Multi-Label Inference for Crowdsourcing

Jing Zhang
School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, Jiangsu, China
jzhang@njust.edu.cn

Xindong Wu
School of Computing and Informatics
University of Louisiana at Lafayette
Lafayette, Louisiana, U.S.
xwu@louisiana.edu

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, one task is 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 a novel probabilistic method, which includes a *multi-class multi-label dependency* (MCMLD) model, to address this problem. The proposed method assumes that the label-correlation exists in both unknown true labels and noisy crowdsourced labels. Thus, it introduces a mixture of multiple independently multinoulli distributions to capture the correlation among the labels. Finally, the unknown true values of the multiple labels of each task, together with a set of confusion matrices modeling the reliability of the workers, can be jointly inferred through an EM algorithm. Experiments with three simulated typical crowdsourcing scenarios and a real-world dataset consistently show that our proposed MCMLD method significantly outperforms several competitive alternatives. Furthermore, if the labels are strongly correlated, the advantage of MCMLD will be more remarkable.

CCS CONCEPTS

• **Computing methodologies** → **Maximum likelihood modeling**; **Mixture models**; **Latent variable models**; • **Information systems** → **Crowdsourcing**;

KEYWORDS

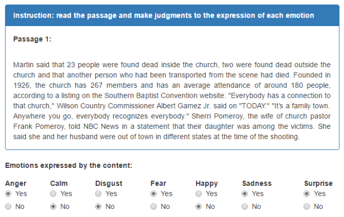
Crowdsourcing, label aggregation, maximum likelihood estimation, mixture models, probabilistic graphical models

This research has been supported by the National Natural Science Foundation of China under grant 61603186, the Natural Science Foundation of Jiangsu Province, China, under grant BK20160843, the China Postdoctoral Science Foundation under grants 2017T100370 and 2016M590457, and the Science Foundation (for Youth) of the Science and Technology Commission of the Central Military Commission (CMC), China, and the US National Science Foundation (NSF) under grant IIS-1613950.

1 INTRODUCTION

Current intelligent systems usually require plenty of human-labeled datasets to build internal models. Human-annotation tasks are often too tedious and time-consuming for individuals to complete. The emergence of crowdsourcing systems provides a convenient and low-cost solution to obtain annotations from the Internet. Crowdsourced annotation acquisition has been used in some important fields (such as data mining [33], machine learning [35], information retrieval [15], and natural language processing [30]) in recent years. However, a significant challenge of using crowdsourcing to acquire labels is that their quality cannot be guaranteed due to the non-expert nature of crowdsourcing. To improve the quality of labels, we usually let each instance be labeled by multiple different heterogeneous workers, which is so-called *repeated labeling*. After sufficient crowdsourced labels are collected, an inference algorithm will be introduced to estimate the true labels of each instance from these noisy labels. This process is also known as *label aggregation* since the estimated true labels are the integration of multiple noisy labels. In addition to the integrated labels, this process may provide more information, such as the estimation of the workers' reliability. 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 *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 [23] and incentive optimization [9].

Truth inference for crowdsourcing started from the study of majority voting [25]—a simple but efficient method. Subsequently, a large number of studies addressed this issue using different techniques. Zheng et al. [37] 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, including *binary-class* and *multi-class* cases. 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 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 [8]. A more general situation is that each label has more than two options (values) as the second application (b) illustrates. In this example, a description of an image contains three



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

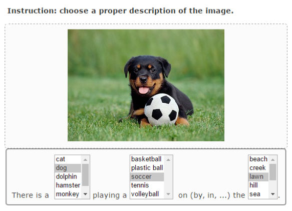
Passage 1:

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

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

placeholders (labels) representing an animal (*dog*), an object (*soccer*), and a place (*lawn*), respectively. All of them have multiple options. We refer this application as *multi-class multi-label* annotation in this paper. The traditional multi-class learning usually transforms a 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.

A critical difference of the multi-label inference in crowdsourcing lies in the exploitation of the correlation among the 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 improve the quality of integrated labels. This paper proposes a novel method to solve the multi-class multi-label inference problem in crowdsourcing. It 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. During the inference, the correlation of the labels is captured and exploited using a mixture of independent multinoulli distributions. Experimental results confirm that our new method not only outperforms other *single-label* alternatives which successively perform inference on each label, but also outperforms our proposed a baseline method that also jointly infers multiple labels.[1]

2 RELATED WORK

Inferring true values of labels from multiple noisy labels is a fundamental research topic in crowdsourcing. Many previous studies modeled the complexity of crowdsourced labeling systems from different aspects. The earliest Dawid & Skene's (DS) algorithm [4] used a confusion matrix to model the reliability of uncertain sources. Some recent approaches are directly derived from the DS model by simplifying the parameters [24], adding constraints to workers [29] or optimizing initial settings [36]. Other approaches model the reliability of workers using fewer parameters [1, 5, 12, 32] but add additional parameters or variables to characterize workers' intention [1, 14] and bias [11, 31]. All above approaches are based on the probabilistic models and can be solved by an EM algorithm, gradient descending, or Gibbs sampling. Some of them [1, 4, 5, 11, 14, 36] can be used in multi-class scenarios, while the others [12, 24, 29, 31, 32] are only suitable for binary cases, but none of them were proposed for multi-label annotation. In addition to the approaches based on the probabilistic models, some studies introduced other techniques to improve the accuracy of inference, including optimization [16, 17, 38], clustering [34], and the injection of expert validation [19].

Facing the diversity of the inference algorithms, we carefully choose among them according to some previous comprehensive empirical studies. Sheshadri et al. [26] developed a benchmark tool SQUARE to evaluate six inference algorithms on ten real-world crowdsourcing datasets. Muhammadi et al. [21] studied the performance of those algorithms in a unified statistical framework. A most recent study [37] evaluated fourteen inference algorithms on many real-world datasets. All these studies have shown some consistent conclusion that the approaches derived from the Dawid & Skene's model [4] have good probabilistic interpretability and exhibit strong robustness in most cases. Thus, our study also adopts the mainstream probabilistic models and uses a set of confusion matrices to model the reliability of crowdsourced workers.

Several studies have touched the topic of multi-label annotation in crowdsourcing. Nowak et al. [22] studied the reliability of workers when they performed multi-label image annotation tasks, compared with expert annotation. As multi-label tasks consume more budgets under the repeated labeling scenario, Deng et al. [6] and Bragg et al. [3] designed some optimal annotation acquisition workflows that leverage the correlation among different labels. Hung et al. [10] proposed a Bayesian nonparametric model to aggregate the labels, which takes advantage of Clustering-based Bayesian Combination of Multi-label Classifiers (cBCMC) [20] to exploit the correlation among labels. However, their model is not entirely unsupervised, requiring a particular portion of ground truths. Li et al. [18] also addressed the inference problem for multi-label annotation using probabilistic models under an active learning paradigm but ignoring the correlation among labels. In this paper, our proposed unsupervised method takes 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

[1]The source code of our proposed method MCMLD is open source and available at https://github.com/wisdomofcrowds/CekaPlus.

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. 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 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.

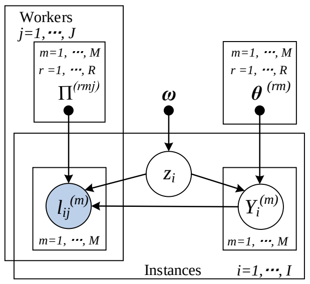


Figure 2: Probabilistic graphical model representation of the `MCMLD` model.

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

4 THE PROPOSED METHOD

In this section, we first present a novel probabilistic model for multi-label inference. Then, we solve the model with an EM algorithm. Finally, we present our inference algorithm.

4.1 Label-Dependent Model

We propose a novel probabilistic generative model, namely *multi-class multi-label dependent* (MCMLD) model, illustrated in Figure 2. 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 ($1 \leq r \leq R$) 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 $\sum_{r=1}^{R} \omega_r = 1$.

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 $\mathrm{P}(\mathbf{y}_i|\Theta) = \prod_{m=1}^{M} \prod_{k=1}^{K} \left(\theta_k^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}$, where $\Theta = [\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(M)}]$ is a set of parameters for all M multinoulli distributions. 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

$$\mathrm{P}(\mathbf{y}_i|\boldsymbol{\Theta}, \boldsymbol{\omega}) = \sum_{r=1}^{R} \omega_r \prod_{m=1}^{M} \prod_{k=1}^{K} \left(\theta_{rk}^{(m)}\right)^{\mathbb{I}(y_i^{(m)}=k)}, \tag{2}$$

where $\boldsymbol{\omega} = [\omega_1, ..., \omega_R]$ are also called the mixing coefficients of multiple multinoulli distributions with the parameter set $\boldsymbol{\Theta} = [\Theta_1, ..., \Theta_R]$.

Generation of crowdsourced labels. In multi-class classification, a 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 to upper-layer applications. 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 $\pi_{kd}^{(mj)} (1 \leq k, d \leq K)$ 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:

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

In this equation, because MCMLD is a mixture model, we sum up all the probability of latent variable z_i, so that z_i disappears. According to the *conditional independency* principle in probabilistic graphical models, adding a latent variable z_i among the unobserved variables $\mathbf{y}_i$ will make them conditional mutually independent, which derives

$$\mathrm{P}\left(y_i^{(1)} = k^{(1)}, ..., y_i^{(M)} = k^{(M)}|\boldsymbol{\Theta}, \boldsymbol{\omega}\right) = \sum_{r=1}^{R} \omega_r \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)}, \tag{4}$$

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

Here, because we need to represent the prior of $\mathbf{y}_i$ under the conditions that each of its elements 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

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

4.2 Inference with EM

To maximize the likelihood of the observed labels $\mathbf{L}$ defined by Eq.(6), we use an expectation-maximization (EM) algorithm. 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:

$$\mathcal{Q}\left(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{old}\right) = \mathbb{E}_{\mathbf{Y},\mathbf{z}|\mathbf{L},\boldsymbol{\Psi}^{old}} \left[\ln \mathrm{P}\left(\mathbf{L}, \mathbf{Y}, \mathbf{z}|\boldsymbol{\Psi}\right)\right], \tag{7}$$

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

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 [2] as follows:

$$\begin{aligned}&\mathrm{P}\left(y_i^{(1)} = k^{(1)}, ..., y_i^{(M)} = k^{(M)}, z_i = r|\mathbf{L}, \boldsymbol{\Psi}\right) \\ &\propto \mathrm{P}\left(L_i|y_i^{(1)} = k^{(1)}, ..., y_i^{(M)} = k^{(M)}, z_i = r, \boldsymbol{\Psi}\right) \\ &\quad \cdot \mathrm{P}\left(y_i^{(1)} = k^{(1)}, ..., y_i^{(M)} = k^{(M)}, z_i = r|\boldsymbol{\Theta}, \boldsymbol{\omega}\right) \\ &= \omega_r \prod_{m=1}^{M} \theta_{rk^{(m)}}^{(m)} \prod_{j=1}^{J} \prod_{d^{(m)}=1}^{K} \left(\pi_{k^{(m)}d^{(m)}}^{(mj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d^{(m)}\right)}.\end{aligned} \tag{8}$$

Eq.(8) provides the expected values of two indicator functions (calculated as two marginal probabilities) that will be used in the M-step as follows:

$$\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)} = k\right)\right] = \sum_{y_i^{(m')}(\forall m' \in \{1,...,M\} \setminus m), z_i} \mathrm{P}\left(y_i^{(m)} = k, \cdots |\mathbf{L}, \boldsymbol{\Psi}\right), \tag{9}$$

$$\mathbb{E}\left[\mathbb{I}\left(z_i = r\right)\right] = \sum_{y_i^{(m)}(\forall m \in \{1,...,M\})} \mathrm{P}\left(z_i = r, \cdots |\mathbf{L}, \boldsymbol{\Psi}\right). \tag{10}$$

M-step. We determine the revised parameter estimate $\boldsymbol{\Psi}$ by maximizing the objective function Eq.(7), i.e., $\boldsymbol{\Psi}^{new} = \mathrm{argmax}_{\boldsymbol{\Psi}}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{old})$. Because we have obtained the posterior probabilities of the latent variables $\mathbf{Y}$ and $\mathbf{z}$, we omit the condition of $\mathbf{L}$ and $\boldsymbol{\Psi}^{old}$ and eliminate the constant factors, i.e.,

$$\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{L}, \mathbf{Y}, \mathbf{z}|\boldsymbol{\Psi}\right)\right] = \sum_{i=1}^{I} \mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i, \mathbf{y}_i, z_i|\boldsymbol{\Theta}, \boldsymbol{\omega}, \boldsymbol{\Pi}_i\right)\right]$$

$$= \sum_{i=1}^{I} \mathbb{E}_{\mathbf{Y},\mathbf{z}}\Big[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i, z_i, \boldsymbol{\Pi}_i\right) \underbrace{\mathrm{P}\left(\mathbf{y}_i|z_i, \boldsymbol{\Theta}\right) \mathrm{P}\left(z_i|\boldsymbol{\omega}\right)}_{\text{constants w.r.t. } \boldsymbol{\Psi} \text{ in derivatives}}\Big].$$

Thus, we only need to maximize $\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln \mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]$, which derives

$$\begin{aligned}&\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\Bigg\{\ln\prod_{r=1}^{R}\Bigg[\omega_r\prod_{m=1}^{M}\prod_{k=1}^{K}\Bigg(\theta_{rk}^{(m)}\\&\qquad\cdot\prod_{j=1}^{J}\prod_{d=1}^{K}\left(\pi_{kd}^{(mj)}\right)^{\mathbb{I}\left(l_{ij}^{(m)}=d\right)}\Bigg)^{\mathbb{I}\left(y_i^{(m)}=k\right)}\Bigg]^{\mathbb{I}(z_i=r)}\Bigg\}\\&=\sum_{i=1}^{I}\sum_{r=1}^{R}\mathbb{E}\left[\mathbb{I}(z_i=r)\right]\Bigg\{\ln\omega_r+\sum_{m=1}^{M}\sum_{k=1}^{K}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\\&\qquad\cdot\left[\ln\theta_{rk}^{(m)}+\sum_{j=1}^{J}\sum_{d=1}^{K}\mathbb{I}\left(l_{ij}^{(m)}=d\right)\ln\pi_{kd}^{(mj)}\right]\Bigg\}.\end{aligned}\tag{11}$$

(1) Using a Lagrange multiplier to optimize Eq.(11) with respect to $\theta_{rk}^{(m)}$, we construct a function

$$\mathcal{F}_1=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln\mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{k=1}^{K}\theta_{rk}^{(m)}-1\right).\tag{12}$$

We let the partial derivative of Eq.(12) 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

$$\frac{\partial\mathcal{F}_1}{\partial\theta_{rk}^{(m)}}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}}{\theta_{rk}^{(m)}}+\lambda=0\tag{13}$$

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

$$\Rightarrow\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]=-\lambda.\tag{14}$$

Plugging Eq.(14) into Eq.(13), we have

$$\hat{\theta}_{rk}^{(m)}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\right\}}{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]}.\tag{15}$$

(2) Using a Lagrange multiplier to optimize Eq.(11) with respect to $\pi_{kd}^{(mj)}$, we construct a function

$$\mathcal{F}_2=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln\mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{k=1}^{K}\pi_{kd}^{(mj)}-1\right).\tag{16}$$

We let the partial derivative of Eq.(16) 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

$$\frac{\partial\mathcal{F}_2}{\partial\pi_{kd}^{(mj)}}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}}{\pi_{kd}^{(mj)}}+\lambda=0\tag{17}$$

$$\Rightarrow\sum_{d=1}^{K}\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}=-\lambda\sum_{d=1}^{K}\pi_{kd}^{(mj)}$$

$$\Rightarrow\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]=-\lambda.\tag{18}$$

Plugging Eq.(18) into Eq.(17), we have

$$\hat{\pi}_{kd}^{(mj)}=\frac{\sum_{i=1}^{I}\left\{\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]\mathbb{I}\left(l_{ij}^{(m)}=d\right)\right\}}{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(y_i^{(m)}=k\right)\right]}.\tag{19}$$

(3) Using a Lagrange multiplier to optimize Eq.(11) with respect to ω_r, we construct a function

$$\mathcal{F}_3=\sum_{i=1}^{I}\mathbb{E}_{\mathbf{Y},\mathbf{z}}\left[\ln\mathrm{P}\left(\mathbf{l}_i|\mathbf{y}_i,z_i,\mathbf{\Pi}_i\right)\right]+\lambda\left(\sum_{r=1}^{R}\omega_r-1\right).\tag{20}$$

We let the partial derivative of Eq.(20) with respect to ω_r be zero. Applying sum-up-to-one condition $\left(\sum_{r=1}^{R}\omega_i=1\right)$, we have

$$\frac{\partial\mathcal{F}_3}{\partial\omega_r}=\frac{\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]}{\omega_i}+\lambda=0,\tag{21}$$

$$\Rightarrow\sum_{r=1}^{R}\sum_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]=-\lambda\sum_{r=1}^{R}\omega_r\Rightarrow I=-\lambda.\tag{22}$$

Plugging Eq.(22) into Eq.(21), we have

$$\hat{\omega}_r=\sum\nolimits_{i=1}^{I}\mathbb{E}\left[\mathbb{I}\left(z_i=r\right)\right]/I.\tag{23}$$

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,

$$\begin{aligned}\hat{y}_i^{(1)},...,\hat{y}_i^{(M)}=\underset{k^{(1)},...,k^{(M)}}{\operatorname{argmax}}&\Bigg\{\sum_{r=1}^{R}\mathrm{P}\Big(y_i^{(1)}=k^{(1)},\\&...,y_i^{(M)}=k^{(M)},z_i=r\Big|\mathbf{L},\mathbf{\Psi}\Big)\Bigg\}.\end{aligned}\tag{24}$$

The *hard* value of z_i also can be determined like Eq.(24).

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 $R=\mathrm{argmin}_{1\le n\le N}\left|\rho-\sum_{b=1}^{n}\eta_b\right|$, where $\rho\in(0,1]$.
4: repeat
5:     E-step: calculate expectation by Eqs.(9) and (10).
6:     M-step: update parameters by Eqs.(15), (19) 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.
```

4.3 Algorithm

To apply the proposed mixture model into a practical usage, we propose a novel algorithm, where a critical issue of setting the value of parameter R under an unsupervised scenario is addressed by exploring 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 MCMLD method on both simulated and real-world datasets.

5.1 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 multi-class multi-label annotation inference algorithm for crowdsourcing. Therefore, we choose four alternatives and another baseline proposed by us to conduct comparisons.

- **MV** is usually served as a baseline for label inference.
- **DS** [4] uses the confusion matrix to model workers and makes inference using maximum likelihood estimation. Previous large-scale empirical studies [21, 26, 37] have shown its robustness and efficiency.
- **SpectralDS** [36] is a variant of DS, where the initial values of the confusion matrices are carefully set.
- **iBCC** [13] is a non-EM based inference method, where a Bayesian generative model of the crowdsourced annotation is solved by Gibbs sampling.
- **MCMLI**[2] is a simple baseline model for multi-label inference, which can jointly infer the M true labels. If we remove node z_i (and its links) from the probabilistic graphic model of MCMLD (in Figure 2), we will obtain MCMLI. It addresses issues 1 and 3 in Section 3.2.

The four alternatives are chosen because they are not only efficient and robust but also suitable for both binary and multi-class cases [37]. 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 revealing the differences among compared algorithms under various predefined conditions, especially, when there are no sufficient real-world datasets to cover different situations.

[2]MCMLI is also proposed by us and has never been published elsewhere. Since it does not directly model the correlation, we only use it as a baseline in this paper. In fact, MCMLI has made an obvious progress as our experiments show. Its source code and document are also available at https://github.com/wisdomofcrowds/CekaPlus.

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.**: number of instances; **#lb.**: number of labels; **#cls.**: number of classes on each label; **#card.** and **#dens.**: *cardinality* and *density* [8].

Datasets for simulation. We selected four *real-world* multi-label classification datasets from the multi-label learning library MULAN[3], 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 multi-label classification. To make a comprehensive investigation, 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, where 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. 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 [26, 37] 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 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, R is set using the method in Section 4.3. Specifically, we use `sklearn.decomposition.pca` in `scikit-learn` 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, 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. 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. We first focus on the performance of jointly inferring the true

[3]http://mulan.sourceforge.net/datasets-mlc.html

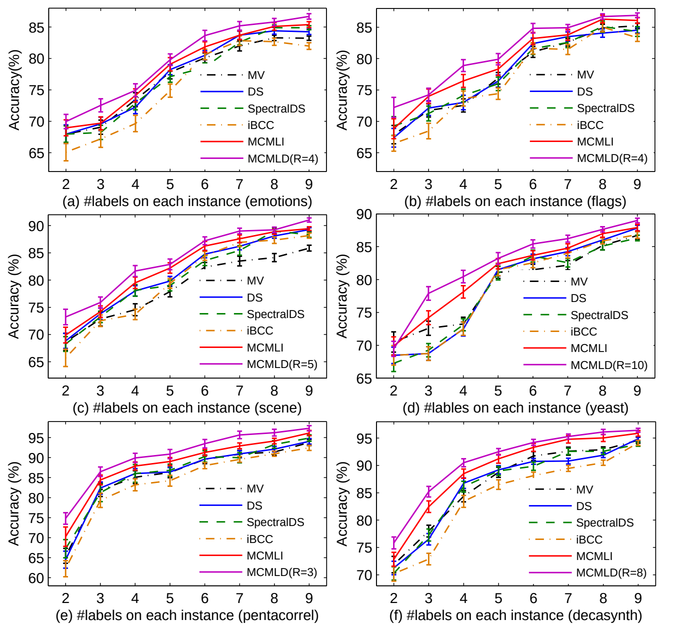


Figure 3: Comparison results under Scenario 1.

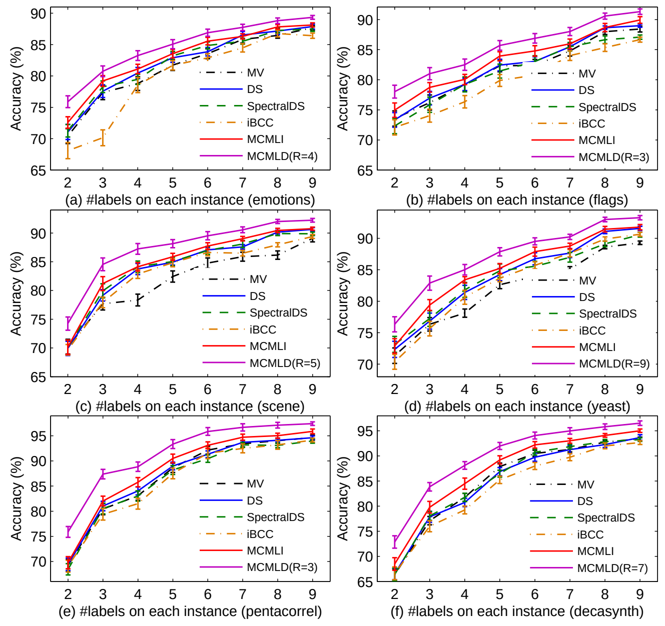


Figure 4: Comparison results under Scenario 2.

labels. Both MCMLD and MCMLI perform joint inference. Clearly, even the simple model MCMLI outperforms DS. The average excess is around 2% (absolute value), and the maximum is over 5% (on *yeast* when #*labels* on each instance is 4). Especially, on two multi-class multi-label datasets, both MCMLD and MCMLI exhibit obvious advantages. Furthermore, MCMLD outperforms MCMLI in most cases with the average excess around 2%. Although errors are uniformly distributed, the crowdsourced labels are generated according to their true values, so that their values may inevitably inherit a certain degree of correlation from their true values. This results in two consequences: (1) jointly inferring multiple labels can achieve a better local optimum, comparing to running a *single-label* inference multiple times on a dataset, which makes both MCMLD and MCMLI outperform the others. (2) MCMLD exploits the correlation and outperforms MCMLI.

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 the four real-world datasets and *decasynth*, 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*, its strong-correlated labels are the first, third and fifth ones. 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 a random correct probability within the range of $[0.75, 0.90]$. The strong-correlated labels can help workers verify their answers, which improves their accuracies. Thus, when label l_a is set to its true value, 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. When workers are aware of label-correlation, the correlation among crowdsourced labels is 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 comparing Figures 4.(b), 4.(c), 4.(d), and 4.(f) with Figures 3.(b), 3.(c), 3.(d), and 3.(f), respectively, we find that under Scenario 1, MCMLD does not significantly outperforms MCMLI at some settings of the numbers of labels on each instance (represented as #*labels* in the following), for examples, on *flags* when $\#labels = 3$ or 8, on *scene* when $\#labels = 5$ or 8, on *yeast* when $\#labels = 2, 5$, or 8, and on *decasynth* when $\#labels = 7$ or 9. Comparatively, under Scenario 2, MCMLD significantly outperforms MCMLI at all points above. The superiority of MCMLD is obviously enlarged on all dataset. Especially, on the strong-correlated dataset *pentacorrel*, the performance of MCMLD exceeds that of MCMLI by more than 4% on average. Thus, MCMLD can benefit a lot from the growth of the crowdsourced label-correlation. In contrast, the advantage of MCMLI is not enlarged under this scenario.

Scenario 3: Biased annotation. The crowdsourced annotation often suffers from the biases of workers. This phenomenon has been extensively studied in recent years [7, 11, 34]. The causes for the biased annotation 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), we let the workers' accuracies on the negative class be within the range of $[0.80, 0.90]$ and those on the positive class be within the range of $[0.40, 0.50]$. 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 $[0.80, 0.90]$ and the others be within the range of $[0.40, 0.50]$. Again, we simulate that there are at least two and at most nine biased workers.

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 accuracies of inference on all datasets consistently decrease. The decrease of the accuracies is more than 15% (absolute value), which suggests that the biased annotation has a negative impact on the performance of inference algorithms. Under this scenario, we find that on the most datasets, except *scene* whose underlying true labels are not correlated, the performance of MCMLI obviously outperforms that of the compared *single-label* inference algorithms. Furthermore, the performance of MCMLD obviously outperforms that of MCMLI in most cases. It seems that 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. Both MCMLD and MCMLI are consistently more advantageous in multi-class annotation. In addition, when the number of labels (i.e. M) is greater, e.g., on *yeast*, MCMLD is more advantageous.

Other observations. SpectralDS does not exhibit obvious advantages over DS, indicating 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 in our simulations.

5.3 Evaluation on real-world datasets

To evaluate the performance of the proposed method in a real-world usage, we utilize the crowdsourcing dataset *affective* [27]. The original *affective* dataset (SemEval) was constructed as the annotation task[28], asking experts 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. 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).

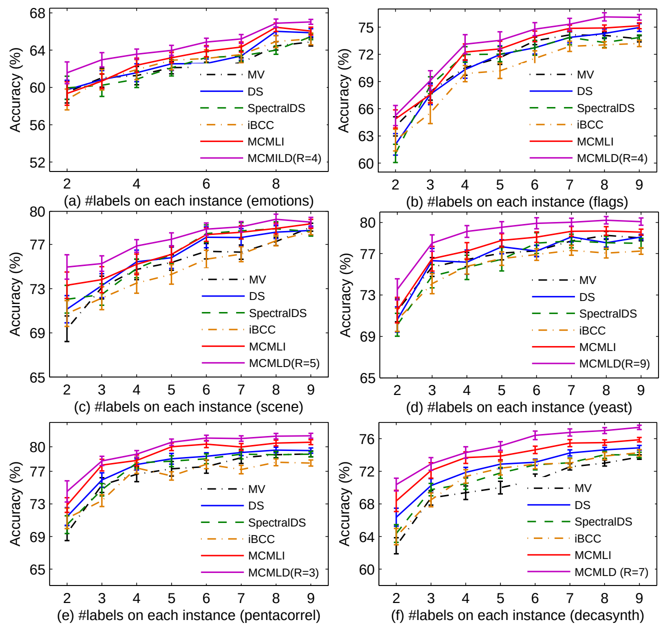


Figure 5: Comparison results under Scenario 3.

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. The value of each label in this dataset is a real number. This 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. Therefore, for the true values (provided by the experts), we let those equaling to zero be the negative class (no emotion) and others be the positive class when creating a binary dataset (i.e., $K = 2$). From its positive class, we extract out the instances with the original true label values greater than 50 to form a new class "strong" emotion. Then, we have a 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 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. The best results with statistical significance (i.e., $p-value < 0.05$) are in bold and the runner-ups are underlined. Obviously, on all datasets, MCMLD significantly outperforms the other methods, and MCMLI is the runner-up. The differences of the average accuracies

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	$\underline{44.49 \pm 0.24}$	$\mathbf{45.52 \pm 0.09}$

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 instances with the crowdsourced labels whose values 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.

6 CONCLUSION

A real-world crowdsourcing annotation application may involve a natural multi-class multi-label scenario. This paper proposed a novel method MCMDL, which can explore and exploit the label-correlation to jointly infer the true values of multiple labels of each instance in an EM procedure as well as models the reliability of crowdsourced workers. 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 MCMLD method significantly outperforms the existing *single-label* alternatives and our baseline method.

REFERENCES

[1] Wei Bi, Liwei Wang, James T Kwok, and Zhuowen Tu. 2014. Learning to Predict from Crowdsourced Data.. In *UAI*. 82–91.

[2] Christopher M Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, Chapter 9, 440–441.

[3] Jonathan Bragg, Daniel S Weld, et al. 2013. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI HCOMP*.

[4] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[5] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*. 469–478.

[6] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. 2014. Scalable multi-label annotation. In *SIGCHI*. 3099–3102.

[7] Carsten Eickhoff. 2018. Cognitive Biases in Crowdsourcing. In *WSDM*.

[8] Eva Gibaja and Sebastián Ventura. 2015. A tutorial on multilabel learning. *Comput. Surveys* 47, 3 (2015), 52.

[9] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing high quality crowdwork. In *WWW*. 419–429.

[10] Nguyen Quoc Viet Hung, Huynh Huu Viet, Nguyen Thanh Tam, Matthias Weidlich, Hongzhi Yin, and Xiaofang Zhou. 2018. Computing Crowd Consensus with Partial Agreement. *IEEE Trans. Knowledge and Data Engineering* 30, 1 (2018), 1–14.

[11] Ece Kamar, Ashish Kapoor, and Eric Horvitz. 2015. Identifying and accounting for task-dependent bias in crowdsourcing. In *Third AAAI HCOMP*.

[12] David R Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *NIPS*. 1953–1961.

[13] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *AISTATS*. 619–627.

[14] Aditya Kurve, David J Miller, and George Kesidis. 2015. Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. *IEEE Trans. Knowledge and Data Engineering* 27, 3 (2015), 794–809.

[15] Matthew Lease and Emine Yilmaz. 2012. Crowdsourcing for information retrieval. In *ACM SIGIR Forum*, Vol. 45. 66–75.

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

[17] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD*. 1187–1198.

[18] Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. 2015. Multi-Label Active Learning from Crowds. *arXiv preprint arXiv:1508.00722* (2015).

[19] Mengchen Liu, Liu Jiang, Junlin Liu, Xiting Wang, Jun Zhu, and Shixia Liu. 2017. Improving Learning-from-Crowds through Expert Validation. In *IJCAI*. 2329–2336.

[20] Pablo G Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Perez-Cruz. 2015. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research* (2015).

[21] Jafar Muhammadi, Hamid R Rabiee, and Abbas Hosseini. 2015. A unified statistical framework for crowd labeling. *Knowledge and Information Systems* 45, 2 (2015), 271–294.

[22] Stefanie Nowak and Stefan Rüger. 2010. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *ICMR*. 557–566.

[23] Vikas C Raykar and Shipeng Yu. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research* 13, Feb (2012), 491–518.

[24] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research* 11, Apr (2010), 1297–1322.

[25] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *SIGKDD*. 614–622.

[26] Aashish Sheshadri and Matthew Lease. 2013. Square: A benchmark for research on computing crowd consensus. In *First AAAI HCOMP*.

[27] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*. 254–263.

[28] Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 70–74.

[29] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *WWW*. 155–164.

[30] Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation* 47, 1 (2013), 9–31.

[31] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *NIPS*. 2424–2432.

[32] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*. 2035–2043.

[33] Guo Xintong, Wang Hongzhi, Yangqiu Song, and Gao Hong. 2014. Brief survey of crowdsourcing for data mining. *Expert Systems with Applications* 41, 17 (2014), 7987–7994.

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

[35] Jing Zhang, Xindong Wu, and Victor S Sheng. 2016. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review* 46, 4 (2016), 543–576.

[36] Yuchen Zhang, Xi Chen, Denny Zhou, and Michael I Jordan. 2014. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *NIPS*. 1260–1268.

[37] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.

[38] Denny Zhou, Sumit Basu, Yi Mao, and John C Platt. 2012. Learning from the wisdom of crowds by minimax entropy. In *NIPS*. 2195–2203.