Multi-Label Answer Aggregation based on Joint Matrix Factorization

Jinzheng Tu[1], Guoxian Yu[1,*], Carlotta Domeniconi[2], Jun Wang[1], Guoqiang Xiao[1], Maozu Guo[3]
[1]College of Computer and Information Sciences, Southwest University, China
[2]Department of Computer Science, George Mason University, USA
[3]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, China
Email: [1]{tujinzheng, gxyu, kingjun, gqxiao}@swu.edu.cn, [2]carlotta@cs.gmu.edu, [3]maozuguo@bucea.edu.cn

***Abstract*—Crowdsourcing is a useful and economic approach to data annotation. To obtain annotation of high quality, various aggregation approaches have been developed, which take into account different factors that impact the quality of aggregated answers. However, existing methods generally focus on single-label (multi-class and binary) tasks, and they ignore the inter-correlation between labels, and thus may have compromised quality. In this paper, we introduce a Multi-Label answer aggregation approach based on Joint Matrix Factorization (ML-JMF). ML-JMF selectively and jointly factorizes the sample-label association matrices collected from different annotators into products of individual and shared low-rank matrices. As such, it takes advantage of the robustness of low-rank matrix approximation to noise, and reduces the impact of unreliable annotators by assigning small (zero) weights to their annotation matrices. In addition, it takes advantage of the correlation among labels by leveraging the shared low-rank matrix, and of the similarity between annotators using the individual low-rank matrices to guide the factorization. ML-JMF pursues the low-rank matrices via a unified objective function, and introduces an iterative technique to optimize it. ML-JMF finally uses the optimized low-rank matrices and weights to infer the ground-truth labels. Our experimental results on multi-label datasets show that ML-JMF outperforms competitive methods in inferring ground truth labels. Our approach can identify unreliable annotators, and is robust against their misleading answers through the assignment of small (zero) weights to their annotation.**

***Index Terms*—Crowdsourcing, Multi-Label Learning, Joint Matrix Factorization, Spammers**

I. Introduction

With the emergence of the internet of things, a large amount of unlabeled data can be easily and cheaply collected. However, annotating such a vast amount of unlabeled data is a difficult challenge, because annotating data with correct and complete labels is time-consuming and often impractical, generally requiring expert knowledge. Crowdsourcing [1] provides an effective and economic solution to collect labels for data from non-expert workers in the open Internet. Label quality of training data plays a crucial role for the performance of machine leaning algorithms, and high-quality labels contribute to reliable performance. Due to significant differences among the crowders (or *workers*) in their knowledge levels, dedications, and evaluation criteria when crowdsourcing, the quality of crowdsourced labels (answers) may be quite different [2], [3]. Furthermore, some workers may simply submit random answers as a mean to earn easy money. Therefore, how to aggregate high-quality answers is a key pursue in croudsourcing [4].

To aggregate high-quality answers, one typical solution is *repeated labeling*, which involves the annotation of the same samples by different workers. Preliminary studies [5] show that the quality of answers can be improved to some degree by integrating repeated labels. Label integration is accomplished by a ground-truth answer inference algorithm on crowdsourced labels, without knowing the features of samples. Many researchers worked on the development of methods to derive high-quality answers from different perspectives, such as the reliability [6], intention [7], difficulty of samples [8], bias of workers [9], and so on.

The aforementioned answer aggregation methods focus on single-label tasks, in which a worker is expected to assign a single label to each sample. However, for many real-world crowdsourcing applications (i.e., image annotations and medical diagnosis [10], [11]), it's common for a sample to be simultaneously associated with several labels. In other words, workers are expected to provide a set of relevant labels for each sample. Assigning several labels to each sample increases the level of noise and bias of crowdsourced labels. In addition, workers no longer either completely agree or disagree on the crowdsourced labels, and thus the consensus becomes partial. Consequently, it's difficult to assess the reliability of workers, since they may provide partially correct and incorrect answers at the same time [12]. Furthermore, the number of possible label combinations is affected by a combinatorial explosion in the case of multi-label samples. For these reasons, multi-label answer aggregation is intrinsically *more challenging* than its single-label counterpart. One simple solution is to independently treat each label and transform the task into multiple binary tasks. Such binary solutions completely ignore the correlation between labels, whose appropriate usage can significantly improve the performance of multi-label learning [13], [14].

To the best of our knowledge, the problem of multi-label answer aggregation in crowdsourcing remains a largely

* Corresponding author: gxyu@swu.edu.cn (Guoxian Yu)

unexplored topic [12], [15]. In this paper, we propose a Multi-Label answer aggregation approach based on Joint Matrix Factorization (ML-JMF). To take advantage of the robustness of low-rank matrix factorization to noise [16], [17], ML-JMF jointly factorizes the sample-label association matrices of respective workers into a set of low-rank matrices for individual workers and a shared low-rank matrix for labels. ML-JMF assigns different weights to these association matrices to further reduce the impact of low quality workers. In addition, ML-JMF defines a term on the shared matrix to employ the correlation between labels and another term on the individual matrices to employ the similarity between workers. These two terms and the objective of matrix factorization are integrated into a unified objective function to guide the factorization. We introduce an iterative solution to optimize the weights, individual low-rank matrices of workers, and the shared low-rank matrix. In the end, ML-JMF uses these weights and optimized low-rank matrices to infer the ground-truth labels.

The main contributions of this paper are summarized as follows:

(1) Our proposed ML-JMF can simultaneously take into account the quality of workers, the noise of crowdsourced labels, correlations between labels, and connections between workers for multi-label answer aggregation.

(2) We introduce an iterative technique to optimize the weights assigned to workers and to pursue the joint matrix factorization.

(3) Our empirical study on benchmark datasets shows that ML-JMF outperforms state-of-the-art competitive methods [18]–[20] for answer aggregation by up to 95% in accuracy, while being robust against spammers. In addition, it can automatically identify low quality workers.

The remainder of this paper is organized as follows. We briefly review related work in Section II, and then elaborate on the proposed algorithm and its optimization in Section III. Section IV provides the experimental results and analysis, and Section VI gives the conclusions and future work.

II. Related Work

The simplest and most efficient answer aggregation method is majority voting (MV) [21]. MV works very well under two prerequisites: 1) the overall accuracy of most workers is larger than 50% in binary labeling tasks, and 2) the error of each worker is uniformly distributed over all class labels. However, these prerequisites do not hold in complicated real-world applications. Due to the lack of expert knowledge, most workers tend to make shallow answers using common sense or simply repeat what others say.

Besides the straightforward MV, researchers are dedicated to many other aggregation solutions from different perspectives [22]. To name a few, Dawid and Skene [23] applied expectation maximization (EM) to model the confusion matrix of each worker and to conduct aggregation from a set of noisy labels. This EM based aggregation algorithm iteratively estimates the labels that are most likely true classes, and then uses these labels to estimate the error rate of each worker and the label distribution. Raykar *et al.* [24] assumed that annotators have biases toward the positive class and negative class, and introduced a Bayesian approach by adding a specific prior for each class. Whitehill *et al.* [25] proposed GLAD (Generative model of Labels, Abilities, and Difficulties) to model both the expertise levels of workers and the difficulties of samples using EM. GLAD treats the probability of a sample being positive as a latent variable, and it can produce high quality results even with many adversarial labelers. Zhang *et al.* [26] proposed a Positive LAbel frequency Threshold (PLAT) algorithm to solve the imbalanced labeling problem caused by the bias of workers via dynamically adjusting the threshold to determine the class membership of an example [26]. Zhang *et al.* [20] introduced adaptive weighted majority voting (AWMV) to utilize the frequency of positive labels in the multiple noisy label sets of each example to estimate a bias rate, and then to assign weights derived from the bias rate to negative and positive labels.

Some researchers have focused on worker behavior or task assignment to improve label quality. Demartini *et al.* [6] assumed that workers act independently and aggregated labels by solving a maximum likelihood estimation problem. Raykar and Yu [27] developed an empirical Bayesian algorithm based on EM to iteratively estimate the ground-truth label and eliminate spammers. Karger *et al.* [28] proposed a belief propagation model to decide which tasks to assign to which workers. This belief model uses task messages to iteratively update worker messages, and vice versa. Next, the true classes are estimated from the information contained in the task messages. Ho and Vaughan [29] developed a two-phase exploration-exploitation algorithm for assigning heterogeneous items to workers with different qualities. Wang *et al.* [30] proposed an approach to obtain high-quality labels from the crowds by distinguishing easy and hard items prior to assigning them to workers.

The multi-label answer aggregation problem has been much less explored [12], [31] than single-label aggregation solutions. Nowak *et al.* [31] studied inter-annotator agreement for multi-label image annotation and found that using the majority vote strategy to generate one annotation set from several annotation sets can filter out noisy judgments of non-experts to some extent. To address the problem of different taxonomies being used in a multi-label domain, Duan *et al.* [15] proposed a probabilistic cascaded method called cascaded estimation with Dawid-Skene (C-DS). C-DS maps label sets in a source taxonomy to label sets in a target taxonomy in terms of the semantic distance between them. Yoshimura *et al.* [19] incorporated GLAD [25] into RAkEL (RAndom k-labELsets) [32] and proposed RAkEL-GLAD to balance the estimation accuracy and computational complexity in multi-label answer aggregation. Hung *et al.* [12] extended the clustering based Bayesian combination of classifiers method [33] for multi-label answer aggregation. This extended solution additionally models the co-occurrence dependency between labels by latent label clusters and the partial consensus between workers by grouping workers with similar answers.

The aforementioned single-label aggregation approaches ignore the interdependence between labels; some of them cannot perform as well as on binary setting, while some other may fail to adapt to multi-label scenarios [34]. On the other hand, multi-label aggregation methods do not differentiate among different types of workers, they do not account for potential noisy annotations and the different biases of individual workers. It's recognized that both label correlation and the types of workers contribute to answer aggregation [19], [35]. Given these observations, we propose an approach called ML-JMF to simultaneously account for label correlations, noisy labels, and quality of individual workers. ML-JMF can differentiate the quality of workers by assigning different weights to their annotation matrices, and reduce noise through low-rank matrix factorization. It further exploits correlation between labels and the similarity between workers to guide the low-rank matrix and weight optimization. Our empirical study shows that ML-JMF achieves superior aggregated labels than other inference algorithms [18]–[20]. ML-JMF can also identify spammers and can selectively aggregate annotations of workers.

III. Problem Formulation

In this section, we first discuss an image annotation task to illustrate the intrinsic challenges of multi-label answer aggregation. Then, we elaborate on ML-JMF and its optimization.

A. Motivation

Table I lists the crowdsourced labels of four images (i_1 - i_4) provided by five workers (w_1 - w_5). For simplicity, these labels are denoted with numbers from 1 to 5. In particular, '-' denotes the fact that the worker thinks the image should not be annotated with the corresponding label.

Table I: Annotation collected from five workers on four images

	w_1	w_2	w_3	w_4	w_5	ground truths	Majority Voting
i_1	{2,3,-4}	{2,3}	{1}	{3}	{2}	{2}	{2,3}
i_2	{3,4}	{-2,3,4}	{2}	{3}	{1,3,4}	{1,3,4}	{3}
i_3	{3,5}	{-1,4}	{4}	{3}	{4,5}	{4,5}	{4}
i_4	{-1,2,3}	{3,4}	{5}	{3}	{2,3,4}	{2,3,4}	{3}

1: grass, 2: lion, 3: sun, 4: tree, 5: river

A straightforward and widely adopted approach to derive aggregated labels is majority voting (MV) [18], [31], which separately considers the five labels. If the number of 'votes' for a particular label of a sample from all workers is the largest (or larger than half workers), this label is included in the aggregated label set. Considering ground truth labels (in practice, often unknown), we have two observations: the aggregated results obtained using MV are (i) partially incorrect (e.g., label 3 should not be assigned to i_1); and (ii) partially incomplete (e.g., labels 1 and 4 should also be assigned to i_4).

This is due to the fact that MV considers all answers as equally important and MV ignores the correlation between labels. In other words, MV assumes that all workers have similar biases and produce answers of equal quality. But in practice, they don't. Kazai *et al.* [35] categorized workers into five groups: (i) Diligent workers (reliable workers) take care of their tasks and may be characterized by a high ratio of useful labels; (ii) Normal workers have general knowledge to give correct answers, but make mistakes occasionally; (iii) Sloppy workers care little about the quality of their work, they may still provide a high fraction of useful labels but with low accuracy; (iv) Incompetent workers lack professional skills or competence, resulting in low accuracy; (v) Spammers may come in different shapes and forms, e.g., they give the same answer to all questions or give random answers. Given the data in Table I, w_5 might be a reliable worker who assigns correct labels; w_1 and w_2 may be normal workers who can give some correct answers; and w_3 and w_4 are spammers. Unlike single-label data, labels of multi-label samples are correlated. For example, we can see that labels 3 and 4 are often assigned to the same images.

From the illustrative example given in Table I, we can conclude that: (i) if the spammers (w_3 and w_4) are removed, the aggregated results for i_1 and i_3 will be correct; (ii) label 4 can be assigned to the same image already tagged with label 3, thus image i_4 can be annotated correctly with its ground truth labels. This illustrates that both *quality* of workers and *correlation* among labels should be considered in multi-label answer aggregation.

B. The Proposed Algorithm

Suppose there are m workers providing labels for n samples, each of which can be annotated with one or more labels, and $\mathcal{L} \triangleq \{1, \ldots, c\}$ is the label set. Each collection of labels provided by a worker as annotation for a sample is a subset of $\mathcal{L}$. Thus, each worker provides a sample-label association matrix for n samples and c distinct labels as follows:

$$\mathbf{A}_w \triangleq \begin{pmatrix} a_{11}^w & \cdots & a_{1c}^w \\ \vdots & \ddots & \vdots \\ a_{n1}^w & \cdots & a_{nc}^w \end{pmatrix} \tag{1}$$

where $a_{il}^w \in \{-1, 0, 1\}$. $a_{il}^w = 1(-1)$ states that the w-th worker annotated the i-th sample with (or without) the l-th ($1 \leq l \leq c$) label, and $a_{il}^w = 0$ means that the worker did not provide an answer for the corresponding label and sample.

The feature information of the n samples may often be shielded from the inference algorithm due to privacy issues [6]. To obtain high-quality aggregated labels from $\{\mathbf{A}_w\}_{w=1}^m$, we advocate the use of the shared information of workers, and also the intrinsic characteristics of individual workers. To this end, and motivated by the robustness to noise of a low-rank approximation of a matrix [17], [36], we jointly factorize $\{\mathbf{A}_w\}_{w=1}^m$ as follows:

$$\min_{\mathbf{U},\mathbf{V}>0} \sum_{w=1}^{m} \boldsymbol{\mu}_w \left\| \mathbf{A}_w - \mathbf{U}_w \mathbf{S} \mathbf{V}^T \right\|_F^2 \tag{2}$$
$$s.t. \sum_{w=1}^{m} \boldsymbol{\mu}_w = 1, \boldsymbol{\mu}_w \geq 0$$

where $\|\cdot\|_F^2$ is the Frobenius norm, $\mathbf{U}_w \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{c \times k}$ are the individual matrix for the w-th worker and the shared low-rank matrix for c labels across m workers;

$k < (n, c)$ is the low-rank size of these two matrices. $\mathbf{S} \in \mathbb{R}^{k \times k}$ is introduced to ensure that the values of $\mathbf{U}_w$ and $\mathbf{V}$ are nonnegative. $\mathbf{V}$ partially encodes the dependency between c distinct labels in the k-dimensional real space. $\boldsymbol{\mu}_w \geq 0$ is the weight assigned to the w-th worker, and it's introduced to reduce the impact of low quality workers (e.g. spammers).

Eq. (2) has a trivial solution $\boldsymbol{\mu}_w = 1$ when $\|\mathbf{A}_w - \mathbf{U}_w \mathbf{S} \mathbf{V}\|_F^2$ gives the smallest approximation loss. To avoid this trivial solution, we add an l_2-norm to Eq. (2) as follows:

$$\begin{aligned} \min \sum_{w=1}^{m} \boldsymbol{\mu}_w \left\| \mathbf{A}_w - \mathbf{U}_w \mathbf{S} \mathbf{V}^T \right\|_F^2 + \lambda \|\boldsymbol{\mu}\|_F^2 \\ s.t. \sum_{w=1}^{m} \boldsymbol{\mu}_w = 1, \boldsymbol{\mu}_w \geq 0 \end{aligned} \tag{3}$$

where $\lambda > 0$ controls the importance of $\|\boldsymbol{\mu}\|_F^2$ and contributes to selectively aggregate the answers from workers by assigning different (possibly zero) weights to $\{\mathbf{A}_w\}_{w=1}^m$. By minimizing the above equation, a larger weight can be assigned to the worker whose annotation matrix $\mathbf{A}_w$ has a smaller approximation loss. On the other hand, if a matrix $\mathbf{A}_w$ cannot be well approximated, it means it's not consistent with the annotations of other workers and may not be reliable. As such, MF-JMF has the potential of automatically removing noisy annotation matrices by crediting zero weights. It can also reduce the impact of partially noisy annotation matrices by assigning smaller weights to them, or by low-rank matrix approximation. Our experimental results confirm the effectiveness of this process.

Unlike single-label answer aggregation, multi-label answer aggregation should account for correlations among labels, since a multi-label sample is often simultaneously annotated with several related but different labels. For example, in Table I, tree and sun are often assigned to the same images, so if a sample is tagged with tree (sun), then it's quite likely to be tagged with sun (tree) also. In fact, this co-occurrence of information is widely adopted in multi-label learning [13]. Given this, we make use of label correlation based on the low-rank representation of c labels as follows:

$$\begin{aligned} \min_{\mathbf{V} \geq 0} \frac{1}{2} \sum_{i,j} \mathbf{C}_{ij} \|\boldsymbol{v}_{i\cdot} - \boldsymbol{v}_{j\cdot}\|_2^2 &= tr(\mathbf{V}^T (\mathbf{D} - \mathbf{C}) \mathbf{V}) \\ &= tr(\mathbf{V}^T \mathbf{L} \mathbf{V}) \end{aligned} \tag{4}$$

where $\mathbf{C} \in \mathbb{R}^{c \times c}$ stores the label correlation between c labels. $\boldsymbol{v}_{i\cdot}$ is the i-th row of $\mathbf{V}$, $tr(\cdot)$ denotes the matrix trace operation, $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{c} \mathbf{C}_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{C}$. For simplicity, we adopt the widely used cosine similarity to quantify the correlation between labels, based on the averaged sample-label association matrix $\sum_{w=1}^{m} \mathbf{A}_w / m$. Other more advanced label correlation measurements can also be adopted [37].

If two workers have similar profiles (levels of knowledge or background), they are likely to give similar answers to the same samples. In other words, the profile similarity can be used to guide the pursue of individual low-rank matrices $\mathbf{U}_w$. Thus, we define a regularization on individual matrices $\mathbf{U}_w$ to take advantage of worker profiles as follows:

$$\begin{aligned} &\min_{\mathbf{U}_w \geq 0} \frac{1}{2} \sum_{w \neq p} \mathbf{R}_{wp} ||\mathbf{U}_w - \mathbf{U}_p||_F^2 \\ &\quad = \sum_{w \neq p} \mathbf{R}_{wp} tr((\mathbf{U}_w - \mathbf{U}_p)^T (\mathbf{U}_w - \mathbf{U}_p)) \\ &\mathbf{R}_{wp} = \frac{tr(\widetilde{\mathbf{A}}_w \widetilde{\mathbf{A}}_p)}{\sqrt{tr(\widetilde{\mathbf{A}}_w \widetilde{\mathbf{A}}_w) tr(\widetilde{\mathbf{A}}_p \widetilde{\mathbf{A}}_p)}} \\ &s.t. \quad \widetilde{\mathbf{A}}_w = \mathbf{A}_w \mathbf{A}_w^T - diag(\mathbf{A}_w \mathbf{A}_w^T) \end{aligned} \tag{5}$$

where $\mathbf{R}_{wp}$ is the similarity between worker w and worker p; it's measured using the modified RV-coefficient [38]. The modified RV-coefficient ($\mathbf{R}_{wp}$) is suggested to measure the common information of high-dimensional data matrices; it can probe the similarity between pairs of datasets (or data matrices) in a simple and comprehensive way [39]. The value of $\mathbf{R}_{wp}$ is between 0 and 1: the larger the value, the larger the similarity between the two workers is. $\mathbf{R}_{wp}$ can also be computed via Pearson correlation or cosine similarity. Our investigation shows that ML-JMF combined with the RV-coefficient to estimate worker similarity can achieve a better accuracy than ML-JMF combined with cosine similarity or Pearson correlation.

We combine the constraints on $\mathbf{V}$ and $\mathbf{U}_w$ with the joint matrix factorization in Eq. (2), and form the objective function of ML-JMF as follows:

$$\begin{aligned} \Phi(\mathbf{U}_w, \mathbf{S}, \mathbf{V}, \boldsymbol{\mu}) &= \arg \min_{\mathbf{U}, \mathbf{V} > 0} \sum_{w=1}^{m} \boldsymbol{\mu}_w \left\| \mathbf{A}_w - \mathbf{U}_w \mathbf{S} \mathbf{V}^T \right\|_F^2 \\ &+ \lambda \|\boldsymbol{\mu}\|_F^2 + \alpha \mathrm{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &+ \beta \sum_{w \neq p} \mathbf{R}_{wp} \mathrm{tr}((\mathbf{U}_w - \mathbf{U}_p)(\mathbf{U}_w - \mathbf{U}_p)^T) \\ s.t. &\sum_{w=1}^{m} \boldsymbol{\mu}_w = 1, \boldsymbol{\mu}_w \geq 0, \mathbf{U}_w \geq 0, \mathbf{V} \geq 0 \end{aligned} \tag{6}$$

where the parameters α and β weight the constraints in Eq. (4) and Eq. (5), respectively. Besides the joint matrix factorization, we employ two constraints to guide the pursue of $\mathbf{U}_w$, $\mathbf{V}$, and $\boldsymbol{\mu}_w$, and thus to improve the accuracy and reliability of multi-label answer aggregation. The experiments confirm the advantage of using these two constraints.

After optimizing $\boldsymbol{\mu}_w$, $\mathbf{U}_w$, $\mathbf{S}$, and $\mathbf{V}$, ML-JMF selectively aggregates the annotation matrices of m workers as follows:

$$\mathbf{A}^* = \sum_{w=1}^{m} \boldsymbol{\mu}_w \mathbf{U}_w \mathbf{S} \mathbf{V} \tag{7}$$

The inferred sample-label association matrix $\mathbf{A}^*$ not only can reduce the impact of too noisy annotation matrices by assigning $\boldsymbol{\mu}_w = 0$ to them, but also removes partially noisy annotations in $\mathbf{A}_w$ of selected workers by low-rank matrix approximation.

C. Optimization

The proposed objective function of ML-JMF in Eq. (6) is not convex for all variables $\mathbf{V}$, $\boldsymbol{\mu}_w$, $\mathbf{S}$, and $\mathbf{U}_w$ ($w =$

$1, 2, \cdots, m$) at the same time. Therefore, it is unrealistic to expect to find the global optimum simultaneously. Here, we introduce an alternative update strategy to optimize $\mathbf{V}$, $\boldsymbol{\mu}_w$, $\mathbf{S}$, and $\mathbf{U}_w$. Particularly, we will optimize one variable while fixing the other variables as constants.

1) Optimizing $\mathbf{V}$*:* By fixing $\mathbf{U}_w$, $\boldsymbol{\mu}_w$ ($\forall w$), and $\mathbf{S}$, we can optimize $\mathbf{V}$ as follows:

$$\min\ \Phi_1(\mathbf{V}) = \sum_{w=1}^{m} \boldsymbol{\mu}_w \|\mathbf{A}_w - \mathbf{U}_w\mathbf{S}\mathbf{V}^T\|_F^2 + \alpha tr(\mathbf{V}^T\mathbf{L}\mathbf{V}) \quad s.t.\ \ \mathbf{V} \geq 0 \tag{8}$$

The derivative of $\Phi_1(\mathbf{V})$ with respect to $\mathbf{V}$ is

$$\frac{\partial \Phi_1}{\partial \mathbf{V}} = \sum_{w=1}^{m} \boldsymbol{\mu}_w (2\mathbf{V}\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S} - 2\mathbf{A}_w^T\mathbf{U}_w\mathbf{S}\mathbf{V}^T) + 2\alpha\mathbf{L}\mathbf{V} \tag{9}$$

Using the Karush-Kuhn-Tucker (KKT) complementary condition [40] for the nonnegativity of $\mathbf{V}$, we can obtain:

$$(\sum\nolimits_{w=1}^{m} \boldsymbol{\mu}_w (\mathbf{V}\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S} - \mathbf{A}_w^T\mathbf{U}_w\mathbf{S}\mathbf{V}^T) + \alpha\mathbf{L}\mathbf{V})_{ij}\mathbf{V}_{ij} = 0 \tag{10}$$

Considering $\mathbf{V} \geq 0$, $\mathbf{S}$ and $\mathbf{L}$ may take a positive or negative sign, we decompose them as $\mathbf{A}_w^T\mathbf{U}_w\mathbf{S} = (\mathbf{A}_w^T\mathbf{U}_w\mathbf{S})^+ - (\mathbf{A}_w^T\mathbf{U}_w\mathbf{S})^-$ and $\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S} = (\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S})^+ - (\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S})^-$, where the matrices with positive and negative symbols are defined as $\mathbf{O}^+ = \frac{|\mathbf{O}|+\mathbf{O}}{2}$ and $\mathbf{O}^- = \frac{|\mathbf{O}|-\mathbf{O}}{2}$. Then, we can obtain the following updating formula for $\mathbf{V}$:

$$\mathbf{V} \leftarrow \mathbf{V}\sqrt{\frac{\sum\limits_{w=1}^{m} \boldsymbol{\mu}_w (\mathbf{A}_w^T\mathbf{U}_w\mathbf{S})^+ + \sum\limits_{w=1}^{m} \boldsymbol{\mu}_w \mathbf{V}(\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S})^- + \alpha\mathbf{L}^-\mathbf{V}}{\sum\limits_{w=1}^{m} \boldsymbol{\mu}_w (\mathbf{A}_w^T\mathbf{U}_w\mathbf{S})^- + \sum\limits_{w=1}^{m} \boldsymbol{\mu}_w \mathbf{V}(\mathbf{S}^T\mathbf{U}_w^T\mathbf{U}_w\mathbf{S})^+ + \alpha\mathbf{L}^+\mathbf{V}}} \tag{11}$$

2) Optimizing $\mathbf{U}_w$*:* Similarly, we can update the $\mathbf{U}_w$, one by one. For a $\mathbf{U}_w$, given $\mathbf{V}, \mathbf{S}$, $\boldsymbol{\mu}_w$, and $\mathbf{U}_p$, $p \in \{1, 2, \ldots, m\}, w \neq p$, the objective function for optimizing $\mathbf{U}_w$ is:

$$\begin{aligned} \min \Phi_2(\mathbf{U}_w) &= \boldsymbol{\mu}_w \left\|\mathbf{A}_w - \mathbf{U}_w\mathbf{S}\mathbf{V}^T\right\|_F^2 \\ &+ \beta \sum_{w \neq p} \mathbf{R}_{wp} \mathrm{tr}((\mathbf{U}_w - \mathbf{U}_p)(\mathbf{U}_w - \mathbf{U}_p)^T) \\ & s.t.\ \ \mathbf{U}_w \geq 0 \end{aligned} \tag{12}$$

The derivative of Φ_2 with respect to $\mathbf{U}_w$ is

$$\begin{aligned} \frac{\partial \Phi_2}{\partial \mathbf{U}_w} &= \boldsymbol{\mu}_w (2\mathbf{U}_w\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T - 2\mathbf{A}_w\mathbf{V}\mathbf{S}^T) \\ &+ 2\beta \sum\nolimits_{w \neq p} \mathbf{R}_{wp} (\mathbf{U}_w - \mathbf{U}_p) \end{aligned} \tag{13}$$

Using the KKT complementary condition [40] for the nonnegativity of $\mathbf{U}_w$, we can obtain:

$$(\boldsymbol{\mu}_w (\mathbf{U}_w\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T - \mathbf{A}_w\mathbf{V}\mathbf{S}^T) + \beta \sum_{w \neq p} \mathbf{R}_{wp} (\mathbf{U}_w - \mathbf{U}_p))_{ij} (\mathbf{U}_w)_{ij} = 0 \tag{14}$$

Since $\mathbf{S}$ may take any sign, similarly to the computation of Eq. (9), we let $\mathbf{A}_w\mathbf{V}\mathbf{S}^T = (\mathbf{A}_w\mathbf{V}\mathbf{S}^T)^+ - (\mathbf{A}_w\mathbf{V}\mathbf{S}^T)^-$ and $\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T = (\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T)^+ - (\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T)^-$. Thus, Eq. (13) leads to the following update formula for $\mathbf{U}_w$:

$$\mathbf{U}_w \leftarrow \mathbf{U}_w \sqrt{\frac{\boldsymbol{\mu}_w (\mathbf{A}_w\mathbf{V}\mathbf{S}^T)^+ + \boldsymbol{\mu}_w (\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T)^- + 2\beta \sum\limits_{w \neq p} \mathbf{R}_{wp}\mathbf{U}_p}{\boldsymbol{\mu}_w (\mathbf{A}_w\mathbf{V}\mathbf{S}^T)^- + \boldsymbol{\mu}_w (\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^T)^+ + 2\beta \sum\limits_{w \neq p} \mathbf{R}_{wp}\mathbf{U}_w}} \tag{15}$$

3) Optimizing $\mathbf{S}$*:* With $\mathbf{U}_w$, $\mathbf{V}$, and $\boldsymbol{\mu}$ known, optimizing Eq. (6) with respect to $\mathbf{S}$ is equivalent to optimize

$$\min \Phi_3(\mathbf{S}) = \sum_{w=1}^{m} \boldsymbol{\mu}_w \left\|\mathbf{A}_w - \mathbf{U}_w\mathbf{S}\mathbf{V}^T\right\|_F^2 \tag{16}$$

Letting $\frac{\partial \Phi_3}{\partial \mathbf{S}} = 0$ leads to the following updating formula:

$$\mathbf{S} = (\sum_{w=1}^{m} \boldsymbol{\mu}_w (\mathbf{U}_w^T\mathbf{U}_w))^{-1} (\sum_{w=1}^{m} \boldsymbol{\mu}_w (\mathbf{U}_w^T\mathbf{A}_w\mathbf{V}))(\mathbf{V}^T\mathbf{V})^{-1} \tag{17}$$

4) Optimizing $\boldsymbol{\mu}$*:* Next, we view $\mathbf{V}$, $\mathbf{U}_w$, and $\mathbf{S}$ as known, and define the objective function with respect to $\boldsymbol{\mu}$ as follows:

$$\begin{aligned} \min \Phi_4(\boldsymbol{\mu}) &= \sum\nolimits_{w=1}^{m} \boldsymbol{\mu}_w \left\|\mathbf{A}_w - \mathbf{U}_w\mathbf{S}\mathbf{V}^T\right\|_F^2 + \lambda \|\boldsymbol{\mu}\|_F^2 \\ &- \sum\nolimits_{w=1}^{m} \boldsymbol{\zeta}_w \boldsymbol{\mu}_w - \gamma(\sum\nolimits_{w=1}^{m} \boldsymbol{\mu}_w - 1) \end{aligned} \tag{18}$$

where $\boldsymbol{\zeta}_w \geq 0$ and $\gamma \geq 0$ are the introduced Lagrange multipliers for constraints $\boldsymbol{\mu}_w \geq 0$ and $\sum_{w=1}^{m} \boldsymbol{\mu}_w = 1$. Let $\mathbf{h}_w = \|\mathbf{A}_w - \mathbf{U}_w\mathbf{S}\mathbf{V}^T\|_F^2$ be the *approximation loss* for $\mathbf{A}_w$, $\mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m]$. The partial derivative of $\Phi_4(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ is:

$$\frac{\partial \Phi_4}{\partial \boldsymbol{\mu}} = \mathbf{h} + 2\lambda\boldsymbol{\mu} - \boldsymbol{\zeta} - \gamma \tag{19}$$

The optimal $\boldsymbol{\mu}$ should satisfy the following four conditions [40]:

1) Complementary slackness condition: $\boldsymbol{\zeta}_w\boldsymbol{\mu}_w = 0$;
2) Stationary condition: $\mathbf{h}_w + 2\lambda\boldsymbol{\mu}_w - \boldsymbol{\zeta}_w - \gamma = 0$;
3) Feasible condition: $\sum_{w=1}^{m} \boldsymbol{\mu}_w = 1, \boldsymbol{\mu}_w \geq 0$;
4) Dual feasibility condition: $\forall \boldsymbol{\zeta}_w \geq 0$.

From the stationary condition, $\boldsymbol{\mu}_w$ can be computed as:

$$\boldsymbol{\mu}_w = \frac{\boldsymbol{\zeta}_w + \gamma - \mathbf{h}_w}{2\lambda} \tag{20}$$

From Eq. (18), we can see that $\boldsymbol{\mu}_w$ depends on $\boldsymbol{\zeta}_w$ and γ, both of which can be analyzed via the following cases:

1) if $\gamma > \mathbf{h}_w$, then $\boldsymbol{\mu}_w > 0$, because of the complementary slackness $\boldsymbol{\zeta}_w\boldsymbol{\mu}_w = 0$ and the dual feasibility $\forall \boldsymbol{\zeta}_w \geq 0$, $\boldsymbol{\zeta}_w = 0$ and $\boldsymbol{\mu}_w = \frac{\gamma - \mathbf{h}_w}{2\lambda}$.
2) if $\gamma = \mathbf{h}_w$, because of $\boldsymbol{\zeta}_w\boldsymbol{\mu}_w = 0$ and $\boldsymbol{\mu}_w = \frac{\boldsymbol{\zeta}_w}{2\lambda}$, $\boldsymbol{\zeta}_w = 0$ and $\boldsymbol{\mu}_w = 0$.
3) if $\gamma < \mathbf{h}_w$, since $\boldsymbol{\mu}_w \geq 0$, it requires $\boldsymbol{\zeta}_w > 0$; because $\boldsymbol{\zeta}_w\boldsymbol{\mu}_w = 0$, then $\boldsymbol{\mu}_w = 0$.

From the above analysis, we can set $\boldsymbol{\mu}_w$ as:

$$\boldsymbol{\mu}_w = \begin{cases} \frac{\gamma - \boldsymbol{h}_w}{2\lambda}, & \gamma > \boldsymbol{h}_w \\ 0, & \gamma \leq \boldsymbol{h}_w \end{cases} \tag{21}$$

Suppose $\overrightarrow{\mathbf{h}}$ stores the entries of $\mathbf{h}$ in ascending order. For a predefined λ not too large, there exists $q \in \{1, 2, \cdots, m\}$ with $\overrightarrow{\mathbf{h}}_q < \gamma$ and $\overrightarrow{\mathbf{h}}_{q+1} \geq \gamma$, satisfying $\sum_{w=1}^{m} \boldsymbol{\mu}_w = \sum_{w=1}^{q} \frac{\gamma - \overrightarrow{\mathbf{h}}_w}{2\lambda} = 1$. Then $\boldsymbol{\mu}_w$ has the following explicit solution:

$$\boldsymbol{\mu}_w = \begin{cases} \frac{\gamma - \overrightarrow{\mathbf{h}}_w}{2\lambda}, & w \leq q \\ 0, & w > q \end{cases} \tag{22}$$

From $\sum_{w=1}^{m} \boldsymbol{\mu}_w = \sum_{w=1}^{q} \frac{\gamma - \overrightarrow{\mathbf{h}}_w}{2\lambda} = 1$, we can get the value for γ as:

$$\gamma = \frac{2\lambda + \sum_{w=1}^{q} \overrightarrow{\mathbf{h}}_w}{q} \tag{23}$$

From the solution of $\boldsymbol{\mu}$, we find that, if $\overrightarrow{\mathbf{h}}_r > \overrightarrow{\mathbf{h}}_p$ and $\gamma \geq \overrightarrow{\mathbf{h}}_r$, the p-th worker will get a larger weight than the r-th worker. This is the case because the r-th worker may provide noisy annotations, which are inconsistent with other workers, and therefore resulting in a large approximation loss. Therefore, adding an l_2 norm to $\boldsymbol{\mu}$ in Eq. (2) can not only remove noisy (irrelevant) answer matrices by assigning zero weights to them, but also can reduce the impact of partially noisy annotation matrices by crediting reduced weights to them. In addition, because of the robustness of low-rank matrix approximation to noises [17], [36], [41], the joint matrix factorization can also remove noisy annotations, and thus further improve the quality of aggregated labels.

From Eq. (22) and Eq. (23), we see that if λ is set to a very small positive value, $\gamma \approx \sum_{w=1}^{q} \overrightarrow{\mathbf{h}}_q / q$, and then ML-JMF will select at least one annotation matrix. On the other hand, if λ is fixed to a very large value, then all the annotation matrices will be used and credited nearly equal weights. To find a value of q that satisfies Eq. (22), we decrease q from m to 1, step by step, and specify the search procedure in Algorithm 1. The whole ML-JMF approach is summarized in Algorithm 2.

Algorithm 1 A method to seek q and compute $\boldsymbol{\mu}_w$

Input: Sorted $\overrightarrow{\mathbf{h}}_w, w \in \{1, 2, \dots m\}$ in ascending order, λ
Output: $q, \boldsymbol{\mu}_w$
Initialize $q = m, \gamma = 0$.
While $q > 0$ **do**
  $\gamma = \frac{2\lambda + \sum_{w=1}^{q} \overrightarrow{\mathbf{h}}_w}{q}$
  If $\gamma - \overrightarrow{\mathbf{h}}_q > 0$ **then**
    break.
  Else
    $q \leftarrow q - 1$.
  End If
End While
$\boldsymbol{\mu}_w \leftarrow \frac{\gamma - \overrightarrow{\mathbf{h}}_w}{2\lambda}$, for $w = 1, \cdots, q$.
$\boldsymbol{\mu}_w \leftarrow 0$, for $w = q+1, \cdots, m$.

Algorithm 2 ML-JMF: Multi-label Answer Aggregation based on Joint Matrix Factorization

Input:
$\{\mathbf{A}_w\}_{w=1}^{m}$: Annotation matrices of m workers;
α, β, λ: input parameters of ML-JMF;
tol: tolerance threshold for iterative optimization;
$maxIter$: maximum number of iterations.
Output:
$\boldsymbol{\mu}$: weights assigned to m workers;
$\mathbf{A}^*$: the aggregated sample-label association matrix;
Initialize $\mathbf{U}_w$, $\mathbf{S}$, $\mathbf{V}$ and $\boldsymbol{\mu}$ in random;
Compute $\mathbf{R}$ via Eq. (5) and $\mathbf{C}$ via cosine similarity;
Compute the initial value of $\Phi^1(\mathbf{U}_w, \mathbf{S}, \mathbf{V}, \boldsymbol{\mu})$ via Eq. (6);
$t = 0$, $\Phi^0 = 0$;
While $|\Phi^t - \Phi^{t+1}| > tol$ & $t < maxIter$
  $t = t + 1$;
  Update the matrix $\mathbf{U}_w$,$\mathbf{V}$, $\mathbf{S}$ via Eq. (15), Eq. (11), Eq. (17), respectively;
  Update $\boldsymbol{\mu}$ using **Algorithm** 1;
  Compute the value of $\Phi^{t+1}(\mathbf{U}_w, \mathbf{S}, \mathbf{V}, \boldsymbol{\mu})$;
End While
Return the aggregated label matrix $\mathbf{A}^*$ via Eq. (7)

IV. Experimental Setup

Datasets: To study the performance of ML-JMF in aggregating crowdsourced labels of multi-label samples, we carry out experiments on five real-world datasets. The statistics of these datasets are listed in Table II. *Movie* is a movie category classification dataset used in [19]. The other four real-world datasets were used by Duan *et al.* [42] in emotion classification. The candidate label sets are taken from the Ekman's taxonomy [43] and the Nakamura's taxonomy [44].

Comparing Algorithms: We compare ML-JMF against two state-of-the-art multi-label methods RA*k*EL-GLAD [19] and C-DS [15], the classifical MV [18], and two representative single-label methods AWMV [20] and PLAT [26] (all discussed in the related work Section). RA*k*EL-GLAD has two parameters: k (number of labels in a label subset) and M (number of random label subsets); we set $k = 2$ and $M = \frac{c\times(c-1)}{2}$ for the experiments. To facilitate the comparison with AWMV and PLAT, we decompose the multi-label answer aggregation problem into multiple binary-label aggregation problems. For example, the *"Apple" Ekman* dataset has six labels, AWMV is separately applied on each label and each label has 2340 tasks (30 *workers/Instance* $\times$ 78 *Instances*). In addition, we introduce ML-JMF(A), a variant of ML-JMF, which uses the optimized weights, but also the original annotation matrices to infer labels; namely, $\mathbf{A}^* = \sum_{w=1}^{m} \boldsymbol{\mu}_w \mathbf{A}_w$. The input parameters of MV, AWMV and PLAT, and C-DS are specified or optimized as the authors suggested in their code or papers. ML-JMF and its variant set $\alpha = 10^3$, $\beta = 0.01$, $\lambda = 10^4$, and $k = \lceil c/2 \rceil + 1$. The parameter sensitivity analysis for ML-JMF is also presented.

Evaluation Metrics: In multi-label answer aggregation, results can be partially correct. We therefore rely on the set-based definition of *Accuracy* to evaluate the individual correctness on n samples. The accuracy is defined as follows [12], [45]:

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} \frac{|\mathcal{T}_i \bigcap \mathcal{T}_i^*|}{|\mathcal{T}^*|}, \tag{24}$$

where $\mathcal{T}_i$ and $\mathcal{T}_i^*$ are the set of true labels and the set of aggregated labels of the i-th sample, respectively.

We also use the *RankingLoss*, a representative evaluation metric in multi-label learning, to evaluate the average fraction of label pairs that are not correctly ranked for the sample. The formal definition of RankingLoss is:

$$Rankingloss = \sum_{j=1}^{n} \frac{1}{c} \sum_{i=1}^{c} \frac{|\mathcal{R}_i|}{|\mathcal{T}_i||\overline{\mathcal{T}}_i|} \tag{25}$$

where $\mathcal{R}_i = \{(c_1, c_2) \in \mathcal{T}_i \times \overline{\mathcal{T}}_i | \mathbf{A}^*(i, c_1) \leq \mathbf{A}^*(i, c_2)\}$, $\mathcal{T}_i \in \mathcal{L}$ is the set of labels associated with the i-th sample, $\overline{\mathcal{T}}_i$ is the complementary set of $\mathcal{T}_i$ in $\mathcal{L}$. $c_1 \in \mathcal{L}_i, c_2 \in \overline{\mathcal{T}}_i$ are the relevant labels and irrelevant labels, respectively. The smaller the RankingLoss is, the better the performance is. The performance is perfect when the RankingLoss is zero [13].

To be consistent with Accuracy, we report the 1-RankingLoss in the following experiments. The initially obtained aggregated labels are expressed as real-numbers and need to be converted into binary labels for computing the Accuracy. In the experiments, we choose the labels with the highest probabilities as the aggregated labels of the sample

Table II: Statistics of five real-world datasets used in the experiments

Dataset	*Workers*	*Instances*	*Labels*	*Tasks*	*Annotations*	*workers/Instance*	*Label per instance*
Movie	89	100	19	3500	6811	35	1.95
AppleEkman	68	78	6	2340	2978	30	1.27
AppleNakamura	57	78	10	2340	2768	30	1.18
LoveEkman	54	63	6	1890	1890	30	1.05
LoveNakamura	41	63	10	2583	3965	41	1.53

according to the number of ground truth labels per sample for all methods. For example, if the i-th sample has two true labels, all the methods consider the two labels with the highest values of their respective label likelihood vectors as the aggregated labels. The RankingLoss directly uses the initially aggregated labels without such conversion.

A. Results of Multi-Label Answer Aggregation

Table III shows the results of different answer aggregation methods on five real-world datasets. Since ML-JMF initializes the matrices $\mathbf{U}_w$, $\mathbf{S}$, and $\mathbf{V}$ randomly, we independently run ML-JMF ten times and report the average results and variance. The five comparing methods are deterministic.

From Table III, we can clearly see that ML-JMF generally outperforms the comparing methods on different datasets. Both RAkEL-GLAD and ML-JMF consider label correlation of multi-label samples while C-DS not. RAkEL-GLAD generally obtains a better Accuracy than C-DS, but they all lose to ML-JMF. This is because ML-JMF takes into account the quality variance of workers and reduces the impact of noisy annotations via matrix factorization. This result corroborates the fact that the quality of workers should be considered when aggregating crowdsourcing labels. ML-JMF(A) assigns different weights to $\mathbf{A}_w$; it outperforms MV, but always loses to ML-JMF. The achieved performance margin between ML-JMF and ML-JMF(A) provides support to the robustness of a low-rank matrix approximation, corroborating its use.

MV, PLAT, and AWMV convert the multi-label answer aggregation problem into multiple single-label problems. They ignore the correlation between labels; as such they are outperformed by ML-JMF and RAkEL-GLAD, which take advantage of label correlations. This comparison suggests that label correlations should be considered in multi-label answer aggregation. Although AWMV and PLAT are signal-label methods, AWMV achieves a better performance than PLAT. A possible reason is that AWMV assigns different weights to different types of labels. We also report the 1-RankingLoss of ML-JMF, RAkEL-GLAD, and C-DS in Table IV. We can see that ML-JMF generally has a larger 1-RankingLoss than these two comparing methods and ML-JMF(A).

In summary, these experimental results not only prove the effectiveness of ML-JMF in aggregating labels of multi-label samples in crowdsourcing, but also confirms that both label correlation and quality of workers should be considered in fusing crowdsourcing labels.

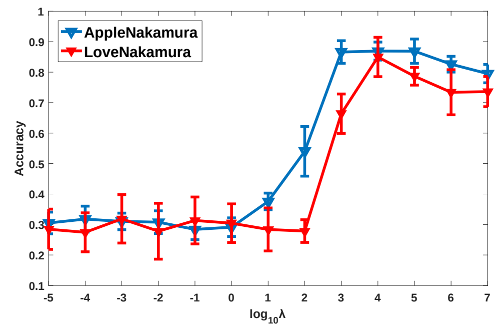

Figure 1: Accuracy of ML-JMF under different input values of λ.

B. Component Analysis of $\boldsymbol{\mu}$

To account for the quality variance of different workers, ML-JMF attaches a weight $\boldsymbol{\mu}_w$ to the w-th worker, which is expected to be small for a low-quality worker and large for a high-quality one. From the explicit solution of $\boldsymbol{\mu}$ in Eq. (22), we can see that once λ is specified, the weight assigned to $\boldsymbol{\mu}_w$ can be derived from the approximation loss of $\mathbf{A}_w$. To find a feasible value of λ, we vary λ in $\{10^{-5}, 10^{-4}, \cdots, 10^{6}, 10^{7}\}$. Furthermore, to investigate the capability of ML-JMF of identifying spammers, we additionally append 20 spam workers, who randomly select a label for all the samples of AppleEkman and AppleNakamura datasets. The Accuracy of ML-JMF under each value of λ is revealed in Figure 1. In practice, we also separately investigated the 20 spammers who assign a random label to each sample, and the 20 spammers who randomly assign the average number of annotations of all workers to samples of the dataset. These two investigations give the similar results as revealed in Figure 1.

ML-JMF has the highest Accuracy when $\lambda \approx 10^4$, the lowest Accuracy when $\lambda < 10$, and gradually reduced Accuracy when $\lambda \geq 10^5$. To further investigate these results, we take the *AppleEkman* dataset, and report the weights ($\boldsymbol{\mu}_w$) assigned to all the annotation matrices when $\lambda = 10$, $\lambda = 10^4$, and $\lambda = 10^5$ in Figure 2. We have several interesting observations. (i) When $\lambda = 10$, only a very small portion of annotation matrices are selected; when $\lambda = 10^5$, all annotation matrices are selected and assigned nearly equal weights. This is expected from Eq. (6); a (too) small λ value does not have a sufficient regularization effect on the weights assigned to individual matrices, and thus only few data matrices are selected. On the other hand, a (too) large λ value inflicts a strong regularization effect and forces similar weight assignments to all matrices. (ii) Since complementary information is spread across the annotation matrices of different workers, ML-JMF with $\lambda = 10^5$ and with $\lambda = 10^4$ obtains a significantly better

Table III: Accuracy of ML-JMF and comparing methods

	Movie	LoveNakamura	LoveEkman	AppleNakamura	AppleEkman
MV	0.9275	0.8726	0.8697	0.8510	0.8622
PLAT	0.8968	0.8839	0.8818	0.8662	0.8868
AWMV	0.9326	0.9126	0.8857	0.8703	0.8953
RAkEL-GLAD	0.9430	0.9363	0.9202	0.9317	0.9295
C-DS	0.9423	0.9267	0.9023	0.9276	0.9363
ML-JMF(A)	0.9317±0.0017	0.8396±0.0021	0.8427±0.0000	0.8313±0.0124	0.8617±0.0031
ML-JMF	**0.9458 ± 0.0028**	**0.9505 ± 0.0127**	**0.9235 ± 0.0121**	**0.9530 ± 0.0208**	**0.9513 ± 0.0228**

Table IV: 1-RankingLoss of ML-JMF and comparing methods

	Movie	LoveNakamura	LoveEkman	AppleNakamura	AppleEkman
RAkEL-GLAD	0.9978	**0.9911**	0.7249	0.9681	0.9701
C-DS	0.9932	0.9729	0.8174	0.9694	0.9687
ML-JMF(A)	0.9879 ± 0.0000	0.9519 ± 0.0000	0.9127 ± 0.0017	0.9656 ± 0.0001	0.9577 ± 0.0000
ML-JMF	**0.9979 ± 0.0000**	0.9791 ± 0.0017	**0.9358 ± 0.0000**	**0.9703 ± 0.0013**	**0.9716 ± 0.0000**

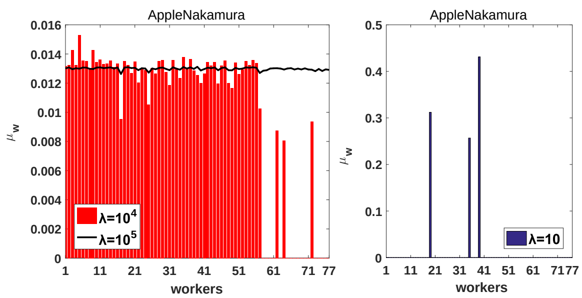

Figure 2: Weights assigned to 77 (57 workers + 20 spammers) annotation matrices of AppleNakamura dataset.

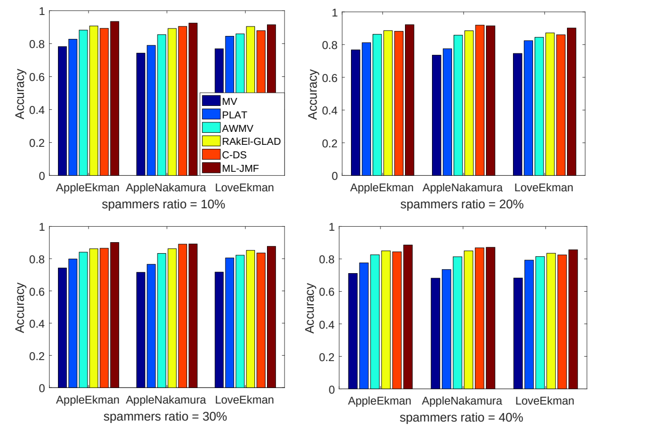

Figure 3: Accuracy under different ratios of spammers on AppleEkamn, AppleNakamura and LoveEkman.

performance than with $\lambda = 10$. (iii) Even if ML-JMF with $\lambda = 10^5$ combines the annotations of 20 spammers, it still obtains a better performance than ML-JMF with $\lambda = 10$; this is because the low-rank matrix approximation can eliminate the noise of annotation matrices. For a similar reason, ML-JMF with $\lambda = 10^4$ occasionally does not assign zero weights to several spammers.

In summary, these experimental results corroborate the fact that ML-JMF can identify spammers and can selectively integrate different annotation matrices via joint matrix factorization. Based on these experimental results, we adopt $\lambda = 10^4$ for the experiments.

C. Robustness to Spammers

Spammers always exist in crowdsourcing platforms. Previous studies show that the proportion of spammers could be up to 40% [3], [46]. As a result, it is important to investigate how each aggregation technique performs when the workers are not trustworthy. For this investigation, we artificially injected $\{10\%, 20\%, 30\%, 40\%\}$ spammers into the worker population and report the performance of the comparing methods under different ratios of spammers in Figure 3. Here, each spammer randomly selects a label from the label space, and then assigns the chosen label to all the samples of the dataset.

As the ratio of spammers increases, all the aggregation methods have reduced Accuracy. This pattern is expected, since more spammers bring in more noisy annotations, which may even surpass the correct ones and make the aggregation task more difficult. ML-JMF generally outperforms the other five methods, especially on the *AppleEkman* dataset. MV is the most sensitive to spammers, since it assumes that all workers (including spammers) provide answers of equal quality, and ignores label correlation. When adding 40% spammers, ML-JMF can still hold an accuracy $> 80\%$ and is more robust to spammers than the other comparing methods. This advantage can be attributed to three factors: (i) unlike existing aggregation approaches, ML-JMF makes use of the robustness of low-rank matrix factorization to reduce the impact of noisy annotations of respective workers (including spammers); (ii) ML-JMF can jointly and selectively integrate the annotation matrices of workers, and can explicitly reduce the impact of spammers by assigning lower (or zero) weights to spammers; (iii) ML-JMF explicitly uses label correlation and worker profile similarity to optimize the weights and the approximation of the respective annotation matrices. For these reasons, RAkEL-GLAD has a lower Accuracy than ML-JMF, although it also takes into account label correlation.

D. Parameter sensitivity analysis

The four parameters α, β, λ, and the rank k of $\mathbf{S}$ may affect the performance of ML-JMF. We conduct additional experiments to study the sensitivity of ML-JMF with respect to α, β, and k. The sensitivity of λ was studied in Subsection V-B (see Figure 1 and Figure 2).

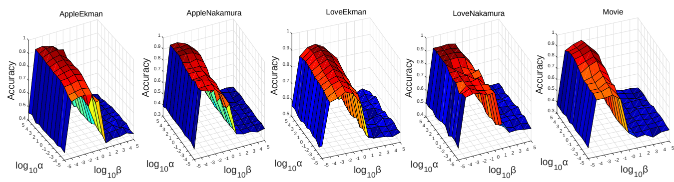

Figure 4: Accuracy of ML-JMF under different combinations of α and β.

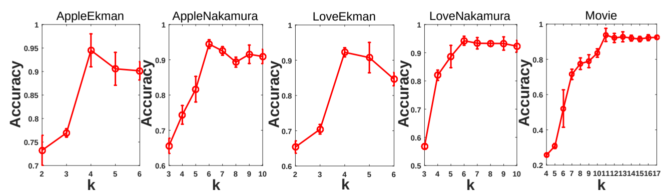

Figure 5: Accuracy of ML-JMF under different low-rank sizes (k).

Figure 4 shows the results of ML-JMF under different combinations of α and β with k set to $\lceil c/2 \rceil + 1$. Under a fixed α, ML-JMF with $\beta \in [10^{-3}, 10]$ has a superior performance than ML-JMF with β in other intervals. This is because a too small value of β underweights the contribution of profile similarity of workers, and a too large value of β overweights the profile similarity of workers. In fact, the annotation matrix $\mathbf{A}_w$ is very sparse. To save costs, the repeated labels of the samples are still very scarce, and thus the user profile matrix $\mathbf{U}_w$ is sparse and share low similarity with other workers. For this reason, the setting of β to a large value drags down the performance of ML-JMF. With a fixed β, ML-JMF achieves a relatively stable performance when $\alpha > 10^2$ and has reduced performance when $\alpha \in [10^{-5}, 10^2]$. This is because a small α does not make sufficient usage of label correlation, which can often boost the performance of multi-label learning. We can conclude that both label correlation and profile similarity of workers can boost the accuracy of multi-label answer aggregation. Based on the above analysis, we set $\alpha = 10^4$ and $\beta = 0.01$ for experiments.

Figure 5 shows the results for ML-JMF under different values of k. As k increases, the performance of ML-JMF increases at first, and then turns to be stable or decreased when $k > \lceil c/2 \rceil + 1$. ML-JMF with $k \approx \lceil c/2 \rceil + 1$ often holds comparable (or better) performance to MJ-JMF with $k = c$. This fact suggests that it's feasible to encode c labels via a low-rank ($k < c$) matrix. Based on the above analysis, we set $k = \lceil c/2 \rceil + 1$ for the experiments.

E. Complexity analysis

We plot the overall loss of $\Phi(\mathbf{U}_w, \mathbf{S}, \mathbf{V}, \boldsymbol{\mu})$ in each iteration (see Eq. (6)) for the AppleEkman dataset in Figure 6. We can clearly see that ML-JMF converges quickly in no more than 20 iterations. The overall loss in each iteration on the other datasets also provide similar results.

ML-JMF and the other comparing methods (whose code was provided by the authors) are implemented using different

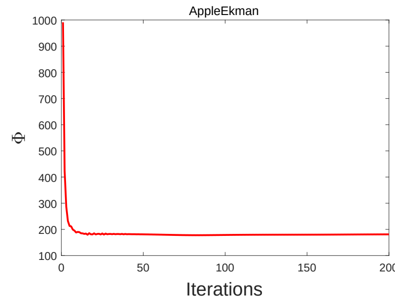

Figure 6: Convergence curve of ML-JMF on the AppleEkman dataset.

languages; as such, it's not meaningful to compare their empirical runtime costs. Therefore, we give the theoretical computational complexity of three multi-label answer aggregation approaches (ML-JMF, RAkEL-GLAD, and C-DS). RAkEL-GLAD takes $O(mnc)$ to create a power set of each label and $O(mnM2^k)$ (where k is the number of labels in a label subset, and M is the number of random label subsets) to calculate the average likelihood of each label, so its complexity is $O(mnc+mnM2^k)$. The computational complexity of C-DS is $O(mnc^2+mc^3+c^3)$. C-DS takes $O(mnc^2+mc^3)$ to compute the joint distribution over the source label vectors and target labels, and $O(c^3)$ to compute the probability of each label for each sample. ML-JMF takes $O(mnk^2)$, $O(nck)$, $O(mnck)$ to iteratively update the low-rank matrices $\mathbf{V}, \mathbf{U}_w, \mathbf{S}$, respectively, and $O(tm)$ to update $\boldsymbol{\mu}$. Thus, the computational complexity of ML-JMF is $O(tmnk^2 + tmnck + tm)$, where t is the number of iterations. The three single-label answer aggregation methods (MV, PLAT, and AWMV) have a lower complexity than multi-label methods, since they separately aggregate answers for each label. Since $k < \{c, n\}$, ML-JMF has a lower complexity than RAkEL-GLAD and C-DS. The code of ML-JMF will be made publicly available.

V. Conclusion

This paper studies how to aggregate the labels of multi-label samples collected via crowdsourcing, and introduces a Multi-Label answer aggregation approach based on Joint Matrix Factorization (ML-JMF). ML-JMF jointly factorizes the sample-label association matrices obtained from different workers into the product of individual low-rank matrices and a shared low-rank matrix, and selectively integrates them by assigning different weights to their answer data matrices. It further integrates the correlation between labels based on the shared matrix and connections between workers by individual matrices to guide the matrix factorization and weights. Experimental results on five real-world datasets show that ML-JMF can identify spammers and achieve higher accuracy than related methods. Our study suggests that both label correlation and quality of workers should be considered in aggregating the labels of multi-label samples. The code of ML-JMF is available at http://mlda.swu.edu.cn/codes.php?name=MLJMF.

Like existing solutions, ML-JMF currently depends on the input parameters; how to reduce the number of input parameters, and how to automatically determine their optimal values are future issues to be pursued.

Acknowledgments

We thank the authors who kindly shared their source code and datasets with us for the experiments, and the reviewers for their comments on improving this paper. This research is supported by NSFC (61872300, 61741217, 61873214 and 61871020). Natural Science Foundation of CQ CSTC (cstc2018jcyjAX0228, cstc2016jcyjA0351 and CSTC2016SHMSZX0824), the Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP-2017A05).

References

[1] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[2] G. Kazai, J. Kamps, M. Koolen, and N. Milic-Frayling, "Crowdsourcing for book search evaluation: Impact of hit design on comparative system ranking," in *SIGIR*, 2011, pp. 205–214.

[3] J. Vuurens, A. P. de Vries, and C. Eickhoff, "How much spam can you take? an analysis of crowdsourcing results to increase accuracy," in *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*, 2011, pp. 21–26.

[4] G. Kazai, J. Kamps, and N. Milic-Frayling, "The face of quality in crowdsourcing relevance labels: Demographics, personality and labeling accuracy," in *CIKM*, 2012, pp. 2583–2586.

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

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

[7] M. Venanzi, J. Guiver, P. Kohli, and N. R. Jennings, "Time-sensitive bayesian information aggregation for crowdsourcing systems," *JAIR*, vol. 56, pp. 517–545, 2016.

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

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

[10] Q. Abbas, M. E. Celebi, C. Serrano, I. F. GarcíA, and G. Ma, "Pattern classification of dermoscopy images: A perceptually uniform model," *Pattern Recognition*, vol. 46, no. 1, pp. 86–97, 2013.

[11] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu, "Corleone: Hands-off crowdsourcing for entity matching," in *SIGMOD*, 2014, pp. 601–612.

[12] N. Q. V. Hung, H. H. Viet, N. T. Tam, M. Weidlich, H. Yin, and X. Zhou, "Computing crowd consensus with partial agreement," *TKDE*, vol. 30, no. 1, pp. 1–14, 2018.

[13] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *TKDE*, vol. 26, no. 8, pp. 1819–1837, 2014.

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

[15] L. Duan, S. Oyama, M. Kurihara, and H. Sato, "Crowdsourced semantic matching of multi-label annotations," in *IJCAI*, 2015, pp. 3483–3489.

[16] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[17] D. Meng and F. De La Torre, "Robust matrix factorization with unknown noise," in *ICCV*, 2013, pp. 1337–1344.

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

[19] K. Yoshimura, Y. Baba, and H. Kashima, "Quality control for crowdsourced multi-label classification using rakel," in *ICONIP*, 2017, pp. 64–73.

[20] J. Zhang, V. S. Sheng, Q. Li, J. Wu, and X. Wu, "Consensus algorithms for biased labeling in crowdsourcing," *Information Sciences*, vol. 382, pp. 254–273, 2017.

[21] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "Crowddb: answering queries with crowdsourcing," in *SIGMOD*, 2011, pp. 61–72.

[22] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: a survey," *AIR*, vol. 46, no. 4, pp. 543–576, 2016.

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

[24] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *JMLR*, vol. 11, no. 2, pp. 1297–1322, 2010.

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

[26] J. Zhang, X. Wu, and V. S. Sheng, "Imbalanced multiple noisy labeling," *TKDE*, vol. 27, no. 2, pp. 489–503, 2015.

[27] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *JMLR*, vol. 13, no. 1, pp. 491–518, 2012.

[28] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal crowdsourcing using low-rank matrix approximations," in *49th Annual Allerton Conference on Communication, Control, and Computing*, 2011, pp. 284–291.

[29] C. J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets," in *AAAI*, 2012, pp. 45–51.

[30] W. Wang, X.-Y. Guo, S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Obtaining high-quality label by distinguishing between easy and hard items in crowdsourcing," in *IJCAI*, 2017, pp. 2964–2970.

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

[32] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *TKDE*, vol. 23, no. 7, pp. 1079–1089, 2011.

[33] P. G. Moreno, A. Artés-Rodríguez, Y. W. Teh, and F. Perez-Cruz, "Bayesian nonparametric crowdsourcing," *JMLR*, 2015.

[34] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *WISE*, 2013, pp. 1–15.

[35] G. Kazai, J. Kamps, and N. Milic-Frayling, "Worker types and personality traits in crowdsourcing relevance labels," in *CIKM*, 2011, pp. 1941–1944.

[36] K. Konstantinides, B. Natarajan, and G. S. Yovanof, "Noise estimation and filtering using block-based singular value decomposition," *TIP*, vol. 6, no. 3, pp. 479–483, 1997.

[37] M. L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *KDD*, 2010, pp. 999–1008.

[38] A. K. Smilde, H. A. Kiers, S. Bijlsma, C. M. Rubingh, and M. J. van Erk, "Matrix correlations for high-dimensional data: the modified rv-coefficient." *Bioinformatics*, vol. 25, no. 3, pp. 401–405, 2009.

[39] A. K. Smilde, M. J. van der Werf, S. Bijlsma, B. J. van der Werff-van der Vat, and R. H. Jellema, "Fusion of mass spectrometry-based metabolomics data," *Analytical Chemistry*, vol. 77, no. 20, pp. 6729–6736, 2005.

[40] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[41] M. Zitnik and B. Zupan, "Data fusion by matrix factorization," *TPAMI*, vol. 37, no. 1, pp. 41–53, 2015.

[42] L. Duan, S. Oyama, H. Sato, and M. Kurihara, "Separate or joint? estimation of multiple labels from crowdsourced annotations," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5723–5732, 2014.

[43] P. Ekman, "An argument for basic emotions." *Cognition & Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[44] A. Nakamura, "Kanjo hyogen jiten [dictionary of emotive expressions]," *Tokyodo*, 1993.

[45] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *PAKDD*, 2004, pp. 22–30.

[46] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms." in *CrowdSearch*, 2012, pp. 26–30.