Multi-Label Learning from Crowds

Shao-Yuan Li, Yuan Jiang, Nitesh V. Chawla, and Zhi-Hua Zhou, *Fellow, IEEE,*

Abstract—We consider multi-label crowdsourcing learning in two scenarios. In the first scenario, we aim at inferring instances' groundtruth given the crowds' annotations. We propose two approaches NAM/RAM (Neighborhood/Relevance Aware Multi-label crowdsourcing) modeling the crowds' expertise and label correlations from different perspectives. Extended from single-label crowdsourcing methods, NAM models the crowds' expertise on individual labels, but based on the idea that for rational workers, their annotations for instances similar in the feature space should also be similar, NAM utilizes information from the feature space and incorporates the local influence of neighborhoods' annotations. Noting that the crowds tend to act in an effort-saving manner while labeling multiple labels, i.e., rather than carefully annotating every proper label, they would prefer scanning and tagging a few most relevant labels, RAM models the crowds' expertise as their ability to distinguish the relevance between label pairs. In the second scenario, we care about cost-efficient crowdsourcing where the labeling and learning process are conducted in tandem. We extend NAM/RAM to the active paradigm and propose instance, label and worker selection criteria such that the labeling cost is significantly saved compared to passive learning without labeling control. The proposals' effectiveness are validated on both simulated and real datasets.

Index Terms—multi-label, crowdsourcing, label correlation, labeling cost, active selection

1 Introduction

In many real world tasks, one example can be associated with multiple concepts, e.g., in image annotation, one image may be tagged with terms such as *urban* and *road* [1]; in bioinformatics, the gene sequence may have multiple functions such as *metabolism*, *transcription* and *protein synthesis* [2]. To deal with such tasks, multi-label learning has received significant attention [3], [4].

Conventional multi-label learning assumes that the groundtruth labels are given for the training data, which are expensive and limited resources. On the one hand, labeling groundtruth requires experts' careful examination over all candidate labels, which is expensive; on the other hand, the availability of experts and labeling budgets can be limited. Rather than resorting to experts for groundtruth, crowdsourcing [5], [6] provides an alternative to collect labels from easy to access and low cost crowds. To alleviate the labeling errors made by crowds, the common wisdom is to distribute the task to multiple workers and estimate higher quality labels using aggregation.

Previous crowdsourcing learning are mostly on single-label tasks in fields such as sentiment classification, medical diagnosis and image tagging [7], [8], [9], whereas using crowds for multi-label tasks is in the primary stage. [10], [11], [12], [13] considered inferring the taxonomy structure of multiple labels, and [11], [14], [15] considered estimating groundtruth labels from crowds' annotations.

In this paper, we consider multi-label crowdsourcing learning in two scenarios. In the first scenario, we concern the annotation collection mode adopted from [11], [14], [15] where given a set of instances, a group of the workers are employed to tag the *proper* labels from a set of candidate labels for the instances they see. After the labeling, the learning is conducted and we wish to obtain an effective classifier and estimate the instances' groundtruth labels. In the second scenario, we also care about labeling cost. We conduct the labeling process and learning in tandem, and wish to learn the classifier with least annotations through active annotation collection.

Treating the *tagged* and *untagged* labels respectively as *positive* and *negative* label annotations, [11], [15] and [14] concerned themselves on extending single-label crowdsourcing methods by considering the label co-occurrence and conditional probabilistic label relationships. The issue is that their label correlations are solicited solely from the noisy annotations, whose qualities are thus affected sensitively by the annotations' quantity and quality, which would even be harmful in case of misleading annotations. More importantly, none of them have noticed the different characteristic of crowds' labeling on multi-label tasks.

For better understanding of the crowds' labeling behavior, we make a comparison between labeling from crowds for multi-label tasks, single-label tasks, and groundtruth labeling. Either for single-label or multi-label tasks, in groundtruth labeling from perfect experts, the untagged labels definitely mean negative labels. For crowds' labeling on single-label tasks, since one proper label tagging is sufficient, thus the untagged labels definitely mean negative annotations. But for multi-label labeling where each instance can be associated with multiple proper labels, this clear distinction is not necessarily true. We observe that while annotating multiple labels, rather than carefully annotating every proper label, the crowds would prefer scanning and tagging a few most relevant labels from their point of view and leave the rest untouched. This may be due to the heavy workload of examining every label, or they just annotate labels they are confident about. We name this as *effort-saving* annotating behavior. In such case, the untagged labels may

- *Shao-Yuan Li, Yuan Jiang and Zhi-Hua Zhou are with the National Key Laboratory for Novel Software Technology, Nanjing Univerisity, Nanjing 210023, China.*
  E-mail: {lisy,jiangy,zhouzh}@lamda.nju.edu.cn
- *Netish V. Chawla is with Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556.*
  E-mail: nchawla@nd.edu

mean uncertain labels or just be untouched. Simply treating them as negative like would be misleading.

Here we give an example. While collecting our experimental data *dataset1* and *dataset2*, we note that the workers rarely annotate more than 3 labels for the images they see, which is surely insufficient for images with a rather larger number of labels. As a validation, we calculate the average number of tagged labels for the images with no less than 3 labels (comprising about 10%, 19% of the data), and obtain 2.1 and 2.5 for *dataset1* and *dataset2*. Such characteristic was not noticed by previous crowdsourcing works.

In this paper, we propose two different approaches NAM (Neighborhood Aware Multi-label crowdsourcing) and RAM (Relevance Aware Multi-label crowdsourcing) for multi-label crowdsourcing learning. Similar as [11], [14], [15], NAM treats the *tagged* and *untagged* labels respectively as *positive* and *negative* annotations, and extends single-label methods considering label correlation exploration. But to alleviate the high sensitivity of label correlations to annotations' noise, based on the idea that instances similar in the feature space should also get similar annotations, NAM incorporates the local influence of neighborhoods' annotations.

Differently, emphasizing on modeling the crowds' *effort-saving* annotating behavior, RAM treats the tagged labels as more relevant than the untagged labels, and models each worker's expertise as its ability to distinguish the correct relevance between label pairs, which also naturally captures the relevance comparison relationship between labels.

Our experiments show that both methods perform good. Ignoring the crowds' *effort-saving* annotating behavior, NAM's none degenerated performance should be due to the label sparsity of the experimental data, i.e., the fraction of examples associated with rather large number of labels is rather small. But for applications with large label density, the *effort-saving* manner should not be ignored. For performance improvement, we can even combine the strongness of NAM's label correlations and RAM's *effort-saving* behavior modeling.

We also extend NAM and RAM to the active paradigm and conduct adaptive selection over instances, labels and workers during the labeling process, such that the most reliable workers are queried for the most valuable instances and labels. Based on the prediction of NAM and RAM, we design criteria to evaluate and select instances considering the prediction uncertainty/query diversity, labels considering the prediction uncertainty/probability of being positive, and workers with high expertise.

Note that in this paper, we assume that most workers are acting with good will, i.e., they are willing to provide good annotations, and we do not pay special attention to adversarial crowds. This is the base assumption in crowdsourcing such that the problems are learnable. In the following we start with a brief review of some related work in Section 2, then propose our approaches in Section 3, and report the experiments in Section 4. Finally, we conclude the paper.

2 Related Work

In this section, we briefly review some related work in multi-label learning and crowdsourcing learning.

Multi-label Learning During the past decade, various approaches have been proposed to deal with multi-label learning tasks [3], [4]. The most straightforward strategy is to decompose the multi-label task into a series of binary classification problems, one for each label and thus single-label learning methods can be applied [16]. This strategy neglects considering the explicit or implicit label relationship, which is widely believed to contain important information to help learning. The observation in text categorization that the labels are organized in a hierarchical structure has motivated approaches to exploit external label structures. For example, [17] considered the tree structure of labels; [18] considered the tree and dag-structure of labels. Besides using explicit label structure information, implicit label relationship were also explored, such as the label ranking idea which transforms the multi-label classification problem into a label ranking problem, and predicts the positive labels in front of the negative labels [2], [19]; and the feature space enhancement idea which constructs meta-level features using label information [20], [21]. Typical multi-label learning requires the groundtruth labels to be given for the training examples, which are expensive resources in real applications.

When the crowds are acting in the *effort-saving* manner, the annotations from crowds can be kind of incomplete. Another possible related field is partial label learning, where a partial set of groundtruth labels are given for the training examples. The target is to recover the complete groundtruth exploiting the benefits of instance-label correlation and label relationship, such as label propagation based on manifold assumption [22], [23], label ranking based on group lasso [24] and label completion based on low rank structure [25], [26]. Regarding the annotations from all crowds form a low rank matrix, partial label learning can be first applied to recover the unknown annotations of crowds, and then majority voting can be applied to predict the groundtruth. Though partial label learning successfully accounts in the instance-label correlation and label relationship, they treat the annotations equally without considering the crowds' expertise variance.

Crowdsourcing Learning With the advent of crowdsourcing platforms such as Amazon Mechanical Turk (AMT), crowdsourcing has been an economic way to collect supervised information. One main focus is to aggregate the imperfect annotations from the crowds to infer groundtruth labels. Previous works mostly focus on single-label tasks, modeling the crowds' expertise from different perspectives using measures with explicit explanations such as classification accuracy [27], [28], confusion matrix [8], [29], [30], [31], and more complex multidimensional vectors [9].

Recently, building hierarchies of labels [10], [11], [12], [13] and inferring groundtruth label from crowds [11], [14], [15] for multi-label tasks were also explored. [10], [11] collected annotations for items and deployed the annotation co-occurrence to infer the hierarchy structure; [12], [13] queried crowds the 'ascendant-descendant' relationship between two labels to reconstruct the label hierarchy. To infer the groundtruth labels, works extending the single-label crowdsourcing methods by taking into label correlations were also studied. [11], [15] incorporated the label co-occurrence dependencies between labels; [14] considered three dependency relationships among all label power set,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

label set of two labels, and label dependency considering conditional independence. Computed solely from the annotations, the quality of the label correlation relies heavily on the availability and reliability of the crowds. Besides, they also ignore that the crowds' annotating behavior on multi-label tasks can be different from that on single-label tasks. To alleviate this, our approach exploit the rich feature information for label correlations calculating, and consider the crowds' specific annotating behavior.

Considering that the labeling budget is often limited in real applications, to reduce the crowdsourcing labeling cost, works on selectively query and learn from the most valuable annotations were also explored, either for single-label tasks [32], [33], [34] or multi-label tasks [11], [35]. [32] exploited the active learning paradigm by actively querying annotations from the most reliable workers for the most uncertain items, where the crowds' reliability is defined as their labeling accuracy and the tasks' uncertainty is defined as their label prediction entropy. [33] extended [32] to the worker scarcity case through transferring knowledge from auxiliary domains. [34] dynamically sampled subsets of the crowds based on an exploration/exploitation criterion to approximate the majority opinion of all crowds. Designed for single-label tasks, the above works ignore the label correlations and crowds' specific annotating behavior. While learning the hierarchy structure for large numbers of labels, to reduce the labeling cost, [11] also explored the assignment of labels to data items using the label prediction entropy measures, but did not pay attention to instance and worker selection. Our previous work [35] extended the idea of [32] to multi-label tasks by incorporating the local neighborhoods' label correlation for *classifier* learning and improved the instance selection strategy. In this paper, we extend [35] by 1) improving the crowdsourcing learning method by also considering label correlations for *crowds* modeling; 2) proposing another crowdsourcing learning model RAM considering the crowds' specific annotating behavior; and 3) designing respective active selection strategies for the two methods considering the characteristics of the two methods and the special property of multi-label tasks.

3 SCENARIO 1: CROWDS AGGREGATION

3.1 Problem Formulation

We use $D = \{(\mathbf{x}_1, \{\mathbf{y}_{1j}\}), \ldots, (\mathbf{x}_N, \{\mathbf{y}_{Nj}\})\}$ to denote the set of N instances annotated by M annotators, where $\mathbf{x}_i$ is the d-dimensional feature vector representation for instance i, $\mathbf{y}_{ij}$ is the annotation results of instance i given by worker j. M^i, M_i^l are used to denote the annotator set labeling instance i and annotating label l for instance i, N^j to denote the instance set annotated by annotator j. Our target is to estimate the groundtruth labels $Z = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N) \in \{+1, -1\}^{L \times N}$ for X given D.

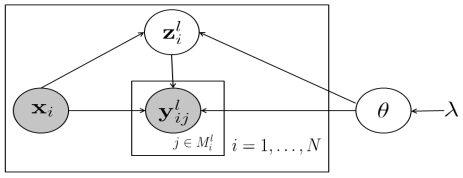
To use one unified general representation for the annotations such that the same notation can be used for the first scenario and active crowdsourcing, we define $\mathbf{y}_{ij} \in \{+1, -1, 0\}^{L \times 1}$. In the active crowdsourcing scenario where labels are selected to query annotation, $\mathbf{y}_{ij}^l = 1(-1)$ means label l is tagged as positive (negative) by worker j for $\mathbf{x}_i$, and $\mathbf{y}_{ij}^l = 0$ means label l is not queried for $\mathbf{x}_i$ from worker j . For the first learning scenario where the annotations are pre-collected through querying the *proper* labels for each instance, the annotations reduce to $\mathbf{y}_{ij} \in \{+1, 0\}^{L \times 1}$ or $\mathbf{y}_{ij} = \{0\}^{L \times 1}$, with $\mathbf{y}_{ij}^l = 1(0)$ denoting worker j tags (doesn't tag) label l as *proper* for $\mathbf{x}_i$, and $\mathbf{y}_{ij} = \{0\}^{L \times 1}$ denoting worker j doesn't annotate instance i.

Compared with traditional multi-label learning, we can see that the key challenges lie in that the labels provided by each annotator can be noisy and the annotators' expertise can be various. Treating the untagged labels as negative label annotations, i.e., reformulating $\mathbf{y}_{ij}^l = 0$ as $\mathbf{y}_{ij}^l = -1$ for the tagged instances with $\mathbf{y}_{ij} \neq \{0\}^{L \times 1}$, previous multi-label crowdsourcing methods extends the single-label crowdsourcing methods by considering the label correlation, whose qualities however are affected sensitively by the quality of annotations. Besides, they also ignore the workers' specific annotating behavior on multi-label tasks.

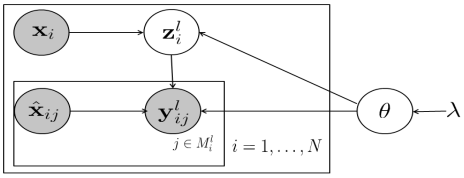
Considering the above, we propose two probabilistic approaches NAM/RAM(Neighborhood/Relevance Aware Multi-label crowdsourcing) modeling the crowds' expertise and utilizing the label correlations from two different perspectives. Specifically, like previous works, by reformulating $\mathbf{y}_{ij}^l = 0$ as $\mathbf{y}_{ij}^l = -1$ for instances with $\mathbf{y}_{ij} \neq \{0\}^{L \times 1}$, NAM extends the single-label crowdsourcing methods and models the crowds' expertise on each individual label, but incorporates the rich information in the feature space to consider the local influence of neighborhoods' label correlations. For RAM, it defines the crowds' expertise as their ability to distinguish the relevance between the tagged and untagged labels, which simultaneously captures the crowds' *effort-saving* annotating manner and the relevance ranking relationship between pairs of labels.

3.2 Method 1:NAM

We first describe one classic single label crowdsourcing model, then extend it by encoding the local influence of neighborhoods' label correlations to multi-label problem.



(a) Single Label Model



(b) Our Multi-Label Model

Fig. 1. (a) The single label probabilistic model for instances $\{\mathbf{x}_i\}$, annotations $\{\mathbf{y}_{ij}^l\}$ and unknown groundruth $\{\mathbf{z}_i^l\}$; (b) Our multi-label probabilistic model for instances $\{\mathbf{x}_i\}$, $\{\hat{\mathbf{x}}_{ij}\}$, annotations $\{\mathbf{y}_{ij}^l\}$ and unknown groundruth $\{\mathbf{z}_i^l\}$ on label l. Here $\hat{\mathbf{x}}_{ij}$ denotes the enhanced representation of $\mathbf{x}_i$ for worker j, which will be explained in Eq. 2.

Figure1(a) illustrates the classic probabilistic graphical model on some label l over the instances $\{\mathbf{x}_i\}$, the annotations $\{\mathbf{y}_{ij}^l\}$, and the unknown groundruth labels $\{\mathbf{z}_i^l\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

Assuming that each annotation $\mathbf{y}_{ij}^l$ depends both on the instance $\mathbf{x}_i$ and its groundtruth $\mathbf{z}_i^l$, using θ to denote the involved parameters and $p_r(\theta)$ its prior probability, the joint distribution of $\{\mathbf{y}_{ij}^l, \mathbf{z}_i^l\}$ can be represented as:

$$\begin{aligned} & P(\{\mathbf{y}_{ij}^l\}_{ij}, \{\mathbf{z}_i^l\}_i | \{\mathbf{x}_i\}_i, \theta) \\ = & \prod_i p(\mathbf{z}_i^l | \mathbf{x}_i, \theta) \prod_{j \in M_i^l} p(\mathbf{y}_{ij}^l | \mathbf{z}_i^l, \mathbf{x}_i, \theta) p_r(\theta). \end{aligned} \quad (1)$$

Here $p(\mathbf{z}_i^l|\mathbf{x}_i, \theta)$ can be regarded as the classifier of label $\mathbf{z}^l$. The second term $p(\mathbf{y}_{ij}^l|\mathbf{z}_i^l, \mathbf{x}_i, \theta)$ models worker j's labeling process of annotating $\mathbf{y}_{ij}^l$ to $\mathbf{x}_i$ with groundtruth $\mathbf{z}_i^l$. The unknown groundtruth $\{\mathbf{z}_i^l\}$ are used in this model as latent variables, whose probability distribution and the parameters θ will be iteratively estimated during learning. Defining one or more variables modeling the crowds' labeling expertise in the second term, this probabilistic model is widely adopted in crowdsourcing learning, e.g., crowds' expertise defined as labeling accuracy [27], [28], precision/recall/confusion matrix [8], [29], [30], [31], and complex multidimensional vectors [9].

Treating the multi-label task as multiple independent binary tasks, the above model and single-label methods can be directly applied. The problem is that the correlation between labels are ignored by such methods. We exploit the local influence of neighborhoods' annotations in our model in Figure1(b). Different from Figure1(a), *enhanced representation* of each instance $\hat{\mathbf{x}}_{ij}$ for each worker j is used,

$$\hat{\mathbf{x}}_{ij} = [\mathbf{x}_i, \mathbf{c}_{ij}], \quad \mathbf{c}_{ij} = \frac{1}{k} \sum_{\mathbf{x}_{i'} \in knn_j(\mathbf{x}_i)} \mathbf{y}_{i'j}, \quad (2)$$

i.e., $\hat{\mathbf{x}}_{ij}$ is the concatenation of $\mathbf{x}_i$ and the *local code* $\mathbf{c}_{ij}$ of instance i for worker j. Here $knn_j(\mathbf{x}_i)$ denotes the k nearest neighbors of x_i among the N^j instances annotated by worker j, $\mathbf{c}_{ij}$ is computed as the average mean annotations of its k nearest neighbors given by worker j.

Based on the idea that, from each worker j's perspective, its annotations to instances similar in the feature space are supposed to be similar, we add the neighborhoods' annotation information as extra features which may imply important information for the crowds labeling prediction. For implementation simplicity, we exploit the average mean statistic of neighbors' annotations in this paper. Other statistical values reflecting the annotation information of neighbors also worthy study, e.g., the median values.

Using the *enhanced representation* of instances for each worker, we then conduct the crowdsourcing learning on each label separately. NAM can be roughly regarded as a two step approach, in the first step, we preprocess the instance taking into account label correlations and construct enhanced instance representations, in the second step, the learning is conducted on each label separately using the new instance representations. On each label l, the probabilistic graphical model in Figure1(b) can be represented as:

$$\begin{aligned} & P(\{\mathbf{y}_{ij}^l\}_{ij}, \{\mathbf{z}_i^l\}_i | \{\mathbf{x}_i\}_i, \hat{\mathbf{x}}_{ij}, \theta) \\ = & \prod_i p(\mathbf{z}_i^l | \mathbf{x}_i, \theta) \prod_{j \in M_i^l} p(\mathbf{y}_{ij}^l | \mathbf{z}_i^l, \hat{\mathbf{x}}_{ij}, \theta) p_r(\theta). \end{aligned} \quad (3)$$

For implementation simplicity, we utilize labeling accuracy as [27], [28] to model the crowds' expertise. For each annotator j, we define an expertise variable $\mathbf{e}_{ij}^l$ as the probability that annotator j provides the correct label for instance i, i.e.,

$$\mathbf{e}_{ij}^l := p(\mathbf{y}_{ij}^l = \mathbf{z}_i^l | \hat{\mathbf{x}}_{ij}, \theta), \quad (4)$$

Here the groundtruth $\mathbf{z}_i^l$ are unknown latent variables and will be inferred in the learning. Using $I(\cdot)$ to denote the indicator function, then the distribution $p(\mathbf{y}_{ij}^l|\mathbf{z}_i^l, \hat{\mathbf{x}}_{ij}, \theta)$ can be formulated as the following Bernoulli distribution,

$$p(\mathbf{y}_{ij}^l|\mathbf{z}_i^l, \hat{\mathbf{x}}_{ij}, \theta) = (1 - \mathbf{e}_{ij}^l)^{[1-I(\mathbf{y}_{ij}^l = \mathbf{z}_i^l)]} (\mathbf{e}_{ij}^l)^{[I(\mathbf{y}_{ij}^l = \mathbf{z}_i^l)]}. \quad (5)$$

Exploiting the *logistic sigmoid* $\sigma(z) = 1/(1+exp(-z))$ acting on some function $f_{j,\theta}^l$ and $f_{0,\theta}^l$ over instances to model $\mathbf{e}_{ij}^l$ and $p(\mathbf{z}_i^l|\mathbf{x}_i, \theta)$, we get

$$\mathbf{e}_{ij}^l := p(\mathbf{y}_{ij}^l = \mathbf{z}_i^l | \hat{\mathbf{x}}_{ij}, \theta) = \sigma(f_{j,\theta}^l(\hat{\mathbf{x}}_{ij})), \quad (6)$$

$$p(\mathbf{z}_i^l = 1 | \mathbf{x}_i, \theta) = \sigma(f_{0,\theta}^l(\mathbf{x}_i)),$$

$$p(\mathbf{z}_i^l = -1 | \mathbf{x}_i, \theta) = \sigma(-f_{0,\theta}^l(\mathbf{x}_i)). \quad (7)$$

While any function can be used to implement $f_{0,\theta}^l$ and $f_{j,\theta}^l$, for ease of exposition, we consider the linear discriminating functions $f_{0,\theta}^l(\mathbf{x}_i) = (\mathbf{w}_0^l)' \mathbf{x}_i$, $f_{j,\theta}^l(\hat{\mathbf{x}}_{ij}) = (\mathbf{w}_j^l)' \hat{\mathbf{x}}_{ij}$. Given the above specifications, the parameters become the classifier/annotator parameters $\theta = \{\mathbf{w}_0^l, \{\mathbf{w}_j^l\}\}$. To overcome overfitting, we further introduce a zero-mean λ-variance Gaussian prior for $\{\mathbf{w}_0^l\}$ and $\{\mathbf{w}_j^l\}$, i.e.,

$$\begin{aligned} & p_r(\theta) = p(\mathbf{w}_0^l | \lambda) \prod_{j \in M_i^l} p(\mathbf{w}_j^l | \lambda) \\ & p(\mathbf{w}_0^l | \lambda), p(\mathbf{w}_j^l | \lambda) \propto \mathcal{N}(\mathbf{0}, \lambda^{-1} \mathbf{I}) \end{aligned} \quad (8)$$

To estimate the parameters $\{\mathbf{w}_0^l, \{\mathbf{w}_j^l\}\}$, we exploit the maximum likelihood criterion and use Expectation-Maximization (EM) [36] with missing variables $\{\mathbf{z}_i^l\}$ and observed variables $\{\mathbf{y}_{ij}^l\}_{ij}$.

E-step: Given current estimation of the parameters $\theta = \{\mathbf{w}_0^l, \{\mathbf{w}_j^l\}\}$ from last M step, the posterior probability of ground truth label $\{\mathbf{z}_i^l\}$ is computed:

$$\begin{aligned} p(\mathbf{z}_i^l) & = p(\mathbf{z}_i^l | \mathbf{x}_i, \hat{\mathbf{x}}_{ij}, \{\mathbf{y}_{ij}^l\}_{ij}, \mathbf{w}_0^l, \{\mathbf{w}_j^l\}) \\ & \propto p(\mathbf{z}_i^l | \mathbf{x}_i, \mathbf{w}_0^l) \prod_{j \in M_i^l} p(\mathbf{y}_{ij}^l | \mathbf{z}_i^l, \hat{\mathbf{x}}_{ij}, \mathbf{w}_j^l). \end{aligned} \quad (9)$$

Substituting Eq.5 and Eq.7 into Eq.9, E-step reduces to:

$$p(\mathbf{z}_i^l = 1) \propto \sigma({\mathbf{w}_0^l}' \mathbf{x}_i) \prod_{\mathbf{y}_{ij}^l \neq 0} \sigma(\mathbf{y}_{ij}^l {\mathbf{w}_j^l}' \hat{\mathbf{x}}_{ij}) \quad (10)$$

$$p(\mathbf{z}_i^l = -1) \propto \sigma(-{\mathbf{w}_0^l}' \mathbf{x}_i) \prod_{\mathbf{y}_{ij}^l \neq 0} \sigma(-\mathbf{y}_{ij}^l {\mathbf{w}_j^l}' \hat{\mathbf{x}}_{ij}) \quad (11)$$

M-step: To estimate the parameters $\theta = \{\mathbf{w}_0^l, \{\mathbf{w}_j^l\}\}$, we maximize the expectation of the joint log-likelihood $Q(\theta)$ of $(\{\mathbf{y}_{ij}^l\}_{ij}, \{\mathbf{z}_i^l\}_i)$ over θ, with respect to the posterior probabilities of $\{\mathbf{z}_i^l\}$ computed by last E step:

$$\begin{aligned} Q(\theta) & = E_{\mathbf{z}}[\ln P(\{\mathbf{y}_{ij}^l\}_{ij}, \{\mathbf{z}_i^l\}_i | \{\mathbf{x}_i\}_i, \{\hat{\mathbf{x}}_{ij}\}_{ij}, \mathbf{w}_0^l, \{\mathbf{w}_j^l\}) p_r(\theta)] \\ & = \sum_i E_{\mathbf{z}}[\ln p(\mathbf{z}_i^l | \mathbf{x}_i, \mathbf{w}_0^l) p(\mathbf{w}_0)] + \\ & \quad \sum_{ij} E_{\mathbf{z}}[\ln p(\mathbf{y}_{ij}^l | \mathbf{z}_i^l, \hat{\mathbf{x}}_{ij}, \mathbf{w}_j^l) p(\mathbf{w}_j^l)] \\ & = Q(\mathbf{w}_0^l) + \frac{\lambda}{2} \|\mathbf{w}_0^l\|^2 + \sum_j [Q(\mathbf{w}_j^l) + \frac{\lambda}{2} \|\mathbf{w}_j^l\|^2]. \end{aligned} \quad (12)$$

Algorithm 1 The NAM Approach

Input: data D; parameter neighbor size $k = 5$, $\lambda = 1e^{-3}$
Output: multi-label classifier $\{\mathbf{w}_0^l\}$, workers' parameters $\{\mathbf{w}_j^l\}$
Algorithm:

1: For each worker j, get the respective enhanced representation $\hat{\mathbf{x}}_{ij} = [\mathbf{x}_i, \mathbf{c}_{ij}]$ of each instance $\mathbf{x}_i$ by Eq. 2
2: **Repeat**
3: *E-step*: For each label l, given $\mathbf{w}_0^l$, $\{\mathbf{w}_j^l\}$ computed from last M-step, estimate the posterior probability $p(\mathbf{z}_i^l)$ by Eq. 10.
4: *M-step*: For each label l, given $p(\mathbf{z}_i^l)$ computed from last E-step, estimate $\mathbf{w}_0^l$, $\{\mathbf{w}_j^l\}$ by maximizing Eq. 13 and 14 using GA
5: **until** the maximum number of iterations is reached

Substituting Eq.5,7 into Eq.12, $Q(\mathbf{w}_0^l)$, $Q(\mathbf{w}_j^l)$ reduces to:

$$Q(\mathbf{w}_0^l) = \sum_i [{\mathbf{w}_0^l}' \mathbf{x}_i p(\mathbf{z}_i^l = 1) - \ln(1 + exp({\mathbf{w}_0^l}' \mathbf{x}_i))] \quad (13)$$

$$Q(\mathbf{w}_j^l) = \sum_i [\mathbf{y}_{ij}^l {\mathbf{w}_j^l}' \hat{\mathbf{x}}_{ij} p(\mathbf{z}_i^l = 1) - \ln(1 + exp(\mathbf{y}_{ij}^l {\mathbf{w}_j^l}' \hat{\mathbf{x}}_{ij}))] \quad (14)$$

Therefore $\mathbf{w}_0^l, \mathbf{w}_j^l$ can be independently optimized by maximizing $Q(\mathbf{w}_0^l)$, $Q(\mathbf{w}_j^l)$, for which we use gradient ascent. Algorithm 1 summarizes the overall process of NAM.

3.3 Method 2:RAM

Rather than treating the crowds' annotations in the same way as single-label tasks, i.e., taking the tagged/untagged labels as positive/negative annotations and modeling the crowds' behavior over each individual label, RAM emphasizes on modeling the crowds' *effort-saving* annotating behavior. From the point of label relevance view, RAM treats annotator j tagging label s but not tagging t to instance $\mathbf{x}$ as that, j thinks label s is more *relevant* than label t to $\mathbf{x}$. From this, each annotator's expertise is defined as its ability to differentiate the correct relevance between label pair (s, t).

Formally, for easiness of understanding, given annotator j and its annotations $\mathbf{y}_{ij} \in \{+1, -1, 0\}^{L \times 1}$ on instance $\mathbf{x}_i$, we construct the label relevance aware examples for it as:

$$\mathbf{xr}_{ij} = \{\mathbf{xr}_{ij}^{st}\} = \{(\mathbf{x}_i, s, t) | \mathbf{y}_{ij}^s = 1, \mathbf{y}_{ij}^t = -1/0\}. \quad (15)$$

Here $\mathbf{xr}_{ij}^{st}$ denotes the relevance aware example over label s and t that, annotator j believes on example $\mathbf{x}_i$ the positively tagged label s is more relevant than the negatively or untagged label t; and $\mathbf{xr}_{ij}$ denotes the set of relevance aware examples over the whole label set on example $\mathbf{x}_i$ in annotator j's opinion. The induced relevance aware examples for each annotator j and the whole data are:

$$\mathbf{Xr}_j = \{\mathbf{xr}_{ij} | i \in N^j\}, \quad \mathbf{XR} = \{\mathbf{Xr}_j\}. \quad (16)$$

Accordingly, using the unknown groundtruth labels $\mathbf{z}$ as latent variables, which will be inferred in the learning process, we define the groundtruth relevance aware positive and negative examples for each $\mathbf{x}_i$ as following:

$$\mathbf{xgp}_i = \{\mathbf{xgp}_i^{st}\} = \{(\mathbf{x}_i, s, t) | \mathbf{z}_i^s = 1, \mathbf{z}_i^t = -1\}, \quad (17)$$

$$\mathbf{xgn}_i = \{\mathbf{xgn}_i^{st}\} = \{(\mathbf{x}_i, s, t) | \mathbf{z}_i^s = -1, \mathbf{z}_i^t = 1\}. \quad (18)$$

$\mathbf{xgp}_i^{st}(\mathbf{xgn}_i^{st})$ denotes that on example $\mathbf{x}_i$, in groundtruth, the positive (negative) label s is more (less) relevant than the negative (positive) label t; and $\mathbf{xgp}_i(\mathbf{xgn}_i)$ denotes the set of groundtruth relevance aware positive (negative) examples over the whole label set on example $\mathbf{x}_i$. The whole groundtruth relevance aware examples are:

$$\mathbf{XGP} = \{\mathbf{xgp}_i\}, \quad \mathbf{XGN} = \{\mathbf{xgn}_i\}. \quad (19)$$

We can see the above examples actually represent the relevance comparison between one pair of labels on particular examples. Based on the above definition, each annotator j's expertise q_{ij}^{st} on example $\mathbf{x}_i$ over label s and t is defined as how annotator j's relevance comparison between label s and t agrees with the groundtruth:

$$\begin{aligned} q_{ij}^{st} &:= P((\mathbf{x}_i, s, t) \in \mathbf{xr}_{ij} | (\mathbf{x}_i, s, t) \in \mathbf{xgp}_i) \quad (20) \\ &:= 1 - P((\mathbf{x}_i, s, t) \in \mathbf{xr}_{ij} | (\mathbf{x}_i, s, t) \in \mathbf{xgn}_i). \end{aligned}$$

We also define the probability of the groundtruth relevance comparison between labels as:

$$\begin{aligned} p_i^{st} &:= P((\mathbf{x}_i, s, t) \in \mathbf{xgp}_i) \quad (21) \\ &:= 1 - P((\mathbf{x}_i, s, t) \in \mathbf{xgn}_i). \end{aligned}$$

Here p_i^{st} denotes the probability that in groundtruth, label s is more relevant than label t on example $\mathbf{x}_i$. Given Eq.20, 21, the likelihood of each example in Eq.15 can be calculated as:

$$P((\mathbf{x}_i, s, t) \in \mathbf{xr}_{ij}) = q_{ij}^{st} p_i^{st} + (1 - q_{ij}^{st})(1 - p_i^{st}). \quad (22)$$

We note that while our Eq. 22 shares the similar ranking idea as that in [37], our problem is quite different, and the induction procedure stemming from the crowds' annotating behavior is unique. Treating each $\{\mathbf{xr}_{ij}^{st}\}$ independently, the likelihood of all examples $\mathbf{XR}$ can be formulated as:

$$\begin{aligned} P(\mathbf{XR}|\theta) &= \prod_{j=1}^{M} \prod_{i \in N^j} \prod_{s,t} P((\mathbf{x}_i, s, t) \in \mathbf{xr}_{ij}) \quad (23) \\ &= \prod_{j=1}^{M} \prod_{i \in N^j} \prod_{s,t} q_{ij}^{st} p_i^{st} + (1 - q_{ij}^{st})(1 - p_i^{st}). \end{aligned}$$

Here θ denotes the parameters involved in the equation.

Defining the annotators' expertise as in Eq.20, we achieve several advantages: first, we avoid the complexity of modeling annotators' expertise variances on individual labels; meanwhile, the annotators' *effort-saving* behavior and the label pair relevance relationship is naturally captured.

Assuming each annotator's expertise over the label relevance depends both on the instance they observe and the class labels, here we exploit a logistic sigmoid function acting on two linear discriminating functions to model q_{ij}^{st}:

$$\begin{aligned} q_{ij}^{st} &= f(\mathbf{x}_i, \alpha_j^s, \alpha_j^t) = \frac{\exp(\alpha_j^{s\prime} \mathbf{x}_i)}{\exp(\alpha_j^{s\prime} \mathbf{x}_i) + \exp(\alpha_j^{t\prime} \mathbf{x}_i)} \\ &= \sigma[(\alpha_j^s - \alpha_j^t)' \mathbf{x}_i]. \quad (24) \end{aligned}$$

Here α_j^s, α_j^t are d-dimensional coefficient vectors modeling annotator j's understanding over label s and t. The probability is reflected through the sigmoid function $\sigma(x) = 1/(1 + exp(-x))$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

Similarly, the groundtruth relevance comparison probability between labels p_i^{st} is defined as:

$$p_i^{st} = f(\mathbf{x}_i, \mathbf{w}^s, \mathbf{w}^t) = \frac{\exp(\mathbf{w}^{s\prime}\mathbf{x}_i)}{\exp(\mathbf{w}^{s\prime}\mathbf{x}_i) + \exp(\mathbf{w}^{t\prime}\mathbf{x}_i)} = \sigma[(\mathbf{w}^s - \mathbf{w}^t)'\mathbf{x}_i]. \quad (25)$$

Here $\mathbf{w}^l$ is the d-dimensional coefficient vector for label l. $\mathbf{f}_l(\mathbf{x}_i) = \mathbf{w}^{l\prime}\mathbf{x}_i$ can be regarded as the classifier model defined on label l and predict the groundtruth for $\mathbf{x}_i$.

Learning Objective Function To learn the annotators' and classifiers' parameters $\theta = \{\mathbf{w}, \{\alpha_j\}\}$, we maximize the log-likelihood of Eq. 23:

$$\hat{\theta} = \{\hat{\mathbf{w}}, \{\hat{\alpha}_j\}\} = \arg\max_{\theta}\{ \ln P(\mathbf{XR}|\theta) \}. \quad (26)$$

Regarding the underlying groundtruth relevance in Eq. 17, 18 as latent variables, we use Expectation-Maximization (EM) [36] to iteratively estimate $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$ given θ in the E-step, and estimate θ given $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$ in the M-step.

E-step: Given the estimation of $\theta = \{\mathbf{w}, \{\alpha_j\}\}$ from last M step, the posteriors probability of $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$, $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgn}_i\}$ are:

$$P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\} \quad (27)$$
$$\{\propto \prod_{j\in M^i}\prod_{s,t} P(\mathbf{xr}_{ij}^{st} \in \mathbf{xr}_{ij}|(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i, \theta) \cdot P((\mathbf{x}_i, s, t) \in \mathbf{xgp}_i|\theta)\} \propto \prod_{j\in M^i}\prod_{s,t} q_{ij}^{st} p_i^{st},$$
$$P\{(\mathbf{x}_i, s, t) \in \mathbf{xgn}_i\} \quad (28)$$
$$\{\propto \prod_{j\in M^i}\prod_{s,t} P(\mathbf{xr}_{ij}^{st} \in \mathbf{xr}_{ij}|(\mathbf{x}_i, s, t) \in \mathbf{xgn}_i, \theta) \cdot P((\mathbf{x}_i, s, t) \in \mathbf{xgn}_i|\theta)\} \propto \prod_{j\in M^i}\prod_{s,t} (1 - q_{ij}^{st})(1 - p_i^{st}).$$

Substituting Eq.24, 25 into Eq.27, 28, E-step reduces to:

$$P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\} \propto \prod_{j\in M^i}\prod_{s,t} \sigma[(\alpha_j^s - \alpha_j^t)'\mathbf{x}_i]\sigma[(\mathbf{w}^s - \mathbf{w}^t)'\mathbf{x}_i], \quad (29)$$
$$P\{(\mathbf{x}_i, s, t) \in \mathbf{xgn}_i\} \propto \prod_{j\in M^i}\prod_{s,t} \sigma[-(\alpha_j^s - \alpha_j^t)'\mathbf{x}_i]\sigma[-(\mathbf{w}^s - \mathbf{w}^t)'\mathbf{x}_i]. \quad (30)$$

M-step: To estimate θ, we maximize the expectation of the joint log-likelihood of $(\mathbf{xr}_{ij}^{st}, (\mathbf{x}_i, s, t))$ over $\theta = \{\mathbf{w}, \{\alpha_j\}\}$, with respect to the posterior probability of $(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i$ computed by last E step:

$$\theta = \arg\max_{\theta} QE(\theta),$$

$$QE(\theta) = E_{(\mathbf{x}_i,s,t)\in\mathbf{xgp}_i}[\ln P(\mathbf{XR}, \{(\mathbf{x}_i, s, t)\}_i^{st})|\theta)] = QE(\{\mathbf{w}\}) + \sum_{j=1}^{M} QE(\alpha_j), \quad (31)$$

Algorithm 2 The RAM Approach

Input: data D; parameter $C = 100$, $\gamma_o = 5 * 1e^{-4}$
Output: multi-label classifier $\mathbf{w}$, workers' parameters $\{\alpha_j\}$
Algorithm:

1: **Repeat**
2: *E-step*: Given $\mathbf{w}$, $\{\alpha_j\}$ computed from last M-step, estimate the posterior probability $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$ by Eq. 29, 30 .
3: *M-step*: Given $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$ computed from last E-step, estimate $\mathbf{w}$, $\{\alpha_j\}$ by maximizing Eq. 32 and 33 using SGD stated in Eq. 34- 37
4: **until** the value Eq. 23 converges or maximum number of iterations is reached

where $QE(\mathbf{w})$, $QE(\alpha_j)$ are induced as,

$$QE(\mathbf{w}) = \sum_{i=1}^{N}\sum_{s,t}\{\ln\sigma[(\mathbf{w}^s - \mathbf{w}^t)'\mathbf{x}_i]P_{oldE} + \ln\sigma[-(\mathbf{w}^s - \mathbf{w}^t)'\mathbf{x}_i](1 - P_{oldE})\}, \quad (32)$$
$$QE(\alpha_j) = \sum_{i\in N^j}\sum_{s,t}\{\ln\sigma[(\alpha_j^s - \alpha_j^t)'\mathbf{x}_i]P_{oldE} + \ln\sigma[-(\alpha_j^s - \alpha_j^t)'\mathbf{x}_i](1 - P_{oldE})\}. \quad (33)$$

We introduce P_{oldE} to denote $P\{(\mathbf{x}_i, s, t) \in \mathbf{xgp}_i\}$ computed by the last E step for the convenience of presentation.

Based on Eq. 32 and 33, the multi-label classifier $\mathbf{w}$ and worker parameter α_j can be independently inferred by maximizing $QE(\mathbf{w})$, $QE(\alpha_j)$ with respect to $\mathbf{w}$, α_j. One solution is alternatively maximizing $QE(\mathbf{w})$, $QE(\alpha_j)$ with respect to one class associated parameters, e.g., $\mathbf{w}^l$, α_j^l, with the others fixed. In the multi-label crowdsourcing tasks, this strategy would be very slow when the number of labels and workers are large. In this paper, we employ the more efficient stochastic gradient descent(SGD) [38] approach to minimize $-QE(\mathbf{w})$, $-QE(\alpha_j)$ as following:

To minimize $-QE(\mathbf{w})$, at each iteration of SGD, we randomly sample an instance $\mathbf{x}$, two labels s, t to form a triplet $(\mathbf{x}, l^s, l^t)$ and perform gradient descent on $\mathbf{w}^s$, $\mathbf{w}^t$:

$$\mathbf{w}^{s(T+1)} = \mathbf{w}^{s(T)} - \gamma^{(T+1)}\mathbf{x}(P_{oldE} - P_{oldw}) \quad (34)$$
$$\mathbf{w}^{t(T+1)} = \mathbf{w}^{t(T)} - \gamma^{(T+1)}\mathbf{x}(P_{oldw} - P_{oldE}) \quad (35)$$

To minimize $-QE(\alpha_j)$, at each iteration of SGD, we randomly sample an instance $\mathbf{x}$, one positive tagged label s and one negative or untagged label t such that $\mathbf{y}_{ij}^s = 1$ and $\mathbf{y}_{ij}^t = -1/0$ to form a triplet $(\mathbf{x}_i, s, t)$ and perform gradient descent on α_j^s, α_j^t:

$$\alpha_j^{s(T+1)} = \alpha_j^{s(T+1)} - \gamma^{(T+1)}\mathbf{x}(P_{oldE} - P_{oldal}) \quad (36)$$
$$\alpha_j^{t(T+1)} = \alpha_j^{t(T+1)} - \gamma^{(T+1)}\mathbf{x}(P_{oldal} - P_{oldE}) \quad (37)$$

Here $\gamma^{(T+1)}$=$\gamma_o/(T+1)$ is the step size updated in each iteration of SGD, γ_o is the initialized step size value. $P_{oldw} = \sigma[(\mathbf{w}^{s(T)} - \mathbf{w}^{t(T)})'\mathbf{x}]$, $P_{oldal} = \sigma[(\alpha_j^{s(T)} - \alpha_j^{t(T)})'\mathbf{x}]$ are introduced for the convenience of presentation. After the SGD update, to overcome overfitting, $\mathbf{w}^s, \mathbf{w}^t, \alpha_j^s, \alpha_j^t$ are normalized to have a L2 norm smaller than a constant C. Algorithm 2 summarizes the overall process of RAM.

4 SCENARIO 2: ACTIVE CROWDSOURCING

Considering that the labeling budget is often limited, whereas annotating the whole dataset may lead to unnecessary information redundancy, we extend NAM and RAM to the active crowdsourcing learning scenario to query and learn from the most valuable annotations. We consider the pool-based active learning case and first give the formal description of the learning setting and process.

Different from the previous scenario with pre-collected annotations, we are given a small number of N_l initially labeled examples with groundtruth available $D_l = \{(\mathbf{x}_1, \mathbf{z}_1, \{\mathbf{y}_{1j}\}), \ldots, (\mathbf{x}_{N_l}, \mathbf{z}_{N_l}, \{\mathbf{y}_{N_l j}\})\}$, with $\mathbf{z}_i$ denoting the groundtruth of $\mathbf{x}_i$, $\mathbf{y}_{ij}$ denoting worker j's annotation for $\mathbf{x}_i$. Besides, a pool of N_u unlabeled examples $D_u = \{\mathbf{x}_{N_l+1}, \ldots, \mathbf{x}_{N_l+N_u}\}$ are available. During the learning process with M workers available for annotating, we iteratively alternate between the two steps: we select the most valuable annotation to query from the workers, and update the crowdsourcing classifier given the annotations.

Based on the previous learning process, NAM and RAM can respectively get an estimation of each worker's expertise $\mathbf{e}_{ij}^l$, q_{ij}^{st} over individual label and label pairs, as implemented in Eq.6, Eq.24, and the classifier $f_{0,\theta}^l(\mathbf{x}_i) = (\mathbf{w}_0^l)^{'}\mathbf{x}_i$, $\mathbf{f}_l(\mathbf{x}_i) = \mathbf{w}^{l'}\mathbf{x}_i$ on each label. In the following, we introduce the instance, label, and worker selection criteria computed based on the prediction results of NAM and RAM.

Instance Selection To select the most informative instance for the current multi-label classifier, typical methods defined some uncertainty measure for the instances considering the prediction uncertainty over labels, such as the average or minimal margin of all labels to their respective svm classifier [39], [40]. In this work, driven by our crowdsourcing query setting, we adopt the query diversity regularized uncertainty measure QCI defined in [41], which was extended from the LCI (Label Cardinality Inconsistency) measure proposed in [42], for our instance selection.

LCI was motivated by the label cardinality observation that the multi-label instances usually have similar number of positive labels. Defined as the inconsistency between the number of predicted positive labels of instances and the average number of positive labels of the initial labeled data, LCI was formulated as:

$$\max_{\mathbf{x}_i} LCI(\mathbf{x}_i) = |\sum_{l=1}^{L} I(\hat{\mathbf{z}}_i^l = 1) - \frac{1}{N_l}\sum_{j=1}^{N_l}\sum_{l=1}^{L} I(\mathbf{z}_j^l = 1)|. \quad (38)$$

Here $\hat{\mathbf{z}}_i^l$ is the label estimation of instance $\mathbf{x}_i$ on l. To avoid always querying the most uncertain instances, Huang *et al.* [41] extended LCI to QCI as:

$$\max_{\mathbf{x}_i} QCI(\mathbf{x}_i) = \frac{|\sum_{l=1}^{L} I(\hat{\mathbf{z}}_i^l = 1) - \frac{1}{N_l}\sum_{j=1}^{N_l}\sum_{l=1}^{L} I(\mathbf{z}_j^l = 1)|}{\max\{0.5, \mathsf{anno}(\mathbf{x}_i)\}}. \quad (39)$$

Here $\mathsf{anno}(\mathbf{x}_i)$ denotes the queried times of instance $\mathbf{x}_i$ and 0.5 is used avoid zero divisor.

Different from conventional active learning where each label is queried at most one time, in crowdsourcing, each label can be queried for multiple times from several workers. Thus the query regularization is meaningful to give more chance to the less queried instances which could contain more unknown information. In the experiment, we give some analysis over the instance selection criteria.

Algorithm 3 The Active Crowdsourcing Procedure

Input: data $D = \{D_l, D_u\}$
Train:
1: Initialize the crowdsourcing data D_c as D_l, learn the crowdsourcing model on D_c using NAM/RAM
2: **Repeat**:
3: Get label predictions for each $\mathbf{x} \in D_u$
4: Select instance $\mathbf{x}_{i^*}$ by Eq. 39
5: Select the labels L_l^* for $\mathbf{x}_{i^*}$ by Eq. 40
6: Select worker j^* for $\mathbf{x}_{i^*}$ by Eq. 41/ Eq. 42 for NAM/RAM
7: Query $\{\mathbf{y}_{i^*j^*}^{l^*}\}$ for $\mathbf{x}_{i^*}$ on $l^* \in L_l^*$ from workers j^*
8: Add $\{\mathbf{y}_{i^*j^*}^{l^*}\}$ to the crowdsourcing data D_c
9: Update crowdsourcing model on D_c by NAM/RAM
10: **Until** The maximum number of queries is reached
Test:
1: For instance $\mathbf{x}_t$, get the label predictions using the multi-label classifier

Label Selection Previous multi-label active learning mainly select labels which are most uncertain such as labels closest to the decision boundary [41], or which improve the classifier best like minimizing the generalization error [43]. Different from them, we consider the *label sparsity* property of multi-label tasks, i.e., the number of *positive* labels for each example is far less than the number of the *negative* counterparts, and their information is critical for learning. Given the selected instance $\mathbf{x}_*$ by Eq.39, we propose to query its l most possibly positive labels, i.e., the top-l ranked labels predicted by the classifier:

$$L_l^* = \{\ l^* \mid \mathbf{w}^{l^*\prime}\mathbf{x}_{i^*} \text{ ranks top } l \text{ among label predictions } \}. \quad (40)$$

We test l with varying values in the experiments.

Annotator Selection Given the selected instance and label $(\mathbf{x}_{i^*}, l^*)$ by Eq. 39 and Eq. 40, we need to select the most reliable annotator to collect high quality annotations. For NAM, Eq. 6 provides each worker's expertise $\mathbf{e}_{ij}^l$ over each label l, thus we select the most reliable worker by

$$NAM:\ j^* = \arg\max_j \mathbf{e}_{i^*j}^{l^*} = \sigma((\mathbf{w}_j^{l^*})^{'}\mathbf{x}_{i^*}). \quad (41)$$

For RAM, Eq.24 provides each worker's expertise q_{ij}^{st} over each label pair. To select the most reliable worker for the most possibly positive label l^*, we select the worker that gives the most possibly positive annotation:

$$RAM:\ j^* = \arg\max_j \exp(\alpha_j^{l^*\prime}\mathbf{x}_{i^*}). \quad (42)$$

After the instances, labels and annotators are selected, the annotations are queried and added to the training data to update the crowdsourcing model. The overall process of the active crowdsourcing framework is summarized in Algorithm 3.

5 COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, we give the computational complexity of the proposed methods. For NAM and RAM, in each EM iteration, their computation complexity in the E-step and M-step are respectively $O(LNMd)$, $O(tLNMd)$ and

$O(NL^2Md), O(tMd)$, here t is used to denote the maximum iteration number of GD/SGD for NAM/RAM in the M-step. Besides, the enhanced representation computation for NAM is $O(N^2(d+Mk))$(computing the distance between all instances costs N^2d, finding the k nearest neighbors for each instance among N instances costs kN). Thus the computation complexity of NAM and RAM are respectively $O(N^2(d+Mk)+TtLNMd)$ and $O(T(NL^2+t)Md)$, here T denotes the maximum EM iteration number. In the experiment, we use LIBLINEAR [44] to initialize the classifier and workers' parameters and set $T = 100$, which show good performance. For the active crowdsourcing, the computation composes of the update of the model, and the active selection by Eq. 39- 42, which is $O(N_u + M)Ld$.

6 Experiment 1: Groundtruth Estimation

In this section, we conduct experiments estimating the groundtruth labels from the given crowdsourcing annotations. We experiment on several simulated and real world datasets with a few baselines.

Baselines We compare with four multi-label crowdsourcing methods **D-DS**, **P-DS**, **ND-DS** [14] and **MLNB** [11]. We also adopt four state of the art single-label crowdsourcing methods **MV**, **DS** [29], **MaxEn** [31] **Yutc** [8] as baselines by applying them on each individual label.

Besides, from the partial label view, the unknown annotations of each worker on its untagged images can be seen as missing entries of the annotation matrix $Y \in \{1, -1, 0\}^{N\times(L\times M)}$, where $Y(i, (j-1)\times L + l) = (-1)1$ denotes worker j annotated image i and (did not)tag(ged) label l as positive, $Y(i, (j-1)\times L + l) = 0$ denotes worker j did not annotate image i. Considering entries in Y are repetition annotations over the same label set, thus a low rank structure assumption of Y is reasonable and we adopt the low rank assumption based partial label recovery method **Maxide** [26] to recover the missing annotations. Then majority voting is used to get the groundtruth estimation. Maxide exploited instance features as side information and was theoretically and experimentally validated to be superior than alternatives [25].

The parameters for the proposed methods are tuned on a small subset of instances with their groundtruth labels available on our real data *dataset1* and fixed during the experiments. For NAM, the neighbor size k is tuned in $[1, 2, ..., 10]$ and set as 5, the prior distribution parameter λ is tuned in $\{10^{-5}, 10^{-4}, ..., 10^{2}\}$ and set as 10^{-3}. For RAM, the initial step size of SGD is set as 5×10^{-4}, and the norm upper bound C is tuned in $\{0.1, 1, ..., 10^5\}$ and set as 100. For practical usage, if the groundtruth labels are not available, their estimations by state of the art methods such as MV and MaxEn[31] can be used as substitution. For baselines, implementation code are provided by their authors and the recommended parameters values are used. Note that in the above baselines, except for **Yutc** and **Maxide** utilizing the instance features during learning, other baselines infer the groundtruth labels only from the annotations.

We report the *macroF1* (MacF1) classification results, which evaluates the macro average results of the F1 classification score on each class label, with larger value indicating better performance. As RAM currently learns no threshold value to separate the positive and negative predictions, to evaluate the classification performance, we treat the top l ranked labels as positive and other labels as negative. Here l is the number of groundtruth positive labels of each example. For fair of comparison, the best result among this strategy and the heuristic real value threshold strategy (0.5 for crowdsourcing baselines' probabilistic output and 0 for Maxide's signed real valued output) for baselines are used. Similar results on other measures such as *microF1*, *average precision* and *ranking loss* are also obtained and omitted here.

TABLE 1
The MacF1 scores of generated workers in **Setting I**

		w1	*w2*	*w3*	*w4*	*w5*	*w6*	*w7*
Em.	p = 90%	.501	.519	.521	.519	.520	.516	.499
	p = 30%	.431	.387	.421	.441	.425	.413	.386
Sc.	p = 90%	.590	.590	.595	.586	.587	.592	.592
	p = 30%	.530	.487	.531	.541	.512	.533	.518
Ye.	p = 90%	.603	.603	.609	.604	.603	.605	.607
	p = 30%	.602	.596	.585	.587	.592	.590	.595

6.1 Simulation Data

In this subsection, we conduct experiments on three multi-label benchmarks from text, image and biology fields.

Text Categorization The *Emotions* (*Em.*) [45] dataset is a collection of 593 songs that are classified into 6 classes of emotions. Each song has on average 1.9 ± 0.7 labels.

Image Annotation The *Scene* (*Sc.*) [46] dataset contains 2407 natural scene images and 6 possible labels. Each image has on average 1.1 ± 0.2 labels.

Gene Functional Classification The *Yeast* (*Ye.*) [2] dataset contains 2417 genes and 14 possible class labels. Each gene has on average 4.2 ± 1.6 labels.

We study two types of annotation generating process. In **Setting I**, as binary crowdsourcing tasks, the annotations are generated for each label separately. To simulate workers presenting similar annotations for instances with similar features, we randomly hold out 50% instances of the whole dataset for crowdsourcing learning, whose annotations of each worker are given by the LIBLINEAR [44] classifier trained on each label using random p fraction data of the other half instances. Multiple annotators are simulated by repeating this process for several times. We generate a moderate number of 7 workers for each dataset, and test $p = 90\%, 30\%$ to simulate different worker expertise levels.

In **Setting II**, treating the multi-label tasks different from multiple binary tasks, we simulate the crowds' *effort-saving* behavior of just annotating the few most relevant labels. We use the classic RankSVM [47] multi-label classifier which predicts the relevance ranking of labels. Similar as Setting I, we generate a moderate number of 7 workers for each dataset with two p values.

For both settings, the average and standard deviation performances over 10 times data generation are recorded respectively in Table 3, 4. We also give the average rank of methods on each dataset over different p settings.

Before diving into the massive result details, we first take an overall look at the annotations' quality generated by the above process. For the three datasets and the two annotation generating setting, the *MacF1* scores of the workers are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
MacF1 results (the larger, the better) (mean ± std.) for **Setting I** on the simulated datasets. • (○) /▼ (▽) indicates that NAM/ RAM is significantly better(worse) than the compared method (paired t-tests at 95% significance level).

		p = 90%	p = 30%	*avk.*		p = 90%	p = 30%	*avk.*		p = 90%	p = 30%	*avk.*
NAM		.620±.016	.628±.005	**1.6**		.737±.008	.750±.004	**1.0**		.607±.003	.617±.003	**2.6**
RAM		.617±.003	.631±.003	**2.0**		.724±.008	.742±.004	**2.0**		.610±.018	.616±.000	**2.3**
D-DS		.520±.011•▼	.551±.007•▼	**8.6**		.675±.009•▼	.636±.001•▼	**8.3**		.597±.004•▼	.605±.001•▼	**5.3**
P-DS		.259±.014•▼	.300±.006•▼	**11.0**		.474±.008•▼	.432±.001•▼	**10.0**		.469±.003•▼	.471±.001•▼	**10.0**
ND-DS		.514±.007•▼	.552±.004•▼	**8.3**		.674±.005•▼	.637±.000•▼	**8.6**		.595±.003•▼	.607±.001•▼	**5.0**
MLNB	Em.	.334±.025•▼	.344±.005•▼	**10.0**	Sc.	.168±.002•▼	.174±.005•▼	**11.0**	Ye.	.306±.002•▼	.311±.002•▼	**11.0**
MV		.606±.014•▼	.610±.005•▼	**7.0**		.685±.007•▼	.657±.001•▼	**7.0**		.628±.007○▽	.636±.002○▽	**1.0**
DS		.608±.009•▼	.618±.013•▼	**4.3**		.705±.009•▼	.672±.008•▼	**5.3**		.576±.007•▼	.595±.004•▼	**7.0**
MaxEn		.614±.007•▼	.623±.007•▼	**4.6**		.704±.005•▼	.678±.006•▼	**5.6**		.560±.005•▼	.575±.003•▼	**8.3**
Yutc		.611±.012•▼	.624±.003•▼	**4.6**		.718±.006•▼	.715±.001•▼	**3.0**		.514±.001•▼	.583±.003•▼	**8.6**
Maxide		.611±.004•▼	.624±.002•▼	**3.6**		.708±.005•▼	.713±.002•▼	**4.0**		.603±.001•▼	.608±.001•▼	**4.6**

TABLE 4
MacF1 results (the larger, the better) (mean ± std.) for **Setting II** on the simulated datasets. • (○) /▼ (▽) indicates that NAM/ RAM is significantly better(worse) than the compared method (paired t-tests at 95% significance level).

		p = 90%	p = 30%	*avk.*		p = 90%	p = 30%	*avk.*		p = 90%	p = 30%	*avk.*
NAM		.615±.009	.634±.015	**1.6**		.734±.006	.737±.004	**1.0**		.632±.007	.618±.002	**1.6**
RAM		.624±.005	.628±.006	**2.0**		.727±.008	.731±.007	**2.0**		.630±.002	.608±.003	**2.6**
D-DS		.520±.012•▼	.553±.006•▼	**9.0**		.594±.007•▼	.653±.003•▼	**9.0**		.587±.005•▼	.592±.002•▼	**5.6**
P-DS		.289±.005•▼	.327±.031•▼	**11.0**		.391±.006•▼	.448±.008•▼	**10.0**		.442±.011•▼	.467±.005•▼	**10.0**
ND-DS		.529±.008•▼	.560±.012•▼	**8.0**		.599±.008•▼	.659±.009•▼	**8.0**		.600±.002•▼	.582±.003•▼	**5.0**
MLNB	Em.	.344±.016•▼	.333±.017•▼	**10.0**	Sc.	.168±.011•▼	.181±.014•▼	**11.0**	Ye.	.306±.003•▼	.305±.001•▼	**11.0**
MV		.612±.006•▼	.628±.006•	**4.0**		.632±.012•▼	.674±.010•▼	**7.0**		.638±.002○▽	.615±.004▽	**1.6**
DS		.596±.012•▼	.616±.010•▼	**7.0**		.645±.009•▼	.691±.011•▼	**5.6**		.568±.019•▼	.609±.016•▼	**6.0**
MaxEn		.615±.013▼	.620±.007•▼	**3.0**		.652±.009•▼	.693±.007•▼	**5.3**		.544±.006•▼	.569±.015•▼	**8.3**
Yutc		.613±.009▼	.619±.013•▼	**5.3**		.714±.007•▼	.732±.005•	**3.0**		.554±.014•▼	.526±.036•▼	**8.6**
Maxide		.603±.006•▼	.622±.005•▼	**5.0**		.705±.005•▼	.710±.008•▼	**4.0**		.604±.003•▼	.584±.002•▼	**5.3**

TABLE 2
The MacF1 scores of generated workers in **Setting II**

		w1	*w2*	*w3*	*w4*	*w5*	*w6*	*w7*
Em.	p = 90%	.498	.478	.437	.442	.492	.449	.450
	p = 30%	.481	.392	.472	.447	.473	.434	.513
Sc.	p = 90%	.533	.449	.520	.535	.460	.466	.435
	p = 30%	.530	.500	.529	.498	.510	.524	.493
Ye.	p = 90%	.598	.569	.599	.595	.574	.597	.601
	p = 30%	.565	.575	.582	.578	.597	.574	.587

shown in Table 1 and Table 2. Comparing Table 1 and Table 2, we can see that the annotations generated by the two processes are quite different. When $p = 90\%$, the *MacF1* scores of the workers in Setting II are universally lower than that in the Setting I on all datasets, whereas with $p = 30\%$, there is no such monotone phenomenon. Besides, scores of the workers generated by Setting II are more diverse compared to that in Setting I.

Now look at the aggregation results in Table 3 and 4, the proposed NAM and RAM always rank the top two, followed by Maxide and the single-label baselines. Assuming a low rank structure among the annotations, Maxide achieves not bad performance here, which should be explained by that the training data used to generate the workers are largely overlapped thus making the crowds' annotations differ not too much. Among the multi-label baselines, MLNB ranks the worst. There are possibly two reasons for this phenomenon, first, MLNB considers the label co-occurrence correlation to learn the multi-label taxonomies, for which the label co-occurrence is specially pertinent, but for problems lacking label hierarchies, the label co-occurrence is not as common; second, MLNB uses the same sensitivity and specificity parameter for all workers without modeling crowds' expertise variance. Comparing D-DS, P-DS extending DS [29] to consider label relationship among all label powersets, label set of two labels, ND-DS performs the most effective by using conditional independent label dependency, which overcomes the label sparsity.

Results on Yeast is a bit different, showing that except for the proposed methods, Maxide, MV and D-DS, ND-DS, the rest methods perform even worse than the original annotations. This may be explained by the relative high difficulty of the Yeast dataset, for which each instance is associated with 4 to 5 positive labels thus correctly tagging all proper labels is more difficult.

Another notable thing is that although in Table 1 and 2, each of the workers generated using $p = 30\%$ training data is less reliable than those generated using 90%, there is no such consequential results in Table 3 and 4, i.e., the aggregation performance for $p = 30\%$ are not necessarily worse than $p = 90\%$. This should be explained by the larger diversity among workers generated in the $p = 30\%$ setting. This may suggest that similar as ensemble learning, the diversity among workers also plays a role.

6.2 Real Data

To get real data, we distributed two image annotation tasks on the AMT platform, and ask the workers to annotate the proper labels for the image they see.

dataset1 The *dataset1* contains 700 images with 6 candidate labels {*desert*, *beach*, *sea*, *mountain*, *tree*, *sunriseset*}. On average each image has 1.2 ± 0.4 labels. The images with more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

TABLE 5
MacF1 results (the larger, the better) (mean ± std.) on *dataset1*. nw denotes the number of workers.● (○) /▼ (▽) indicates that NAM/ RAM is significantly better(worse) than the compared method (paired t-tests at 95% significance level).

	Alg.	$nw=1$	$nw=3$	$nw=5$	$nw=7$	$nw=9$	$nw=11$	$nw=13$	$nw=15$	$avg.rk.$
	NAM	.745±.131	.823±.027	.862±.018	.878±.016	.890±.013	.895±.005	.897±.004	.900±.003	**1.2**
	RAM	.726±.148	.836±.020	.858±.018	.869±.011	.883±.012	.892±.008	.894±.008	.900±.004	**1.8**
	D-DS	.466±.341●▼	.723±.065●▼	.768±.053●▼	.797±.050●▼	.826±.037●▼	.846±.016●▼	.855±.003●▼	.860±.004●▼	**6.0**
	P-DS	.243±.183●▼	.350±.064●▼	.365±.045●▼	.381±.040●▼	.397±.028●▼	.408±.013●▼	.414±.004●▼	.417±.003●▼	**10.0**
	ND-DS	.467±.341●▼	.734±.070●▼	.780±.053●▼	.812±.049●▼	.836±.037●▼	.857±.017●▼	.868±.008●▼	.875±.004●▼	**4.8**
	MLNB	.141±.041●▼	.192±.014●▼	.189±.011●▼	.186±.006●▼	.207±.016●▼	.204±.017●▼	.204±.011●▼	.206±.008●▼	**11.0**
	MV	.465±.280●▼	.703±.062●▼	.766±.057●▼	.798±.047●▼	.829±.035●▼	.859±.014●▼	.870±.008●▼	.881±.006●▼	**5.6**
dataset1	DS	.496±.259●▼	.718±.064●▼	.737±.036●▼	.748±.044●▼	.777±.031●▼	.801±.027●▼	.815±.008●▼	.844±.016●▼	**7.5**
	MaxEn	.505±.259●▼	.709±.048●▼	.778±.050●▼	.820±.049●▼	.846±.038●▼	.865±.017●▼	.876±.010●▼	.880±.008●▼	**4.1**
	Yutc	.631±.185●▼	.729±.021●▼	.758±.014●▼	.768±.024●▼	.779±.018●▼	.799±.012●▼	.801±.010●▼	.809±.008●▼	**6.9**
	Maxide	.729±.093▼	.757±.027●▼	.764±.020●▼	.767±.012●▼	.771±.002●▼	.767±.010●▼	.766±.006●▼	.765±.006●▼	**7.0**

TABLE 6
MacF1 results (the larger, the better) (mean ± std.) on *dataset2*. nw denotes the number of workers.● (○)/▼ (▽) indicates that NAM/ RAM is significantly better(worse) than the compared method (paired t-tests at 95% significance level).

	Alg.	$nw=1$	$nw=3$	$nw=5$	$nw=7$	$nw=9$	$nw=11$	$nw=13$	$nw=15$	$avg.rk.$
	NAM	.406±.143	.750±.055	.775±.044	.799±.042	.839±.012	.857±.005	.859±.004	.865±.000	**1.1**
	RAM	.389±.145	.745±.073	.759±.049	.788±.028	.817±.006	.831±.049	.828±.006	.836±.002	**2.6**
	D-DS	.076±.016●▼	.569±.220●▼	.621±.161●▼	.670±.100●▼	.752±.028●▼	.777±.008●▼	.784±.006●▼	.793±.000●▼	**6.7**
	P-DS	.014±.015●▼	.131±.210●▼	.142±.152●▼	.148±.110●▼	.157±.030●▼	.157±.009●▼	.162±.007●▼	.161±.000●▼	**10.1**
	ND-DS	.021±.013●▼	.657±.190●▼	.678±.147●▼	.705±.107●▼	.767±.028●▼	.779±.008●▼	.786±.008●▼	.601±.000●▼	**5.3**
	MLNB	.077±.003●▼	.104±.011●▼	.106±.012●▼	.108±.009●▼	.109±.008●▼	.115±.004●▼	.114±.009●▼	.113±.009●▼	**10.6**
	MV	.134±.014●▼	.555±.188●▼	.612±.153●▼	.674±.114●▼	.773±.039●▼	.815±.011●▼	.835±.007●	.850±.000●▽	**5.3**
	DS	.141±.015●▼	.578±.184●▼	.629±.143●▼	.686±.101●▼	.759±.032●▼	.765±.005●▼	.760±.005●▼	.770±.003●▼	**5.9**
dataset2	MaxEn	.141±.013●▼	.575±.195●▼	.631±.161●▼	.690±.119●▼	.786±.038●▼	.826±.007●▼	.836±.004●▽	.844±.000●▽	**3.8**
	Yutc	.302±.073●▼	.564±.074●▼	.601±.033●▼	.604±.032●▼	.634±.020●▼	.635±.028●▼	.640±.022●▼	.655±.023●▼	**7.4**
	Maxide	.530±.041○▽	.613±.040●▼	.583±.050●▼	.567±.024●▼	.560±.030●▼	.541±.010●▼	.574±.031●▼	.544±.000●▼	**7.4**

than one label comprise about 23%. Annotations from 18 workers each annotating no less than 70 images are kept for experiment. On average each worker annotated 267 ± 201 images, each image was annotated by 6.9 ± 2.3 workers.

dataset2 The *dataset2* contains 1495 images with 16 candidate labels {*desert*, *beach*, *sea*, *boat*, *mountain*, *flower*, *tree*, *garden*, *waterfall*, *building*, *city*, *car*, *person*, *indoor*, *sunriseset*, *sky*}. On average each image has 1.8 ± 0.9 labels. The images with more than one label comprise about 61%. Annotations from 15 workers each annotating no less than 100 images are kept. On average each worker annotated 397 ± 453 images, each image was annotated by 10.1 ± 1.4 workers.

We use the two tasks with different instance and label sizes to test the algorithms. The groundtruth labels are annotated by human volunteers, and a 1248-dim fisher vector is extracted as feature representation for each image.

To get some rough idea about the crowds' annotation quality, we conduct some initial analysis. For each worker, we calculate its macroF1 classification result on its corresponding annotated example set: on *dataset1* and *dataset2*, the macroF1 scores for their 18 and 15 workers are respectively {0.62, 0.69,0.76,0.77,0.78, 0.79, 0.80, 0.80, 0.80, 0.82, 0.83, 0.83, 0.83, 0.83, 0.84, 0.85, 0.85, 0.91 }, {0.60, 0.71, 0.75, 0.76, 0.76, 0.76, 0.76, 0.77, 0.77, 0.78, 0.78, 0.80, 0.80, 0.83, 0.85 }. We can see that the crowds' macroF1 scores are mainly around [0.70,0.80], indicating that most workers are intending to provide good annotations.

Following [9], [28], we show the approaches' varying performance affected by the numbers of annotations each example receives. Workers are added one by one randomly. To alleviate the effects of worker order randomness, the process is repeated for 10 times. The average and standard deviation results as the number of workers varies from 1 to 15 for the two datasets are shown in Table 5, 6 .

It can be seen that in most cases, NAM and RAM achieve significant better performance than baselines. Though the partial label learning approach Maxide performs nice when the number of workers is small (less than 5), it fails to gain more benefits when the number of annotations increases, which should be due to its ignorance of modeling the crowds' expertise. This indicates that crowdsourcing learning itself is different from other problems, for which careful attention to crowds' labeling expertise should be paid.

For the crowdsourcing baselines, we can see an overall monotone performance increasing trend as the number of workers grows from small, and tend to reach a flat level as the number of workers become large. Similar as results on the simulation data, MLNB ranks the worst. Comparing D-DS, P-DS and ND-DS, ND-DS achieves the best performance by using conditional independent label dependency properties to overcome the data sparsity problem.

Among the single-label baselines, when the number of workers is less than 11, there is an obvious gap between them and our proposed methods, suggesting that to get the same good performance, NAM and RAM needs much fewer annotations. While MV and MaxEn increase faster as annotations grow, other baselines increase slower and achieve relatively inferior results. Ignoring label correlations and the crowds' specific annotating expertise on multi-label tasks, treating multi-label crowdsourcing tasks as repetitions of single-label problem shows inferiority.

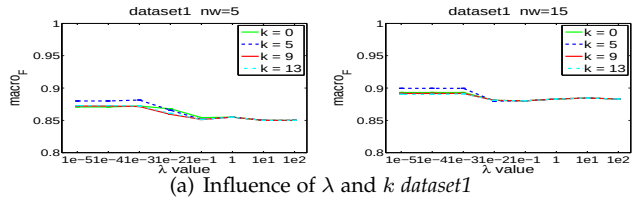
Taking into account the local influence of neighborhoods'

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

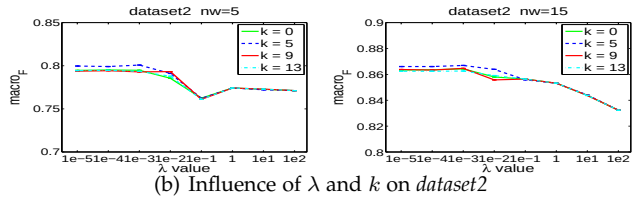
label correlations, and crowds' effort-saving annotating behavior, our NAM and RAM achieves the best performance.

6.3 Parameter Study

In the above results, parameters are fixed for NAM and RAM. Here we explore the effects of the two parameters λ and neighbor size k to NAM, and the norm upper bound C to RAM. Tuning λ in $\{10^{-5}, 10^{-4}, ..., 10^{2}\}$ and k in $\{0, 5, 9, 13\}$ for NAM, tuning C in $\{0.1, 1, ..., 10^{5}\}$ for RAM, we report the *MacroF1* results on *dataset1* and *dataset2* with 5 and 15 workers in Figure 2 and Figure 3.



(a) Influence of λ and k *dataset1*



(b) Influence of λ and k on *dataset2*

Fig. 2. Influence of the two parameters λ and neighborsize k for NAM with two different number of workers $nw = 5$ and $nw = 15$.

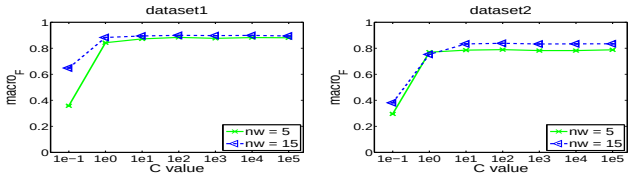


Fig. 3. Influence of parameter C for RAM on *dataset1* and *dataset2* with two different number of workers $nw = 5$ and $nw = 15$.

From Figure 2, we can see that compared to the mild effect of k, the value of λ affects the learning more significantly. With all four k values, the performance of NAM is stably good (degenerates much) for λ no larger than 10^{-3} (no smaller than 10^{-2}). For the neighbor size k, the performance improves as k increases from 0 to 5, and degenerates as k becomes large. This is reasonable as the effects of far away neighbors become weak. Given the results in Figure 2, we suggest $\lambda = 10^{-3}$ and $k = 5$ in practical use.

From Figure 3, we can see C shows a similar effect on RAM as λ on NAM, with stably good performance when no smaller than 100, the value we use in the experiments. It's not strange that same preference for C are shared over datasets since the performance variance due to the varying of data size (number of instances, labels, and annotations) is normalized out using SGD optimization.

7 Experiment 2: Active Crowdsourcing

7.1 Setup

In this section, we study the effect of active crowdsourcing strategies on the two real datasets. For each dataset, we randomly partition the instances into three parts comprising 5%, 70% and 25% of the whole data to construct the initial labeled annotated training data, the unlabeled training data and the test data. At each query, the (instance, label, worker) triple(s) are selected and their annotations are added into the data to update the learning model. The average performances over five times random data partition is reported.

Comparison I We conduct two sets of comparisons. First, we fix the instance selection strategy as QCI [41] defined in Eq. 39, and test the following label and worker selection strategies for NAM and RAM: 1)LP, 2)LU, 3)LR which select label that is most possibly to be Positive (using Eq. 40), select label that is most Uncertain(label probability closest to 0.5), and select label Randomly; 1)WE, 2)WR which select worker that is most Expertised (using Eq. 41/ Eq. 42 for NAM/RAM), and select workers Randomly. To get an intuition on the relative performance of the proposed method, we also incorporate the Single-label Active Crowdsourcing (SAC) method [32] as baseline. SAC selects the most uncertain instance (with one label whose probability is closest to 0.5) to query from the most reliable worker (with highest labeling accuracy).

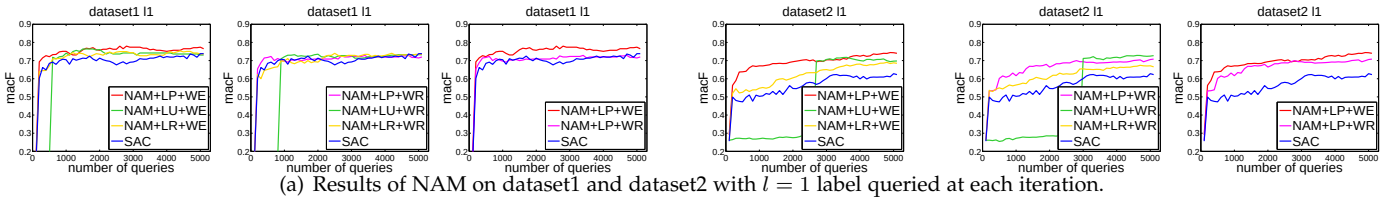
Comparison II In the second comparison, we demonstrate the effectiveness of the QCI instance selection criterion we adopted from [41]. Fixing the label and worker selection strategy as LP and WE, we investigate the effect of three instance selection strategies: 1)the QCI strategy in Eq. 39, 2)the LCI in Eq. 38, and 3) Random instance selection.

We test the number of queried labels $l = 1, 3$. To take advantage of the newly added annotations, the *enhanced instance representations* for NAM are updated after every 50 queries. The results for the two kinds of comparisons are shown in Figure 4-5 and Figure 6.

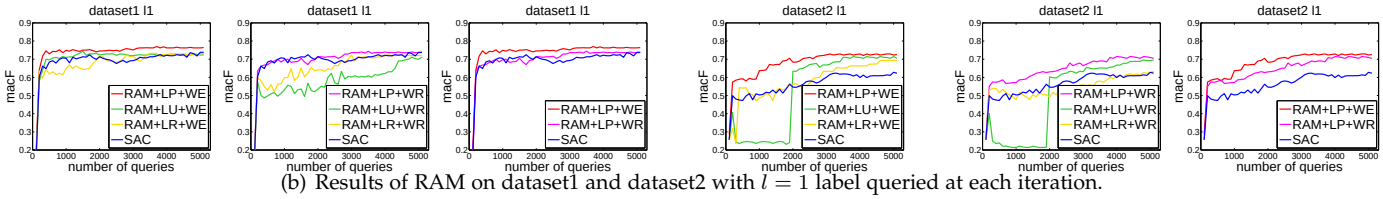
7.2 Results and Analysis

Results of Comparison I We first look at the results for Comparison I with $l = 1$ in Figure 4(a) and 4(b). To give a clear demonstration, we show three plots for each data, representing the results of the three label selection strategies while fixing the worker selection strategy as WE, WR, and the comparison of WE and WR with the label selection fixed as LP. Comparing the results of NAM on *dataset1* in the left three plots of Figure 4(a), we can see that: 1) either with WE or WR, comparing LP, LU and LR, a) in most time, LP is much better than the LR random selection, b) LU sometimes does not performs good until enough annotations are collected, for example, 500, 800 on *dataset1* with worker selection strategy WE, WR. This phenomenon is more obvious on *dataset2* which concerns 16 labels. This could be due to the label sparsity property where the queried annotations are mostly negative, whose contribution to modeling the crowds maybe misleading; 2) fixing the label selection strategy as LP, the comparison of WE and WR in the third figure of the left (right) three plots validates that WE always performs better than WR. In the right three plots of Figure 4(a) for NAM on *dataset2* with larger number of labels, more obvious gaps between comparison methods are observed. With the same setting, the comparison of RAM with different label and worker selection strategies are similarly shown in Figure 4(b), from which the proposed LP and WE strategies are most effective.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2857766, IEEE Transactions on Knowledge and Data Engineering

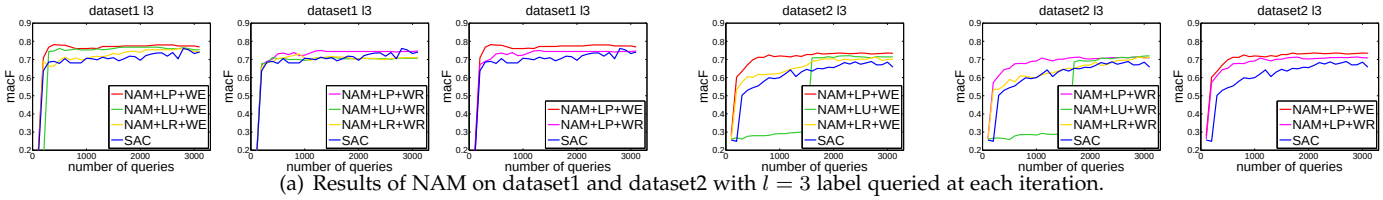


(a) Results of NAM on dataset1 and dataset2 with $l = 1$ label queried at each iteration.

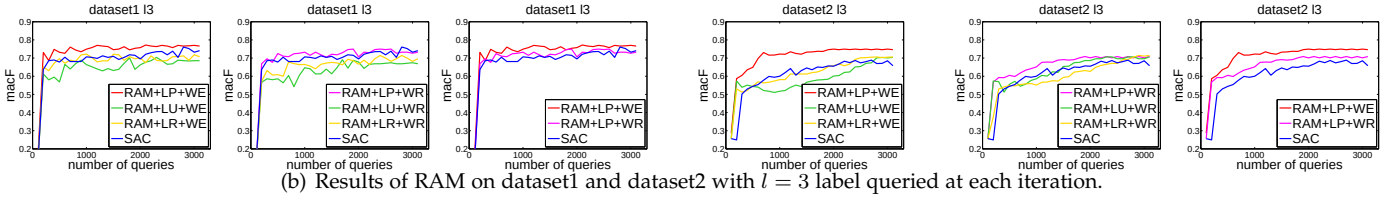


(b) Results of RAM on dataset1 and dataset2 with $l = 1$ label queried at each iteration.

Fig. 4. Results for label strategies {LP, LU, LR} and worker strategies {WE, WR} with queried labels $l = 1$. Instances selection is fixed as QCI.



(a) Results of NAM on dataset1 and dataset2 with $l = 3$ label queried at each iteration.



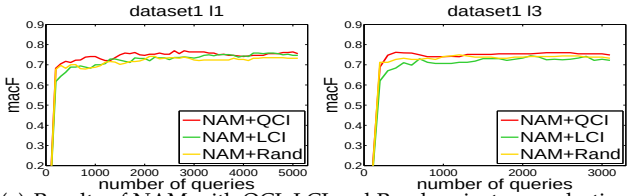
(b) Results of RAM on dataset1 and dataset2 with $l = 3$ label queried at each iteration.

Fig. 5. Results for label strategies {LP, LU, LR} and worker strategies {WE, WR} with queried labels $l = 3$. Instances selection is fixed as QCI.

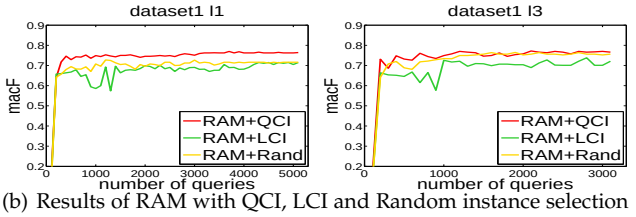
For the results of $l = 3$ in Figure 5, we get similar comparison between the strategies, but with smoother and faster converged results (3000 queries compared to 5000 for $l = 1$), especially on *dataset2* with a rather larger label set. From this we recommend $l = 3$ for practical usage, which is a moderate number of labels for the workers to tag, and at the same time makes the learning process more efficient.

Results of Comparison II The results of Comparison II are shown in Figure 6. We omit the similar results for *dataset2* due to space limitation. From Figure 6 we can see that, with label and worker selection fixed as LP and WE, QCI consistently performs better than random and LCI, but not that significant as the effect of label and worker selection. It's noteworthy that random selection here is a rather strong choice, which actually is not rare case for multi-label active learning tasks. The reason is probably because the informativeness of multi-label data could be ambiguously distributed over the instance set, due to the intrinsic simultaneous concern of multiple label, and random selection gives even chance to instances which may capture more valuable information.

Comparing the effects of instance, label, and worker strategies, from Figure 4-5, 6, we can see that the label and worker strategies play significant roles in finding the helpful annotations. With label and worker selection fixed as LP and WE, the instance can be selected with more flexibility. For example, in scenarios where efficiency is more concerned, random instance can be used which is very fast compared to active instance selection.



(a) Results of NAM with QCI, LCI and Random instance selection



(b) Results of RAM with QCI, LCI and Random instance selection

Fig. 6. Instance selection strategy study for NAM and RAM on *dataset1*, with top ranked queried label $1, 3$.

8 Conclusions and Future Work

In this paper, we deal with multi-label crowdsourcing learning and propose two approaches NAM/RAM (Neighborhood/Relevance Aware Multi-label crowdsourcing) modeling the crowds' expertise and label correlations from different perspectives. Based on the idea that instances similar in the feature space should share similar annotations, NAM

models the crowds' expertise on each individual label and utilizing the local influence of neighborhoods' label correlations. RAM captures the workers' *effort-saving* annotating behavior and models their expertise as their ability to distinguish the label relevance. Considering that the labeling budget is always limited, we also extend NAM and RAM to the active learning paradigm, and propose label and worker query strategies to select the most possibly positive labels to query from the most reliable workers.

Currently, we do not pay special attention to spammer workers which provide no beneficial annotations during the labeling process, for future work, we would like to deal with spammer worker filtering. Besides, a number of issues should be concerned in scenarios of large label crowdsourcing: firstly, simply presenting the whole label set to the crowds would be too low-efficient and result in uncontrollable labeling errors; secondly, the computational complexity is a big problem. Thus efficient and quality guaranteed label collection modes and learning algorithms are worth study directions, e.g., richer information collection including but not limited to membership labels, strategies on label grouping or problem reduction to transform the problem into smaller easy to handle subproblems.

ACKNOWLEDGMENTS

Authors want to thank reviewers for their helpful comments and suggestions. This research was supported by the National Science Foundation of China (61333014), the CSC-IBM Y-100 Program. Zhi-Hua Zhou is the corresponding author.

REFERENCES

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., 2001, pp. 681–687.

[3] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, M. Oded and R. Lior, Eds. Berlin,Germany: Springer, 2010, pp. 667–685.

[4] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[5] E. Horvitz, "Reflections on challenges and promises of mixed-initiative interaction," *AI Magazine*, vol. 28, no. 2, pp. 13–22, 2007.

[6] D. Weld, C. Lin, and J. Bragg, "Artificial intelligence and collective intelligence," in *The Collective Intelligence Handbook*, T. Malone and M. Bernstein, Eds., 2015.

[7] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, "Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, 2008, pp. 254–263.

[8] V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[9] P. Welinder, S. Branson, S. Belongie, and P. Perona, "The multidimensional wisdom of crowds," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 2024–2432.

[10] L. Chilton, G. Little, D. Edge, D. Weld, and J. Landay, "Cascade: Crowdsourcing taxonomy creation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 1999–2008.

[11] J. Bragg, Mausam, and D. Weld, "Crowdsourcing multi-label classification for taxonomy creation," in *First AAAI conference on human computation and crowdsourcing*, 2013.

[12] Y. Sun, A. Singla, D. Fox, and A. Krause, "Building hierarchies of concepts via crowdsourcing," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 844–853.

[13] L. Duan, S. Oyama, M. Kurihara, and H. Sato, "Crowdsourced semantic matching of multi-label annotations." in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 3483–3489.

[14] 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.

[15] N. T. Tam, H. H. Viet, N. Q. V. Hung, M. Weidlich, H. Yin, and X. Zhou, "Multi-label answer aggregation for crowdsourcing," *Technique Report*, 2016.

[16] F. D. Comite, R. Gilleron, and M. Tommasi, "Learning multi-label altenating decision tree from texts and data," in *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, 2003, pp. 35–49.

[17] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Hierarchical classification: Combining bayes with SVM," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006, pp. 177–184.

[18] W. Bi and J. Kwok, "Multi-label classification on tree- and dag-structured hierarchies," in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, 2011, pp. 17–24.

[19] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[20] Y. Yang and S. Gopal, "Multilabel classification with meta-level features in a learning-to-rank framework," *Machine Learning*, vol. 88, no. 1-2, pp. 47–68, 2012.

[21] M.-L. Zhang, "Lift: Multi-label learning with label-specific features," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, Spain, 2011, pp. 1609–1614.

[22] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *Proceedings of the 24th AAAI Conference on Artificial Intelligences*, Atlanta, GE, 2010, pp. 593–598.

[23] F. Zhao and Y. Guo, "Semi-supervised multi-label learning with incomplete labels," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 4062–4068.

[24] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011, pp. 2801–2808.

[25] A. Goldberg, X. Zhu, B. Recht, J.-M. Xu, and R. Nowak, "Transduction with matrix completion: Three birds with one stone," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., 2010, pp. 757–765.

[26] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, Z. Ghahramani, and K. Weinberger, Eds., 2013, pp. 2301–2309.

[27] 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 *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., 2009, pp. 2035–2043.

[28] D. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1953–1961.

[29] A. P. Dawid and A. M. Skene, "Maximum likeihood estimation of observer error-rates using the em algorithm," *Journal of the Royal Statistical Society*, vol. 28, no. 1, pp. 20–28, 1979.

[30] V. S. Sheng, F. J. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers." in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, 2008, pp. 614–22.

[31] D. Zhou, S. Basu, Y. Mao, and J. Platt, "Learning from the wisdom of crowds by minimax entropy," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 2195–2203.

[32] Y. Yan, R. Rosales, G. Fung, and J. Dy, "Active learning from crowds," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 1161–1168.
[33] M. Fang, J. Yin, and D. Tao, "Active learning for crowdsourcing using knowledge transfer," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 1809–1815.
[34] S. Ertekin, C. Rudin, and H. Hirsh, "Approximating the crowd," *Data Mining and Knowledge Discovery*, vol. 28(5-6), pp. 1189–1221, 2014.
[35] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Multi-label active learning from crowds," *CoRR*, vol. abs/1508.00722, 2015.
[36] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data," *Journal of the Royal Statistical Society - B*, vol. 39, no. 1, pp. 1–38, 1977.
[37] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz, "Pairwise ranking aggregation in a crowdsourced setting," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, Italy, 2013, pp. 193–202.
[38] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.
[39] M. Singh, E. Curran, and P. Cunningham, "Active learning for multi-label image annotation," in *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*, 2008.
[40] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *ICML*, 2000.
[41] S.-J. Huang and Z.-H. Zhou, "Active query driven by uncertainty and diversity for incremental multi-label learning," in *Proceedings of the 13th IEEE International Conference on Data Mining*, 2013, pp. 1079–1084.
[42] X. Li and Y. Guo, "Active learning with multi-label svm classification," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 1479–1485.
[43] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, "Two-dimensional active learning for image classification," in *Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.
[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
[45] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multilabel classification of music into emotions," in *International Conference on Music Information Retrieval*, 2008.
[46] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
[47] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., 2001, pp. 681–687.

Shao-Yuan Li is a PhD candidate in computer science from Nanjing University. Her research interest is mainly on machine learning and data mining. She is currently working on learning with incomplete/nonperfect information, specifically on multi-label and multi-view problems. She has won the Grand and SME Segment winner of PAKDD 2012 Data Mining Competition.

Yuan Jiang received the PhD degree in computer science from Nanjing University, China, in 2004. She was selected in program for New Century Excellent Talents in University of Ministry of Education of China in 2009. Currently she is a full professor in the department of Computer Science and Technology at Nanjing University. Her research interests are mainly on machine learning and data mining. She is currently working on multi-label learning , multi-view learning and multi-instance learning. She has published over 50 papers in leading international/national journals and conferences.

Nitesh V. Chawla is the Frank M. Freimann Professor of Computer Science and Engineering at University of Notre Dame. He is the director of the Notre Dame Interdisciplinary Center for Network Science (iCeNSA). He is the recipient of the 2015 IEEE CIS Outstanding Early Career Award for 2015, the IBM Watson Faculty Award in 2012, and the IBM Big Data and Analytics Faculty Award in 2013, the National Academy of Engineering New Faculty Fellowship, the Rodney Ganey Award in 2014 and Michiana 40 Under 40 in 2013. He has also received and nominated for a number of best paper awards. He serves on editorial boards of a number of journals and organization/program committees of top-tier conferences. He is also the director of ND-GAIN Index, Fellow of the Reilly Center for Science, Technology, and Values, the Institute of Asia and Asian Studies, and the Kroc Institute for International Peace Studies at Notre Dame.

Zhi-Hua Zhou (S'00-M'01-SM'06-F'13) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as an Assistant Professor in 2001, and is currently Professor, Head of the Department of Computer Science and Technology and Dean of the School of Artificial Intelligence; he is also the Founding Director of the LAMDA group. His research interests are mainly in artificial intelligence, machine learning and data mining. He has authored the books *Ensemble Methods: Foundations and Algorithms* and *Machine Learning* (in Chinese), and published more than 150 papers in top-tier international journals or conference proceedings. He has received various awards/honors including the National Natural Science Award of China, the CCF WangXuan Award, the PAKDD Distinguished Contribution Award, the IEEE ICDM Outstanding Service Award, the IEEE CIS Outstanding Early Career Award, the Microsoft Professorship Award, etc. He also holds 22 patents. He is an Executive Editor-in-Chief of the *Frontiers of Computer Science*, Associate Editor-in-Chief of the *Science China Information Sciences*, Action or Associate Editor of the *Machine Learning*, *IEEE Trans. Pattern Analysis and Machine Intelligence*, *ACM Trans. Knowledge Discovery from Data*, etc. He served as Associate Editor-in-Chief for *Chinese Science Bulletin* (2008-2014), Associate Editor for Associate Editor for *IEEE Trans. Knowledge and Data Engineering* (2008-2012), *IEEE Trans. Neural Networks and Learning Systems* (2014-2017), *ACM Trans. Intelligent Systems and Technology* (2009-2017), *Neural Networks* (2014-2016), etc. He founded ACML (Asian Conference on Machine Learning), served as Advisory Committee member for IJCAI (2015-2016), Steering Committee member for ICDM, PAKDD and PRICAI, and Chair of various conferences such as General co-chair of PAKDD 2014 and ICDM 2016, Program co-chair of SDM 2013 and IJCAI 2015 Machine Learning Track, and Area chair of NIPS, ICML, AAAI, IJCAI, KDD, etc. He is/was the Chair of the IEEE CIS Data Mining Technical Committee (2015-2016), the Chair of the CCF-AI(2012-), and the Chair of the Machine Learning Technical Committee of CAAI (2006-2015). He is a foreign member of the Academy of Europe, and a Fellow of the ACM, AAAI, AAAS, IEEE, IAPR, IET/IEEE, CCF and CAAI.