Multi-Label Truth Inference for Crowdsourcing Using Mixture Models

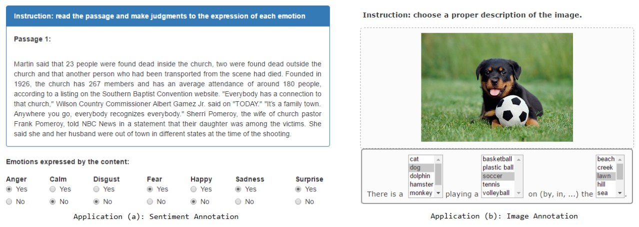
Jing Zhang, *Senior Member, IEEE,* and Xindong Wu, *Fellow, IEEE*

Abstract—When acquiring labels from crowdsourcing platforms, a task may be designed to include multiple labels and the values of each label may belong to a set of various distinct options, which is the so-called multi-class multi-label annotation. To improve the quality of labels, requesters usually let one task be independently completed by a group of heterogeneous crowdsourced workers. Then, the true values of the multiple labels of each task are inferred from these repeated noisy labels. In this paper, we propose two novel probabilistic models MCMLI and MCMLD to address the multi-class multi-label inference problem in crowdsourcing. MCMLI assumes that the labels of each task are mutually independent and MCMLD utilizes a mixture of multiple independently multinoulli distributions to capture the correlation among the labels. Both models can jointly infer multiple true labels of each instance as well as estimate the reliability of crowdsourced workers modeled by a set of confusion matrices with an expectation–maximization algorithm. Experiments with three typical crowdsourcing scenarios and a real-world dataset show that our proposed models significantly outperform existing competitive alternatives. When the labels are strongly correlated, MCMLD substantially outperforms MCMLI. Furthermore, our models can be easily simplified to the one-coin models, which show more advantageous when errors are uniformly distributed, or labels are sparse.

Index Terms—Crowdsourcing, maximum likelihood estimation, mixture models, probabilistic graphical models, statistical inference.

1 Introduction

Current intelligent systems usually require plenty of human-labeled datasets to build internal models. These human-annotation tasks used to be completed by domain experts, which cannot meet the requirement of fast evolution of current AI systems due to its expensive and time-consuming characteristics. The emergence of crowdsourcing systems provides a convenient, fast and low-cost solution to obtain annotations from the Internet, where internet users undertake the task and get a small amount of monetary reward after completing them. In recent years, acquisition of annotations from crowdsourcing has been widely adopted by many application domains in today's big data era [1], such as machine learning [2], [3], data mining [4], computer vision [5], information retrieval [6], and natural language processing [7]. However, one of the great challenges for label acquisition with crowdsourcing is that the quality of labels cannot be guaranteed due to its non-expert nature of crowdsourced workers. A straightforward approach to improve the quality of labels is *repeated labeling* scheme, where each instance is labeled by multiple heterogeneous crowdsourced workers. After sufficient repeated labels are collected, a truth inference algorithm will be introduced to estimate the true labels of each instance from those noisy labels. This process is also known as *label aggregation* since an estimated true label can be viewed as an integration of multiple noisy labels. Thus, an estimated true label is also called an *integrated* label. In addition, this process may provide more information, such as estimates of workers' reliability and distribution of classes. In crowdsourcing, the term *truth inference* is often used as a synonym of *label aggregation* as in this paper, but we should know that using the term *inference* implies that the process involves probabilistic modeling. A general inference algorithm in crowdsourcing should be agnostic, assuming that no prior knowledge can be exploited, because one of its objectives is to provide information for further usages, such as spammer detection [8] and incentive optimization [9].


Instruction: read the passage and make judgments to the expression of each emotion

Passage 1:

Martin said that 23 people were found dead inside the church, two were found dead outside the church and that another person who had been transported from the scene had died. Founded in 1926, the church has 267 members and has an average attendance of around 180 people, according to a listing on the Southern Baptist Convention website. "Everybody has a connection to that church," Wilson Country Commissioner Albert Gamez Jr. said on "TODAY." "It's a family town. Anywhere you go, everybody recognizes everybody." Sherri Pomeroy, the wife of church pastor Frank Pomeroy, told NBC News in a statement that their daughter was among the victims. She said she and her husband were out of town in different states at the time of the shooting.

Emotions expressed by the content:

Anger	Calm	Disgust	Fear	Happy	Sadness	Surprise
[✓ Yes]	Yes	Yes	[✓ Yes]	Yes	[✓ Yes]	[✓ Yes]
No	[✓ No]	[✓ No]	No	[✓ No]	No	No

Application (a): Sentiment Annotation

Instruction: choose a proper description of the image.

There is a [cat / [✓ dog] / dolphin / hamster / monkey] playing a [basketball / plastic ball / [✓ soccer] / tennis / volleyball] on (by, in, ...) the [beach / creek / [✓ lawn] / hill / sea].

Application (b): Image Annotation

Fig. 1. Two examples involving *multi-label* annotation in crowdsourcing.

Truth inference for crowdsourcing started from the study of majority voting [10]—a simple but efficient method. Subsequently, a large number of studies addressed the truth inference issue using different techniques. Zheng et al. [11] reviewed ten-year efforts in this field by an empirical study, and observed that dozens of inference algorithms only focus on the *single-label* scenario, where the single label of each task has at least two (*binary-class*) or more (*multi-class*) options (values). However, in certain areas, it is quite common that each task associates with a set of appropriate labels. Figure 1 illustrates two examples of such applications. In

• *J. Zhang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.*
E-mail: jzhang@njust.edu.cn
• *X. Wu is with the Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei 230009, China, the Institute of Big Knowledge Science, Hefei University of Technology, Hefei 230009, China, and Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China.*

Manuscript received Jan. 1, 2019; revised ?? ??, 2019.

the first application (a), workers are required to provide their judgments to the presence of seven emotions conveyed through the passages. The workers only need to take a binary option—*yes* or *no*—as with the problem settings in traditional *multi-label* classification [12], [13]. A more general situation is that each label has more than two options as the second application (b) illustrates. In this example, a description of an image contains three placeholders (labels) representing an animal (*dog*), an object (*soccer*), and a location (*lawn*), respectively. All of these labels have multiple options. We refer this application as the *multi-class multi-label* annotation in this paper. When a label has multiple values in a dataset, many traditional multi-class learning algorithms transform this multi-class dataset to various binary ones. However, in crowdsourcing, it will be more user-friendly and efficient if multi-class and binary-class multi-label inference are handled uniformly.

One critical difference between the multi-label inference and the single-label one lies in the exploitation of the correlation among labels. For example, in the above sentiment annotation application, negative emotions such as fear, disgust, and sadness are usually strongly correlated. Leveraging this correlation may provide opportunities to further improve the quality of integrated labels. This paper proposes two novel probabilistic models to solve the multi-class multi-label inference problem in crowdsourcing. Experimental results under different scenarios confirm that our new models outperform other competitive alternatives. The main contributions of the paper are three-fold :

- the proposed models can jointly infer the unknown true values of the multiple labels of instances as well as the reliability of workers parameterized by confusion matrices in an EM procedure, confirming that joint inference outperforms repeatedly running single-label inference algorithms on each label;
- our second proposed model can explore and exploit the correlation among labels using a mixture of multiple independent multinoulli distributions, confirming that the mixture model substantially improves the performance of recovering true values, when labels are correlated;
- our proposed models can easily be simplified to the one-coin models, which uses one parameter to model the reliability of workers and has a better performance when annotation errors are uniformly distributed or the labels are sparse.

Some parts of this paper were published in KDD-2018 [14]. Compared with the previous version, the main extensions of the paper lie in: In addition to MCMLD, we also proposed another novel simpler model MCMLI to address the multi-label inference; We reduce both the proposed MCMLI and MCMLD models into the multi-class one-coin forms, namely MCMLI-OC and MCMLD-OC, which only use one parameter to model the reliability of each worker; We conducted additional experiments to evaluate the performance of MCMLI-OC and MCMLD-OC.

The remainder of the paper is organized as follows. Section 2 briefly reviews the related studies for the inference in crowdsourcing. Section 3 formalizes the problem definition and highlights three critical issues to be addressed. Section 4 presents two novel probabilistic models and the EM algorithms. Section 5 evaluates the performance of the proposed models on both simulated and real-world datasets. Section 6 simplifies the proposed models to one-coin models and investigate their performance. Section 7 concludes the paper and discusses some future work.

2 Related Work

Inferring true values of labels from multiple noisy labels is a fundamental research topic in crowdsourcing. Truth inference algorithms are usually unsupervised (agnostic), i.e., latent variables are only estimated through the observations without any prior knowledge, which is different from some *truth discovery* approaches [15], [16]. Many previous studies modeled the complexity of crowdsourced labeling systems from different aspects. As early as in 1979, Dawid & Skene proposed a statistical model (DS) for the consensus calculation from multiple uncertain sources [17], where the reliability of each source is modeled by a confusion matrix. Ipeirotis et al. [18] found that the DS model shows its effectiveness in crowdsourcing. Thus, some recent approaches are directly derived from the DS model by introducing priors to the parameters [19], adding constraints to workers [20], or optimizing initial settings [21]. Other approaches model the reliability of workers using fewer parameters [22], [23], [24], [25] but add additional parameters or variables to characterize workers' intention [22], [26] and bias [27], [28]. All these approaches are based on the probabilistic models and can be solved by an EM algorithm, gradient descending, or Markov Chain Monte Carlo (including Gibbs sampling). Among them, some [17], [21], [22], [23], [26], [27] can be used in multi-class scenarios, while the others [19], [20], [24], [25], [28] are only suitable for binary cases, but none was proposed for the multi-label annotation. In addition to the approaches based on the probabilistic models, some recent studies introduced other techniques to improve the accuracy of inference, including optimization [29], [30], [31], clustering [32], [33], noise correction [34], and injection of experts' validation [35].

Facing the diversity of the inference algorithms, we carefully choose among them according to several previous comprehensive empirical studies. Sheshadri and Lease [36] first developed a benchmark tool SQUARE and used it to evaluate six inference algorithms on ten real-world crowdsourcing datasets. Muhammadi et al. [37] studied the performance of eleven inference algorithms on three real-world and one synthetic datasets in a unified statistical framework. The most recent study [11] evaluated fourteen inference algorithms on five real-world datasets under various evaluation metrics, such as accuracy, F1-score, MAE (Mean Absolute Error)and RMSE (Root Mean Square Error). All these studies have shown some consistent conclusion that those approaches derived from the DS model [17] usually have a good probabilistic interpretability and exhibit strong robustness in most cases. Therefore, in our study, we still adopt the mainstream probabilistic models and use a set of confusion matrices to model the reliability of workers.

Several studies have touched the topic of multi-label annotation in crowdsourcing. For example, the early work [38] compared the reliability of crowdsourced workers with that of domain experts when they performed multi-label image annotation tasks. When talking about multi-label annotation, label correlation is one of the critical factors that affect inference performance, even though it also can help in a single-label scenario. For example, Fang et al. [39] proposed a two-round truth inference method which takes the label similarity (a relation between labels) into account to improve the accuracy of image annotation. Fang et al. [40] studied a kind of context-aware answer aggregation method, where a hidden Markov model is introduced to model the label correlation. Han et al. [41] incorporated the semantic relations among labels from external knowledge bases into the inference models. For multi-label annotation, as it consumes more budgets under the repeated labeling scheme, Bragg et al. [42] and Deng et al. [43] designed some optimal annotation acquisition workflows that leverage the correlation among different labels to reduce the costs. Hung et al. [44] proposed a Bayesian nonparametric model to aggregate the noisy labels from crowdsourcing, which takes advantage of Clustering-based Bayesian Combination of Multi-label Classifiers (cBCMC) [45] that also exploits the correlation among labels. However, their model is not entirely unsupervised (agnostic), requiring a particular portion of ground truths. Li et al. [46] also addressed the inference problem for the multi-label annotation using probabilistic models under an active learning paradigm, but they completely ignored the correlation among labels and assumed that each label has only two options. In this paper, our proposed unsupervised methods take into account the correlation among labels and jointly infer the true values for all labels of each instance.

3 Problem Statement

In this section, we first formalize multi-label annotation in crowdsourcing and then discuss three critical issues to be addressed in this paper.

3.1 Multi-Label Annotation

The dataset posted on a crowdsourcing system to collect repeated noisy labels is denoted by $\mathcal{D} = \{< \mathbf{x}_i, \mathbf{y}_i >\}_{i=1}^{I}$, where each instance $\mathbf{x}_i$ associates with an unknown true class label set $\mathbf{y}_i$. Each $\mathbf{y}_i$ consists of M labels, which is denoted by $\mathbf{y}_i = [y_i^{(1)}, ..., y_i^{(M)}]$. All true label sets of the entire dataset are denoted by $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_I\}$. In a real-world application, because all instances are usually created from the same template, the present order of the labels on each instance is the same. Thus, we can simply say "$y_i^{(m)}$ is the mth label of instance $\mathbf{x}_i$." For different labels, they usually have different numbers of values because each label reflects a specific aspect of the instance. For example, in Figure 1.(b), the first label (*animal*) may include dozens of values, but the third label (*place*) may include only several values. To simplify the formalization, we let each label associate with exactly K values, i.e., $y_i^{(m)} \in \{1, ..., K\}$, where K is the maximum number of values that the labels have. This does not have any side effect in practice. For example, given $K = 5$, if the mth label only has three values, we can treat the $y_i^{(m)} = 4$ and $y_i^{(m)} = 5$ as two unobserved values. A probabilistic model will guarantee that all probabilities with respect to the unobserved values are always zero (i.e., $\mathrm{P}(y_i^{(m)} = 4) = \mathrm{P}(y_i^{(m)} = 5) = 0$), which does not affect the results of inference. Note that, in this simplification, the $k \in \{1, ..., K\}$ is just an index of that value because values of different labels represent different things. However, we can easily make a discrimination of them by adding superscripts , i.e., $k^{(m)} \neq k^{(n)}$ if $m \neq n$.

There are totally J crowdsourced workers labeling the dataset. All collected noisy labels for each instance $\mathbf{x}_i$ form a matrix $L_i^{J\times M} = [\mathbf{l}_{i1}, ..., \mathbf{l}_{iJ}]^T$, where each element (noisy label) $l_{ij}^{(m)} = k \in \{0\} \cup \{1, ..., K\}$ in a row vector $\mathbf{l}_{ij}$ represents that worker j provides value k to the mth label of instance $\mathbf{x}_i$. Particularly, $l_{ij}^{(m)} = 0$ means that worker j does not provide any value to the mth label of instance $\mathbf{x}_i$. All crowdsourced labels of the entire dataset are denoted by $\mathbf{L} = \{L_1, ..., L_I\}$.

3.2 Three Critical Issues

Our study aims to solve three critical issues in multi-class multi-label inference for crowdsourcing.

Issue 1: Jointly inferring truth labels. The primary objective of an inference algorithm for crowdsourcing is to infer the true labels $Y^{I\times M}$ for all instances from the collected noisy labels $\mathbf{L}^{I\times J\times M}$ and simultaneously minimize the overall inference error rate as follows:

$$\epsilon = \min\left\{\frac{1}{I\cdot M}\sum_{i=1}^{I}\sum_{m=1}^{M}\mathbb{I}\left(\hat{y}_i^{(m)} \neq y_i^{(m)}\right)\right\}, \qquad (1)$$

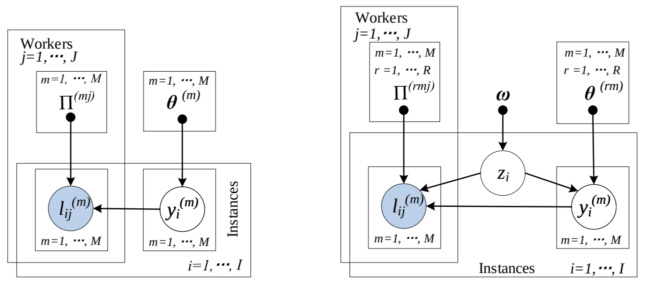
where $\mathbb{I}$ is an indicator function. It outputs 1 when the test condition satisfies. Otherwise, it outputs 0.

A simple approach to solve this issue is successively running a *single-label* inference algorithm on each label of all instances. That is, the *single-label* inference algorithm totally runs M rounds on the same dataset. In the mth round, the inference error rate on the mth label is minimized. However, there exists a question that has never been answered. If we can jointly infer all M labels of each instance at once, will the total error rate be less than that obtained in the above manner?

Issue 2: Exploring and exploiting the correlation among the labels. In multi-label annotation, the correlation among the labels exhibits great complexity. On the one hand, there naturally exists correlation among the true labels. On the other hand, this correlation, in turn, has some impact on the distribution of the crowdsourced labels. Workers can mutually check their own answers on different labels to improve the label quality if they are aware of the correlation among the true labels.

For instance $\mathbf{x}_i$ and the J workers labeling it, if we use a joint probability of $\mathrm{P}(\mathbf{l}_i, \mathbf{y}_i)$ to model the correlation among both the crowdsourced labels and true labels, and use the chain rule of conditional probability to solve it, it will results in a complexity of $O(c \cdot I \cdot J \cdot K^M \cdot (K^J)^M)$ in total, which is computationally infeasible. Therefore, the second question is "Is there a simple method to model the correlation among the labels and does it improve the inference accuracy?"

Issue 3: Modeling the reliability of the workers. Estimates of the reliability of crowdsourced workers is an important function of an inference algorithm. A multi-class



(a) The MCMLI model (b) The MCMLD model

Fig. 2. Probabilistic graphical model representation of the `MCMLI` and `MCMLD` models.

multi-label inference algorithm must model the reliability of the workers on M labels with $M \cdot K^2$ values. Furthermore, the upper-layer applications require the inference algorithm to provide fine-grained information as far as possible.

4 The Proposed Methods

In this section, we first present two novel probabilistic model for multi-class multi-label inference. Then, we solve the models with EM algorithms. Finally, we discuss the parameter settings in the models.

4.1 Label-Independent Model

We first propose a novel probabilistic generative model, namely *multi-class multi-label independent* (MCMLI) model, under an assumption that the labels are mutually independent. The probabilistic graphical model representation of MCMLI is illustrated in Figure 2.(a).

Generation of true labels. For multi-class classification, each true label (supposed to be the mth label in multi-label annotation) is independently drawn from a multinoulli distribution with parameters $\boldsymbol{\theta}^{(m)} = [\theta_1^{(m)}, ..., \theta_K^{(m)}]$, where $\sum_{k=1}^{K} \theta_k^{(m)} = 1$. That is, for the mth label of an instance $\mathbf{x}_i$, we have $\mathrm{P}(y_i^{(m)} | \boldsymbol{\theta}^{(m)}) = \prod_{k=1}^{K} \left(\theta_k^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}$. Consequently, the probability of the M-dimensional true label vector $\mathbf{y}_i$ of the instance is

$$\mathrm{P}(\mathbf{y}_i | \Theta) = \prod_{m=1}^{M} \prod_{k=1}^{K} \left(\theta_k^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}, \tag{2}$$

where $\Theta = [\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(M)}]$ is a set of parameters for all M multinoulli distributions.

Generation of crowdsourced labels. In multi-class classification, the confusion matrix is a powerful tool that can comprehensively depict the distribution of a classifier's capability over all pairs of classes, providing fine-grained information. In MCMLI, each worker j independently provides value to each label of an instance $\mathbf{x}_i$. We use a set of confusion matrices $\boldsymbol{\Pi}^{(j)} = [\Pi^{(1j)}, ..., \Pi^{(Mj)}]$ to model the reliability of worker j with respect to M labels. We denote all sets of confusion matrices of totally J workers by a parameter set $\widetilde{\boldsymbol{\Pi}} = \{\boldsymbol{\Pi}^{(1)}, ..., \boldsymbol{\Pi}^{(J)}\}$. In matrix $\Pi^{(mj)}$, each element $\pi_{kd}^{(mj)} (1 \le k, d \le K)$ represents the probability of worker j labeling (true) class k as class d on the mth label, which derives $\mathrm{P}(l_{ij}^{(m)} = d | y_i^{(m)} = k) = \pi_{kd}^{(mj)}$. That is, $l_{ij}^{(m)}$ conditioning on $y_i^{(m)} = k$ obeys a multinoulli distribution with parameters $[\pi_{kd}^{(mj)}]_{d=1}^{K}$ and $\sum_{d=1}^{K} \pi_{kd}^{(mj)} = 1$.

Consider each instance $\mathbf{x}_i$ is independently labeled by J workers, the likelihood of all observed noisy labels on the instance can be calculated as follows:

$$\mathrm{P}\left(L_i | \widetilde{\boldsymbol{\Pi}}, \Theta\right) = \sum_{k^{(1)}=1}^{k^{(1)}=K} \cdots \sum_{k^{(M)}=1}^{k^{(M)}=K} \Big[\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)} | \Theta\right) \cdot \mathrm{P}\left(L_i | y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, \widetilde{\boldsymbol{\Pi}}\right) \Big], \tag{3}$$

where $(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}) \overset{\text{def}}{=} (y_i^{(1)} = k^{(1)}, ..., y_i^{(M)} = k^{(M)})$. Because we assume that the true labels of an instance are mutually independent, we have

$$\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)} | \Theta\right) = \theta_{k^{(1)}}^{(1)} \cdots \theta_{k^{(M)}}^{(M)}, \tag{4}$$

$$\mathrm{P}\left(L_i | y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, \widetilde{\boldsymbol{\Pi}}\right) = \prod_{j=1}^{J} \mathrm{P}\left(l_{ij}^{(1)}, ..., l_{ij}^{(M)} | y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, \widetilde{\boldsymbol{\Pi}}\right)$$
$$= \prod_{j=1}^{J} \left(\prod_{m=1}^{M} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(mj)}\right)^{\mathbb{I}(l_{ij}^{(m)}=d^{(m)})} \right). \tag{5}$$

Here, because we need to present the prior of $\mathbf{y}_i$ under the conditions that each of its element is assigned with a different value, we use the superscript (m) to distinguish these ks on different labels. (Refer to Section 3.1 for the physical meaning of the superscript.) Similarly, the variable d is also decorated by the superscript (m).

Finally, the log-likelihood of all crowdsourced labels of the entire dataset is

$$\ln \mathrm{P}(\mathbf{L} | \Theta, \widetilde{\boldsymbol{\Pi}}) = \sum_{i=1}^{I} \ln \Bigg(\sum_{k^{(1)}=1}^{k^{(1)}=K} \cdots \sum_{k^{(M)}=1}^{k^{(M)}=K} \Big[\prod_{m=1}^{M} \theta_{k^{(m)}}^{(m)} \cdot \prod_{j=1}^{J} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(mj)}\right)^{\mathbb{I}(l_{ij}^{(m)}=d^{(m)})} \Big] \Bigg). \tag{6}$$

4.2 Label-Dependent Model

The second proposed model illustrated in Figure 2.(b), namely *multi-class multi-label dependent* (MCMLD) model, considers the correlation among the labels. In this model, the correlation among the (true) labels is modeled by *soft* clusters. The instances belonging to the same cluster suggest that their labels exhibit some correlation. We assign each instance $\mathbf{x}_i$ a latent variable z_i to indicate its cluster membership. Given the total number R of clusters, the probability of instance $\mathbf{x}_i$ being in cluster r ($1 \le r \le R$) is $\mathrm{P}(z_i = r) = \omega_r$. Thus, for each instance $\mathbf{x}_i$, z_i is drawn from a multinoulli distribution, i.e., $\mathrm{P}(z_i | \boldsymbol{\omega}) = \prod_{r=1}^{R} (\omega_r)^{\mathbb{I}(z_i=r)}$, where $\boldsymbol{\omega} = [\omega_1, ..., \omega_R]$ are its parameters. Obviously, a *soft* cluster means that each instance belongs to cluster r with a probability ω_r and $\sum_{r=1}^{R} \omega_r = 1$. In this soft cluster setting, the generation of noisy label $l_{ij}^{(m)}$ is also can be distributed to R clusters according to the proportion of the instance lying in each cluster. That is, we have $\mathrm{P}(l_{ij}^{(m)} = d | y_i^{(m)} = k, z_i = r) = \pi_{kd}^{(rmj)}$, where $\sum_{r=1}^{R} \pi_{kd}^{(rmj)} = \pi_{kd}^{(mj)}$. Accordingly, we also have $\sum_{d=1}^{K} \pi_{kd}^{(rmj)} = \omega_r$ and $\sum_{r=1}^{R} \sum_{d=1}^{K} \pi_{kd}^{(rmj)} = 1$.

Generation of true labels. In MCMLD, each true label vector $\mathbf{y}_i$ is generated from R independent clusters with the probabilities $[\omega_1, ..., \omega_R]$, which is a *mixture of multiple independent multinoulli* distributions. Thus, the probability of $\mathbf{y}_i$ of instance $\mathbf{x}_i$ has the form

$$\mathrm{P}(\mathbf{y}_i|\boldsymbol{\Theta}, \boldsymbol{\omega}) = \sum_{r=1}^{R} \omega_r \prod_{m=1}^{M} \prod_{k=1}^{K} \left(\theta_{rk}^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}, \quad (7)$$

where $\boldsymbol{\omega} = [\omega_1, ..., \omega_R]$ are also called the mixing coefficients of multiple multinoulli distributions with the parameter set $\boldsymbol{\Theta} = [\Theta_1, ..., \Theta_R]$. For instance $\mathbf{x}_i$, the joint probability of variables $\mathbf{y}_i$ and z_i can be calculated as follows:

$$\begin{aligned}\mathrm{P}(\mathbf{y}_i, z_i) &= \mathrm{P}(z_i)\,\mathrm{P}(\mathbf{y}_i|z_i)\\ &= \prod_{r=1}^{R}\left[\omega_r \prod_{m=1}^{M}\prod_{k=1}^{K}\left(\theta_{rk}^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}\right]^{\mathbb{I}(z_i=r)}. \quad (8)\end{aligned}$$

Generation of crowdsourced labels. The generation of crowdsourced labels in MCMLD is similar to that in MCMLI. Under this mixture model, the probability of the true labels with fixed known values (i.e., Eq.(4) in MCMLI) has the form

$$\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}|\boldsymbol{\Theta}, \boldsymbol{\omega}\right) = \sum_{r=1}^{R} \omega_r \theta_{rk^{(1)}}^{(1)} \cdots \theta_{rk^{(M)}}^{(M)}. \quad (9)$$

Accordingly, the log-likelihood of all crowdsourced labels of the entire dataset can be calculated as follows:

$$\begin{aligned}\ln \mathrm{P}(\mathbf{L}|\boldsymbol{\Theta}, \boldsymbol{\omega}, \widetilde{\boldsymbol{\Pi}}) = \sum_{i=1}^{I} \ln \Bigg(\sum_{k^{(1)}=1}^{k^{(1)}=K} \cdots \sum_{k^{(M)}=1}^{k^{(M)}=K} \Big[\sum_{r=1}^{R} \omega_r \\ \cdot \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)} \cdot \prod_{j=1}^{J} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(rmj)}\right)^{\mathbb{I}(l_{ij}^{(m)}=d^{(m)})} \Big]\Bigg). \\ (10)\end{aligned}$$

4.3 Inference with EM for MCMLI

Our optimization objective is to maximize the likelihoods of the observed data $\mathbf{L}$ defined by Eq.(6), which can be achieved by expectation-maximization (EM) algorithm, which iteratively applies E-step and M-step.

E-step. We calculate the expected value of the log likelihood function, with respect to the conditional distribution of $\mathbf{Y}$ given the observed noisy labels $\mathbf{L}$ under the current estimates of parameters $\boldsymbol{\Psi}_1^{old}$:

$$\mathcal{Q}_1\left(\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_1^{old}\right) = \mathbb{E}_{\mathbf{Y}|\mathbf{L}, \boldsymbol{\Psi}_1^{old}}\left[\ln \mathrm{P}\left(\mathbf{L}, \mathbf{Y}|\boldsymbol{\Psi}_1\right)\right], \quad (11)$$

where $\boldsymbol{\Psi}_1 = \{\Theta, \widetilde{\boldsymbol{\Pi}}\}$. In fact, this will simply result in the calculation of the joint posterior probability of $\mathbf{y}_i$ for each instance $\mathbf{x}_i$ by applying Bayes' theorem as follows:

$$\begin{aligned}\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}|\mathbf{L}, \boldsymbol{\Psi}_1\right) &= \prod_{m=1}^{M} \mathrm{P}\left(y_i^{(m)} = k^{(m)}|\mathbf{L}, \boldsymbol{\Psi}_1\right)\\ &\propto \prod_{m=1}^{M} \theta_{k^{(m)}}^{(m)} \prod_{j=1}^{J} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(mj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d^{(m)}\right)}. \quad (12)\end{aligned}$$

Eq.(12) directly provides the expected value of an indicator function (calculated as a marginal probability) that will be used in M-step as follows:

$$\begin{aligned}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] &= \mathrm{P}\left(y_i^{(m)} = k\right)\\ &= \sum_{y_i^{(m')}(\forall m' \in \{1,...,M\} \backslash m)} \mathrm{P}\left(y_i^{(m')} = k, \cdots |\mathbf{L}, \boldsymbol{\Psi}_1\right) \quad (13)\end{aligned}$$

M-step. We determine the revised parameter estimate $\boldsymbol{\Psi}_1$ by maximizing the objective function $\mathcal{Q}_1$ formed as Eq.(11), i.e., $\boldsymbol{\Psi}_1^{new} = \mathrm{argmax}_{\boldsymbol{\Psi}_1}\, \mathcal{Q}_1(\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_1^{old})$. The parameters are updated as follows:[1]

$$\hat{\theta}_k^{(m)} = \sum\nolimits_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] \Big/ I, \quad (14)$$

$$\hat{\pi}_{kd}^{(mj)} = \frac{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] \mathbb{I}\left(l_{ij}^{(m)} = d\right)}{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] \mathbb{I}\left(l_{ij}^{(m)} \neq 0\right)}. \quad (15)$$

After convergence. The integrated labels of instance $\mathbf{x}_i$ should be the sequence of class values with the maximum posterior probability. That is,

$$\hat{y}_i^{(1)}, ..., \hat{y}_i^{(M)} = \underset{k^{(1)},...,k^{(M)}}{\mathrm{argmax}} \left\{\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}|\mathbf{L}, \boldsymbol{\Psi}_1\right)\right\}. \quad (16)$$

4.4 Inference with EM for MCMLD

To maximize the likelihood of the observed labels $\mathbf{L}$ defined by Eq.(10), we use an EM algorithm again. The difficulty of applying EM to MCMLD lies in that there are two types of latent variables $\mathbf{Y}$ and $\mathbf{z}$. (Traditionally, a mixture model only has one type of latent variables.)

E-step. We first construct an expectation function of the log-likelihood of the observed data $\mathbf{L}$ conditioning on latent variables $\mathbf{Y}$ and $\mathbf{z}$ as follows:

$$\mathcal{Q}\left(\boldsymbol{\Psi}_2, \boldsymbol{\Psi}_2^{old}\right) = \mathbb{E}_{\mathbf{Y},\mathbf{z}|\mathbf{L}, \boldsymbol{\Psi}_2^{old}}\left[\ln \mathrm{P}\left(\mathbf{L}, \mathbf{Y}, \mathbf{z}|\boldsymbol{\Psi}_2\right)\right], \quad (17)$$

where $\boldsymbol{\Psi}_2 = \{\boldsymbol{\Theta}, \widetilde{\boldsymbol{\Pi}}, \boldsymbol{\omega}\}$.

For the instance $\mathbf{x}_i$ with two types of latent variables $\mathbf{y}_i$ and z_i, when the crowdsourced labels $\mathbf{L}$ are observed, we must calculate the joint posterior probability of $\mathbf{y}_i$ and z_i by Bayes' theorem as follows:

$$\begin{aligned}&\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, z_i = r|\mathbf{L}, \boldsymbol{\Psi}_2\right)\\ &\propto \mathrm{P}\left(L_i|y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, z_i = r, \boldsymbol{\Psi}_2\right) \mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}, z_i = r|\boldsymbol{\Theta}, \boldsymbol{\omega}\right)\\ &= \omega_r \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)} \prod_{j=1}^{J} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(rmj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d^{(m)}\right)}. \quad (18)\end{aligned}$$

Eq.(18) provides the expected values of two indicator functions (calculated as two marginal probabilities) that will be used in the M-step as follows:

$$\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] = \sum_{y_i^{(m')}(\forall m' \in \{1,...,M\} \backslash m), \forall z_i} \mathrm{P}\left(y_i^{(m)} = k, \cdots |\mathbf{L}, \boldsymbol{\Psi}_2\right), \quad (19)$$

1. The detailed derivations of the equations in the M-steps of MCMLI and MCMLD are in the appendix of this paper.

$$\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]=\sum_{y_i^{(m)}(\forall m\in\{1,\ldots,M\})}\mathrm{P}\left(z_i=r,\cdots\big|\mathbf{L},\mathbf{\Psi}_2\right). \quad (20)$$

M-step. We determine the revised parameter estimate $\mathbf{\Psi}_2$ by maximizing the objective function $\mathcal{Q}_2$ formed as Eq.(17), i.e., $\mathbf{\Psi}_2^{new}=\mathrm{argmax}_{\mathbf{\Psi}_2}\,\mathcal{Q}_2(\mathbf{\Psi}_2,\mathbf{\Psi}_2^{old})$. The parameters are updated as follows:

$$\hat{\theta}_{rk}^{(m)}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}}{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]}, \quad (21)$$

$$\hat{\omega}_r=\sum\nolimits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]/I. \quad (22)$$

The updated function of parameter $\pi_{kd}^{(rmj)}$ has the similar form as Eq.(15) in MCMLI but only considers the instances in cluster r:

$$\hat{\pi}_{kd}^{(rmj)}=\frac{\sum\limits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)}{\sum\limits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}\neq 0\right)}. \quad (23)$$

After convergence. The integrated labels of instance $\mathbf{x}_i$ should be the sequence of class values with the maximum marginal posterior probability. That is,

$$\hat{y}_i^{(1)},\ldots,\hat{y}_i^{(M)}=\underset{k^{(1)},\ldots,k^{(M)}}{\mathrm{argmax}}\left\{\sum_{r=1}^{R}\mathrm{P}\Big(y_i^{(\cdot)}\overset{1..M}{=}k^{(\cdot)},z_i=r\big|\mathbf{L},\mathbf{\Psi}_2\Big)\right\} \quad (24)$$

The latent variable z_i can be estimated by maximizing the marginal posterior probability as follows:

$$\hat{z}_i=\underset{r}{\mathrm{argmax}}\left\{\sum_{k^{(1)}=1}^{k^{(1)}=K}\cdots\sum_{k^{(M)}=1}^{k^{(M)}=K}\mathrm{P}\Big(y_i^{(\cdot)}\overset{1..M}{=}k^{(\cdot)},z_i=r\big|\mathbf{L},\mathbf{\Psi}_2\Big)\right\}. \quad (25)$$

4.5 Setting Parameter R in MCMLD

One critical issue of the proposed MCMLD model is the setting of the parameter R under an unsupervised condition. Algorithm.MCMLD below presents a novel method of setting its value, which explores the correlation among the crowdsourced labels.

First, it combines all crowdsourced labels into a two-dimensional matrix $U^{(I\cdot J)\times M}$. Each row in U can be viewed as a data point, whose features consist of M labels provided by a worker. Then, we apply the principal components analysis (PCA) algorithm on matrix U. The dimension of the features will be reduced from M to N ($N\leq M$), which stands for the number of irrelevant labels. In practice, we use the normalized eigenvalues ($\boldsymbol{\eta}$) of the covariance matrix in PCA, which present the contributions of the components. If $\boldsymbol{\eta}$ is in the descending order, step 3 in the algorithm will determine the value of R, which means that R components contribute approximately ρ portion of the variance. Note that setting ρ is much easier than setting R. For example, in our experiments, we simply deem that if the R components contribute 70% of the variance, they are irrelevant.

5 Experiments

In this section, we evaluate the performance of the proposed methods on both simulated and real-world datasets.

Algorithm 1 The MCMLD Inference for Crowdsourcing

```
Input: Multi-Class Multi-Label Crowdsourced Dataset $\mathcal{D}$, $\rho$
Output: Integrated labels $\hat{\boldsymbol{y}}_i$ for each instance $\mathbf{x}_i$
1: Create a two-dimensional matrix $U^{(I\cdot J)\times M}$ from $\mathcal{D}$.
2: Apply PCA on $U$, obtaining normalized eigenvalues of the covariance matrix $\boldsymbol{\eta}=\{\eta_1,...,\eta_N\}$ in descending order.
3: Set $R=\mathrm{argmin}_{1\leq n\leq N}|\rho-\sum_{b=1}^{n}\eta_b|$, where $\rho\in(0,1]$.
4: repeat
5:     E-step: calculate expectation by Eqs.(19) and (20).
6:     M-step: update parameters by Eqs.(21), (22) and (23).
7: until convergence
8: For each instance $\mathbf{x}_i$, calculate $\hat{\boldsymbol{y}}_i$ by Eq.(24).
9: return all $\hat{\boldsymbol{y}}_i$s.
```

5.1 Evaluation metric and algorithms in comparisons

Evaluation metric. Because our primary objective is to accurately infer the true values of all labels on each instance, we merely use the label-based accuracy (i.e., the inverse of Eq.(1)) as the evaluation metric.

Algorithms in comparisons. To the best of our knowledge, there is no entirely unsupervised algorithm proposed for the multi-class multi-label annotation in crowdsourcing. Therefore, we choose four alternative *single-label* algorithms to conduct comparisons.

- **MV** (majority voting) is usually served as a baseline for label inference.
- **DS** [17] uses the confusion matrix to model workers and makes inference using maximum likelihood estimation. Previous large-scale empirical studies [11], [36], [37] have shown its robustness and efficiency.
- **SpectralDS** [21] is a variant of DS, where the initial values of the confusion matrices are carefully set.
- **iBCC** [47] is a non-EM based inference method, where a Bayesian generative model of the crowdsourced annotation is solved by Gibbs sampling.

The algorithms are chosen because they are not only efficient and robust but also suitable for both binary and multi-class cases [11]. For each of these *single-label* algorithms, from the 1st labels of all instances to the Mth labels, we repeatedly run the algorithm to infer their true values.

5.2 Evaluation by simulations

The advantage of simulation lies in its ability of comprehensively revealing the differences among the compared algorithms under various predefined conditions, especially, when there are no sufficient real-world crowdsourcing datasets to cover different situations.

Datasets for simulation. We selected four *real-world* multi-label classification datasets from the multi-label learning library MULAN[2] to conduct our simulations. Using the real-world datasets is because their labels naturally exhibit some degree of correlation. However, these labels, collected from experts and served as the ground truth, only have binary values, since they were created for the traditional

2. http://mulan.sourceforge.net/datasets-mlc.html

TABLE 1
Six multi-label datasets for simulations

Dataset	#inst.	#lb.	#cls.	card.	dens.
emotions	593	6	2	1.869	0.311
flags	194	7	2	3.392	0.485
scene	2407	6	2	1.074	0.179
yeast	2417	14	2	4.237	0.303
pentacorrel	600	5	3	5.00	1.000
decasynth	1000	10	4	10.00	1.000

Note: **#inst.** represents the number of instances; **#lb.** represents the number of labels on each instance; **#cls.** represents the number of classes on each label; **#card.** and **#dens.** represent *cardinality* and *density* [12], respectively.

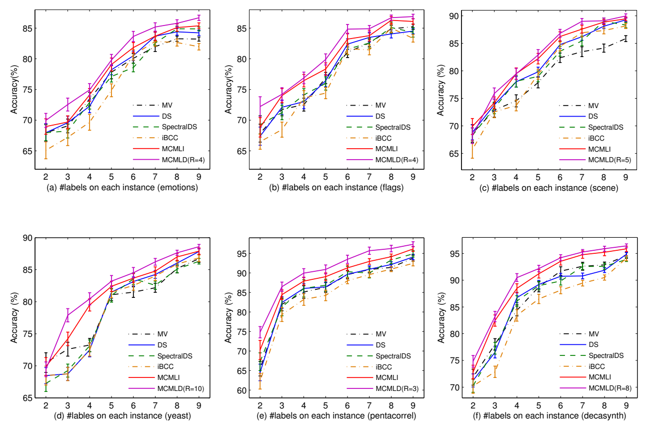


Fig. 3. Comparison results on six datasets under Scenario 1.

multi-label classification. To make a comprehensive investigation to the multi-class multi-label inference, we created another two datasets *pentacorrel* and *decasynth*, in which each label has three and four values, respectively. Dataset *pentacorrel* was created with strong label-correlation. In this dataset, the values of the first, third and fifth labels always appear three patterns $(1, 2, 3)$, $(2, 3, 1)$ and $(3, 1, 2)$. For dataset *decasynth*, we relaxed the correlation of labels. The values of labels are approximately uniformed distributed and generated regardless of others. The detailed information of these datasets for simulation is listed in Table 1.

Scenario 1: Annotation errors are uniformly distributed. In our simulation, the overall accuracy of each worker is independently drawn from a uniform distribution $U(0.60, 0.80)$. Previous empirical studies [11], [36] have shown that the accuracies of workers are normally within this range. Under Scenario 1, we simulate that workers are never aware of the label-correlation. That is, their annotation errors errors are uniformly distributed. When workers provide values to the mth labels of all instances, they randomly select a fixed portion of the instances to provide true values. Furthermore, on a multi-class label, the errors are also uniformly distributed across the incorrect values. For MCMLD, the parameter R is set using the PCA-based method in Section 4.5. Specifically, we use `sklearn.decomposition.pca` in `scikit-learn` (a machine learning library written in Python) to perform PCA. Then, R is set to the value that the first R components of the returned result `explained_variance_ratio_` contribute 70% variance. For all confusion matrices that model the reliability of workers, their diagonal elements are randomly initialized with the values within the range of $[0.70, 0.90]$ and their non-diagonal elements are set to the remainder (i.e., one minus the diagonal value) averaged by $K - 1$. The number of workers increases from 2 to 9. For each setting, we first generate a synthetic dataset and then run the six algorithms on it. The procedure is repeated 20 times and the means of the accuracies and their standard deviations are reported.

Figure 3 shows the comparison results under Scenario 1. Intuitively, we might think that the performance of the DS model running multiple times (abbreviated as *multiple-DS*) on all labels of a dataset should be the same as that of the MCMLI model. Actually, we find that in most cases, MCMLI outperforms DS. The average excess is round 2% (absolute value), and the maximum is over 5% (on *yeast* when #labels on each instance is 4). The relationship between multiple-DS and MCMLI is like that between coordinate descending and gradient descending. Suppose each instance has M labels. Multiple-DS optimizes the maximum likelihood estimate for each label one by one. First, it finds the MLE for the first label, and then, the second label, and so on, like the coordinate descending performs optimization against only one coordinate each time. After the M-th label is processed, the final status of the overall objective function (suppose to be the sum of M sub-objective functions) is not necessarily the same as the MLE for the joint of M labels. Comparably, MCMLI performs MLE in the space defined by the joint of M labels, like the gradient descending performs optimization against the gradient (calculated in a space with M coordinate) each time. That is, when MCMLI converges, its objective function is definitely in a local optimum, while that of multiple-DS is not. Only under an extremely independent condition do they perform the same. Thus, MCMLI is usually better than multiple-DS.

In our simulation, the crowdsourced labels are independently generated according to their true values, so that their values may inevitably inherit a certain degree of correlation from their true values. Just because of this point, not only MCMLI outperforms multiple-DS, but also MCMLD sightly outperforms MCMLI in most cases. In addition, on two multi-class multi-label datasets, especially on *decasynth*, both MCMLI and MCMLD obviously present advantages. On the dataset with a stronger label-correlation *pentacorrel*, the gaps between MCMLD and MCMLI are larger than that on *decasynth*. Therefore, jointly inferring multiple labels can achieve a better local optimum, comparing to running a *single-label* inference multiple times on a dataset.

There are also some other observations. SpectralDS does not obviously exhibit advantages over the traditional DS, which indicates that it is not easy to find optimal initial settings for an EM algorithm suitable for different circumstances. iBCC, which uses Gibbs sampling for inference, performs worst. This indicates that as a brutal approximate algorithm, Gibb sampling usually performs worse than EM does in our simulations.

Scenario 2: Annotation with strong label-correlation. Under this scenario, workers are aware of the label-correlation. When labeling, the workers can mutually check their own answers on different labels with the help of the correlation. We strengthen the correlation as follows. For

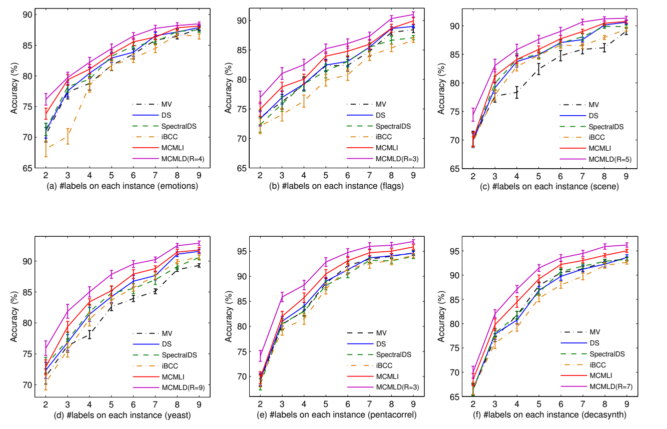


Fig. 4. Comparison results on six datasets under Scenario 2.

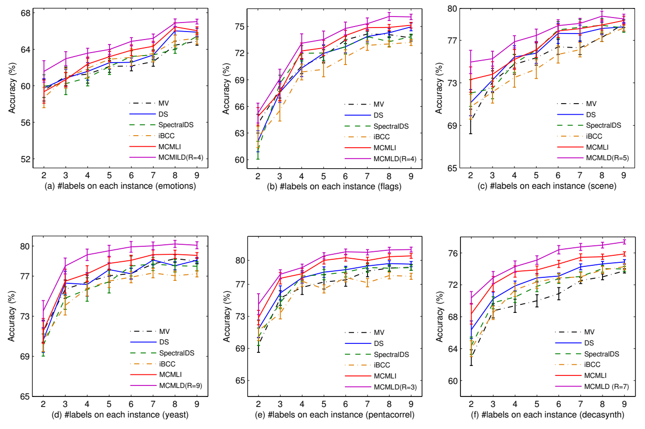


Fig. 5. Comparison results on six datasets under Scenario 3.

the four real-world datasets and *decasynth* in Table 1, we run PCA on the true label matrix $Y^{I\times M}$ to obtain a correlation matrix, then we select $t = (30\% \times \#lb.)$ pairs of labels with the top-t large correlation coefficients from the matrix (i.e., for *emotions*, *flags*, and *scene*, $t = 2$, for *yeast*, $t = 4$, and for *decasynth*, $t = 3$). For dataset *pentacorrel*, we have already known that the strong-correlated labels are the first, third and fifth ones. The crowdsourced label simulation is as follows. Suppose (l_a, l_b) is a pair of strong-correlated labels on an instance. If the value of label l_a has not been generated, we generate its value with the correct probability within the range of $[0.75, 0.90]$. The strong-correlated labels can help workers easily verify their judgments, which improves their accuracies. When label l_a is set to its true value on the instance, label l_b has 95% probability of being its true value. Although the workers have higher accuracies when labeling the strong-correlated labels, we still let their overall accuracies within the range of $[0.60, 0.80]$ by tuning the accuracies on the non-correlated labels.

Figure 4 shows the comparison results under Scenario 2. Note that there are two types of correlation: correlation among the true labels and that among the crowdsourced labels. Under Scenario 1, we assume that when labeling instances, workers are never aware of the existence of label-correlation which can help them mutually check their own answers. However, because the correlation is implied in the truth labels, the crowdsourced labels also exhibit a certain degree of correlation. Under Scenario 2, we simulate that the workers are aware of the correlation and mutually check their own answers to improve their correct rates. The consequence is that the correlation among crowdsourced labels is further strengthened, which directly results in the decline of the calculated values of the parameter R. On datasets *flags*, *yeast* and *decasynth*, the values of Rs individually reduce one compared with those values under Scenario 1. When we compare Figures 4.(b), 4.(c), 4.(d), and 4.(f) with Figures 3.(b), 3.(c), 3.(d), and 3.(f), respectively, it is easy to find that under Scenario 2, MCMLD significantly outperforms MCMLI under the most settings of the numbers of labels on each instance (represented as $\#labels$ in the following) on these datasets. Under Scenario 1, MCMLD does not significantly outperforms MCMLI at some points, for examples, on *flags* when $\#labels = 3, 4,$ or 8, on *scene* when $\#labels = 4, 5,$ or 8, on *yeast* when $\#labels = 5, 6$ or 8, and on *decasynth* when $\#labels = 7, 8,$ or 9. Comparatively, under Scenario 2, at all above points, the performance of MCMLD is significantly better than that of MCMLI with the increments within the range of $1\% \sim 4\%$ (absolute value). In addition, when we examine the performance of MCMLI, comparing with the multiple-DS, we find that the difference of their performance is not obviously enlarged. It seems that the performance of MCMLI is only relevant to the correlation of the true labels while not to the correlation of the crowdsourced labels. These results suggest that the proposed MCMLD model can benefit from the growth of the crowdsourced label correlation. As for other compared models, the performance of SpectralDS is similar to that of DS, and iBCC is not robust in this simulation again. Consistently, the proposed MCMLI and MCMLD both outperform the compared alternatives.

Scenario 3: Biased annotation. The crowdsourced annotation often suffers from the biases of workers. Many studies keep focusing on the biased annotation issue in recent years, from the previous inference models [27], [32] to the most recently cognitive mechanism [48]. The causes for the biased annotation are complicated. It may be due to workers' subjective social reality, lack of expertise, common sense decisions, and even the spam submissions. In our study, we simply explain the biased annotation as a phenomenon that the workers tend to provide some classes overwhelming the others. For the binary annotation, we assume that workers tend to provide the negative class, which results in the increase of their accuracies on the true negative labels and the decrease of their accuracies on the true positive ones. For datasets *emotions*, *flags*, *scene*, and *yeast*, whose $K = 2$ (binary annotation for each label), we let the workers' accuracies on the negative class be within the range of $[0.8, 0.9]$ and those on the positive class be within the range of $[0.4, 0.5]$. For the multi-class datasets *pentacorrel* (*decasynth*), we let the workers' accuracies on the first class (the first two classes for *decasynth*) be within the range of $[0.8, 0.9]$ and the others be within the range of $[0.4, 0.5]$. Again, we simulate that there are at least two and at most nine biased workers labeling all instances.

Figure 5 shows the comparison results under Scenario 3. We notice that although the overall accuracies of the workers are within the same range as the those of workers under Scenarios 1 and 2, the inference accuracies on all datasets consistently decrease. The decrease of the inference accuracies is more than 15% (absolute value), which suggests that the biased annotation has negatively impact on the performance of inference algorithms. Under this scenario, we find that on the most datasets, expect *scene* whose underline true labels are not correlated, the performance of MCMLI obviously outperforms that of the compared *single-label* alternatives. Furthermore, the performance of MCMLD also obviously outperforms that of MCMLI in the most cases. Also, some other interesting details can be found. First, the biased annotation actually increases the degree of the crowdsourced label correlation, which results in MCMLD's significantly outperforming MCMLI. This fact is also indicated by the calculation of the parameter R on *yeast* (also on *decasynth*), whose values are 10 and 9 (8 and 7 for *decasynth*) under Scenarios 1 and 3, respectively. Second, both the proposed MCMLI and MCMLD are more advantageous in multi-class annotation, which can be observed under all the three scenarios. Finally, when the number of labels (i.e. M) is greater, e.g., on *yeast*, MCMLD is more advantageous.

Summary. Under three simulated typical crowdsourcing scenarios, our proposed models consistently outperform the compared *single-label* alternatives. When labels are strongly correlated, MCMLD performs better.

5.3 Evaluation on real-world datasets

To evaluate the performance of the proposed models on real-world crowdsourcing datasets, we utilize the crowdsourcing dataset *affective* [49]. The original *affective* dataset (SemEval) was constructed as the annotation task proposed in [50]. In their work, experts were asked to provide numeric judgments rating the headline of a piece of news for six emotions: anger, disgust, fear, joy, sadness, and surprise. These expert judgments serve as the ground truth. Snow et al. [49] selected a 100-headline sample from the original dataset and collected 1000 affect labels from the Amazon Mechanical Turk for each of six label types, where each instance was labeled by ten unique workers from six aspects (emotions). For each emotion, the workers were asked to provide scores within the range of [0, 100].

Conversion to multi-class multi-label datasets. In above *affective* dataset, each instance obtained six labels from a worker, which forms a typical multi-label dataset. Different from our simulation designed for classification, the value of each label in this dataset is a real number. This just provides an opportunity to change the dataset into several classification datasets by dividing the real label value into intervals. The analysis of the scores in this dataset shows that the scores provided by experts are rather strict, compared with the crowdsourced ones. Therefore, for the true values (provided by the experts), we let those equaling to zero be the negative class (no such an emotion) and others be the positive class when creating a binary labeling dataset (i.e., $K = 2$). From the positive instances in this binary dataset, we extract out the instances with the original true label values greater than 50 to form a new class ″strong″ emotion. Then, we have the second dataset, where each label has three choices (negative, moderate, and strong), i.e., $K = 3$. We continue to divide the instances into two and three categories by averaging the interval of the class ″moderate″ emotion, obtaining two datasets with $K = 4$ and $K = 5$, respectively. For the scores provided by the crowdsourced workers, we just simply divide the values that are greater than zero into two, three, and four intervals, forming three, four, and five classes, respectively. Finally, we have four multi-class multi-label datasets whose the numbers of labels are the same (i.e., $M = 6$) but the values of each label are different (i.e., $K = 2, 3, 4,$ and 5, respectively).

Results. We repeatedly run the compared algorithms 20 times on these real-world datasets. The experimental results are listed in Table 2. In the table, the best results with statistical significance (i.e., $p{-}value < 0.05$) are in bold. The runner-ups with statistical significance are underlined. First, we still used our PCA-based method to set the value of R in the MCMLD model. Obviously, on all datasets, MCMLD significantly outperforms the other methods, and MCMLI is the runner-up. The differences of the average accuracies between our MCMLD (MCMLI) and the best one among the existing *single-label* alternatives (i.e., DS on *D2*, *D3*, *D5*, and SpectralDS on D4) are 3.45 (1.77), 3.12 (1.93), 4.62 (2.26) and 1.19 (0.16) on the four datasets, respectively. Therefore, the observations on the real-world datasets are consistent with those in our simulation. Furthermore, we find that if we divide the values of labels into more intervals, the quality of the crowdsourced labels will decrease. Because on the one hand there will be fewer and fewer instances with the crowdsourced labels whose values just lies in the interval of true values, on the other hand, each interval of true values will contain fewer instances. That is why on *D5* all methods have low accuracies, and MCMLI cannot significantly outperform the compared alternatives.

6 Reduction to One-Coin Model

Although our proposed MCMLI and MCMLD models provide fine-grained information about the capability of crowdsourced workers, they include a large number of parameters in the multi-label scenario, which may suffer from the sparse of the observed data. In this section, we illustrate that our proposed models can be easily reduced to multi-class one-coin models. Then, we present that the one-coin models exhibit some advantages under specific circumstances.

6.1 One-Coin Model for Multi-Class Multi-Label Inference

In one-coin model, the reliability of workers is modeled only be one parameter on each label, which represents the overall accuracy of a worker on that label. Formally, we parameterize the reliability of worker j as $\boldsymbol{\zeta}_j = [\zeta_j^{(1)}, ..., \zeta_j^{(M)}]$. We denote all sets of reliability of totally J workers by a parameter set $\widetilde{\boldsymbol{\zeta}} = \{\boldsymbol{\zeta}_1, ..., \boldsymbol{\zeta}_J\}$. When worker j labels instance $\mathbf{x}_i$ on its mth label given that its true value is class k, the generative probability of the crowdsourced label is

$$\mathrm{P}\left(l_{ij}^{(m)} | y_i^{(m)} = k\right) = \left(\zeta_j^{(m)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=k\right)} \left(\frac{1-\zeta_j^{(m)}}{K-1}\right)^{\mathbb{I}\left(l_{ij}^{(m)}\neq k\right)}. \tag{26}$$

TABLE 2
Accuracy on the real-world datasets with different label values (in percent)

Dataset	MV	iBCC	DS	SpectralDS	MCMLI	MCMLD($R=4$)
D2 ($K=2$)	69.71 ± 0.29	70.87 ± 0.98	72.83 ± 0.11	71.71 ± 1.05	$\underline{74.60 \pm 0.21}$	$\mathbf{76.28 \pm 0.26}$
D3 ($K=3$)	63.33 ± 0.13	63.83 ± 0.46	64.21 ± 0.12	63.41 ± 0.59	$\underline{66.14 \pm 0.34}$	$\mathbf{67.33 \pm 0.31}$
D4 ($K=4$)	53.85 ± 0.10	53.50 ± 0.36	52.96 ± 0.24	54.06 ± 0.32	$\underline{56.32 \pm 0.11}$	$\mathbf{58.68 \pm 0.21}$
D5 ($K=5$)	43.33 ± 0.12	44.16 ± 0.14	44.33 ± 0.09	43.88 ± 0.22	$\underline{44.49 \pm 0.24}$	$\mathbf{45.52 \pm 0.09}$

As Eq.(26) illustrates, the multi-class one-coin model assumes that if the worker provides a wrong label (i.e., $l_{ij}^{(m)} \neq k$), the probability of providing the other $K-1$ wrong labels is uniformly distributed. Under the binary case (i.e., $K = 2$), Eq.(26) is a standard Bernoulli distribution. The generations of the true labels are the same as the above.

Corresponding to MCMLI, we can derive its variants under the one-coin setting, namely *Multi-Class Multi-Label Independent One-Coin* (MCMLI-OC) model. To derive the log-likelihood of all crowdsourced labels of the entire dataset, we just need to replace term $\prod_{d^{(m)}=1}^{K}\left(\pi_{k^{(m)}d^{(m)}}^{(mj)}\right)^{\mathbb{I}(l_{ij}^{(m)}=d^{(m)})}$ in Eq.(10) with Eq.(26), which gives

$$\ln \mathrm{P}(\mathbf{L}|\Theta,\widetilde{\boldsymbol{\zeta}}) = \sum_{i=1}^{I} \ln \left(\sum_{k^{(1)}=1}^{k^{(1)}=K} \cdots \sum_{k^{(M)}=1}^{k^{(M)}=K} \Big[\prod_{m=1}^{M} \theta_{k^{(m)}}^{(m)} \cdot \prod_{j=1}^{J} \left(\zeta_j^{(m)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=k\right)} \left(\frac{1-\zeta_j^{(m)}}{K-1}\right)^{\mathbb{I}\left(l_{ij}^{(m)}\neq k\right)} \Big] \right). \quad (27)$$

We still use an EM algorithm to solve the model. In the E-step, the joint probability of true labels conditioning on the observed data is

$$\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}|\mathbf{L},\Theta,\widetilde{\boldsymbol{\zeta}},\right) = \prod_{m=1}^{M} \mathrm{P}\left(y_i^{(m)} = k^{(m)}|\mathbf{L},\Theta,\widetilde{\boldsymbol{\zeta}}\right) \propto \prod_{m=1}^{M} \theta_{k^{(m)}}^{(m)} \prod_{j=1}^{J} \left(\zeta_j^{(m)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=k\right)} \left(\frac{1-\zeta_j^{(m)}}{K-1}\right)^{\mathbb{I}\left(l_{ij}^{(m)}\neq k\right)}. \quad (28)$$

Then, we can still use Eq.(13) to obtain $\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]$. In the M-step, the parameter $\theta_k^{(m)}$ that generates true labels is updated using Eq.(14), and the parameter of the reliable of worker j is updated as follows:

$$\hat{\zeta}_j^{(m)} = \frac{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right] \mathbb{I}\left(l_{ij}^{(m)}=k\right)}{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right] \mathbb{I}\left(l_{ij}^{(m)}\neq 0\right)}. \quad (29)$$

Similarly, Corresponding to the MCMLD model, we can derive its variants under the one-coin setting, namely *Multi-Class Multi-Label Dependent One-Coin* (MCMLD-OC) model. The log-likelihood of all crowdsourced labels of the entire dataset can be calculated as follows:

$$\ln \mathrm{P}(\mathbf{L}|\boldsymbol{\Theta},\boldsymbol{\omega},\widetilde{\boldsymbol{\zeta}}) = \sum_{i=1}^{I} \ln \left(\sum_{k^{(1)}=1}^{k^{(1)}=K} \cdots \sum_{k^{(M)}=1}^{k^{(M)}=K} \Big[\sum_{r=1}^{R} \omega_r \cdot \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)} \prod_{j=1}^{J} \left(\zeta_j^{(rm)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=k\right)} \left(\frac{1-\zeta_j^{(rm)}}{K-1}\right)^{\mathbb{I}\left(l_{ij}^{(m)}\neq k\right)} \Big] \right). \quad (30)$$

We still use an EM algorithm to solve the model. In the E-step, the joint probability of true labels conditioning on the observed data is

$$\mathrm{P}\left(y_i^{(\cdot)} \overset{1..M}{=} k^{(\cdot)}|\mathbf{L},\boldsymbol{\Theta},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\omega}\right) = \prod_{m=1}^{M} \mathrm{P}\left(y_i^{(m)} = k^{(m)}|\mathbf{L},\boldsymbol{\Theta},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\omega}\right) \propto \omega_r \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)} \prod_{j=1}^{J} \left(\zeta_j^{(rm)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=k\right)} \left(\frac{1-\zeta_j^{(rm)}}{K-1}\right)^{\mathbb{I}\left(l_{ij}^{(m)}\neq k\right)}. \quad (31)$$

Then, we can still use Eqs.(19) and (20) obtain $\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]$ and $\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]$, respectively. In the M-step, the parameters $\theta_{rm}^{(m)}$ and ω_r that generate true labels are updated using Eqs.(21) and (22) respective, and the parameter ζ_j^{rm} is updated as follows:

$$\hat{\zeta}_j^{(rm)} = \frac{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right] \mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right] \mathbb{I}\left(l_{ij}^{(m)}=k\right)}{\sum_{i=1}^{I} \mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right] \mathbb{I}\left(l_{ij}^{(m)}\neq 0\right)}. \quad (32)$$

6.2 Evaluation

We conduct some experiments to evaluate the performance of the MCMLI-OC and MCMLD-OC models, compared with the original MCMLI and MCMLD models, respectively. In the experiments, the initial value of each reliability parameter $\zeta_j^{(m)}$ in both MCMLI-OC and MCMLD-OC model is drawn from a uniform distribution $U(0.70, 0.90)$.

Figure 6 shows the experimental results on datasets *emotions*, *yeast*, *pentacorrel* and *decasynth* under the scenario that annotation errors are uniformly distributed across the classes and the labels. That is, the experimental settings are almost the same as they are in Scenario 1 in Section 5. On two binary-class datasets (*emotions* and *yeast*), both MCMLI-OC and MCMLD-OC do not perform worse than their counterparts under all $\#labels$ settings. Especially, when $\#labels$ is less than and equal to six, both MCMLI-OC and MCMLD-OC outperform their counterparts, respectively. On the multi-class dataset *pentacorrel*, both MCMLI-OC and

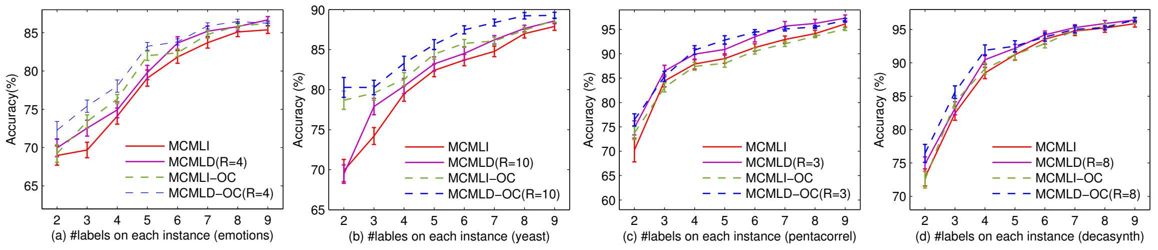


Fig. 6. Comparison results on datasets *emotions*, *yeast*, *pentacorrel* and *decasynth* when annotation errors are uniformly distributed across the classes and the labels.

MCMLD-OC perform almost the same as their counterparts. However, at some points, i.e., $\#labels = 2, 4, 5$, and 6, MCMLD-OC slightly outperforms MCMLD. Moreover, on dataset *decasynth*, when $\#labels = 2, 3, 4$, and 5, MCMLD-OC slightly outperforms MCMLD. On this dataset, MCMLI-OC performs the same as MCMLI does. On the other binary-class datasets *flags* and *scene*, the experimental results are similar to those on datasets *emotions* and *yeast*. In summary, we may draw the conclusion: under the scenario that errors are uniformly distributed across the classes and the labels (1) both MCMLI-OC and MCMLD-OC do not perform worse than the MCMLI and MCMLD models, especially, on the binary-class datasets, MCMLI-OC and MCMLD-OC may perform better than the original models because Bernoulli distribution fits the binary-class data well; (2) when the crowdsourced labels are sparse, the one-coin models show more advantageous because more parameters in MCMLI and MCMLD may require more crowdsourced labels to make accurate estimation.

Another different annotation scenario is that workers exhibit bias when perform the tasks as our Scenario 3 in Section 5 simulates. We compare the MCMLI-OC and MCMLD-OC models with their counterparts under biased annotation scenarios. Figure 7 shows the experimental results on datasets *emotions*, *yeast*, *pentacorrel* and *decasynth*. Obviously, under biased annotation scenario, both MCMLI-OC and MCMLD-OC perform worse than their counterparts on all datasets (including *flags* and *scene*). The reason is clear that one reliability parameter of each worker is not enough to capture worker's bias towards the different classes. In contrast, with confusion matrix MCMLI and MCMLD can easily capture the bias and results in better results.

7 Conclusion

A real-world crowdsourcing annotation application may involve a natural multi-class multi-label scenario. This paper has presented our first effort to exploit both worker reliability and label dependency to perform the ground truth inference under this scenario. We have proposed two different models MCMLI and MCMDL. Both of them can jointly infer the true values of multiple labels of each instance in an EM procedure. We conducted a set of simulations under three typical crowdsourcing scenarios with different numbers of labels and classes on each label as well as a set of real-world dataset evaluations. Both the simulations and real-world evaluations have shown that our MCMLI and MCMLD models significantly perform better in recovering ground truth labels than the existing alternatives of successively running inference algorithms on each label. Furthermore, the MCMLD model can use a mixture of multiple independent multinoulli distributions to capture the dependency of the labels, which makes it outperform the MCMLI model in most cases, especially, when labels are highly correlated. In addition, we also demonstrated that the proposed two models are easily reduced into the one-coin models, which have fewer parameters and perform well under some circumstances.

Acknowledgments

This work was partially supported by the National Key Research and Development Program of China under grant 2016YFB1000901, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT17R32, the National Natural Science Foundation of China under grants 91846104, 91746209, 61603186, the Natural Science Foundation of Jiangsu Province, China, under grant BK20160843, CMC GF funding, and the China Postdoctoral Science Foundation under grant 2017T100370. Jing Zhang was also supported by the 2018 sponsorship of Jiangsu Overseas Visiting Scholar Program for University Prominent Young & Middle-aged Teachers and Presidents.

References

[1] T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, "The big data system, components, tools, and technologies: a survey," *Knowledge and Information Systems*, pp. 1–81, 2019.
[2] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: a survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 543–576, 2016.
[3] E. Toch, B. Lerner, E. Ben-Zion, and I. Ben-Gal, "Analyzing large-scale human mobility data: a survey of machine learning methods and applications," *Knowledge and Information Systems*, vol. 58, no. 3, pp. 501–523, 2019.
[4] G. Xintong, W. Hongzhi, Y. Song, and G. Hong, "Brief survey of crowdsourcing for data mining," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7987–7994, 2014.
[5] A. Kovashka, O. Russakovsky, L. Fei-Fei, K. Grauman *et al.*, "Crowdsourcing in computer vision," *Foundations and Trends® in Computer Graphics and Vision*, vol. 10, no. 3, pp. 177–243, 2016.
[6] M. Lease and E. Yilmaz, "Crowdsourcing for information retrieval," in *ACM SIGIR Forum*, vol. 45, no. 2, 2012, pp. 66–75.
[7] A. Wang, C. D. V. Hoang, and M.-Y. Kan, "Perspectives on crowdsourcing annotations for natural language processing," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 9–31, 2013.
[8] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 491–518, 2012.
[9] C.-J. Ho, A. Slivkins, S. Suri, and J. W. Vaughan, "Incentivizing high quality crowdwork," in *WWW*, 2015, pp. 419–429.

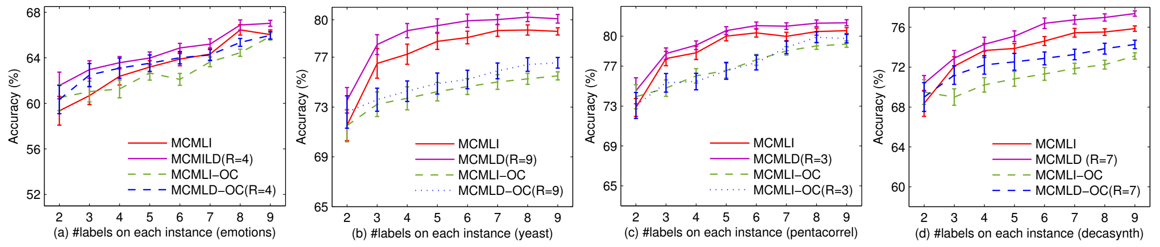


Fig. 7. Comparison results on datasets *emotions*, *yeast*, *pentacorrel* and *decasynth* when annotation exhibits bias.

[10] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *SIGKDD*, 2008, pp. 614–622.

[11] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *Proceedings of the VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[12] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, p. 52, 2015.

[13] S. Burkhardt and S. Kramer, "Multi-label classification using stacked hierarchical dirichlet processes with reduced sampling complexity," *Knowledge and Information Systems*, vol. 59, no. 1, pp. 93–115, 2019.

[14] J. Zhang and X. Wu, "Multi-label inference for crowdsourcing," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2738–2747.

[15] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, "A survey on truth discovery," *ACM Sigkdd Explorations Newsletter*, vol. 17, no. 2, pp. 1–16, 2016.

[16] W. Liu, J. Liu, B. Wei, H. Duan, and W. Hu, "A new truth discovery method for resolving object conflicts over linked data with scale-free property," *Knowledge and Information Systems*, vol. 59, no. 2, pp. 465–495, 2019.

[17] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.

[18] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 2010, pp. 64–67.

[19] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1297–1322, 2010.

[20] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi, "Community-based bayesian aggregation models for crowdsourcing," in *WWW*, 2014, pp. 155–164.

[21] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet em: A provably optimal algorithm for crowdsourcing," in *NIPS*, 2014, pp. 1260–1268.

[22] W. Bi, L. Wang, J. T. Kwok, and Z. Tu, "Learning to predict from crowdsourced data." in *UAI*, 2014, pp. 82–91.

[23] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *WWW*, 2012, pp. 469–478.

[24] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *NIPS*, 2011, pp. 1953–1961.

[25] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, 2009, pp. 2035–2043.

[26] A. Kurve, D. J. Miller, and G. Kesidis, "Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention," *IEEE Trans. Knowledge and Data Engineering*, vol. 27, no. 3, pp. 794–809, 2015.

[27] E. Kamar, A. Kapoor, and E. Horvitz, "Identifying and accounting for task-dependent bias in crowdsourcing," in *Third AAAI HCOMP*, 2015.

[28] P. Welinder, S. Branson, P. Perona, and S. J. Belongie, "The multidimensional wisdom of crowds," in *NIPS*, 2010, pp. 2424–2432.

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

[30] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *SIGMOD*, 2014, pp. 1187–1198.

[31] D. Zhou, S. Basu, Y. Mao, and J. C. Platt, "Learning from the wisdom of crowds by minimax entropy," in *NIPS*, 2012, pp. 2195–2203.

[32] J. Zhang, V. S. Sheng, J. Wu, and X. Wu, "Multi-class ground truth inference in crowdsourcing with clustering," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1080–1085, 2016.

[33] J. Zhang, V. S. Sheng, and T. Li, "Label aggregation for crowdsourcing with bi-layer clustering," in *ACM SIGIR*. ACM, 2017, pp. 921–924.

[34] J. Zhang, V. S. Sheng, T. Li, and X. Wu, "Improving crowdsourced label quality using noise correction," *IEEE trans. Neural Networks and Learning Systems*, 2017.

[35] M. Liu, L. Jiang, J. Liu, X. Wang, J. Zhu, and S. Liu, "Improving learning-from-crowds through expert validation," in *IJCAI*, 2017, pp. 2329–2336.

[36] A. Sheshadri and M. Lease, "Square: A benchmark for research on computing crowd consensus," in *First AAAI HCOMP*, 2013.

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

[38] S. Nowak and S. Rüger, "How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation," in *ICMR*, 2010, pp. 557–566.

[39] Y.-L. Fang, H.-L. Sun, P.-P. Chen, and T. Deng, "Improving the quality of crowdsourced image labeling via label similarity," *Journal of Computer Science and Technology*, vol. 32, no. 5, pp. 877–889, 2017.

[40] Y. Fang, H. Sun, G. Li, R. Zhang, and J. Huai, "Context-aware result inference in crowdsourcing," *Information Sciences*, vol. 460, pp. 346–363, 2018.

[41] T. Han, H. Sun, Y. Song, Y. Fang, and X. Liu, "Incorporating external knowledge into crowd intelligence for more specific knowledge acquisition." in *IJCAI*, vol. 2016, 2016, pp. 1541–1547.

[42] J. Bragg, D. S. Weld *et al.*, "Crowdsourcing multi-label classification for taxonomy creation," in *First AAAI HCOMP*, 2013.

[43] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei, "Scalable multi-label annotation," in *SIGCHI*, 2014, pp. 3099–3102.

[44] N. Q. V. Hung, H. H. Viet, N. T. Tam, M. Weidlich, H. Yin, and X. Zhou, "Computing crowd consensus with partial agreement," *IEEE Trans. Knowledge and Data Engineering*, vol. 30, no. 1, pp. 1–14, 2017.

[45] P. G. Moreno, A. Artés-Rodríguez, Y. W. Teh, and F. Perez-Cruz, "Bayesian nonparametric crowdsourcing," *Journal of Machine Learning Research*, 2015.

[46] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Multi-label active learning from crowds," *arXiv preprint arXiv:1508.00722*, 2015.

[47] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *AISTATS*, 2012, pp. 619–627.

[48] C. Eickhoff, "Cognitive biases in crowdsourcing," in *WSDM*, 2018.

[49] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *EMNLP*, 2008, pp. 254–263.

[50] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007, pp. 70–74.

Jing Zhang is an Associate Professor in the School of Computer Science and Engineering at the Nanjing University of Science and Technology. He received his M.S. degree in Computer Science from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2006, and the Ph.D. degree in Computer Science from the Hefei University of Technology, Hefei, China, in 2015. His research interests include data mining, machine learning, and their applications in business and industry. He has published dozens of articles in prestigious journals, such as TKDE, TNNLS, TCYB, TMM, JMLR, and top tier conferences, such as SIGKDD, AAAI, SIGIR, ICDM, CIKM, etc.

Dr. Zhang is a Senior Member of IEEE. He serves as a PC member for a number of international conferences and a reviewer for more than 20 international journals.

Xindong Wu is a Professor of Computer Science with the School of Computing and Informatics, the University of Louisiana at Lafayette, Lafayette, LA, USA, and a Yangtze River Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. He received his Bachelor's and Master's degrees in Computer Science from the Hefei University of Technology, China, and his Ph.D. degree in Artificial Intelligence from the University of Edinburgh, United Kingdom. His research interests include data mining, knowledge engineering, and Web information exploration.

Dr. Wu is a Fellow of the IEEE and the AAAS. He is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), the Editor-in-Chief of Knowledge and Information Systems (KAIS, by Springer), and an Editor-in-Chief of the Springer Book Series on Advanced Information and Knowledge Processing (AI & KP). He was the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering (TKDE) between 2005 and 2008. He served as a program committee chair/co-chair for ICDM 2003 (the 3rd IEEE International Conference on Data Mining), KDD 2007 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining), CIKM 2010 (the 19th ACM Conference on Information and Knowledge Management, and ICBK 2017 (the 8th IEEE International Conference on Big Knowledge).

APPENDIX A. DERIVATION DETAILS IN M-STEP

M-step finds new values of the parameters that maximize the objective functions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ defined by Eqs. (11) and (17). Because we have obtained the posterior probabilities of the latent variables, in the following derivations, we omit the conditions of the observed label $\mathbf{L}$ and old values of parameters $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$.

A.1 M-Step for MCMLI

The objective is to maximize the quantity as follows:

$$\mathbb{E}_{\mathbf{Y}}\left[\ln \mathrm{P}\left(\mathbf{L},\mathbf{Y}|\mathbf{\Psi}_1\right)\right]=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln \mathrm{P}\left(\mathbf{l}_i,\mathbf{y}_i|\Theta,\mathbf{\Pi}_i\right)\right]=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln\left(\mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,\mathbf{\Pi}_i\right)\underbrace{\mathrm{P}\left(\mathbf{y}_i|\Theta\right)}_{constant}\right)\right]. \tag{33}$$

We omit the constant factor with respect to the parameters $\mathbf{\Psi}_1$ in the partial derivatives. Thus, we only need to maximize the term $\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,\mathbf{\Pi}_i\right)\right]$, which derives

$$\begin{aligned}&\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln\prod_{m=1}^{M}\prod_{k=1}^{K}\left(\theta_k^{(m)}\prod_{j=1}^{J}\prod_{d=1}^{K}\left(\pi_{kd}^{(mj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d\right)}\right)^{\mathbb{I}\left(y_i^{(m)}=k\right)}\right]\\&=\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\left[\ln\theta_k^{(m)}+\sum_{j=1}^{J}\sum_{d=1}^{K}\mathbb{I}\left(l_{ij}^{(m)}=d\right)\ln\pi_{kd}^{(mj)}\right].\end{aligned} \tag{34}$$

(1) Using a Lagrange multiplier to optimize Eq.(34) with respect to $\theta_k^{(m)}$, we construct a function

$$\mathcal{F}_1=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{k=1}^{K}\theta_k^{(m)}-1\right). \tag{35}$$

We let the partial derivative of Eq.(35) with respect to $\theta_k^{(m)}$ be zero. Applying sum-up-to-one condition $\left(\sum_{k=1}^{K}\theta_k^{(m)}=1\right)$, we have

$$\frac{\partial\mathcal{F}_1}{\partial\theta_k^{(m)}}=\frac{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]}{\theta_k^{(m)}}+\lambda=0 \tag{36}$$

$$\Rightarrow\sum_{k=1}^{K}\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]=-\lambda\sum_{k=1}^{K}\theta_k^{(m)}\Rightarrow I=-\lambda. \tag{37}$$

Plugging Eq.(37) into Eq.(36), we have

$$\hat{\theta}_k^{(m)}=\sum\nolimits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\Big/I.$$

(2) Using a Lagrange multiplier to optimize Eq(34) with respect to $\pi_{kd}^{(mj)}$, we construct a function

$$\mathcal{F}_2=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{k=1}^{K}\pi_{kd}^{(mj)}-1\right). \tag{38}$$

We let the partial derivative of Eq.(38) with respect to $\pi_{kd}^{(mj)}$ be zero. Applying sum-up-to-one condition $\left(\sum_{d=1}^{K}\pi_{kd}^{(mj)}=1\right)$, we have

$$\frac{\partial\mathcal{F}_2}{\partial\pi_{kd}^{(mj)}}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}}{\pi_{kd}^{(mj)}}+\lambda=0 \tag{39}$$

$$\begin{aligned}&\Rightarrow\sum_{d=1}^{K}\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}=-\lambda\sum_{d=1}^{K}\pi_{kd}^{(mj)}\\&\Rightarrow\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}\neq 0\right)=-\lambda.\end{aligned} \tag{40}$$

Plugging Eq.(40) into Eq.(39), we have

$$\hat{\pi}_{kd}^{(mj)}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}}{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}\neq 0\right)}.$$

A.2 M-Step for MCMLD

The objective is to maximize the quantity as follows:

$$\begin{aligned}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{L},\mathbf{Y},\mathbf{z}|\mathbf{\Psi}_2\right)\right] &= \sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i,\mathbf{y}_i,z_i|\mathbf{\Theta},\boldsymbol{\omega},\mathbf{\Pi}_i\right)\right] \\ &= \sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\Bigg[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\underbrace{\mathrm{P}\left(\mathbf{y}_i|z_i,\mathbf{\Theta}\right)\mathrm{P}\left(z_i|\boldsymbol{\omega}\right)}_{constants}\Bigg]. \end{aligned} \tag{41}$$

We omit the constant factors with respect to the parameters $\mathbf{\Psi}_2$ in the partial derivatives. Thus, we only need to maximize the term $\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]$, which derives

$$\begin{aligned}&\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left\{\ln\prod_{r=1}^{R}\left[\omega_r\prod_{m=1}^{M}\prod_{k=1}^{K}\left(\theta_{rk}^{(m)}\cdot\prod_{j=1}^{J}\prod_{d=1}^{K}\left(\pi_{kd}^{(rmj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d\right)}\right)^{\mathbb{I}\left(y_i^{(m)}=k\right)}\right]^{\mathbb{I}(z_i=r)}\right\} \\ &= \sum_{i=1}^{I}\sum_{r=1}^{R}\mathbb{E}\left[\mathbb{I}(z_i=r)\right]\left\{\ln\omega_r+\sum_{m=1}^{M}\sum_{k=1}^{K}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\left[\ln\theta_{rk}^{(m)}+\sum_{j=1}^{J}\sum_{d=1}^{K}\mathbb{I}\left(l_{ij}^{(m)}=d\right)\ln\pi_{kd}^{(rmj)}\right]\right\}. \end{aligned} \tag{42}$$

(1) Using a Lagrange multiplier to optimize Eq.(42) with respect to $\theta_{rk}^{(m)}$, we construct a function

$$\mathcal{F}_3=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{k=1}^{K}\theta_{rk}^{(m)}-1\right). \tag{43}$$

We let the partial derivative of Eq.(43) with respect to $\theta_{rk}^{(m)}$ be zero. Applying sum-up-to-one condition $\left(\sum_{k=1}^{K}\theta_{rk}^{(m)}=1\right)$, we have

$$\frac{\partial\mathcal{F}_3}{\partial\theta_{rk}^{(m)}}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}}{\theta_{rk}^{(m)}}+\lambda=0 \tag{44}$$

$$\begin{aligned}&\Rightarrow\sum_{k=1}^{K}\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}=-\lambda\sum_{k=1}^{K}\theta_{rk}^{(m)} \\ &\Rightarrow\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]=-\lambda. \end{aligned} \tag{45}$$

Plugging Eq.(45) into Eq.(44), we have

$$\hat{\theta}_{rk}^{(m)}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}}{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]}.$$

(2) Using a Lagrange multiplier to optimize Eq.(42) with respect to ω_r, we construct a function

$$\mathcal{F}_4=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{r=1}^{R}\omega_r-1\right). \tag{46}$$

We let the partial derivative of Eq.(46) with respect to ω_r be zero. Applying sum-up-to-one condition $\left(\sum_{r=1}^{R}\omega_i=1\right)$, we have

$$\frac{\partial\mathcal{F}_4}{\partial\omega_r}=\frac{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]}{\omega_i}+\lambda=0, \tag{47}$$

$$\Rightarrow\sum_{r=1}^{R}\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]=-\lambda\sum_{r=1}^{R}\omega_r\Rightarrow I=-\lambda. \tag{48}$$

Plugging Eq.(48) into Eq.(47), we have

$$\hat{\omega}_r=\sum\nolimits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]/I.$$

The updated function of parameter $\pi_{kd}^{(rmj)}$ and its derivation are similar with those of the MCMLI model but only consider the instances in cluster r, which is Eq.(23) in the main body.