80.17	77.47	70.98 ◦	76.28	69.89 ◦	77.12	71.18 ◦	84.44	79.14 ◦
credit-a	64.50	81.16 •	65.10	83.16 •	66.78	83.46 •	65.90	81.42 •	64.67	81.52 •	64.78	81.94 •	70.83	84.19 •
credit-g	63.61	65.79 •	63.89	70.18 •	65.29	67.52 •	65.11	68.14 •	63.39	65.72 •	63.72	66.96 •	66.91	70.87 •
diabetes	60.35	60.70	60.43	63.39 •	61.46	65.40 •	60.47	62.04 •	59.99	60.96	60.23	61.50	64.06	65.46 •
heart-c	61.65	69.64 •	61.23	71.45 •	61.39	74.39 •	62.48	73.20 •	61.42	69.57 •	61.39	70.79 •	67.50	78.91 •
heart-h	59.52	73.88 •	60.10	76.77 •	60.51	78.06 •	63.23	79.05 •	59.83	73.84 •	59.80	74.59 •	65.68	81.46 •
heart-statlog	59.44	67.45 •	59.59	68.82 •	61.63	74.41 •	61.96	72.89 •	59.85	68.26 •	59.63	68.11 •	65.52	78.70 •
hepatitis	55.61	60.52 •	53.48	59.68 •	56.00	65.35 •	50.26	56.00	55.35	60.06 •	55.10	60.13 •	58.77	65.81 •
horse-colic	60.54	65.90 •	60.73	67.85 •	61.98	68.26 •	58.59	63.83 •	60.49	65.90 •	60.62	66.66 •	63.64	71.96 •
hypothyroid	91.24	86.93 ◦	91.88	88.38 ◦	92.81	95.78 •	90.33	88.57	91.17	86.53 ◦	91.44	88.27 ◦	94.30	97.11 •
ionosphere	62.02	65.07	63.28	68.38 •	66.07	69.26	60.28	68.35 •	61.65	64.56	62.08	65.81 •	68.63	67.72
iris	92.27	80.20 ◦	93.00	80.27 ◦	94.27	90.13	93.80	84.80 ◦	92.47	80.27 ◦	92.73	80.87 ◦	96.33	91.93 ◦
kr-vs-kp	65.05	72.24 •	67.22	77.86 •	68.41	77.01 •	66.35	74.54 •	65.05	72.22 •	65.80	73.37 •	71.57	75.86 •
labor	63.86	64.74	62.46	64.39	69.30	70.70	73.69	64.74	64.03	63.51	63.68	63.68	66.84	68.77
letter	46.63	95.64 •	40.92	96.02 •	60.67	99.03 •	46.99	96.56 •	46.49	95.65 •	42.94	95.72 •	62.65	96.52 •
lymph	61.96	69.39 •	62.50	68.92 •	63.92	72.30 •	62.84	69.12 •	61.62	68.92 •	62.09	69.12 •	67.50	73.51 •
mushroom	66.90	87.06 •	73.26	88.47 •	78.06	89.09 •	64.54	88.09 •	66.77	87.02 •	68.59	87.44 •	90.25	88.84
segment	98.10	93.52 ◦	98.17	94.21 ◦	97.78	98.60 •	98.20	95.09 ◦	98.07	93.28 ◦	98.17	93.82 ◦	91.52	93.56 •
sick	74.39	60.39 ◦	80.79	69.16 ◦	79.82	69.04 ◦	85.96	63.86 ◦	74.25	60.15 ◦	75.73	61.91 ◦	48.02	60.09
sonar	58.56	61.73 •	58.27	61.25 •	59.09	63.32 •	57.07	60.96	58.46	62.26 •	58.27	61.78 •	60.72	62.93
spambase	70.09	76.09 •	74.54	79.34 •	73.81	80.04 •	69.78	81.51 •	70.07	76.10 •	71.35	77.01 •	68.71	75.98 •
tic-tac-toe	65.26	64.14	66.02	68.42	67.25	66.78	67.10	66.87	65.36	64.77	65.66	65.40	67.58	67.61
vehicle	88.64	77.83 ◦	89.03	78.13 ◦	87.99	79.01 ◦	88.75	78.68 ◦	88.57	77.39 ◦	88.75	77.91 ◦	75.38	68.51 ◦
vote	62.74	74.46 •	64.69	78.00 •	65.59	81.31 •	65.10	77.93 •	62.21	74.07 •	62.97	75.20 •	79.13	85.86 •
vowel	72.76	82.88 •	73.55	83.08 •	73.18	93.30 •	77.82	87.26 •	72.57	82.49 •	72.53	82.72 •	71.38	89.02 •
waveform	86.08	89.19 •	86.85	89.55 •	88.84	86.76 ◦	87.82	90.16 •	86.04	89.11 •	86.30	89.51 •	87.46	86.50 ◦
zoo	59.31	81.29 •	59.21	81.39 •	59.31	80.79 •	61.98	80.70 •	58.91	81.19 •	59.31	81.29 •	67.92	80.79 •
Mean	68.06	73.07	68.84	74.88	70.12	77.34	69.62	75.29	67.95	73.01	68.16	73.59	71.69	77.21
$W/T/L$	20/7/7		22/5/7		23/7/4		19/9/6		20/7/7		21/6/7		20/9/5	

Table 4: Integration accuracy (%) comparisons of seven label integration algorithms after label completion by WSLC and LDLLC on the Gaussian distribution, respectively.

	MV		IWMV		MNLDP		AALI		DEWSMV		LAGNN		DVLI	
Dataset	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC	WSLC	LDLLC
anneal	82.14	78.33 ◦	82.24	79.41 ◦	85.46	82.81 ◦	81.58	80.51	82.07	78.64 ◦	82.25	79.12 ◦	84.18	73.27 ◦
audiology	52.79	69.16 •	52.79	68.58 •	52.61	68.19 •	52.61	71.90 •	52.79	69.16 •	52.88	68.63 •	52.92	71.50 •
autos	77.22	69.85 ◦	77.22	70.10 ◦	78.29	69.22 ◦	76.88	71.07 ◦	77.03	69.90 ◦	77.22	69.85 ◦	78.39	66.34 ◦
balance-scale	69.22	74.21 •	69.09	76.38 •	73.66	82.51 •	69.50	75.89 •	68.96	74.62 •	69.14	75.28 •	71.79	83.09 •
biodeg	64.88	62.08	64.87	72.70	66.02	63.39	65.01	66.65	64.85	62.12	64.85	63.42	66.11	65.38
breast-cancer	58.32	60.66	58.39	61.68	58.78	63.15 •	58.50	65.56 •	58.39	60.91 •	58.36	60.70 •	58.53	65.07 •
breast-w	61.30	68.99 •	61.33	71.82 •	62.63	79.72 •	61.35	75.15 •	61.32	68.76 •	61.33	69.71 •	62.66	91.46 •
car	72.28	68.55 ◦	72.31	70.87	74.76	79.03 •	72.23	69.46 ◦	72.26	68.36 ◦	72.33	70.02 ◦	73.67	78.45 •
credit-a	61.42	77.91 •	61.23	80.35 •	62.90	81.51 •	62.32	78.83 •	61.41	77.49 •	61.39	78.33 •	62.80	82.45 •
credit-g	59.60	61.38 •	59.64	67.17 •	61.03	64.80 •	59.82	64.56 •	59.66	62.06 •	59.63	62.79 •	60.20	65.89 •
diabetes	62.16	62.00	62.27	64.51	63.52	66.13 •	62.36	63.09	62.11	62.23	62.21	62.46	63.22	67.80 •
heart-c	59.25	69.11 •	59.08	70.20 •	59.90	71.62 •	59.94	73.90 •	59.21	68.32 •	59.11	69.54 •	60.76	78.55 •
heart-h	60.41	72.75 •	60.27	76.12 •	61.70	77.89 •	60.95	76.90 •	60.34	72.52 •	60.31	73.37 •	61.67	80.27 •
heart-statlog	60.85	67.00 •	60.70	67.34 •	62.59	71.37 •	60.74	72.22 •	61.04	67.41 •	60.78	67.41 •	62.85	74.81 •
hepatitis	54.84	55.03	54.84	55.55	54.71	57.74	52.58	51.94	54.58	54.90	54.77	55.42	54.71	57.03
horse-colic	61.88	68.07 •	61.71	69.16 •	62.58	68.45 •	61.99	65.00	61.88	67.39 •	61.68	68.29 •	63.04	72.06 •
hypothyroid	91.52	89.73 ◦	91.55	92.93 •	93.99	96.69 •	91.31	91.39	91.55	89.60 ◦	91.55	91.86 •	92.61	97.95 •
ionosphere	61.62	63.48	62.39	64.59	62.68	67.29 •	62.34	65.90	61.68	63.50	61.85	64.25	65.21	67.66
iris	73.80	82.60 •	72.73	82.47 •	85.93	90.80 •	74.47	86.20 •	73.93	82.53 •	73.47	83.27 •	81.80	92.93 •
kr-vs-kp	63.32	72.90 •	63.34	77.53 •	64.36	76.84 •	63.71	74.60 •	63.31	72.77 •	63.34	74.00 •	64.14	75.66 •
labor	66.49	72.63 •	66.49	74.74 •	67.90	77.19 •	66.84	72.98 •	66.49	72.46 •	66.67	73.16 •	68.25	80.00 •
letter	96.52	95.87 ◦	96.52	96.15 ◦	98.47	98.99 •	96.59	96.72 •	96.52	95.79 ◦	96.52	95.85 ◦	97.29	96.52 ◦
lymph	59.93	67.23 •	59.86	67.16 •	61.22	72.43 •	60.88	68.72 •	59.80	66.89 •	59.86	67.23 •	61.15	74.73 •
mushroom	62.29	80.49 •	62.31	84.50 •	63.36	85.15 •	62.32	81.67 •	62.32	80.43 •	62.31	81.22 •	63.69	87.69 •
segment	92.47	93.58 •	92.43	93.98 •	96.08	98.50 •	92.79	95.22 •	92.48	93.38 •	92.47	93.74 •	94.08	93.16 ◦
sick	63.05	55.81 ◦	63.16	60.59	65.10	60.37	62.95	61.93	63.03	55.74 ◦	63.14	57.73	64.65	47.48 ◦
sonar	57.84	57.21	58.66	57.12	57.50	60.72 •	55.05	57.74 •	57.60	57.21	57.93	56.88	58.36	56.92
spambase	63.04	74.26 •	63.07	78.42 •	64.01	78.39 •	63.21	80.38 •	63.04	74.16 •	63.05	75.36 •	63.76	73.74 •
tic-tac-toe	58.37	59.88 •	58.29	63.79 •	59.23	61.98 •	58.57	62.73 •	58.37	59.61	58.31	60.80 •	58.75	62.64 •
vehicle	84.91	79.63 ◦	84.40	80.16 ◦	90.14	81.66 ◦	85.41	80.31 ◦	84.87	79.48 ◦	84.82	79.82 ◦	86.42	71.37 ◦
vote	60.69	74.51 •	60.90	78.97 •	61.77	82.41 •	61.15	75.10 •	60.80	74.19 •	60.80	75.45 •	63.06	86.53 •
vowel	85.67	84.58 ◦	85.69	85.28	87.93	94.12 •	86.22	87.97 •	85.65	84.23 ◦	85.69	84.56 ◦	86.42	89.97 •
waveform	80.61	89.02 •	80.08	89.44 •	87.76	86.78	81.60	90.15 •	80.56	88.98 •	80.58	89.43 •	86.58	86.17
zoo	54.76	82.18 •	54.76	82.18 •	57.33	82.58 •	54.36	81.88 •	54.66	82.48 •	54.85	82.28 •	57.63	82.97 •
Mean	67.51	72.37	67.49	74.47	69.59	76.48	67.59	74.54	67.49	72.30	67.51	72.98	69.16	76.40
$W/T/L$	20/6/8		21/9/4		27/4/3		23/8/3		20/6/8		22/6/6		23/5/6	

Table 5: Integration accuracy (%) comparisons of seven label integration algorithms after label completion by TLLC and LDLLC on the Gaussian distribution, respectively.

	MV		IWMV		MNLDP		AALI		DEWSMV		LAGNN		DVLI	
Dataset	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC	TLLC	LDLLC
anneal	89.17	78.34 ◦	89.68	79.48 ◦	87.84	82.81 ◦	90.08	80.16 ◦	88.99	78.69 ◦	89.32	79.12 ◦	90.53	73.33 ◦
audiology	46.24	68.90 •	46.02	68.58 •	43.19	68.19 •	49.29	71.99 •	46.06	68.58 •	46.06	68.63 •	38.67	71.46 •
autos	70.29	70.20	70.78	70.05	66.24	69.22	69.90	70.68	69.95	70.10	70.24	69.85	66.88	66.34
balance-scale	69.66	74.35 •	70.70	76.27 •	71.49	82.51 •	69.97	76.24 •	69.38	73.79 •	69.87	75.28 •	75.89	83.22 •
biodeg	63.97	62.16	65.91	72.71	65.55	63.39	64.87	66.44	63.95	62.00	64.56	63.42	66.61	65.34
breast-cancer	59.33	60.84	59.23	61.47	61.26	63.15	64.79	64.75	59.44	60.39	59.30	60.70	61.71	64.93
breast-w	66.05	68.66	68.80	71.92	69.67	79.72 •	77.25	74.95	66.22	69.17 •	66.60	69.71	90.17	91.39
car	74.39	68.61 ◦	76.43	70.84 ◦	78.63	79.03	75.66	69.50 ◦	74.16	68.27 ◦	74.87	70.02 ◦	82.16	78.59 ◦
credit-a	62.39	77.68 •	62.97	80.51 •	62.83	81.51 •	62.81	78.97 •	62.26	77.45 •	62.55	78.33 •	67.97	82.39 •
credit-g	62.91	61.58	63.16	67.28 •	64.08	64.80	63.84	64.62	62.84	61.78	62.99	62.79	65.71	65.82
diabetes	61.61	61.97	61.76	64.35 •	62.42	66.13 •	63.30	63.13	61.33	61.90	61.58	62.46	65.91	67.75 •
heart-c	61.03	68.75 •	60.86	70.79 •	62.28	71.62 •	61.65	73.83 •	60.76	68.55 •	60.86	69.54 •	67.10	78.19 •
heart-h	61.84	72.48 •	61.50	75.85 •	63.95	77.89 •	65.03	77.21 •	61.87	73.23 •	61.60	73.37 •	68.88	80.20 •
heart-statlog	59.74	66.63 •	60.11	67.52 •	60.59	71.37 •	59.78	71.92 •	59.33	66.52 •	59.33	67.41 •	64.30	75.04 •
hepatitis	50.97	54.77	50.77	56.19 •	51.23	57.74	46.77	52.97	51.10	55.68	51.35	55.42	51.55	57.23
horse-colic	61.77	67.91 •	62.12	68.67 •	61.74	68.45 •	59.59	64.81 •	61.77	67.47 •	61.90	68.29 •	65.79	71.90 •
hypothyroid	89.19	89.72	90.41	92.93 •	91.66	96.69 •	88.75	91.50 •	89.13	89.62	89.54	91.86 •	93.02	97.92 •
ionosphere	62.31	63.19	61.97	65.02	65.07	67.29	60.74	66.07 •	62.54	63.19	62.36	64.25	72.19	67.83
iris	94.13	82.60 ◦	94.93	82.87 ◦	95.73	90.80 ◦	95.87	86.60 ◦	94.20	82.33 ◦	94.47	83.27 ◦	96.20	93.13 ◦
kr-vs-kp	66.03	72.86 •	67.95	77.49 •	69.26	76.84 •	67.80	74.63 •	65.93	72.92 •	66.72	74.00 •	72.52	75.71 •
labor	72.10	72.98	73.16	75.09	72.10	77.19 •	67.72	73.33	70.88	72.98	71.40	73.16	71.40	79.65
letter	47.55	95.86 •	41.58	96.14 •	62.57	98.99 •	47.71	96.70 •	47.33	95.84 •	46.35	95.85 •	63.53	96.52 •
lymph	61.49	67.16 •	61.28	67.36 •	64.73	72.43 •	62.90	69.53 •	61.49	67.30 •	61.55	67.23 •	66.62	74.26 •
mushroom	66.81	80.50 •	72.20	84.51 •	76.95	85.15 •	60.51	81.97 •	66.72	80.43 •	68.33	81.22 •	89.49	87.68 ◦
segment	98.14	93.55 ◦	98.23	93.98 ◦	97.91	98.50 •	98.26	95.08 ◦	98.08	93.33 ◦	98.21	93.74 ◦	90.62	93.17 •
sick	73.36	55.64 ◦	79.71	60.60	79.18	60.38 ◦	80.98	61.45 ◦	73.44	55.70 ◦	74.69	57.73 ◦	45.89	47.54
sonar	56.54	57.12	57.50	57.11	57.31	60.72	56.30	58.17	56.20	57.31	56.68	56.88	57.02	57.26
spambase	70.26	74.35 •	74.57	78.43 •	73.42	78.39 •	67.66	80.28 •	70.19	74.24 •	71.48	75.36 •	67.79	73.75 •
tic-tac-toe	60.61	59.66	60.99	63.75	61.32	61.98	62.48	62.70	60.75	60.02	60.78	60.80	62.81	62.75
vehicle	89.14	79.67 ◦	89.85	80.16 ◦	87.84	81.66 ◦	89.31	80.44 ◦	88.96	79.64 ◦	89.50	79.82 ◦	75.45	71.53 ◦
vote	64.21	74.21 •	65.36	78.83 •	66.51	82.41 •	66.00	75.82 •	64.07	74.19 •	64.57	75.45 •	81.13	86.53 •
vowel	76.19	84.68 •	77.48	85.30 •	76.81	94.12 •	81.82	88.41 •	76.21	84.39 •	76.27	84.56 •	73.95	89.96 •
waveform	86.02	89.05 •	86.84	89.43 •	89.22	86.78 ◦	88.03	90.20 •	85.87	88.97 •	86.26	89.43 •	87.33	86.18 ◦
zoo	60.60	81.98 •	60.10	82.18 •	60.30	82.58 •	62.48	81.49 •	60.30	82.48 •	60.59	82.28 •	70.50	82.77 •
Mean	68.12	72.31	68.97	74.52	70.03	76.48	69.12	74.60	67.99	72.31	68.32	72.98	71.39	76.39
$W/T/L$	16/12/6		20/9/5		20/9/5		18/10/6		17/11/6		17/11/6		17/11/6	