Neurocomputing 151 (2015) 891–904

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning classifiers from dual annotation ambiguity via a min–max framework

Jinbo Bi *, Xin Wang

Department of Computer Science and Engineering, University of Connecticut, 371 Fairfield Way U4155, Storrs, CT 06269-4155, USA

ARTICLE INFO

Article history:
Received 28 April 2014
Received in revised form
16 August 2014
Accepted 3 October 2014
Communicated by J. Kwok
Available online 14 October 2014

Keywords:
Classification
Data annotation ambiguity
Learning from multiple data annotators
Multiple instance learning

ABSTRACT

Many pattern recognition problems confront two sources of annotation ambiguity where (1) multiple annotators have provided their versions of a class label which may not be consistent with one another, which forms multi-labeler learning; (2) and meanwhile a class label is associated with a bag of input vectors or instances rather than each individual instance and a bag is positive for a class label as long as one of its instances shows an evidence of that class, which is often referred to as multi-instance learning. Existing methods for multi-labeler learning and multi-instance learning only address one source of the labeling ambiguity. They are not trivially feasible to tackle the dual ambiguity problem. We hence propose a novel optimization framework by modifying the hinge loss to employ the weighted consensus of different labelers' labels and further generalizing the notion of loss functions to bags of multiple instances. The proposed formulation can be approximately solved by two mathematically tractable models that accommodate two types of labeling bias. An alternating optimization algorithm has been derived to efficiently solve the two models. The proposed algorithms outperform existing methods on benchmark data sets collected for document classification, real-life crowd-sourced data sets, and a medical problem of heart wall motion analysis with diagnoses from multiple radiologists.

1. Introduction

In a variety of real-world problems, ambiguous and inconsistent annotations of data exist inevitably and bring an important set of machine learning problems associated with the efficient modeling and utilization of ambiguous supervision. Data annotation becomes ambiguous often due to both the labor-intensive and time-consuming nature in the labeling process and the difficulty of the annotation tasks themselves. The mechanism that causes labeling ambiguity varies from problem to problem, and multiple causes of ambiguity can exist in a single problem in many practical domains.

Human linguistic annotation is crucial for many natural language processing (NLP) tasks but can be expensive and time-consuming. Crowdsourcing methods, such as Amazon Mechanical Turk sys tem [1] and Crowdflower system [2], recruit non-expert annotators to label the text documents with a relatively low cost. Internet annotators, however, can provide notoriously inconsistent labels to a document [3,4]. In medical applications, medical images are often assessed by multiple radiologists to improve the diagnostic accuracy of abnormality. Studies have shown a significant variation between annotators in the interpretation of diagnostic images [5]. This inter-labeler variation leads to a source of ambiguity in the supervision labels.

In document classification with respect to a focused topic, a document may contain multiple passages that either cover the corresponding topic or only relate to other topics. Consider a document as a bag comprising several passages as its instances. A document is often assigned to a topic category as long as one of its passages or instances is relevant to the topic. When we try to classify a document, it would be very important to identify the specific passage in the document that corresponds to the given topic. An image can be represented as a bag of different regions and can be associated with the objects that each region dictates. This type of ambiguous annotation leads to the so-called multiple instance learning problem and usually has labeling bias in between positive and negative classes as positive labels are commonly based on evidence validation whereas negative labels indicate either true negative or lack of knowledge.

Many practical learning problems present multi-instance examples that are labeled by multiple annotators. For instances, a document can be labeled by many internet labelers in terms of whether it is relevant to a particular topic. Some labelers may recognize the passages in the document that correspond to the topic, whereas others may not, resulting in inconsistent annotati ons from these labelers. Moreover, if a document is labeled negative

* Corresponding author. Tel.: +1 860 486 1458; fax: +1 860 486 4817.
E-mail address: jinbo@engr.uconn.edu (J. Bi).

http://dx.doi.org/10.1016/j.neucom.2014.10.005
0925-2312/

for a specific topic, it may be truly absent of the topic but can also indicate that the labeler fails the evidence search. When multiple labelers annotate if an image contains a specific object, they may perceive different regions of the image. Hence, some may give positive labels whereas others label it negative for the object, leading to a disagreement in the annotation. An example's true label becomes a latent variable, and multiple versions of its value are given. In this paper, we solve the problem of constructing a classifier based on the different versions of a class label to predict if a multi-instance example is associated with the class label, and to identify the instances responsible for the class membership. Fig. 1 illustrates this challenging problem.

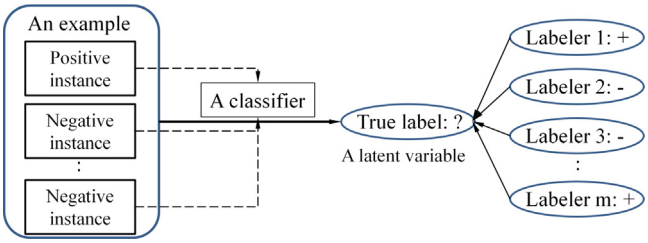


Fig. 1. The problem of constructing a classifier from a training data set where each example is annotated by multiple labelers and the true label is unknown. This classifier is also expected to identify the instances of the example that are responsible for the class assignment (i.e., the positive instances).

To the best of our knowledge, existing multiple instance learning algorithms [6–9] do not cope with the labeling inconsistency if multiple human experts have labeled the multi-instance examples. Our problem also differs from the multi-instance multi-label (MIML) learning problems [10–12] in which each bag as an example may correspond to multiple labels when the example is labeled based on different concepts. These labels are all considered as accurate labels for the example: for instance, in image annotation tasks, a picture of landscapes may contain the sky, mountain or trees simultaneously, so this picture may correspond to all these labels. In contrast, for our problem, an example is labeled based on a single concept, corresponding to one class label, but multiple inconsistent versions of this class label are given rather than an accurate label (which we call the true label). Note that some versions may be *incorrect* labels. The state of the art in multi-labeler learning methods [13–17] can estimate a true label from the different versions of the label given by different labelers, but are not feasible to cope with examples of multiple instances, especially when different examples consist of different numbers of instances. In summary, none of the existing methods have addressed the dual ambiguity issue. Therefore, in this paper we propose an approach to integrate expertise from multiple labelers and build classifiers that are able to classify bags of instances, or multi-instance examples with respect to the estimated true label and identify true positive instances for positive bags. The major contributions of this paper are as follows:

- Propose a mechanism to learn the consensus label from multiple labelers by modifying the hinge loss which is commonly used in support vector machines [18].
- Extend the modified hinge loss to bags of multiple instances with a theoretical analysis to the resulted optimization problem.
- Two relaxation models are derived that properly approximate the original optimization formulation based on different assumptions on labeling bias of different labelers.
- Develop an alternating optimization algorithm to solve the two models which show superior performance in solving the dual annotation ambiguity problem.

The paper is organized as follows. We briefly review existing methods in the relevant areas: multi-instance learning (MIL), multi-instance multi-label learning (MIML), and learning from multiple data annotators in Section 2. Section 3 is dedicated to the derivation of our proposed approach. In Section 3.1, we propose a bi-convex program to simultaneously estimate reliabilities of labelers and construct classifiers by taking weighted consensus of labels from different labelers. Section 3.2 gives a min–max optimization program that effectively deals with labels given only at the bag level instead of the instance level. Section 3.3 derives two tractable approximation models to the proposed formulation and describes an alternating optimization algorithm that solves the two models efficiently. Extensive computational experiments have been conducted and results are included in Section 4 to demonstrate the performance of the proposed approach. We give conclusion and discussions in Section 5.

2. Related works

Although there has been an increasing amount of literatures related to the ambiguous annotation problem, no existing method can effectively address the dual annotation ambiguity problem of our concern.

2.1. Existing methods for multiple instance learning

To date, many methods have been proposed to solve MIL problems (see a recent review in [19]). These methods can be roughly divided into two categories: generative and discriminative approaches. Early methods are dominated by generative approaches which aim at locating a target region in the instance feature space so that all positive instances lie in its vicinity and all negative instances are far away from it. These methods include axis-parallel rectangles [20], the diverse density (DD) method [21], EM-DD (the expectation–maximization alternative of DD) [22], and the generalized EM-DD [23] together with their related theoretical results [24]. The main drawback against generative approaches is the use of single or a limited number of regions (prototypes) to represent the target concept for the positive class, which may not be valid in practice.

Later methods work primarily towards a discriminative scheme that adapts standard supervised learning approaches to the multiple instance setting. The k-nearest neighbor (kNN and Bayesian kNN [25]), neural networks [26], boosting approaches ([27], decision trees ([28], logistic regression [29], and support vector machine (SVM) [30,9,31,32] have all been generalized from their single instance counterparts to ambiguously supervised multiple instance learning. Although the generalized formulations perform competitively on MIL problems, none of them can simultaneously estimate the true labels from different labeler-annotations when building multi-instance classifiers. These methods often assume a distribution model over the instances in an example and link this model to the (accurate) class labels of each bag (example). The statistical model will not work in the scenario when true labels are not given.

2.2. Existing methods for multiple instance multiple label learning

Multi-instance multi-label learning is firstly formalized in [10], where each training example is associated with multiple instances and multiple class labels. All of the labels are truly associated with the examples. MIML learning aims to correctly predict all of the labels and tries to find every corresponding label for an example.

The early methods [10] either transform the multiple labels of an example into binary labels, which indicate whether an individual label is assigned to an example, and then solve the MIML

problem using regular MIL methods, or transform the multi-instance examples to be single instances and then solve the MIML problem using methods that construct multiple classifiers jointly for all labels. Many algorithms have been proposed for MIML since then. MIML problems have been addressed by statistical models [33,34], by metric learning [11], or via the revision of SVM algorithms by exploring the connections between the instances and labels of an MIML example [35–37]. Recent work studies the situations in which there are unlabeled examples [38] or the examples with partially untagged labels [39].

Even though the MIML learning algorithms allow a single example to have more than one corresponding labels, these labels are based on different concepts, and are all accurate labels that can be directly used in a learning formulation. They cannot address noisy labels where the true labels need to be derived or estimated from disagreeing versions of labelers' labels.

2.3. Existing methods for multiple labeler learning

The work related to learning from multiple annotators can be divided into two sub-areas. One area of the work focuses on modeling of the annotation process and estimating true labels and error rates of the labelers independent of any classifiers. The early statistical methods [40–43] on error rate estimation for repeated but conflicting test results, and the recent work on evaluating the quality of the crowdsourced annotation [3,4,44], fall in this area. The latest work ranks annotators to identify spammers [44], uses Multinomial probabilistic models to quantify the competency of each labeler [45], and parameterizes labeler expertise or reliabilities as well as the difficulty of an annotation task in order to model human annotation process [46–48,13]. However, these methods do not aim to construct a classifier that is based on the features of the examples to predict their labels.

Multi-labeler learning (MLL) has recently moved to classifier estimation from multi-labeler-annotated data. Repeated labeling methods [49–51] identify the labels that should be reacquired from some labelers in order to improve classification performance or data quality. A recent theoretical work [52], however, argues that the repeated labeling *negatively* impacts the relative size of the training sample. Another set of approaches [53,54] assumes the existence of prior knowledge relating the different labelers, and the prior is used to identify the samples for each labeler that are appropriate to be used in the classifier estimation. The latest methods [55,14,15,56], however, assume neither that labels can be reacquired, nor existence of any prior on labeler relations. These approaches rely on certain data distribution, such as Bernoulli model or Gaussian model on the true labels [56] or two-coin model for annotators [15], and design an expectation–maximization algorithm on logistic regression classifiers. All these methods are not trivially feasible to deal with multi-instance examples so that an example's estimated true (bag-level) label can be linked to appropriate instances in the example, especially when different examples (bags) contain different numbers of instances.

3. Material and methods

In this section, we propose a novel optimization framework that can efficiently address the dual annotation ambiguity problem by modifying the hinge loss to employ the weighted consensus of different labelers' labels and further generalizing the notion of loss functions to bags of multiple instances.

3.1. A bi-convex program for learning classifiers from multiple annotators

In the problem of learning from multiple annotators, an input example $\mathbf{x}_i$ in training data is annotated with multiple versions $\{y_i^1, y_i^2, \ldots, y_i^m\}$ of the label y_i. Let $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ comprise the n training examples, where $\mathbf{x}_i \in R^d$. We focus on the problem of binary classification where $y_i \in \{-1, 1\}$. The labels from different labelers $y_i^j \in \{-1, 1\}$, $j \in \{1, 2, \ldots, m\}$. We derive a new learning model by altering the hinge loss $\xi_i = [1 - y_i(\mathbf{x}_i^\top \mathbf{w} + b)]_+ = \max\{0, 1 - y_i(\mathbf{x}_i^\top \mathbf{w} + b)\}$ commonly used in SVMs where $\mathbf{w}$ is the weight vector and b is the offset of the linear model to be determined.

We approximate an example's golden standard y_i by a weighted combination of each labeler's labels. In other words, we estimate y_i by $\hat{y}_i = \sum_{j=1}^m r_j y_i^j$ and each labeler j is associated with a reliability factor r_j where $0 \le r_j \le 1$. If the reliability factors of all labelers are equal, this combination amounts to the majority voting. If we require additionally $\sum_j r_j = 1$, we approximate y_i by a convex combination of labelers' opinions. These different ways of combinations may all be reasonable, and the most appropriate one may be problem-specific. If the weighted consensus of all labelers $\sum_j r_j y_i^j > 0$, the example i is more likely to be in the class of $y=1$; or otherwise, it likely has a true label of $y=-1$.

We modify the hinge loss by replacing the true labels y_i, which are unknown during classifier training, by the weighted consensus. Thus,

$$\xi_i = \left[1 - \left(\sum_j r_j y_i^j\right)(\mathbf{x}_i^\top \mathbf{w} + b)\right]_+ . \tag{1}$$

When the consistency is high among the labels given by different labelers, especially by reliable labelers, the magnitude of $\sum_j r_j y_i^j$ tends to be large regardless of its sign, showing high annotation confidence for $\mathbf{x}_i$. Minimizing the modified hinge loss Eq. (1) implies to penalize strongly the errors made on the examples $\mathbf{x}_i$ with highly agreed labels. When the labeling consistency is low among reliable labelers for some examples, assigning these examples to either class can be a vague guess. The modified hinge loss, as how it is defined, will give small errors for these examples, and hence the classification performance on these ambiguous examples is not emphasized.

To regularize the empirical hinge loss, we minimize an objective function defined as $\lambda \|\mathbf{w}\|^2 + \sum_i [1 - (\sum_j r_j y_i^j)(\mathbf{x}_i^\top \mathbf{w} + b)]_+$ subject to the bound constraints on $0 \le \mathbf{r} \le 1$ where λ is a tuning parameter to balance between empirical errors and the regularization term $\|\mathbf{w}\|^2$. It is easy to verify that the objective function is bi-convex (i.e., convex with respect to $(\mathbf{w}, b)$ for fixed $\mathbf{r}$ and convex with respect to $\mathbf{r}$ for fixed $(\mathbf{w}, b)$) and the bound constraints give a convex feasible region. This problem forms a special case of bi-convex optimization. Even when we include the additional constraint $\sum_j r_j = 1$ for convex combinations of labelers' opinions. This constraint is affine and hence bi-affine. The resulting problem is still bi-convex. To form a canonical form of the optimization problem, the hinge loss is translated into a constraint $(\sum_j r_j y_i^j)(\mathbf{x}_i^\top \mathbf{w} + b) \ge 1 - \xi_i$ for each example i where $\xi_i \ge 0$, and both r and $(\mathbf{w}, b)$ are now variables to be determined in the optimization problem. Overall, we search for the best $\mathbf{w}, b, \mathbf{r}$ by optimizing the following problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi},\mathbf{r}} \quad \lambda \|\mathbf{w}\|^2 + \sum_i \xi_i \\ &\text{s.t.} \quad \left(\sum_j r_j y_i^j\right)(\mathbf{w}^\top \mathbf{x}_i + b) \ge 1 - \xi_i, \\ &\xi_i \ge 0, \quad 0 \le r_j \le 1, \\ &i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m \end{aligned} \tag{2}$$

where we simply use the bound constraints on $\mathbf{r}$ (other constraints can be used if appropriate). Eq. (2) is also a quadratically constrained quadratic optimization problem but with one of its constraints bi-convex. Due to the bi-convexity, efficient algorithms can be derived to approximate an optimal solution. We will discuss an algorithm based on alternating optimization in Section 3.3.

3.2. A min–max program for learning with dual annotation ambiguity

We now derive a learning formulation to address the dual labeling ambiguity issue where multiple experts or non-experts are utilized to annotate training examples, each of which consists of a varying number of instances. We extend the modified hinge loss equation (1) from the instance level to assessing the loss occurred on a bag. In the binary classification MIL, a bag is labeled positive if at least one instance in it is positive, and negative if all the instances in it are negative. The goal of a MIL problem is to distinguish positive bags from negative bags. It is also important to infer the labels for the instances. In the dual annotation ambiguity problem, a bag B_k is labeled with m versions of the bag-level label y_k^j, $j=1,\ldots,m$ and $k=1,\ldots,n$. If we associate with each labeler a reliability factor r_j, the true label of B_k is estimated by $\hat{y}_k=\sum_j r_j y_k^j$. If the combined consensus of all labelers' opinions $\hat{y}_k>0$ for a bag B_k, then B_k is considered to be a positive bag; or otherwise, B_k is a negative bag.

We propose a min–max framework that aims to infer the labels of instances from the estimated bag-level true labels by generalizing the notion of loss functions to bags of multiple instances and minimizing the loss on bags directly. Let B contain the indices of the instances in a bag. Due to the asymmetric logic in the MIL labeling process, if a bag B is "negative", then $y_i=-1,\ \forall i\in B$, which corresponds to an "AND" logic among all of the instances in the bag. If a bag B is "positive", then $\exists i\in B$, such that $y_i=+1$, which corresponds to an "OR" logic among instances in the bag.

Now, let us pre-label all instances in a bag with the bag's estimated true label, or the consensus bag-level label of all labelers. Let ξ_{ik} be the hinge loss of the i-th instance of the k-th bag defined as $\xi_{ik}=[1-(\sum_j r_j y_k^j)(\mathbf{x}_{ik}^\top\mathbf{w}+b)]_+$. If $\xi_{ik}=0$, the i-th instance is correctly classified with respect to $\hat{y}_k$. If $\xi_{ik}>0$, the i-th instance is misclassified or classified without a proper margin. For a *negative* bag, the AND operation requires all instances in the bag to be correctly classified, which requires all hinge errors to be 0. In other words, $\max_{i\in B}\xi_{ik}=0$. For a *positive* bag, the OR operation only requires one ξ to be 0, which amounts to $\min_{i\in B}\xi_{ik}=0$. The min or the max function conditioned on a bag's label can serve as an objective to be minimized for determining instance-level labels.

We thus construct a classifier by minimizing the integrated and regularized loss function for the best parameters $(\mathbf{w},b,\mathbf{r})$, i.e.,

$$\min_{\mathbf{w},b,\mathbf{r},\boldsymbol{\xi}}\ \lambda\|\mathbf{w}\|^2+\sum_{k\in\{k:\hat{y}_k>0\}}\min_{i\in B_k}\{\xi_{ik}\}+\sum_{k\in\{k:\hat{y}_k\le 0\}}\max_{i\in B_k}\{\xi_{ik}\} \tag{3}$$

where $\hat{y}_k=\sum_j r_j y_k^j$ is the bag-level label estimated from different labelers' labels for a bag B_k. This formulation classifies bags by calculating bag-level losses, but ultimately, it infers the labels of instances in a positive bag B_k as $y_p=+1,\ \forall p\in\{p\in B_k|\xi_{pk}=\min_{i\in B_k}\{\xi_{ik}\}\}$, and otherwise $y_p=-1$.

Eq. (3) is, however, mathematically intractable since (a) the index sets involved in the two summation terms rely on the estimated bag labels $\hat{y}_k$, or more precisely, the labeler reliabilities $\mathbf{r}$ that themselves are to be determined; and (b) to evaluate the objective function, it requires the evaluation of the minimum and maximum operations in the two summation terms.

We first prove that the evaluation of the minimum and maximum values in (3) can be completely omitted once estimated true labels are given (or in other words, once the two index sets are determined). Then, we develop relaxed forms of (3) that are tractable formulations and can effectively determine the index sets used in the two summations. Note that when the reliability of a labeler is known and fixed, the estimate of the true labels, $\hat{y}_k=\sum_j r_j y_k^j$, is determined and can be used to distinguish positive bags from negative bags. Then different treatments (min or max) will be used to compute their losses. We prove an equivalence between (3) and the following problem when $\mathbf{r}$ is fixed:

$$\begin{aligned}\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\mu}}\ & \lambda\|\mathbf{w}\|^2+\sum_{k\in\{k:\hat{y}_k>0\}}\sum_{i\in B_k}\mu_{ik}\xi_{ik}+\sum_{k\in\{k:\hat{y}_k\le 0\}}\eta_k\\ \text{s.t.}\ & \hat{y}_k(\mathbf{w}^\top\mathbf{x}_{ik}+b)\ge 1-\xi_{ik},\quad \xi_{ik}\ge 0\\ & \mu_{ik}\ge 0,\quad \sum_{i\in B_k}\mu_{ik}=1,\quad \text{if } \hat{y}_k>0,\\ & \eta_k\ge\xi_{ik},\quad i\in B_k,\quad \text{if } \hat{y}_k\le 0,\\ & i\in B_k,\quad k=1,2,\ldots,n.\end{aligned} \tag{4}$$

where n is the total number of bags in the training set. The proof of this equivalence highlights the equivalence between the logic OR, i.e., $\min_{i\in B_k}\xi_{ik}$, and the convex combination of hinge losses ξ_{ik} in the bag B_k.

Theorem 3.1. *Any optimal solution $(\hat{\mathbf{w}},\hat{b})$ of Problem (3) is optimal to Problem (4) and vice versa when $\mathbf{r}$ is fixed.*

Proof. We first prove that an optimal solution of (4) has nonzero μ's only on the instances for which the classifier $\mathbf{w}^\top\mathbf{x}+b$ achieves $\min\{\xi_{ik},i\in B_k\},\ \forall k\in\{k:\hat{y}_k>0\}$.

Let $(\hat{\mathbf{w}},\hat{b},\hat{\boldsymbol{\xi}},\hat{\boldsymbol{\mu}})$ be the optimal solution of (4). For notational convenience, denote the objective function of (4) as $\mathcal{J}(\mathbf{w},b,\boldsymbol{\xi},\mu)$. Then let $\hat{\mathcal{J}}$ be the objective value attained at the optimal solution. Notice that the hinge loss $\hat{\boldsymbol{\xi}}$ is uniquely determined by $(\hat{\mathbf{w}},\hat{b})$ as $\hat{\xi}_{ik}=\max\{0,1-\hat{y}_k(\mathbf{w}^\top\mathbf{x}_{ik}+b)\}$ for each instance $\mathbf{x}_{ik}\in B_k$.

If $\exists k_0\in\{k:\hat{y}_k>0\}$, and $\exists i_0\in B_{k_0}$, such that $\hat{\mu}_{i_0k_0}>0$ but $\hat{\xi}_{i_0k_0}\ne\min\{\xi_{ik_0},i\in B_{k_0}\}$. Then let $\tilde{\xi}_{pk_0}=\min\{\xi_{ik_0},i\in B_{k_0}\}<\hat{\xi}_{i_0k_0}$. Then, re-set $\tilde{\mu}_{pk_0}=1$ and $\tilde{\mu}_{i_0k_0}=0$. Now, $\tilde{\mathcal{J}}=\hat{\mathcal{J}}-\hat{\mu}_{i_0k_0}\hat{\xi}_{i_0k_0}+\tilde{\mu}_{pk_0}\tilde{\xi}_{pk_0}<\hat{\mathcal{J}}$. This contradicts to the optimality of $(\hat{\mathbf{w}},\hat{b},\hat{\boldsymbol{\xi}},\hat{\boldsymbol{\mu}})$.

By this contradiction, $\forall i,k$, if $\mu_{ik}>0$, the corresponding ξ_{ik} has to be the minimum loss that the classifier achieves on the k-th bag. This implies that at the optimality, the objective of (4) is exactly equal to

$$\mathcal{J}=\lambda\|\mathbf{w}\|^2+\sum_{k\in\{k:\hat{y}_k>0\}}\min_{i\in B_k}\{\xi_{ik}\}+\sum_{k\in\{k:\hat{y}_k\le 0\}}\max_{i\in B_k}\{\xi_{ik}\}.$$

To prove the other direction, let $(\hat{\mathbf{w}},\hat{b},\hat{\boldsymbol{\xi}})$ be the optimal solution of (3). We can simply define $\mu_{ik}=1$, if ξ_{ik} achieves the smallest hinge loss over the bag B_k, or otherwise $\mu_{ik}=0$, for all bags where $\hat{y}_k>0$. Following the same line of thought, we can prove that $\boldsymbol{\mu}$ is optimal to (4), and the solution $(\hat{\mathbf{w}},\hat{b},\hat{\boldsymbol{\xi}},\hat{\boldsymbol{\mu}})$ is optimal to (4). □

Eq. (4) is also a bi-convex quadratic program where the objective function is bi-convex in the sense that it is convex with respect to $(\mathbf{w},b)$ for fixed $\boldsymbol{\mu}$ and is convex with respect to $\boldsymbol{\mu}$ for fixed $(\mathbf{w},b)$. All constraints of (4) are affine and hence bi-affine.

3.3. Solving the proposed formulation

If we design an iterative algorithm to optimize the proposed formulation (3), the most significant challenge is how to tackle the stochastic nature of the objective function. The objective function of (3) is stochastic (not deterministic) because the min and max functions are calculated on different sets of bags in different iterations if $\mathbf{r}$ varies in the iterations. The difficulty hence lies in the determination of r's because varying their values would alter the decision of a bag's label, and correspondingly alter the objective

function. We hence develop relaxed forms of (3) that are tractable, and approximate but effective solutions can be efficiently obtained.

In an alternating optimization process, we solve (3) by alternating between solving two sub-problems: one sub-problem is optimized for the best classifier characterized by $(\mathbf{w}, b)$ with a fixed choice of reliabilities $\mathbf{r}$; the other sub-problem is optimized for the best $\mathbf{r}$ after obtaining a classifier. The slack variables $\boldsymbol{\xi}$ that measure the hinge losses will need to be optimized in both sub-problems because they vary when either $(\mathbf{w}, b)$ or $\mathbf{r}$ is changed.

3.3.1. Sub-problem 1: building a MIL classifier when labeler reliabilities are known

If the reliability r of a labeler is known and fixed, Eq. (4) is optimized for the best classifier $(\mathbf{w}, b)$. The parameters $\boldsymbol{\mu}$ are also correspondingly optimized in order to calculate proper bag-level hinge losses. An alternating optimization procedure can be developed to solve (4) that alternates between solving two smaller sub-problems: one is to fix $\boldsymbol{\mu}$ in (4) for the best $(\mathbf{w}, b)$ and the other is to fix $(\mathbf{w}, b)$ in (4) for the best $\boldsymbol{\mu}$. The first sub-problem is a convex quadratic program similar to the standard SVM, and can hence be solved efficiently. The second sub-problem has an analytical solution and the optimal $\boldsymbol{\mu}$ can be directly obtained by searching for the smallest ξ_{ik} for each bag B_k with $\hat{y}_k > 0$ and setting the corresponding $\mu_{ik} = 1$ and other μ's to 0.

3.3.2. Sub-problem 2: determining a labeler's reliability when the instance-level predictions are known

If we fix the classifier parameters $(\mathbf{w}, b)$, the predicted value $\mathbf{w}^\top \mathbf{x} + b$ of every instance $\mathbf{x}$ is hence determined. The only variables in (3) comprise the reliabilities of each labeler that ultimately determine which bag should use the min loss and which bag should use the max loss. Converting (3) into a canonical optimization formulation yields

$$\begin{aligned}
&\min_{\mathbf{r},\boldsymbol{\xi},\boldsymbol{\mu}} \quad \sum_{k\in\{k:\hat{y}_k>0\}} \sum_{i\in B_k} \mu_{ik}\xi_{ik} + \sum_{k\in\{k:\hat{y}_k\le 0\}} \eta_k \\
&\text{s.t.} \quad \left(\sum_j r_j y_k^j\right)(\mathbf{w}^\top \mathbf{x}_{ik} + b) \ge 1 - \xi_{ik}, \quad \xi_{ik} \ge 0 \\
&\mu_{ik} \ge 0, \quad \sum_{i\in B_k} \mu_{ik} = 1, \quad \text{if } \hat{y}_k > 0, \\
&\eta_k \ge \xi_{ik}, \quad i \in B_k, \quad \text{if } \hat{y}_k \le 0, \\
&i \in B_k, \quad k = 1, 2, \ldots, n. \qquad (5)
\end{aligned}$$

Eq. (5) is still difficult to solve as the index sets on $\boldsymbol{\mu}$ and $\boldsymbol{\xi}$ depend on the values of $\mathbf{r}$. Two options exist to approximately solve Sub-problem 2. The first option, which we call the *all_min* model, is to estimate r's so all the bags' labels reflect at least the predicted value (by the current classifier) of one of its instances, which corresponds to applying an "OR" operation to each bag regardless the estimated bag labels.

The second option is to compute r's in such a way that we continue to apply the "OR" operation to those bags estimated to be positive (i.e., $\hat{y}_k = \sum_j r_j y_k^j > 0$) in Sub-problem 1, and apply the "AND" operation to those estimated to be negative. It implies that we will choose r's so that the new consensus bag labels can be tuned towards what the current classifier predicts. This option is a commonly used strategy in iterative algorithms, which means we fix $\hat{y}_k$ in (5) by the one obtained in the previous iteration $\hat{y}_k^{old}$. We name this option the *selective min_max* model.

For the *all_min* option, only the single instance with the smallest hinge loss from each bag is used for optimizing r's. For the *selective min_max* option, based on the current estimated labels in Sub-problem 1, each positive bag yields one instance to be used in Sub-problem 2 whereas all instances in the negative bags are used in updating r's. This option employs the min and max losses according to the estimated bag labels used in Sub-problem 1. Hence, it will update the labelers' reliabilities to best reflect the current estimate of bag labels. These two relaxation options of (3) lead to mathematically tractable problems that can be solved efficiently, and their solutions reflect different assumptions on the labeling bias of the labelers.

Labeling bias is commonly observed in MIL tasks, the *all_min* option takes the effect that positive labels may be more accurate than negative labels. Positive labels are commonly due to the recognition of an evidence for a class label. If a labeler annotates a bag with $y_k = +1$, it is likely that this labeler has witnessed an evidence from the bag. In contrast, a negative label may be only due to an insufficient evidence search. The *all_min* model treats any bag as potentially a positive bag if one of its labelers gives $+1$ and one of its instances satisfies $\mathbf{w}^\top \mathbf{x}_{ik} + b > 0$. It amounts to requiring an instance-level "OR" logic over all bags so as to minimize only the smallest hinge loss occurred on each bag regardless of their labels. This leads us to the following optimization problem:

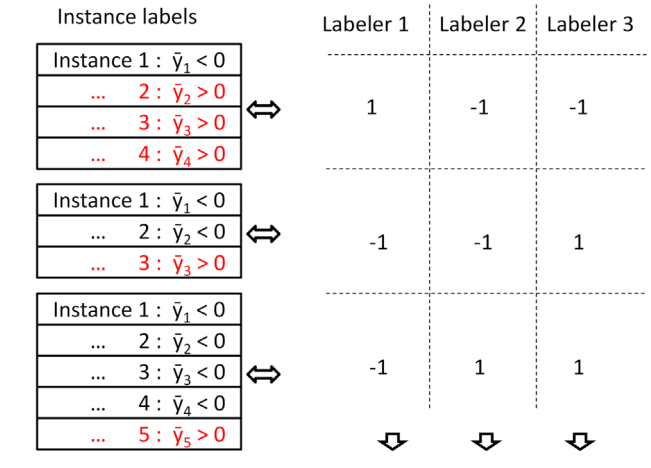
$$\begin{aligned}
&\min_{\mathbf{r},\boldsymbol{\xi}} \quad \sum_{k=1}^{n} \min_{i\in B_k} \xi_{ik} \\
&\text{s.t.} \quad \left(\sum_j r_j y_k^j\right)(\mathbf{w}^\top \mathbf{x}_{ik} + b) \ge 1 - \xi_{ik}, \quad \xi_{ik} \ge 0, \\
&i \in B_k, \quad k = 1, 2, \ldots, n. \qquad (6)
\end{aligned}$$

However, the *all_min* model is a strong relaxation to the original formulation where for a negative bag, all its instances need to agree with the bag label. The *selective min_max* model takes the effect to leverage the classifier outputs (i.e., $\mathbf{w}^\top \mathbf{x} + b$) of all instances in a currently estimated negative bag. An instance's predicted label is determined as $+1$ if $\mathbf{w}^\top \mathbf{x} + b > 0$ or -1 otherwise. The *selective min_max* model enforces the consensus labels, i.e., the estimate of groundtruth, to be as consistent as possible to the predicted labels of all instances in the negative bags returned by Sub-problem 1.

3.3.3. Demonstration of the different model effects

We have implemented the two models (the *all_min* and *selective min_max* models) to be used in Sub-problem 2. The two models create distinct effects on the estimated true labels. Fig. 2 uses a simple example to illustrate the difference between the results of the two models.

As shown on the right-hand side of Fig. 2, three labelers annotated three bags that contained varying numbers of instances.



Reliabilities	all_min	low	high	fair
	selective min_max	high	fair	low

Fig. 2. Demonstration of the effects of the *all_min* and *selective min_max* models in the estimation of true bag labels. Three bags containing different numbers of instances are each labeled by three labelers.

Assume that a classifier predicts a label (according to whether $\bar{y} > 0$) for each instance as shown on the left-hand side of Fig. 2, and this classifier is built using majority voted labels, or in other words, using the estimated true labels $\sum_j r_j y_k^j$ where r_j's are equal across different labelers. Under this circumstance, the *all_min* model will produce the highest reliability factor for the second labeler by optimizing (6). It is because the second labeler gives labels, $[-1, -1, 1]$, to the bags, that align well with the majority voted labels. The *all_min* model minimizes the sum of the hinge losses occurred on only one of the instances in each bag. These selected instances obtain the smallest hinge loss in each bag. More precisely, these instances have their predicted labels agree with the estimated bag labels. Hence, only the first instance in Bag 1, the first or the second instance in Bag 2 and the last instance in Bag 3 would be used to update the reliability factors.

The *selective min_max* model, however, will raise the first labeler to be most reliable. This model solves (5) (with $\hat{y}_k$ replaced by the currently estimated $\hat{y}_k^{old}$) to update each labeler's reliability. Because the third bag is currently estimated as positive, the *selective min_max* model will use the min loss on this bag which means that the estimated bag label should be consistent with the label of just one of its instances. The estimated labels of the first and the second bags are both $y = -1$. The max loss is used over the instances in each bag, which requires $\sum_j r_j y_k^j$ to be in concordance with the predicted labels of a majority of the instances in the bags. The first labeler gives labels $[1, -1]$ for these two bags, which are consistent with 3 out of 4 instances in Bag 1 and 2 out of 3 instances in Bag 2. The consistency of this labeler is the highest among all labelers. If this labeler's r is raised to a high value but his/her label on the third bag (which is -1) becomes dominating in the sum $\sum_j r_j y_k^j$, then the sign of this sum may flip. In this case, because the min loss is used on the third bag, the *selective min_max* model will select one of the instances, i.e., the one that has been predicted the most negative, which will still lead to a small bag-level loss. Hence, the first labeler will receive a high r value in the *selective min_max* model.

3.3.4. The proposed alternating algorithm

Notice that there are two smaller sub-problems involved in solving Sub-problem 1. We develop an alternating optimization strategy that solves the three sub-problems (two from Sub-problem 1) in turns until reaching a fixed point. The resultant algorithm will output a classifier that can be applied to each instance and at the same time assess the labelers' reliabilities which are used to estimate the true bag-level labels. The first Sub-problem solves for $(\mathbf{w}, b)$ with a fixed $\mathbf{r}$, and the second Sub-problem optimizes with respect to $\mathbf{r}$ when the classifier $(\mathbf{w}, b)$ is fixed. Algorithm 1 depicts the details of our algorithm which is implemented separately for the *all_min* (Problem (6)) and *selective min_max* (Problem (5)) models.

Algorithm 1. An alternating algorithm for dual ambiguity problems.

Input: $\mathbf{X}$ with bag index sets, all y_k^j's and λ
Output: $\mathbf{w}$, b and $\mathbf{r}$

1. Initialize $\mathbf{r} =$ a constant (evenly assigned to labelers).
2. Determine the bag labels by the weighted consensus $\sum_j r_j y_k^j$ based on the current values of r's.
3. Compute μ by finding the smallest ξ for each positive bag and setting the corresponding $\mu = 1$ and other $\mu = 0$. (In the initial step, set μ to $1/|B_k|$ for each positive bag.)
4. Solve (4) with fixed μ for the best $(\mathbf{w}, b)$.
5. Solve the *all_min* model (6) (or the *selective min_max* model (5) with $\hat{y}_k$ replaced by previously obtained $\hat{y}_k^{old}$) with fixed $(\mathbf{w}, b)$ for $\mathbf{r}$.

Repeat steps 2–5 until $(\mathbf{w}, b)$ reaches a fixed point.

Since (3) has a stochastic objective function which varies due to the random effect between taking the min or the max loss determined by the value of the random variables $\mathbf{r}$, convergence analysis of our algorithms is difficult. We will leave it for the future to study techniques in stochastic combinatorial optimization [57,58] to estimate the probability of the algorithm convergence in a theoretical form. Empirically, the proposed algorithms can terminate at a fixed point for the two approximation models within 20 iterations in all our experiments. Although the proposed Algorithm 1 solves two relaxed variants of (3), it produces better classifiers than either regular MIL algorithms or multi-labeler learning algorithms by actively and effectively handling both sources of the ambiguity as shown in our experiments.

We briefly analyze the time complexity of Algorithm 1 by evaluating the computation cost at each iteration. Let ℓ, ℓ^+ and ℓ^- be the numbers of total instances, and the instances in the predicted positive and negative bags, respectively. Let n, n^+ and n^- be the numbers of all bags, and the positive and negative bags predicted at the current iteration, and d be the number of features for each instance. In Algorithm 1, three sub-problems are solved at each iteration. The first sub-problem finds $(\mathbf{w}, b)$ by solving (4). Eq. (4) is a convex quadratic program. It uses one instance in each positive bag and all instances in a negative bag to compute slack variables $\boldsymbol{\xi}$, so there are $n^+ + \ell^-$ slack variables. The problem dimension is $\tilde{d} = d + 1 + n^+ + \ell^-$. The second sub-problem finds $\boldsymbol{\mu}$ once $(\mathbf{w}, b)$ and $\mathbf{r}$ are fixed. This step has an analytical solution which only requires to scan through the instances in positive bags, so it requires a computation cost of $O(\ell^+)$. The last sub-problem is a linear program to optimize $\mathbf{r}$ and update slack variables $\boldsymbol{\xi}$, and the problem dimension is $\tilde{d} = m + n$ for the *all_min* model and $\tilde{d} = m + n^+ + \ell^-$ for the *selective min_max* model. We used the simplex method and a simplex-based active set method in the CPLEX optimization software [59] to, respectively, solve the linear and quadratic programs. Although the simplex method has the exponential worst-case complexity [60], its average-case complexity is only polynomial [61,62]. For instance, by assuming a spherically symmetric distribution on the constraint coefficients, a widely used polynomial upper bound on the complexity of simplex was obtained as $\tilde{d}^{2.5}\tilde{n}^{1/(\tilde{d}-1)}$ where $\tilde{n}$ is the number of constraints in the program [63]. It is well known that the simplex method performs very efficiently in practice, which is the case shown in our empirical study (see Section 4.4). Given Algorithm 1 typically stops after 20 iterations, its time complexity would approximately be a constant times the order of complexity of simplex.

4. Computational results

We implemented Algorithm 1 in Matlab where (4)–(6) were solved by calling CPLEX optimization solvers. We tested the proposed approach against the state of the art on several benchmark data sets from the natural language processing (NLP) domain, real-world crowdsourced data sets generated from human facial expression images and a medical problem that used echocardiograms for heart wall motion analysis (HWMA). First, we validated if an algorithm that deals with two sources of labeling ambiguity would improve multiple instance learning by better integrating experts' varying expertise. Second, we validated if our algorithm that enables multi-labeler learning methods to deal with examples represented by sets of instances will improve the study of some real-life crowdsourced data. Third, we investigated the algorithmic behavior and scalability of the

proposed approach with respect to large quantities of labelers, which simulated the effects of large-scale crowdsourcing.

Since existing MIL methods are unable to deal with more than one version of class labels assigned to an example, following a common practice, we used majority voted labels in these methods to train classifiers. Existing MLL methods cannot easily tackle the situation that different examples have different numbers of instances. We preprocessed the original data so that examples were represented using vectors of the same length. In our experiments, this was achieved by appropriately merging features from the different instances, so all methods were compared on the basis of the same amount of data/information for fair comparison.

4.1. Evaluation data sets

The first set of evaluation data was collected for document categorization, and was widely used for evaluating MIL methods [30]. The second set of data contained facial expression images that were annotated by multiple online labelers. The third data set contained HWMA features that were extracted from ultrasound videos, and was used to diagnose if a human heart had abnormal motion on its left ventricular wall by multiple radiologists. All data sets were used to compare the proposed approach against representative MIL methods. The facial expression image data set and the HWMA data set, both with real crowdsourced labels, were used to validate the proposed approach against existing MLL methods.

4.1.1. NLP benchmark data sets

Three sets of NLP data were used with their summary shown in Table 1.

TREC data sets [30]: Four TREC data sets were downloaded from the website of National Institute of Standards and Technology, http://trec.nist.gov/. They were collected from several years of selected MEDLINE articles. Each article was split into multiple passages using overlapping windows of maximal 50 words in each window. Since the TREC data was extremely sparse, we performed a principal component analysis to reduce the data dimension. We chose the number of principal components that cumulatively explained 75% of the total data variance in each data set, which produced 46, 48, 31 and 48 features for the four TREC data sets. All the four data sets had 400 bags including 200 positive bags. The four data sets contained 3224, 3344, 3246 and 3391 total instances.

Newsgroups data sets [29] were composed from 20 Newsgroups corpus. In this data set, each news post corresponded to an instance. For each of the 20 news categories, each bag was made up by a random number of posts. For positive bags, 3% of the posts were randomly drawn from the target category and the remaining posts were from other categories. Three categories of these data sets, alt.atheism, comp.graphics and sci.med, were used in our experiments. Each of them contained 100 bags including 50 positive bags.

Table 1
Statistics of NLP data sets.

Data sets	Bags	Positive bags	Features	AVG. Ins/Bag[a]
Trec1	400	200	46	8
Trec2	400	200	48	8
Trec3	400	200	31	8
Trec4	400	200	48	8
alt.altheism	100	50	200	54
comp.graphics	100	50	200	31
sci.med	100	50	200	30
GOcomponent	718	359	200	18
GOfunction	770	385	200	17

[a] AVG. Ins/Bag: rounded average instance number per bag.

BioCreative data sets [7,64] were derived from the articles published in biomedical journals based on the names of human proteins, and their relatedness to the gene ontology (GO) codes. The gene ontology consists of 3 hierarchical domains of standardized biological terms referring to cellular components, biological processes and molecular functions, and each term was mapped to a unique GO code. A ⟨protein, article⟩ pair was labeled with a GO code if the article contained text that linked the protein to the GO code. Examples labeled positive for a GO code consisted of documents that were labeled with that GO code. We used two specific data sets, referred to as GOcomponent and GOfunction in our experiments.

4.1.2. Facial expression data sets with real crowdsourced labels

We also tested our methods on the Facial Expression data set previously used in a crowdsourcing study [65]. This data set contained 585 head-shots of 20 users. For each user, images were collected in which the user could be looking at 4 directions: straight, left, right and up, and the user could present four different kinds of facial expressions in each direction: neutral, happy, sad and angry. The images were labeled with respect to the 4 types of facial expressions by totally 27 online labelers at the Amazon Mechanical Turk. On average, each image received labels from 9 labelers. If a labeler did not annotate an image, we set the label to be 0, which corresponded to no evidence search for the corresponding facial expression from the specific labeler, and would not be used by any method.

We performed experiments to classify, based only on image features, if an image contained a happy face. We selected a set of 220 images with users looking straight ahead, left and right. We excluded images in which users wore sunglasses. Twenty-four labelers were involved in labeling the 220 images, of which 55 were associated with true labels ("+1") for the happy facial expression, and others were hence labeled by "−1". An early work in [65] estimated the actual expression labels using majority voting among the 9 labelers, and reported an accuracy of only 63.3% against the true labels for happy expression. Hence, this data set represents a very difficult problem. It can be even more challenging to not only estimate the true class labels but also simultaneously classify these images based on image features to the estimated true class.

Each image was represented by a collection of patches (or instances). Each patch was described by a vector of numerical image features. Since the original image contains large areas of background rather than the human face, we adopted the technique used in [9] to detect salient regions of an image. This has been proved to be a successful technique for MIL tasks [66,67]. We first searched patches with a scale between 20 and 50 pixels. The largest detected region contained mostly the face area. Then, we detected salient regions only on the face areas with the scale varying from 2 to 8 pixels. This step gave us 8 to 32 salient regions (instances) for an image. Fig. 3 presents a sample image of the detected salient regions on the human face. We resized each salient region into 40×40 pixels. The Local Binary Pattern (LBP) method [68] was used to extract features (58 of them) from each patch. The central location and scale of the detected salient region were also used as features. Totally, 61 features were computed for each patch or instance. In order to compare with algorithms that were only able to handle single instance examples, we divided the subregion containing mainly the human face into patches within a grid, and then LBP features were extracted for each patch. The LBP features from all patches were concatenated to form a single-instance example.

4.1.3. Medical image data sets with diagnoses from multiple radiologists

The goal of HWMA was to analyze and predict if a patient heart had abnormal motion based on image features extracted from two

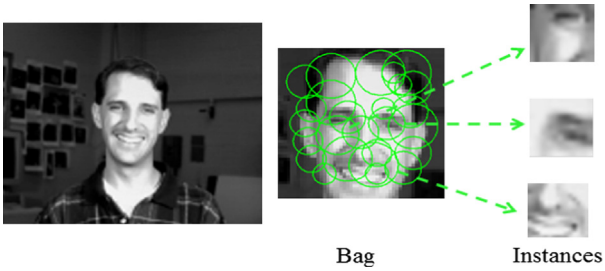


Fig. 3. An exemplar facial expression image which is represented by a bag of multiple instances (patches) as shown in circles.

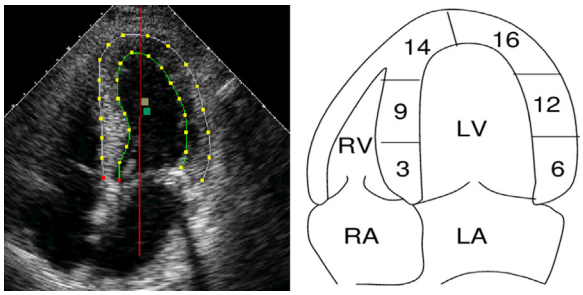


Fig. 4. Left: an ultrasound image of Apical 4 Chamber (A4C) view; right: the 6 heart segments seen from the A4C view.

sets of ultrasound images: base-dose and peak-dose, collected in stress tests. The wall of left ventricle is medically segmented into 16 segments, corresponding to 16 instances. Fig. 4 shows 6 of the 16 wall segments seen from the apical 4 chamber (A4C) view of an ultrasound clip. For each segment, 25 features were extracted. We also concatenated the features from each of the 16 segments to form a single-instance example. Base-dose image set contained 220 heart cases. The peak-dose set had 208 cases where 12 cases were dropped from this set due to poor image quality. The feature extraction process was described in more detail in our early works [69].

Five expert radiologists rated each segment of each heart case in terms of the severity of abnormality ranging from 1 to 5. If a rating was greater than 1, the segment was abnormal, and hence its label $y=+1$, and otherwise $y=-1$. If one of the segments was rated abnormal, the entire heart was rated abnormal. Groundtruth labels are usually difficult to acquire for HWMA. Researchers generally treat the consensus of expert readings as the ground-truth. Hence, the majority voted segment-level labels were used in our experiments as groundtruth, based on which the bag-level groundtruth labels were induced.

4.2. Comparison to existing multi-instance learning algorithms

The NLP data sets were originally designed for testing MIL methods with groundtruth labels. In this paper, however, the goal of the study is to evaluate if dealing with dual annotation ambiguity yields better learning performance. In other words, we deal with the kind of problem where no groundtruth labels are available, and instead multiple versions of a label are given and associated with a bag of instances. The NLP data sets were chosen to use in our experiments because by means of their groundtruth labels we could objectively evaluate the accuracy of our models.

We hence simulated 20 labelers from the groundtruth labels of these NLP data sets. Each labeler's labels were created following the same procedure discussed in [15]. We first specified two parameters for each labeler, the sensitivity α and specificity β. Four of the labelers were specified to have both sensitivity and specificity close to 0.5. In other words, these labelers' performance was close to random guess. Six of the labelers were given equal sensitivity and specificity in the values of 0.6, 0.65, 0.7, 0.75, 0.8 and 0.85. The rest ten labelers were prejudicial in the sense that half of them had higher sensitivity than specificity, i.e., $[\alpha,\beta]=[0.8,0.3]$, $[0.75,0.3]$, $[0.75,0.4]$, $[0.7,0.4]$, and $[0.6,0.4]$, and the other half had the exactly opposite parameter values. Once the parameters were specified for a labeler, a random number was generated uniformly from $[0,1]$ for each example (a bag). When the true label was $+1$ (or -1), if the random number was not bigger than the labeler's α (or β), this labeler chose the original label; or otherwise, (s)he flipped the sign of the label.

Although HWMA data set received crowdsourced diagnoses from five radiologists, we simulated 20 labelers in the same way as described above to increase noise level. Because the facial expression data set already came with 24 labelers' annotation, we did not simulate labelers for this data set. The experiment on facial expression data was done on three selected sets of the images where faces oriented in three different directions. Therefore 220 facial expression images were used and the performance was averaged over the three sets.

We compared our approach with four representative MIL methods listed below. These methods were implemented in PRTools [70] and its extension with MIL toolbox [71] except the MIL method, MIL-hinge, in [72] which was solved using CPLEX:

- mi-SVM [30] based on a mixed integer program, with a linear kernel.
- MILBoost [73] based on AdaBoost, with 100 rounds as the maximum number of iterations.
- MILES [9] based on a conversion to a single-instance example and the application of sparse SVMs.
- The MIL-hinge model in [72] based on a revision to the hinge loss.

These MIL methods cannot handle crowdsourced labels. To show that the ability of estimating true labels is important in the multi-instance learning setting, we used majority voted labels for the MIL methods (which is a common practice if an algorithm can only take one version of the labels), and let our methods automatically estimate the true labels. If the proposed methods perform better than the standard MIL methods, it demonstrates that the estimated labels by our models are more accurate than majority votes, and our models are better alternatives when dual ambiguity exists.

Because MIML learning is related to our learning problem, the following two representative MIML methods were also used in our comparison. (Their open-source codes were obtained from the authors' website.) However, MIML learning solves a different problem that constructs multiple classifiers altogether by jointly considering related MIL classification problems. It is not designed for integrating crowdsourced labels to construct a single classifier for only one target label. In our experiments with the MIML methods, we treated each labeler's label as a target label in the multi-label setting. In other words, if we have 20 labelers, a MIML method will report 20 classifiers, each corresponding to a labeler (although these classifiers were jointly built). Hence, the performance of a MIML method was compared by reporting the accuracy of the best classifier among the 20 classifiers that it constructed.

- M^3MIML [37] based on a quadratic program to maximize the classification margin, with a linear kernel.
- MIMLSVM [10] based on a revision to the SVM formulation, with a linear kernel.

Five-fold cross validation (CV) was performed to test all of the methods. There was no tuning parameter in the MILBoost method except we set its maximum number of iterations to 100. Other methods, mi-SVM, MILES, M^3MIML, MIMLSVM, had a tuning

Table 2
Comparison on TEST accuracies (%) for predicting bag labels between our approach and MIL, MIML methods.

Data sets	mi-SVM	MILBoost	MILES	MIL-hinge	M^{3}MIML	MIMLSVM	*all_min*	*selective min_max*
Trec1	82.0(1.0)	85.3(4.0)	87.8(1.3)	86.3(2.1)	80.0(4.2)	77.5(3.9)	92.0(2.6)	**92.5**(**1.3**)
Trec2	69.2(1.7)	73.0(1.8)	72.5(3.0)	66.0(1.4)	70.2(1.2)	67.5(2.2)	**73.0**(**1.5**)	72.0(1.5)
Trec3	70.7(0.8)	73.7(1.8)	71.8(2.1)	65.3(3.3)	70.0(1.4)	65.5(4.2)	**75.5**(**2.6**)	74.0(1.9)
Trec4	75.5(1.8)	76.5(1.8)	68.0(2.2)	66.0(1.2)	77.5(3.5)	67.5(2.9)	**80.0**(**2.5**)	79.5(0.6)
alt.atheism	64.6(2.1)	62.8(2.2)	58.0(2.4)	63.0(1.6)	63.0(1.4)	62.0(2.7)	64.0(2.9)	**69.0**(**2.6**)
comp.graphic	54.0(2.2)	**58.0**(**3.9**)	54.0(2.6)	54.0(1.1)	57.0(0.7)	56.7(2.8)	56.0(2.1)	55.0(1.9)
sci.med	62.0(3.3)	65.0(3.7)	**68.0**(**3.2**)	62.0(2.1)	59.3(4.3)	59.0(4.6)	**68.0**(**3.4**)	67.0(3.7)
GOcomponent	74.5(3.0)	78.0(1.5)	78.0(3.3)	72.1(1.1)	71.6(4.1)	70.7(2.1)	79.4(2.6)	**80.0**(**2.8**)
GOfunction	80.7(1.7)	77.7(2.5)	77.9(2.2)	74.8(0.6)	69.1(2.5)	65.8(2.1)	**80.9**(**2.2**)	76.6(2.9)
HWMA(base)	69.5(1.3)	70.9(1.5)	69.5(2.6)	70.1(2.6)	70.0(2.4)	69.1(3.4)	**74.6**(**0.9**)	72.7(1.5)
HWMA(peak)	72.1(1.1)	72.6(2.0)	73.4(1.9)	74.2(1.0)	68.3(4.4)	73.6(2.4)	**83.6**(**1.2**)	77.4(2.2)
FacialExpression	56.3(1.3)	57.0(1.3)	60.4(2.2)	54.2(1.1)	54.0(4.0)	53.8(2.2)	**61.1**(**2.6**)	58.6(2.1)

parameter named C. The MIL-hinge method had a tuning parameter γ. The two proposed methods had a hyperparameter λ. All these parameters were tuned in the same procedure within the training data. An internal three-fold CV was performed within each training phase to select a proper hyperparameter value for each of the methods from the choices of $2^k, k=-10,-9,\ldots,6$.

Table 2 shows the comparison on the averaged prediction accuracies for the bag-level labels and the standard deviation based on five separate trials of CV. The highest averaged accuracies were shown in bold fonts. As shown in Table 2, our methods obtained better classification accuracies than MIL methods on 10 out of the 12 experiments. On the rest two data sets, comp.graphic and sci.med, our methods achieved comparable performance with the best results. In particular, the two MIML methods showed worse performance in general than standard MIL methods that used majority voted labels. The MIML methods employed all the 20 versions of labels and used them to learn classifiers jointly for each labeler. As majority of the labelers were simulated with low accuracies, even though we reported the best classifier's accuracy, the performance was contaminated by other labelers' performance. This result provided an evidence that a method for MIML learning would not be a solution for the dual ambiguity problem. All these results demonstrate the effectiveness of our methods and validate the hypothesis that better integration of annotators' expertise can improve multi-instance learning in a crowdsourcing scenario.

Although our approach handles both multi-instance examples and crowdsourced labels, it is worth investigating how the proposed MIL component works by itself, which also sheds light on what causes the performance improvement in Table 2. We performed additional experiments to compare the performance of the four MIL methods with that of our methods when groundtruth labels were provided. Note that when only one version of the labels, i.e. the groundtruth labels, is provided to our methods, the second Sub-problem in Algorithm 1 will be omitted because the only reliability parameter r will be set to 1 automatically. The two models, *all_min* and *selective min_max*, will be identical. In this situation, our approach is treated merely as another MIL approach. In this set of experiments, we observed that our method performed most similar to MIL-hinge with an average test classification accuracy 76.2% and standard deviation 3.1% over the 12 data sets. The four MIL methods, mi-SVM, MILBoost, MILES, and MIL-hinge, reported average classification accuracies of 75.3%(3.2%), 75.7%(2.4%), 76.6%(3.2%), and 76.2%(3.1%), respectively, over the 12 data sets. We see that the performance of our MIL component is comparable to other state-of-the-art MIL methods. This observation confirms that the performance improvement in Table 2 is due to the ability of our methods to handle dual ambiguity.

We also examined the quality of the labeler's reliability factors estimated by our methods. In the experiments, only our methods could report the reliability of each labeler. Spearman's correlation coefficients, showing the agreement between the estimated reliability and the simulated labelers' accuracy, are given in Table 3 as in the ρ columns. A correlation coefficient ranges from -1 to $+1$ with higher values indicating higher levels of agreement. Typically, $\rho=0$ means that the two variables, a labeler's estimated reliability and the simulated accuracy, are not correlated at all. A negative ρ means that the two variables are inversely proportional. All correlation coefficients were positive for both of our methods on all data sets. The consistency is statistically significant on more than half of the data sets as shown in the paired t-test p-values. Using a standard threshold $p \le 0.05$, the *all_min* model and the *selective min_max* model had significant results on 7 and 6 data sets, respectively. Hence, our models are capable of assessing the reliability of a labeler in the absence of knowledge about true labels.

Table 3
Spearman's correlation coefficients ρ between a labeler's accuracy as simulated and the labeler's reliability factor estimated by our methods.

Data sets	*all_min*		*selective min_max*	
	ρ^*	p-value	ρ^*	p-value
Trec1	0.63	0.36e−2	0.64	0.29e−2
Trec2	0.70	0.84e−3	0.69	0.10e−2
Trec3	0.45	0.53e−1	0.46	0.47e−1
Trec4	0.63	0.36e−2	0.62	0.46e−2
alt.atheism	0.59	0.50e−2	0.68	0.88e−3
comp.graphic	0.59	0.50e−2	0.52	0.17e−2
sci.med	0.61	0.50e−2	0.47	0.42e−1
GOcomponent	0.55	0.14e−1	0.53	0.19e−1
GOfunction	0.47	0.42e−1	0.56	0.12e−1
HWMA(base)	0.49	0.33e−1	0.52	0.22e−1
HWMA(peak)	0.67	0.14e−2	0.62	0.36e−2
Facial expression	0.52	0.15e−1	0.47	0.49e−1

* $\rho \in [-1,1]$. A ρ value closer to 1 indicates that the two variables are more correlated.

4.3. Comparison to existing multi-labeler learning algorithms

We compared our methods with three recently published MLL algorithms as listed below. Note that many other learning-from-crowds algorithms do not aim to build classifiers rather than to study the nature of the crowd behaviors. The three MLL methods we chose all construct classifiers by integrating expert expertise and are most suitable for comparison with our methods.

- Expectation–maximization (EM) method with a two-coin model as labeler accuracy prior [15].
- EM method with Gaussian prior on labeler accuracy [56].
- EM method with Bernoulli prior on labeler accuracy [56].

In the multi-labeler learning context, the best possible classifier would be obtained by training a classifier against the true labels if they are

known. A baseline model could be the classifier trained with respect to the simple majority votes across all labelers. Hence, we also built MIL classifiers [72] using the groundtruth labels and majority voted labels as the best possible model and a baseline model.

In the experiments with the facial expression data set, only our methods can run on examples with varying numbers of patches as detected. To make single-instance examples, we resized the face area of each image into 120×120 pixels and split into 6×6 grid cells. Then, each image had the same number of 36 instances. The same 15 LBP features were extracted from each patch. Hence, this process transformed each image into a vector of the same length (with 540 features). Five-fold CV was performed on this data set where our methods used the 36 instances in each bag and other methods used single instances of 540 features. The averaged performance was reported. Table 4 shows the averaged Area-Under-the-ROC-Curve (AUC) and the standard deviations over the five folds in the CV. Since the facial expression data set was extremely difficult, all methods had modest AUCs, but the proposed methods still outperformed other methods.

In the experiments with the HWMA data set, we combined the 16 segments of each heart case to form single-instance examples. Given each segment was represented by 25 image features, we obtained 400 features for each heart example. In order to more closely examine the reliability estimates of the different methods, we used 3 simulated labelers and 2 actual radiologists. The three simulated labelers had sensitivity of [0.6, 0.65, 0.7] and specificity of [0.4, 0.65, 0.7]. The first simulated labeler was the least competent labeler. Based on the groundtruth labels, there were 77 and 71 positive bags, respectively, from the base-dose and peak-dose image sets. The HWMA data set was split by the five radiologists to form a test set for each dose. We draw receiver operating characteristic (ROC) curves [74] to measure the test performance of each classifier.

Table 4
Averaged AUCs over the 5 folds of cross validation on the facial expression data set.

Algorithms	Averaged AUC	Standard deviation
Groundtruth	0.63	0.04
all_min	**0.61**	0.01
selective min_max	**0.59**	0.02
Two-coin model in [15]	0.56	0.02
Gaussian model in [56]	0.55	0.01
Bernoulli model in [56]	0.55	0.01
Majority voting	0.53	0.01

The best two values obtained by the MLL methods are indicated by bold fonts.

Fig. 5 shows the ROC curves of the classifiers built by different methods on HWMA data sets. AUCs were also computed and given in the two figures. Notice that only the models trained with groundtruth, majority voted labels and two of our methods were obtained based on multi-instance examples. The other three methods ran on single-instance examples because they could not handle multi-instance examples. The empirical results on all the data sets show that the classifiers trained with groundtruth performed the best as expected. All classifiers obtained by multi-labeler learning methods performed better than the baseline model.

On both of the facial expression and HWMA data sets, the five MLL methods achieved similar prediction accuracies where our methods performed slightly better. However, only our methods were able to identify the positive instances, including the essential micro patterns that revealed the facial expression, and the abnormal segments that were responsible for the abnormality of a heart whereas other MLL methods could not identify the regions responsible for the positive label of an image. Fig. 6 shows the instances that were identified to be responsible for the "happy" label to three face images by both of our methods.

Figs. 7 and 8 show the reliability estimates of the internet labelers in the Facial Expression data and the radiologists for HWMA, respectively. Shown in the figures are the estimates averaged over the five CV folds. The labeler reliabilities estimated by our methods were comparable to the labeling accuracies estimated by the two-coin model [15]. The two coin model returned us two parameters for each labeler named as α and β, corresponding to the sensitivity and the specificity of a labeler, respectively. With these two parameters and the groundtruth predicted by the two-coin model, labeling accuracies can be computed as $(\alpha n^+ + \beta n^-)/n$ where n^+, n^- and n are the numbers of positive examples, negative examples and total examples, respectively. The estimated accuracies were normalized to be in the same scale with our reliability estimates. The methods with Gaussian and Bernoulli models [56] did not report reliability estimates because they built classifiers for each individual labeler to represent a labeler's reliability [56].

Since the crowdsourced labels of facial expression data set were collected from real labelers, their competency was unknown beforehand. As shown in Fig. 7, both *all_min* and *selective min_max* models reported reliabilities with large variation between different labelers. In comparison, the reliabilities from the two-coin model did not vary that much among different labelers. Overall, there was a certain level of agreement across the three methods in the estimated reliabilities although our methods would perform better to screen out redundant or spammer labelers given the high variance of its reliability estimates.

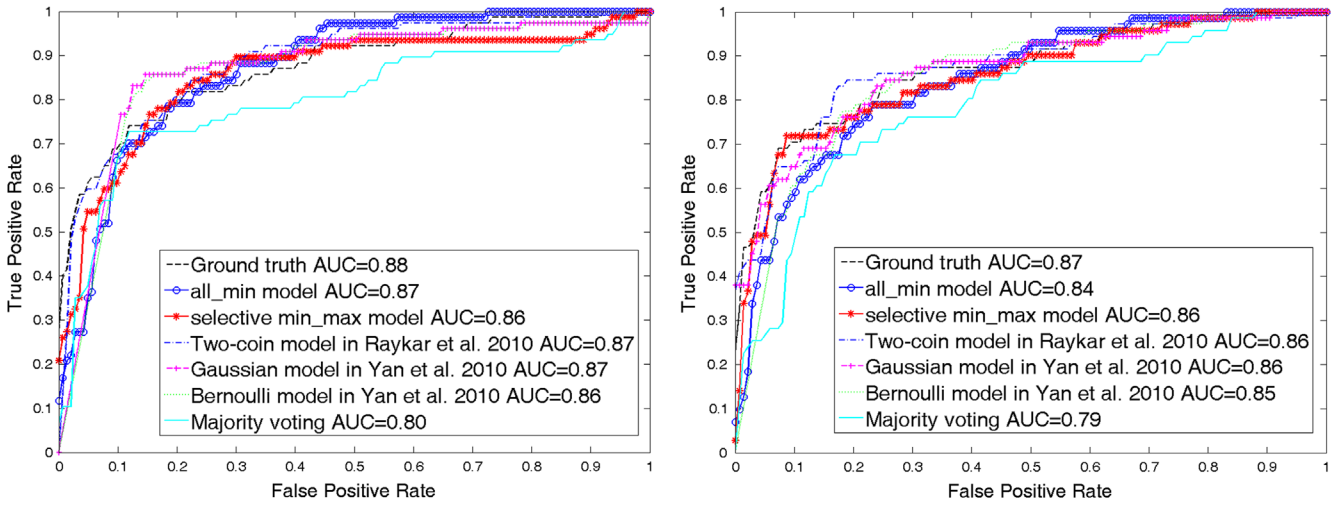


Fig. 5. ROC comparison among different methods for the HWMA base-dose images (left) and peak-dose images (right).

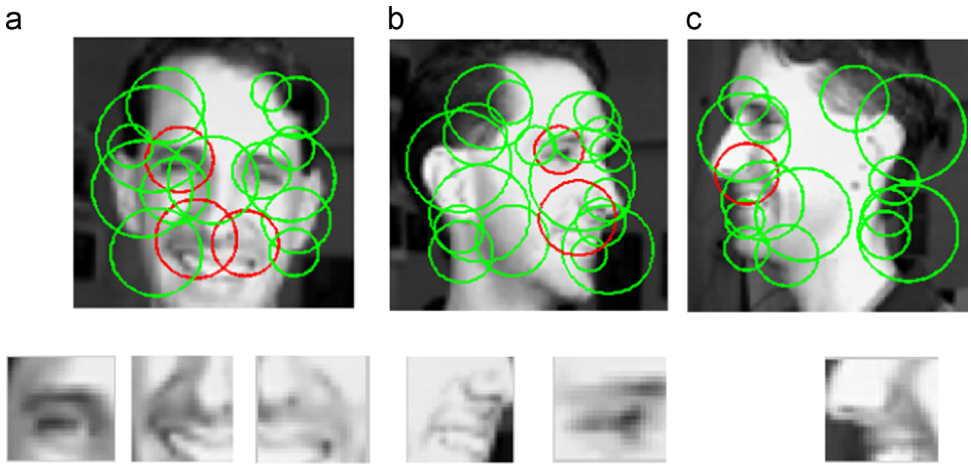


Fig. 6. An example of the instance-level prediction for three facial images where the user looks to different directions. The identified positive instances are shown in red circles and the corresponding patches are enumerated at the bottom.

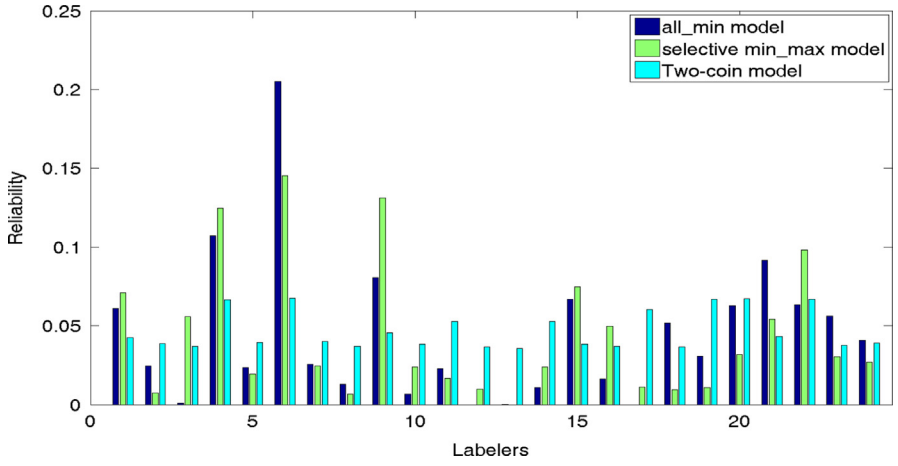


Fig. 7. The reliability estimates by the different methods averaged over the five folds of cross validation on the facial expression images.

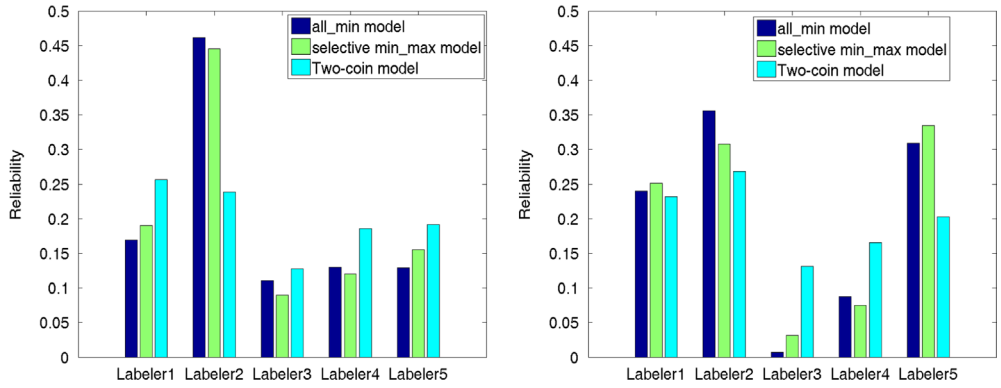


Fig. 8. The reliability estimates by the different methods averaged over the five folds of cross validation. Left: on HWMA base-dose images; right: on HWMA peak-dose images.

For the HWMA labelers, as shown in Fig. 8, the first two labelers (the real radiologists) were rated consistently higher than the third and the forth labelers (simulated labelers) by all models. Among the simulated labelers (the last three labelers), the first one (the least competent labeler) was rated the lowest, and the last one was simulated with good accuracy, and hence received a higher rating.

We examined the quality of the estimated true labels calculated by our methods as $\hat{y}_k = \sum_{j=1}^{m} r_j y_k^j$ based on the averaged reliability estimates. The estimated $\hat{y}$ is a real-valued variable due to the real-valued reliabilities $\mathbf{r}$, and hence can be tested against the true labels via a ROC plot. The estimated $\hat{y}$ can be treated as the likelihood that an example is positive. Fig. 9 shows the ROC curves of the estimated $\hat{y}$ against the true labels on the facial expression data. Fig. 10 shows the curves for HWMA base-dose and peak-dose data. The two models of our approach reported very similar $\hat{y}$ on the two HWMA data sets. On the facial expression data set,

the *selective min_max* model performed slightly better than the *all_min* model. Overall, these models all reached high accuracies in the groundtruth estimation with AUCs over 0.9, demonstrating the effectiveness of our approach.

4.4. The scalability of the proposed methods

For all of the above experiments, the runtime of our algorithm was in seconds. In a general crowdsourcing scenario, many more labelers may be utilized to annotate data. We hence tested the scalability of our models by increasing the number of labelers. We simulated 20, 100, 500, and 1000 labelers, respectively, for the HWMA data sets following the similar procedure as described in Section 4.2. In particular, 15% of the simulated labelers were set to have specificities and sensitivities close to a random guess, and 20% of the labelers were simulated as prejudicial with 10% possessing greater sensitivity than specificity and the rest were in the opposite case. Table 5 shows the AUCs achieved by each of the MLL algorithms using 5-fold CV. Hyper-parameters were tuned within the training data. The proposed *all_min* model and *selective min_max* model consistently achieved the best accuracies when the number of labelers varied.

For the experiments using 100 and 500 labelers, the two-coin model in [15] achieved the same AUC as that of the proposed model. However, when we increased the number of labelers, the two-coin model became numerically unstable. The two-coin model updates the estimated groundtruth denoted by μ, a probability of the true label being $+1$, based on the multiplications of two accuracy parameters $0 \le \alpha < 1$ and $0 \le \beta < 1$ of all labelers, e.g., $\alpha_1\alpha_2\cdots\alpha_m$ and $\beta_1\beta_2\cdots\beta_m$ assuming that there are m labelers. These products become extremely small with large m, consequently, the μ becomes oscillating between 0 and 1 since these two products are used in the numerator and denominator of the updating formula for μ. Experimental results show that great sparsity existed in the reliability factors estimated by our models when a large number of labelers were included, which means that the estimated label $\sum_j r_j y_k^j$ was based on few labelers and information from other labelers might be redundant. Hence, our models could automatically select few labelers whose labels were valid to make accurate estimates of the groundtruth and exclude correlated or redundant labelers.

Fig. 11 shows the runtime for one iteration of each method when the number of labelers varies. Although all methods required longer runtime as the number of labelers was increased, our methods were generally more scalable with the number of labelers (shown as flatter lines). It is likely because increasing the number of labelers only affects the optimization of Sub-problem 2 in our approach which is however a simple linear program. Our algorithms typically terminated within 20 iterations, which was similar to the two coin model in [15] but significantly less than the methods in [56].

To further test the time efficiency of the proposed approach, we used the full set of 585 head-shots in the facial expression data set in an experiment to classify happy faces from the rest. We used the $40 \times 40 = 1600$ raw intensity features for each instance. Each of the 585 bags contained 8 to 32 instances as described in Section 4.1. This created a rather large data set. There were totally 27 labelers involved in labeling these images. We evaluated the runtime of each iteration of our algorithms. The averaged runtime for solving the first Sub-problem was 1.8 s. The runtime for solving the second Sub-problem was 0.7 s for the *all_min* model and 1.5 seconds for the *selective min_max* model. Both models finished the training in 10 iterations, so the training phase of our approach overall required

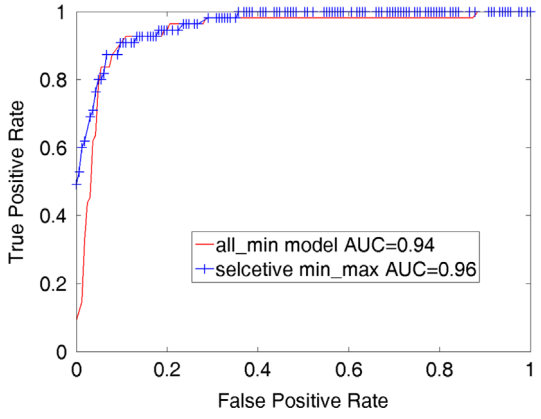


Fig. 9. ROC curves of estimated labels obtained by our methods drawn against groundtruth bag-level labels on facial expression images.

Table 5
AUC comparison between MLL algorithms on the HWMA data set with the number of labelers ranging from 20 to 1000.

Algorithms	20	100	500	1000
HWMA base dose				
Two-coin model in [15]	0.87	**0.91**	**0.90**	0.91
Gaussian model in [56]	0.90	0.87	0.88	0.89
Bernoulli model in [56]	0.89	0.86	0.88	0.88
all_min	**0.91**	**0.91**	**0.90**	**0.92**
selective min_max	0.86	0.88	0.87	0.89
HWMA peak dose				
Two-coin model in [15]	0.86	0.88	0.87	0.88
Gaussian model in [56]	0.85	0.88	0.87	0.86
Bernoulli model in [56]	0.83	0.86	0.86	0.87
all_min	**0.87**	0.88	**0.88**	**0.89**
selective min_max	**0.87**	**0.89**	**0.88**	**0.89**

The best performance obtained on each data set is indicated by bold fonts.

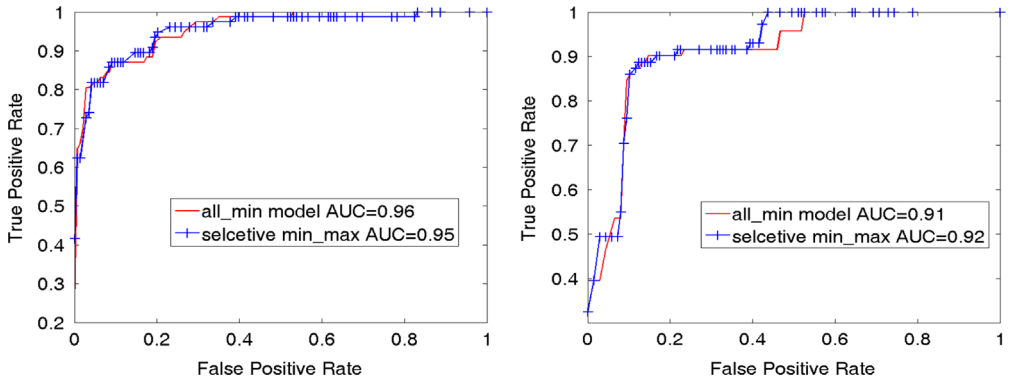


Fig. 10. ROC curves of estimated labels obtained by our methods drawn against groundtruth bag-level labels. Left: on base-dose images; right: on peak-dose images.

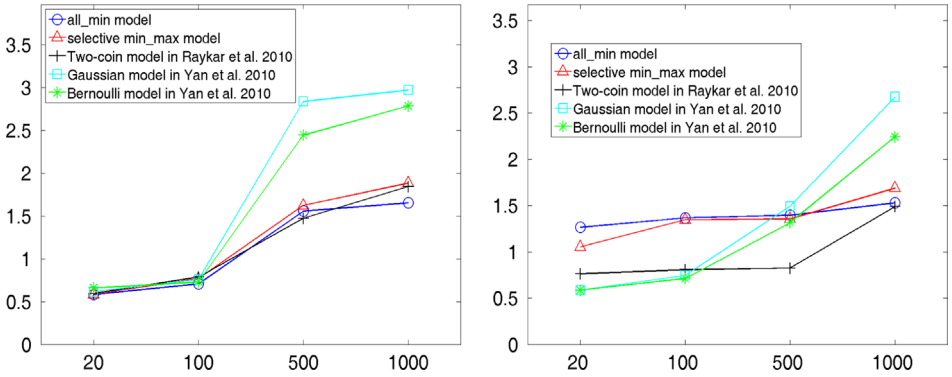


Fig. 11. Runtime (in seconds) comparison between different methods when the number of labelers varies. Left: HWMA base dose; right: HWMA peak dose.

25–33 s. This is considered relatively efficient if we compare it with the cost of the two MIML algorithms when they ran on the exactly same data with both multi-instance examples and multiple labelers' labels. On average, M^3MIML and MIMLSVM spent 260 and 110 s, respectively, to finish their training processes.

5. Conclusion

We have derived an effective approach to construct classifiers when multiple annotators with varying expertise are utilized to label bags of instances. In many practical applications, dual labeling ambiguity presents where one kind of ambiguity comes from the inconsistency of multiple labelers' labels and the other comes from the inability of labeling individual instances of an example. We first modify the hinge loss to employ the weighted consensus of different labelers' labels and then use min–max optimization to extend the loss from instance-level to bag-level, which creates a solution to the dual labeling ambiguity issue. An alternating optimization algorithm is designed to optimize the proposed formulation after relaxing it to two approximation variants. We have compared the proposed models to the state of the art multi-instance learning and multi-labeler learning methods. Empirical results on NLP benchmark data sets and two real-world crowdsourced problems have demonstrated the effectiveness of the proposed approach over existing methods and proved the need for such a technique to address the dual ambiguity problem.

There are several limitations of the current work. We have not examined other potential solvers that explore the bi-convexity property of the proposed formulation (2), which may motivate better relaxations to the integrative formulation (3). Given the stochastic nature of the formulated optimization (3) and its relaxations (e.g., the sub-problem in (5)), we are unable to provide a convergence proof for Algorithm 1. More extensive empirical evaluation on real-life data sets with accessible input features and a large number of labelers may better assess the strength and weakness of the proposed approach. (Notice that many crowdsourced data sets do not provide input features for classifier training). Besides the proposed *all_min* and *selective min_max* models for estimating the true labels and labeler reliabilities, alternative models may exist to further enhance the algorithm.

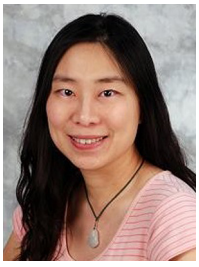
Acknowledgments

This work was supported by the research grant from National Science Foundation of United States (IIS-1320586).

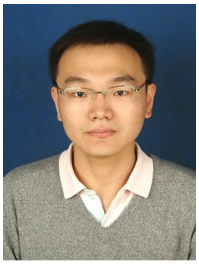
References

[1] K. Fort, G. Adda, K.B. Cohen, Amazon mechanical turk: gold mine or coal mine? Comput. Linguist. 37 (2) (2011) 413–420.
[2] The World's Largest Workforce ⟨http://crowdflower.com⟩.
[3] R. Snow, B. O'Connor, D. Jurafsky, A.Y. 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, 2008, pp. 254–263.
[4] A. Sorokin, D. Forsyth, Utility data annotaton with Amazon Mechanical Turk, in: Proceedings of the First IEEE Workshop on Internet Vision at Computer Vision and Pattern Recognition (CVPR) 08, 2008, pp. 1–8.
[5] R. Hoffmann, T. Marwick, D. Poldermans, H. Lethen, R. Ciani, P.v.d. Meer, H. Tries, P. Gianfagna, P. Fioretti, J.J. Bax, M.A. Katz, R. Erbel, P. Hanrath, Refinements in stress echocardiographic techniques improve inter-institutional agreement in interpretation of dobutamine stress echocardiograms, Eur. Heart J. 23 (2002) 821–829.
[6] O. Maron, A.L. Ratan, Multiple instance learning for natural scene classification, in: Proceedings of the International Conference on Machine Learning, 1998, pp. 341–349.
[7] S. Ray, M. Craven, Supervised versus multiple instance learning: an empirical comparison, in: Proceedings of the International Conference on Machine Learning, 2005, pp. 697–704.
[8] V.C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, B. Rao, Bayesian multiple instance learning: automatic feature selection and inductive transfer, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 808–815.
[9] Y. Chen, J. Bi, J. Wang, MILES: multiple instance learning via embedded instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1–17.
[10] Z.-H. Zhou, M.-L. Zhang, Multi-instance multi-label learning with application to scene classification, in: Advances in Neural Information Processing Systems, vol. 19, 2007, pp. 1609–1616.
[11] R. Jin, S. Wang, Z.-H. Zhou, Learning a distance metric from multi-instance multi-label data, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 896–902.
[12] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, Y.-F. Li, Multi-instance multi-label learning, Artif. Intell. 176 (1) (2012) 2291–2320.
[13] D. Zhou, J. Platt, S. Basu, Y. Mao, Learning from the wisdom of crowds by minimax entropy, in: Advances in Neural Information Processing Systems (NIPS), vol. 25, 2012, pp. 2204–2212.
[14] H. Kajino, H. Kashima, Convex formulations of learning from crowds, Trans. Jpn. Soc. Artif. Intell. 27 (2012) 133–142.
[15] V.C. Raykar, S. Yu, G.H. Valadez, C. Florin, L. Bogoni, L.H. Zhao, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (2010) 1297–1322.
[16] Y. Yan, R. Rosales, G. Fung, J. Dy, Modeling multiple annotator expertise in the semi-supervised learning scenario, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, 2010, pp. 241–248.
[17] V.C. Raykar, S. Yu, L.H. Zhao, A. Jerebko, C. Florin, G.H. Valadez, L. Bogoni, L. Moy, Supervised learning from multiple experts: who to trust when everyone lies a bit, in: Proceedings of the 26th International Conference on Machine Learning, 2009, pp. 96–103.
[18] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[19] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105.
[20] T.G. Dietterich, R.H. Lathrop, T. Lozano-Perez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1–2) (1997) 31.
[21] O. Maron, T. Lozano-Perez, A framework for multiple-instance learning, in: Advances in Neural Information Processing Systems, 1998, pp. 570–576.
[22] Q. Zhang, S.A. Goldman, EM-DD: an improved multiple-instance learning technique, in: Advances in Neural Information Processing Systems, 2002, pp. 1073–1080.

[23] R. Rahmani, S.A. Goldman, H. Zhang, S.R. Cholleti, J.E. Fritts, Localized content-based image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1902–1912.
[24] P.M. Long, L. Tan, PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples, Mach. Learn. 30 (1) (1998) 7–21.
[25] J. Wang, J.D. Zucker, Solving the multiple-instance problem: a lazy learning approach, in: Proceedings of the International Conference on Machine Learning, Morgan Kaufmann, Burlington, MA, 2000, pp. 1119–1125.
[26] J. Ramon, L.D. Raedt, Multi-instance neural networks, in: Proceedings of the Conference on International Conference on Machine Learning (ICML) Workshop Attribute-Value and Relational Learning, 2000.
[27] P. Auer, R. Ortner, A boosting approach to multiple instance learning, in: Proceedings of the European Conference on Machine Learning, 2004, pp. 63–74.
[28] H. Blockeel, D. Page, A. Srinivasan, Multiple instance tree learning, in: Proceedings of the International Conference on Machine Learning, 2005, pp. 321–328.
[29] B. Settles, M. Craven, S. Ray, Multiple-instance active learning, in: Advances in Neural Information Processing Systems (NIPS), MIT Press, Cambridge, MA, 2008, pp. 1289–1296.
[30] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 561–568.
[31] Z. Fu, A. Robles-Kelly, J. Zhou, MILIS: multiple instance learning with instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 958–977.
[32] G. Fung, M. Dundar, B. Krishnapuram, R.B. Rao, Multiple instance learning for computer aided diagnosis, in: Advances in Neural Information Processing Systems, vol. 19, 2007, pp. 425–432.
[33] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, Z. Wang, Joint multi-label multi-instance learning for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[34] S.-H. Yang, H. Zha, B.-G. Hu, Dirichlet–Bernoulli alignment: a generative model for multi-class multi-label multi-instance corpora, in: Advances in Neural Information Processing Systems, 2009, pp. 2143–2150.
[35] N. Nguyen, A new SVM approach to multi-instance multi-label learning, in: Proceedings of the 10th IEEE International Conference on Data Mining, IEEE, 2010, pp. 384–392.
[36] F. Briggs, X.Z. Fern, R. Raich, Rank-loss support instance machines for MIML instance annotation, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, 2012, pp. 534–542.
[37] M.-L. Zhang, Z.-H. Zhou, M3MIML: a maximum margin method for multi-instance multi-label learning, in: Proceedings of the Eighth IEEE International Conference on Data Mining, IEEE, Middlesex, NJ, 2008, pp. 688–697.
[38] S. Feng, D. Xu, Transductive multi-instance multi-label learning algorithm with application to automatic image annotation, Expert Syst. Appl. 37 (1) (2010) 661–670.
[39] S.-J. Yang, Y. Jiang, Z.-H. Zhou, Multi-instance multi-label learning with weak label, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, AAAI Press, Palo Alto, CA, 2013, pp. 1862–1868.
[40] A.P. Dawid, A.M. Skeene, Maximum likelihood estimation of observed error-rates using the EM algorithm, Appl. Stat. 28 (1) (1979) 20–28.
[41] S.L. Hui, S.D. Walter, Estimating the error rates of diagnostic tests, Biometrics 36 (1) (1980) 167–171.
[42] J. Spiegelhalter, P. Stovin, An analysis of repeated biopsies following cardiac transplantation, Stat. Med. 2 (1) (1983) 33–40.
[43] S.L. Hui, X.H. Zhou, Evaluation of diagnostic tests without gold standards, Stat. Methods Med. Res. 7 (4) (1998) 354–370.
[44] V.C. Raykar, S. Yu, Ranking annotators for crowdsourced labeling tasks, in: Advances in Neural Information Processing Systems, vol. 24, MIT Press, Cambridge, MA, 2011, pp. 1809–1817.
[45] C. Liu, Y.-M. Wang, Truelabel+confusion: a spectrum of probabilistic models in analyzing multiple ratings, in: Proceedings of the International Conference on Machine Learning, 2012.
[46] S.B.P. Welinder, S. Branson, P. Perona, The multidimensional wisdom of crowds, in: Proceedings of the 2010 Neural Information Processing Systems (NIPS) Conference, 2010, pp. 2424–2432.
[47] Y. Tian, J. Zhu, Learning from crowds in the presence of schools of thought, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 226–234.
[48] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, J. Movellan, Whose vote should count more: optimal integration of labels from labelers of unknown expertise, in: Proceedings of the 2009 Neural Information Processing Systems (NIPS) Conference, 2009, pp. 2035–2043.
[49] P. Smyth, U. Fayyad, M. Burl, P. Perona, P. Baldi, Inferring ground truth from subjective labeling of Venus images, in: Advances in Neural Information Processing Systems, 1995, pp. 1085–1092.
[50] P. Donmez, J. G. Carbonell, Proactive learning: cost-sensitive active learning with multiple imperfect oracles, in: Proceedings of the Conference on Information and Knowledge Management, 2008, pp. 619–628.
[51] V.S. Sheng, F. Provost, P.G. Ipeirotis, Get another label? Improving data quality and data mining using multiple, noisy labelers, in: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, 2008, pp. 614–622.
[52] O. Dekel, O. Shamir, Good learners for evil teachers, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, New York, NY, USA, 2009, pp. 233–240.
[53] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. Wortman, Learning bounds for domain adaptation, in: Advances in Neural Information Processing Systems, vol. 20, MIT Press, Cambridge, MA, 2008, pp. 129–136.
[54] K. Crammer, M. Kearns, J. Wortman, Learning from multiple sources, J. Mach. Learn. Res. 9 (2) (2009) 1757–1774.
[55] R. Jin, Z. Ghahramani, Learning with multiple labels, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2003, pp. 897–904.
[56] Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, J. Dy, Modeling annotator expertise: learning when everybody knows a bit of something, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 2010, pp. 932–939.
[57] A. Shapiro, D. Dentcheva, A. Ruszczynski, Lectures on Stochastic Programming: Modeling and Theory, SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.
[58] G.B. Dantzig, G. Infanger, Stochastic Programming, vol. 150, Springer, Dordrecht, 2011.
[59] IBM ILOG CPLEX Division, New York, NY, IBM ILOG CPLEX Callable library 12.1 Reference Manual, 2009.
[60] V. Klee, G. J. Minty, How good is the simplex algorithm? in: Inequalities-III, 1972, pp. 159–175.
[61] R. Shamir, Probabilistic analysis in linear programming, Stat. Sci. 8 (1) (1993) 57–64.
[62] D.A. Spielman, S.-H. Teng, Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time, J. ACM 51 (3) (2004) 385–463.
[63] K.H. Borgwardt, The Simplex Method: A Probabilistic Analysis, Algorithms and Combinatorics, vol. 1, Springer-Verlag, New York, 1980.
[64] C. Blaschke, E. Leon, M. Krallinger, A. Valencia, Evaluation of biocreative assessment of task 2, BMC Bioinform. 6 (Suppl 1) (2005) S16.
[65] B. Mozafari, P. Sarkar, M.J. Franklin, M.I. Jordan, S. Madden, Active learning for crowd-sourced databases, ⟨http://dx.doi.org/abs/1209.3686v3⟩.
[66] T. Kadir, M. Brady, Saliency scale and image description, Int. J. Comput. Vis. 45 (2) (2001) 83–105.
[67] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, vol. 2, 2003, pp. 264–271.
[68] T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on kullback discrimination of distributions, in: Proceedings of the 12th International Conference on Pattern Recognition (ICPR), vol. 1, 1994, pp. 582–585.
[69] M. Qazi, G. Fung, S. Krishnan, J. Bi, B. Rao, A. Katz, Automated heart abnormality detection using sparse linear classifiers, IEEE Eng. Med. Biol. 26 (2) (2007) 56–63.
[70] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. Ridder, D. Tax, S. Verzakov, Prtools, A Matlab Toolbox for Pattern Recognition. URL ⟨http://www.prtools.org⟩, 2010.
[71] D. Tax, MIL, A Matlab Toolbox for Multiple Instance Learning. URL ⟨http://prlab.tudelft.nl/david-tax/mil.html⟩, March 2013.
[72] J. Bi, J. Liang, Multiple instance learning of pulmonary embolism detection with geodesic distance along vascular structure, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[73] P. Viola, J. C. Platt, C. Zhang, Multiple instance boosting for object detection, in: Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA. 2006, pp. 1419–1426.
[74] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.



Jinbo Bi received a Ph.D. degree in mathematics from Rensselaer Polytechnic Institute, USA, and a master degree in Electrical Engineering and Automatic Control from Beijing Institute of Technology, China. She is an associate professor of Computer Science and Engineering at the University of Connecticut. Prior to her current appointment, she worked with Siemens Medical Solutions on computer aided diagnosis research and Partners Healthcare on clinical decision support systems. Her research interests include machine learning, data mining, bioinformatics and biomedical informatics.



Xin Wang received a master degree in Control Theory and Control Engineering from Dalian University of Technology, China. He is currently studying at the University of Connecticut toward his Ph.D. degree in Computer Science. His research interests include machine learning, data mining and intelligent systems.