Multiple-Instance Learning from Unlabeled Bags with Pairwise Similarity

Lei Feng, Senlin Shu, Yuzhou Cao, Lue Tao, Hongxin Wei, Tao Xiang, Bo An, Gang Niu

Abstract—In *multiple-instance learning* (MIL), each training example is represented by a bag of instances. A training bag is either negative if it contains no positive instances or positive if it has at least one positive instance. Previous MIL methods generally assume that training bags are fully labeled. However, the exact labels of training examples may not be accessible, due to security, confidentiality, and privacy concerns. Fortunately, it could be easier for us to access the pairwise similarity between two bags (indicating whether two bags share the same label or not) and unlabeled bags, as we do not need to know the underlying label of each bag. In this paper, we provide the first attempt to investigate MIL from only similar-dissimilar-unlabeled bags. To solve this new MIL problem, we first propose a strong baseline method that trains an instance-level classifier by employing an unlabeled-unlabeled learning strategy. Then, we also propose to train a bag-level classifier based on a convex formulation and theoretically derive a generalization error bound for this method. Comprehensive experimental results show that our instance-level classifier works well, while our bag-level classifier even has better performance.

Index Terms—Multiple-Instance Learning, Similar-Dissimilar-Unlabeled Classification, Empirical Risk Minimization

1 Introduction

MACHINE learning [1] has achieved much success in a variety of real-world problems, especially in supervised learning problems. In supervised learning [2], a predictive model is learned from a fully labeled dataset where we know the exact label of each training example. However, it could be quite difficult or even impossible for us to collect a large-scale fully labeled dataset due to the unaffordable annotation costs or confidentiality concerns. Therefore, weakly supervised learning [3] naturally arises, which attempts to learn a model from data with only weak supervision. Due to the difficulty of collecting large-scale fully labeled datasets in many real-world scenarios, weakly supervised learning has attracted increasing attention from machine learning and data mining communities. According to the different types of weak supervision information at hand, weakly supervised learning includes various learning problems, such as semi-supervised learning [4], [5], [6], noisy-label learning [7], [8], [9], partial-label learning [10], [11], [12], [13], positive-unlabeled learning [14], [15], [16], positive-confidence classification [17], [18], similar-dissimilar classification [19], [20], unlabeled-unlabeled learning [21], [22], triplet comparison classification [23], pairwise comparison classification [24], similarity-confidence learning [25].

This paper considers another weakly supervised learning problem called *multiple-instance learning* (MIL) [26], [27], [28], [29], [30], which aims to deal with the binary classification task where each training example is represented by a bag of instances and a binary label (indicating whether the bag is positive or not) is provided for each training bag. A training bag is considered either positive if it has at least one positive instance or negative if it contains no positive instances. Given this kind of data, MIL aims to construct a predictive model that can be used to predict the label of unseen bags. From the problem setting of MIL, we can know that MIL is more difficult than ordinary binary classification because the labels of all the instances in the training set are not accessible. Because of the practical problem setting of MIL, MIL has been frequently used in many real-world problems, such as text categorization [31], face detection [32], medical diagnosis [33], [34], image retrieval [35], [36], [37], visual tracking [38], object detection [39], [40], and drug activity prediction [26].

Up to now, numerous efforts have been made to develop effective methods for MIL. Representative methods include citation k-NN [41], EM-DD [42], MI-SVM [31], MIBoosting [43], MILES [44], MIGraph [45], MIForests [46], and MI-ODM [47]. These methods solve the MIL problem in different ways. For example, MI-SVM [31] formalizes the bag margin of each training bag and applies the maximum margin principle. MIBoosting [43] assumes that all the instances in a bag contribute equally and independently to a the label of the bag. Most of the previous methods achieved satisfactory performance. However, all of them require fully labeled bags for training an effective classifier that can be used to accurately predict the label of any test bags.

In many real-world scenarios, due to unaffordable annotation costs [4], privacy considerations [48], and social bias [49], it could be unlikely for us to collect a fully labeled MIL dataset where the exact label of each training bag is known. For example, in the task of drug activity prediction

Lei Feng, Yuzhou Cao, Hongxin Wei, and Bo An are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: feng0093@e.ntu.edu.sg, nanjing.caoyuzhou@gmail.com, hongxin001@e.ntu.edu.sg, boan@ntu.edu.sg).
Senlin Shu and Tao Xiang are with the College of Computer Science, Chongqing University, China (e-mail: shusenlin@126.com, txiang@cqu.edu.cn).
Lue Tao is with the National Key Lab for Novel Software Technology, Nanjing University, China (e-mail: taol@lamda.nju.edu.cn).
Gang Niu is with the RIKEN Center for Advanced Intelligence Project, Japan (e-mail: gang.niu.ml@gmail.com).
This work was done when Lei Feng was working at Chongqing University.
Corresponding author: Tao Xiang.

[26], the goal is to build a model to predict whether a new molecule is qualified to make a special drug or not, by learning from a set of known molecules. In this task, a molecule is considered as a bag, which contains many low-energy shapes, and each shape is considered as an instance. A bag label indicates whether a specific molecule is qualified to make a special drug, which depends on whether the molecule has some special shapes. It could be difficult for human experts to accurately figure out the exact bag label of each molecule by inspecting its low-energy shapes, because annotating all the low-energy shapes could incur unaffordable monetary costs. Fortunately, it would be much easier to judge whether two molecules share the same bag label, instead of knowing the exact bag label of each molecule. In this case, we refer to two bags that have the same bag label as a *similar* bag pair and two bags that have different bag labels as a *dissimilar* bag pair. Given such kind of weakly supervised data, an important question naturally arises: Is it possible for us to successfully learn an effective bag-level binary classifier from only similar and dissimilar bags without any labeled bags? Moreover, unlabeled bags are cheap and widely available, and many previous studies [4], [19], [50] showed that unlabeled data could be helpful to model training. Therefore, we also consider another important question: Can we exploit unlabeled data to further improve the learning performance?

In this paper, we provide affirmative answers to the above questions. We provide the first attempt to investigate MIL from similar-dissimilar-unlabeled bags and the main contributions of this paper can be summarized as follows:

- We propose a strong baseline method that trains an instance-level classifier by employing an unlabeled-unlabeled learning strategy.
- We also propose a convex formulation that trains a bag-level classifier based on empirical bag risk minimization. We present the detailed derivation processes when suitable loss functions are employed, and theoretically derive a generalization error bound for this method.
- Comprehensive experimental results show that our instance-level classifier works well, while our bag-level classifier is even better.

The rest of this paper is organized as follows. Section 2 introduces preliminary knowledge of related learning problems. Section 3 introduces technical details of the proposed methods for learning an instance-level classifier and learning a bag-level classifier. Section 4 reports experimental results on various datasets. Finally, Section 5 gives a conclusion of this paper.

2 Preliminaries

In this section, we introduce some preliminary knowledge of three related learning problems including binary classification, similar-dissimilar-unlabeled learning, and multiple-instance learning.

2.1 Binary Classification

In binary classification, let us denote by $\mathcal{X} \in \mathbb{R}^d$ the feature space with d dimensions and $\mathcal{Y} = \{-1, +1\}$ the binary label space. Every training example $(\boldsymbol{x}, y)$ is considered to be independently sampled from an unknown data distribution with probability density $p(\boldsymbol{x}, y)$. For binary classification, we aim to learn an instance-level binary classifier f that tries to minimize the following (expected) classification risk:

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\big[\ell(f(\boldsymbol{x}), y)\big], \tag{1}$$

where $\ell : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}_+$ represents a binary loss function and $\mathbb{E}_{p(\boldsymbol{x},y)}[\cdot]$ denotes the expectation over the joint probability density $p(\boldsymbol{x}, y)$. Given a set of training examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ identically and independently sampled from $p(\boldsymbol{x}, y)$, the empirical risk minimization method aims to minimize the following empirical risk

$$\widehat{R}(f) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i).$$

Since $\mathbb{E}_{p(\boldsymbol{x},y)}[\widehat{R}(f)] = R(f)$, we refer to $\widehat{R}(f)$ as an unbiased risk estimator of $R(f)$.

2.2 Similar-Dissimilar-Unlabeled Classification

Recently, the pairwise similarity [20] between two data points has been used to serve as weak supervision information for training an effective binary classifier. Bao et al. [50] showed that we can successfully learn an effective binary classifier, given only similar data pairs and unlabeled data points, where a similar data pair means the two data points share the same label. Later, Shimada et al. [19] further incorporated dissimilar data pairs into model training. Here, we introduce the probability densities of similar, dissimilar, and unlabeled data:

$$\begin{aligned}
p_{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}') &= \frac{\pi^2 p_+(\boldsymbol{x})p_+(\boldsymbol{x}') + (1-\pi)^2 p_-(\boldsymbol{x})p_-(\boldsymbol{x}')}{\pi^2 + (1-\pi)^2},\\
p_{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{x}') &= \frac{1}{2}p_+(\boldsymbol{x})p_-(\boldsymbol{x}') + \frac{1}{2}p_+(\boldsymbol{x}')p_-(\boldsymbol{x}),\\
p_{\mathrm{U}}(\boldsymbol{x}) &= \pi p_+(\boldsymbol{x}) + (1-\pi)p_-(\boldsymbol{x}),
\end{aligned}$$

where $\pi = p(y = +1)$ denotes the (positive) class prior, $p_+(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = +1)$ and $p_-(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = -1)$ denote the probability densities of positive and negative data respectively. Given the above probability densities of similar, dissimilar, and unlabeled data, previous studies [19], [50] showed that the classification risk $R(f)$ can be recovered by the pairwise combination of similar, dissimilar, and unlabeled data, i.e.,

$$\begin{aligned}
R(f) = R_{\mathrm{SD}}(f) &= \pi_{\mathrm{S}}\mathbb{E}_{p_{\mathrm{S}}(\boldsymbol{x},\boldsymbol{x}')}\Big[\frac{\mathcal{L}(f(\boldsymbol{x}), +1) + \mathcal{L}(f(\boldsymbol{x}'), +1)}{2}\Big]\\
&\quad + \pi_{\mathrm{D}}\mathbb{E}_{p_{\mathrm{D}}(\boldsymbol{x},\boldsymbol{x}')}\Big[\frac{\mathcal{L}(f(\boldsymbol{x}), -1) + \mathcal{L}(f(\boldsymbol{x}'), -1)}{2}\Big],\\
R(f) = R_{\mathrm{SU}}(f) &= \pi_{\mathrm{S}}\mathbb{E}_{p_{\mathrm{S}}(\boldsymbol{x},\boldsymbol{x}')}\Big[\frac{\widetilde{\mathcal{L}}(f(\boldsymbol{x})) + \widetilde{\mathcal{L}}(f(\boldsymbol{x}'))}{2}\Big]\\
&\quad + \mathbb{E}_{p_{\mathrm{U}}(\boldsymbol{x})}\Big[\mathcal{L}(f(\boldsymbol{x}), -1)\Big],\\
R(f) = R_{\mathrm{DU}}(f) &= \pi_{\mathrm{D}}\mathbb{E}_{p_{\mathrm{D}}(\boldsymbol{x},\boldsymbol{x}')}\Big[-\frac{\widetilde{\mathcal{L}}(f(\boldsymbol{x})) + \widetilde{\mathcal{L}}(f(\boldsymbol{x}'))}{2}\Big]\\
&\quad + \mathbb{E}_{p_{\mathrm{U}}(\boldsymbol{x})}\Big[\mathcal{L}(f(\boldsymbol{x}), +1)\Big],
\end{aligned}$$

where $\pi_{\mathrm{S}} = \pi^2 + (1-\pi)^2$ denotes the similarity prior, $\pi_{\mathrm{D}} = 1 - \pi_{\mathrm{S}}$ denotes the dissimilarity prior, and

$$\mathcal{L}\big(f(\boldsymbol{x}), t\big) = \frac{\pi}{2\pi - 1}\ell(f(\boldsymbol{x}), t) - \frac{1-\pi}{2\pi - 1}\ell(f(\boldsymbol{x}), -t), \tag{2}$$

$$\widetilde{\mathcal{L}}(f(\boldsymbol{x})) = \frac{1}{2\pi - 1}\ell\big(f(\boldsymbol{x}), +1\big) - \frac{1}{2\pi - 1}\ell\big(f(\boldsymbol{x}), -1\big). \tag{3}$$

Because $R(f)$ can be equivalently represented by $R_{\mathrm{SD}}(f)$, $R_{\mathrm{SU}}(f)$, and $R_{\mathrm{DU}}(f)$, we can learn a binary classifier from similar data pairs sampled from $p_{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}')$, dissimilar data pairs sampled from $p_{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{x}')$, and unlabeled data sampled from $p_{\mathrm{U}}(\boldsymbol{x})$, by minimizing the empirical approximation of $R_{\mathrm{SD}}(f)$, $R_{\mathrm{SU}}(f)$, and $R_{\mathrm{DU}}(f)$.

2.3 Multiple-Instance Learning

In MIL [30], each training example is represented by a bag of instances. Let us denote the MIL training set by $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ where $X_i = \{\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ij}, \ldots, \boldsymbol{x}_{ib_i}\}$ is a bag of b_i instances, with $\boldsymbol{x}_{ij} \in \mathcal{X}$ representing the j-th instance in X_i. The bag label $Y_i = +1$ means that the bag X_i has at least one positive instance, and $Y_i = -1$ contains no positive instances. It is noteworthy that all the instances in the bag are unknown and we only know the binary label of the bag. Given this kind of data, MIL aims to construct a bag-level binary classifier to classify unseen test bags.

In order to obtain a bag-level binary classifier, some representative methods [51], [52] adapted the multiple-instance representations to single-instance algorithms by representation transformation. Following this strategy, in this work, we construct a bag-level linear-in-parameter classifier with a specially designed kernel that represents a bag by a single feature vector. Specifically, the bag-level linear-in-parameter classifier is formulated as follows:

$$g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X), \tag{4}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ is the learning parameters, and $\boldsymbol{\phi}(\cdot) \in \mathbb{R}^n$ is a vector of basis functions defined as

$$\boldsymbol{\phi}(X) = \begin{bmatrix} \widetilde{\mathcal{K}}(X, X_1) \\ \vdots \\ \widetilde{\mathcal{K}}(X, X_n) \end{bmatrix}. \tag{5}$$

Here, we can see that the bag representation relies on a special kernel $\widetilde{\mathcal{K}}$. Gartner et al. [53] proposed multiple-instance kernels, which aims at maping a bag to a feature space. A representative multiple-instance kernel called statistical kernel is defined as $\widetilde{\mathcal{K}}(X, X') := \mathcal{K}(\boldsymbol{s}(X), \boldsymbol{s}(X'))$, where $\mathcal{K}$ is an ordinary kernel function and $\boldsymbol{s}(X)$ is a statistic of the bag X. A common choice of $\boldsymbol{s}(X)$ is the minimax statistic:

$$\boldsymbol{s}(X) := [\min_{\boldsymbol{x}\in X} x^{(1)}, \ldots, \min_{\boldsymbol{x}\in X} x^{(d)}, \max_{\boldsymbol{x}\in X} x^{(1)}, \ldots, \max_{\boldsymbol{x}\in X} x^{(d)}]^\top,$$

where $x^{(i)}$ denotes the i-th element of $\boldsymbol{x}$. Gartner et al. [53] also empirically showed that satisfactory performance can be achieved when the polynomial kernel and the minimax statistic $\boldsymbol{s}(X)$ are used in the statistical kernel $\widetilde{\mathcal{K}}$:

$$\widetilde{\mathcal{K}}(X, X') = \left(\boldsymbol{s}(X)^\top \boldsymbol{s}(X') + 1\right)^p, \tag{6}$$

where p is a hyper-parameter.

In summary, we aim to learn a bag-level classifier $g(X)$ (defined in Eqs. (4), (5), and (6)) for MIL.

3 Multiple-Instance Learning from Similar, Dissimilar, and Unlabeled Bags

In this section, we first define the generation process of similar, dissimilar, and unlabeled bags. Based on this data generation process, we propose a strong baseline method that trains an instance-level classifier and a convex formulation that trains a bag-level classifier.

3.1 Generation Process of Training Bags

Inspired by previous studies [19] on instance-level data generation process of similar, dissimilar, and unlabeled instances, we adopt an analogous generation process of similar, dissimilar, and unlabeled bags. Let us denote by the set of similar and dissimilar bag pairs as $\mathcal{D}_{\mathrm{SD}} = \{(X_i, X_i', Z_i)\}_{i=1}^{N_{\mathrm{SD}}}$ where $Z_i = +1$ if $Y_i = Y_i'$, otherwise $Z_i = -1$. We can obtain the respective sets of similar and dissimilar bag pairs as follows:

$$\begin{aligned}
&\mathcal{D}_{\mathrm{S}} = \{(X_{\mathrm{S}.i}, X_{\mathrm{S}.i}')\}_{i=1}^{N_{\mathrm{S}}} = \{(X, X') \mid (X, X', Z = +1) \in \mathcal{D}_{\mathrm{SD}}\}, \\
&\mathcal{D}_{\mathrm{D}} = \{(X_{\mathrm{D}.i}, X_{\mathrm{D}.i}')\}_{i=1}^{N_{\mathrm{D}}} = \{(X, X') \mid (X, X', Z = -1) \in \mathcal{D}_{\mathrm{SD}}\}, \\
&\mathcal{D}_{\mathrm{S}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{S}}(X, X'), \quad \mathcal{D}_{\mathrm{D}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{D}}(X, X').
\end{aligned}$$

Then, we introduce the following notations representing the priors and conditional probability densities of similar and dissimilar bag pairs:

$$\begin{aligned}
\theta_{\mathrm{S}} &:= p(Y = Y'), \quad p_{\mathrm{S}}(X, X') := p(X, X' \mid Y = Y'), \\
\theta_{\mathrm{D}} &:= p(Y \neq Y'), \quad p_{\mathrm{D}}(X, X') := p(X, X' \mid Y \neq Y').
\end{aligned}$$

By further denoting by the $p(Y = 1) = \theta$ the bag-level (positive) class prior, we have

$$\begin{aligned}
\theta_{\mathrm{S}} &= p(Y = +1)p(Y' = +1) + p(Y = -1)p(Y' = -1) \\
&= \theta^2 + (1 - \theta)^2, \\
\theta_{\mathrm{D}} &= p(Y = +1)p(Y' = -1) + p(Y = -1)p(Y' = +1) \\
&= 2\theta(1 - \theta), \\
p_{\mathrm{S}}(X, X') &= \frac{\theta^2}{\theta_{\mathrm{S}}} p_+(X)p_+(X') + \frac{(1-\theta)^2}{\theta_{\mathrm{S}}} p_-(X)p_-(X'), \\
p_{\mathrm{D}}(X, X') &= \frac{1}{2} p_+(X)p_-(X') + \frac{1}{2} p_+(X')p_-(X).
\end{aligned}$$

In addition, we consider that unlabeled bags are generated as follows:

$$\mathcal{D}_{\mathrm{U}} := \{X_{\mathrm{U}.i}\}_{i=1}^{N_{\mathrm{U}}} \sim p_{\mathrm{U}}(X) = \theta p_+(X) + (1 - \theta)p_-(X).$$

Given the above generation process of similar, dissimilar, and unlabeled bags, we can derive a strong baseline method that learns an instance-level classifier by employing an unlabeled-unlabeled learning strategy and a convex formulation that learns a bag-level classifier based on empirical bag risk minimization.

3.2 Learning An Instance-Level Binary Classifier

Based on the above data generation process, we present a strong baseline method that trains an instance-level binary classifier for MIL from similar-dissimilar-unlabeled bags. By using the instance-level classifier to predict the label of each instance in an unseen test bag, we are able to predict the label of the bag.

Our motivation stems from unlabeled-unlabeled learning [21], [22], which aims to train an instance-level binary classifier from two sets of unlabeled data points with different class priors. In our problem setting, the three sets $\mathcal{D}_{\mathrm{S}}$, $\mathcal{D}_{\mathrm{D}}$, and $\mathcal{D}_{\mathrm{U}}$ can be considered as unlabeled datasets with different (instance-level) class priors. In this way, we are able to learn an instance-level binary classifier from two of them, by using the unlabeled-labeled learning method [21].

Here, the key problem becomes how to figure out the instance-level class priors of $\mathcal{D}_{\mathrm{S}}$, $\mathcal{D}_{\mathrm{D}}$, and $\mathcal{D}_{\mathrm{U}}$. It would be easy for us to verify that the proportion of positive bags in $\mathcal{D}_{\mathrm{S}}$ is $\theta^2/\theta_{\mathrm{S}}$, the proportion of positive bags in

$\mathcal{D}_{\mathrm{D}}$ is $1/2$, and the proportion of positive bags in $\mathcal{D}_{\mathrm{U}}$ is θ. Then, following Bao et al. [54], we consider that the instances in positive bags are drawn from the instance-level marginal distribution $p(\boldsymbol{x})$, where $p(\boldsymbol{x})$ is defined as $p(\boldsymbol{x}) = \pi p(\boldsymbol{x} \mid y = +1) + (1-\pi)p(\boldsymbol{x} \mid y = -1)$ and $\pi = p(y = +1)$ is the instance-level class prior of the set of unlabeled instances. Besides, we also consider that the instances in negative bags are drawn from the instance-level negative class-conditional distribution $p(\boldsymbol{x} \mid y = -1)$. In this way, we can know that the instance-level class priors of $\mathcal{D}_{\mathrm{S}}$, $\mathcal{D}_{\mathrm{D}}$, and $\mathcal{D}_{\mathrm{U}}$ are $\pi\theta^2/\theta_{\mathrm{S}}$, $\pi/2$, and $\theta\pi$, respectively. Since the instance-level class priors of the three sets are different, we can train a binary classifier by minimizing the empirical approximation of the risk provided in the following proposition.

Proposition 1 (Theorem 4 in [21])**.** *Let η and η' be different class priors of two unlabeled datasets (with $\eta > \eta'$), and $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{tr}'}(\boldsymbol{x}')$ be the densities of two datasets of unlabeled data, respectively. The classification risk $R(f)$ in Eq. (1) can be equivalently represented as*

$$
\begin{aligned}
R(f) = R_{\mathrm{UU}}(f) = \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}\Big[&\frac{(1-\eta')\pi}{\eta-\eta'}\ell_+\big(f(\boldsymbol{x})\big) \\
&- \frac{\eta'(1-\pi)}{\eta-\eta'}\ell_-\big(f(\boldsymbol{x})\big)\Big] \\
+ \mathbb{E}_{p_{\mathrm{tr}'}(\boldsymbol{x}')}\Big[&\frac{\eta(1-\pi)}{\eta-\eta'}\ell_-\big(f(\boldsymbol{x}')\big) - \frac{(1-\eta)\pi}{\eta-\eta'}\ell_+\big(f(\boldsymbol{x}')\big)\Big],
\end{aligned}
$$

where

$$
\ell_+\big(f(\boldsymbol{x})\big) = \ell\big(f(\boldsymbol{x}), +1\big), \quad \ell_-\big(f(\boldsymbol{x})\big) = \ell\big(f(\boldsymbol{x}), -1\big).
$$

By incorporating Proposition 1 into our problem setting, we can know that

- By considering $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{D}}$ as two unlabeled datasets with different instance-level class priors (η and η'), we have $\eta = \pi\theta^2/\theta_{\mathrm{S}}$ and $\eta' = \pi/2$.
- By considering $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{U}}$ as two unlabeled datasets with different instance-level class priors (η and η'), we have $\eta = \pi\theta^2/\theta_{\mathrm{S}}$ and $\eta' = \theta\pi$.
- By considering $\mathcal{D}_{\mathrm{U}}$ and $\mathcal{D}_{\mathrm{D}}$ as two unlabeled datasets with different instance-level class priors (η and η'), we have $\eta = \theta\pi$ and $\eta' = \pi/2$.

In this way, we can minimizing the following empirical risks for learning an instance-level classifier:

$$
\begin{aligned}
\widehat{R}^{\mathrm{ins}}_{\mathrm{SD}}(f) &= \frac{1}{|\mathcal{D}_{\mathrm{S}}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{\mathrm{S}}}\Big(\frac{(2-\pi)\theta_{\mathrm{S}}}{2\theta-1}\ell_+\big(f(\boldsymbol{x})\big) - \frac{(1-\pi)\theta_{\mathrm{S}}}{2\theta-1}\ell_-\big(f(\boldsymbol{x})\big)\Big) \\
&+ \frac{1}{|\mathcal{D}_{\mathrm{D}}|}\sum_{\boldsymbol{x}'\in\mathcal{D}_{\mathrm{D}}}\Big(\frac{2(1-\pi)\theta^2}{2\theta-1}\ell_-\big(f(\boldsymbol{x}')\big) - \frac{2(\theta_{\mathrm{S}}-\theta^2\pi)}{2\theta-1}\ell_+\big(f(\boldsymbol{x}')\big)\Big), \\
\widehat{R}^{\mathrm{ins}}_{\mathrm{SU}}(f) &= \frac{1}{|\mathcal{D}_{\mathrm{S}}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{\mathrm{S}}}\Big(\frac{(1-\pi\theta)\theta_{\mathrm{S}}}{\theta^2-\theta\theta_{\mathrm{S}}}\ell_+\big(f(\boldsymbol{x})\big) - \frac{(1-\pi)\theta_{\mathrm{S}}}{\theta-\theta_{\mathrm{S}}}\ell_-\big(f(\boldsymbol{x})\big)\Big) \\
&+ \frac{1}{|\mathcal{D}_{\mathrm{S}}|}\sum_{\boldsymbol{x}'\in\mathcal{D}_{\mathrm{U}}}\Big(\frac{(1-\pi)\theta}{\theta-\theta_{\mathrm{S}}}\ell_-\big(f(\boldsymbol{x}')\big) - \frac{\theta_{\mathrm{S}}-\theta^2\pi}{\theta^2-\theta\theta_{\mathrm{S}}}\ell_+\big(f(\boldsymbol{x}')\big)\Big), \\
\widehat{R}^{\mathrm{ins}}_{\mathrm{DU}}(f) &= \frac{1}{|\mathcal{D}_{\mathrm{U}}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{\mathrm{U}}}\Big(\frac{2-\pi}{2\theta-1}\ell_+\big(f(\boldsymbol{x})\big) - \frac{1-\pi}{2\theta-1}\ell_-\big(f(\boldsymbol{x})\big)\Big) \\
&+ \frac{1}{|\mathcal{D}_{\mathrm{D}}|}\sum_{\boldsymbol{x}'\in\mathcal{D}_{\mathrm{D}}}\Big(\frac{2(1-\pi)\theta}{2\theta-1}\ell_-\big(f(\boldsymbol{x}')\big) - \frac{2(1-\theta\pi)}{2\theta-1}\ell_+\big(f(\boldsymbol{x}')\big)\Big).
\end{aligned}
$$

Given similar, dissimilar, and unlabeled data (i.e., $\mathcal{D}_{\mathrm{S}}$, $\mathcal{D}_{\mathrm{D}}$, $\mathcal{D}_{\mathrm{U}}$) simultaneously, we can train an instance-level binary classifier f by minimizing the following empirical risk:

$$
\widehat{R}^{\mathrm{ins}}_{\mathrm{SDU}}(f) = \alpha_1\widehat{R}^{\mathrm{ins}}_{\mathrm{SD}}(f) + \alpha_2\widehat{R}^{\mathrm{ins}}_{\mathrm{SU}}(f) + \alpha_3\widehat{R}^{\mathrm{ins}}_{\mathrm{DU}}(f), \tag{7}
$$

where α_1, α_2, and α_3 are positive real values and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. By inserting a specific binary loss function (e.g., hinge loss) into Eq. (7), we can obtain a strong baseline method for MIL from similar-dissimilar-unlabeled bags. It is worth noting that MIL from similar-dissimilar-unlabeled bags would reduce to MIL from similar-dissimilar (similar-unlabeled or dissimilar-unlabeled) bags when $\alpha_1 = 1$ ($\alpha_2 = 1$ or $\alpha_3 = 1$) and the details of these three special cases will be empirically investigated in Section 4.2.

It should be noted that the goal of this paper is to predict bag-level labels. Therefore, learning an instance-level classifier could be considered as a more complex solution to our problem than directly learning a bag-level classifier. According to *Ockham's Razor* that the simplest is usually the right one, we can expect that directly learning a bag-level classifier could be superior to learning an instance-level classifier. Therefore, we further present an empirical risk minimization method for directly learning a bag-level binary classifier.

3.3 Learning A Bag-Level Binary Classifier

Based on the data generation process defined in Section 3.1, motivated by previous studies [19], [50] on the derived risks of learning from similar, dissimilar, and unlabeled data (i.e., $R_{\mathrm{SD}}(f)$, $R_{\mathrm{SU}}(f)$, $R_{\mathrm{DU}}(f)$ defined in Section 2.2), we propose to train a bag-level classifier g by minimizing the following empirical risks:

$$
\begin{aligned}
\widehat{R}^{\mathrm{bag}}_{\mathrm{SD}}(g) &= \frac{\theta_{\mathrm{S}}}{2N_{\mathrm{S}}}\sum\nolimits_{i=1}^{N_{\mathrm{S}}}\Big(\mathcal{L}\big(g(X_{\mathrm{S}.i}), +1\big) + \mathcal{L}\big(g(X'_{\mathrm{S}.i}), +1\big)\Big) \\
&+ \frac{\theta_{\mathrm{D}}}{2N_{\mathrm{D}}}\sum\nolimits_{j=1}^{N_{\mathrm{D}}}\Big(\mathcal{L}\big(g(X_{\mathrm{D}.j}), -1\big) + \mathcal{L}\big(g(X'_{\mathrm{D}.j}), -1\big)\Big), \\
\widehat{R}^{\mathrm{bag}}_{\mathrm{SU}}(g) &= \frac{\theta_{\mathrm{S}}}{2N_{\mathrm{S}}}\sum\nolimits_{i=1}^{N_{\mathrm{S}}}\Big(\widetilde{\mathcal{L}}\big(g(X_{\mathrm{S}.i})\big) + \widetilde{\mathcal{L}}\big(g(X'_{\mathrm{S}.i})\big)\Big) \\
&+ \frac{1}{N_{\mathrm{U}}}\sum\nolimits_{j=1}^{N_{\mathrm{U}}}\Big(\mathcal{L}\big(g(X_{\mathrm{U}.j}), -1\big)\Big), \\
\widehat{R}^{\mathrm{bag}}_{\mathrm{DU}}(g) &= \frac{\theta_{\mathrm{D}}}{2N_{\mathrm{D}}}\sum\nolimits_{i=1}^{N_{\mathrm{D}}} -\Big(\widetilde{\mathcal{L}}\big(g(X_{\mathrm{D}.i})\big) + \widetilde{\mathcal{L}}\big(g(X'_{\mathrm{D}.i})\big)\Big) \\
&+ \frac{1}{N_{\mathrm{U}}}\sum\nolimits_{j=1}^{N_{\mathrm{U}}}\Big(\mathcal{L}\big(g(X_{\mathrm{U}.j}), +1\big)\Big),
\end{aligned}
$$

where $\mathcal{L}\big(g(X), t\big)$ ($t \in \{+1, -1\}$) and $\widetilde{\mathcal{L}}\big(g(X)\big)$ are analogously defined in Eqs. (2) and (3). Given similar, dissimilar, and unlabeled bags (i.e., $\mathcal{D}_{\mathrm{S}}$, $\mathcal{D}_{\mathrm{D}}$, and $\mathcal{D}_{\mathrm{U}}$) simultaneously, we can train a bag-level binary classifier g by minimizing the following empirical risk:

$$
\widehat{R}^{\mathrm{bag}}_{\mathrm{SDU}}(g) = \beta_1\widehat{R}^{\mathrm{bag}}_{\mathrm{SD}}(g) + \beta_2\widehat{R}^{\mathrm{bag}}_{\mathrm{SU}}(g) + \beta_3\widehat{R}^{\mathrm{bag}}_{\mathrm{DU}}(g), \tag{8}
$$

where β_1, β_2, and β_3 are positive values and satisfy the condition $\beta_1 + \beta_2 + \beta_3 = 1$. Thus, the bag-level method also has three special cases and the details will be empirically investigated in Section 4.2. It is noteworthy that we need to use a specific binary loss function in the three empirical risks $\widehat{R}^{\mathrm{bag}}_{\mathrm{SD}}(g)$, $\widehat{R}^{\mathrm{bag}}_{\mathrm{SU}}(g)$, and $\widehat{R}^{\mathrm{bag}}_{\mathrm{DU}}(g)$. However, we find that the three empirical risks may not be convex even if a convex binary loss function ℓ (e.g., the hinge loss) is used, and thus the final empirical risk $\widehat{R}^{\mathrm{bag}}_{\mathrm{SDU}}(g)$ may not be convex. Fortunately, as verified by previous studies [19], [50], if the used convex binary loss function ℓ satisfies the following condition:

$$
\ell\big(g(X), +1\big) - \ell\big(g(X), -1\big) = -g(X),
$$

then the three empirical risks are convex, and thus the final empirical risk $\widehat{R}^{\mathrm{bag}}_{\mathrm{SDU}}(g)$ is a convex objective function.

For the final empirical risk $\widehat{R}^{\text{bag}}_{\text{SDU}}(g)$, we can find that the two bags X and X' in the same similar or dissimilar bag pair are symmetric and interchangeable, hence they play the same role. Therefore, we can arrange them together. Specifically, we can equivalently denote the sets by $\mathcal{D}_{\text{S}} = \{X_{\text{S}.i}\}_{i=1}^{2N_{\text{S}}} = \{X_{\text{S}.i}\}_{i=1}^{N_{\text{S}}} \cup \{X'_{\text{S}.i}\}_{i=1}^{N_{\text{S}}}$ and $\mathcal{D}_{\text{D}} = \{X_{\text{D}.j}\}_{j=1}^{2N_{\text{D}}} = \{X_{\text{D}.j}\}_{j=1}^{N_{\text{D}}} \cup \{X'_{\text{D}.j}\}_{j=1}^{N_{\text{D}}}$. In this way, by further substituting specific binary loss functions that satisfy the condition $\ell\big(g(X),+1\big) - \ell\big(g(X),-1\big) = -g(X)$ into the three empirical risks $\widehat{R}^{\text{bag}}_{\text{SD}}(g)$, $\widehat{R}^{\text{bag}}_{\text{SU}}(g)$, and $\widehat{R}^{\text{bag}}_{\text{DU}}(g)$, we can rewrite them in more specific forms:

$$
\begin{aligned}
\widehat{R}^{\text{bag}}_{\text{SD}}(g) &= \frac{\theta_{\text{S}}}{2N_{\text{S}}}\sum\nolimits_{i=1}^{2N_{\text{S}}}\Big(\ell(g(X_{\text{S}.i}),+1) - \frac{1-\theta}{2\theta-1}g(X_{\text{S}.i}))\Big) \\
&\quad + \frac{\theta_{\text{D}}}{2N_{\text{D}}}\sum\nolimits_{j=1}^{2N_{\text{D}}}\Big(\ell(g(X_{\text{D}.j}),-1) + \frac{1-\theta}{2\theta-1}g(X_{\text{D}.j})\Big), \\
\widehat{R}^{\text{bag}}_{\text{SU}}(g) &= \frac{\theta_{\text{S}}}{2N_{\text{S}}}\sum\nolimits_{i=1}^{2N_{\text{S}}} -\frac{1}{2\theta-1}g(X_{\text{S}.i}) \\
&\quad + \frac{1}{N_{\text{U}}}\sum\nolimits_{j=1}^{N_{\text{U}}}\Big(\ell(g(X_{\text{U}.j}),-1) + \frac{1-\theta}{2\theta-1}g(X_{\text{U}.j})\Big), \\
\widehat{R}^{\text{bag}}_{\text{DU}}(g) &= \frac{\theta_{\text{D}}}{2N_{\text{D}}}\sum\nolimits_{i=1}^{2N_{\text{D}}} \frac{1}{2\theta-1}g(X_{\text{D}.i}) \\
&\quad + \frac{1}{N_{\text{U}}}\sum\nolimits_{j=1}^{N_{\text{U}}}\Big(\ell(g(X_{\text{U}.j}),+1) - \frac{1-\theta}{2\theta-1}g(X_{\text{U}.j})\Big).
\end{aligned}
$$

For the used model $g(X) = \boldsymbol{w}^\top\boldsymbol{\phi}(X)$, we can represent the vector of basis function $\boldsymbol{\phi}$ as

$$
\boldsymbol{\phi}(X) = \begin{bmatrix} \widetilde{\mathcal{K}}(X, X_{\text{S}.1}) \\ \vdots \\ \widetilde{\mathcal{K}}(X, X_{\text{S}.2N_{\text{S}}}) \\ \widetilde{\mathcal{K}}(X, X_{\text{D}.1}) \\ \vdots \\ \widetilde{\mathcal{K}}(X, X_{\text{D}.2N_{\text{D}}}) \\ \widetilde{\mathcal{K}}(X, X_{\text{U}.1}) \\ \vdots \\ \widetilde{\mathcal{K}}(X, X_{\text{U}.N_{\text{U}}}) \end{bmatrix},
$$

where $\widetilde{\mathcal{K}}$ is defined in Eq. (6).

Here, we need to consider a specific convex binary loss function ℓ in Eq. (8) that satisfies the condition $\ell(g(X),+1) - \ell(g(X),-1) = -g(X)$ for practical implementation. In this paper, we consider the widely used squared loss and double hinge loss [55].

3.3.1 Practical Implementation with Specific Losses

For convenience, let us first introduce the following notations:

$$
\begin{aligned}
\boldsymbol{X}_{\text{S}} &= [\boldsymbol{\phi}(X_{\text{S}.1}), \ldots, \boldsymbol{\phi}(X_{\text{S}.2N_{\text{S}}})]^\top \in \mathbb{R}^{2N_{\text{S}}\times(2N_{\text{S}}+2N_{\text{D}}+N_{\text{U}})}, \\
\boldsymbol{X}_{\text{D}} &= [\boldsymbol{\phi}(X_{\text{D}.1}), \ldots, \boldsymbol{\phi}(X_{\text{D}.2N_{\text{D}}})]^\top \in \mathbb{R}^{2N_{\text{D}}\times(2N_{\text{S}}+2N_{\text{D}}+N_{\text{U}})}, \\
\boldsymbol{X}_{\text{U}} &= [\boldsymbol{\phi}(X_{\text{U}.1}), \ldots, \boldsymbol{\phi}(X_{\text{U}}.N_{\text{U}})]^\top \in \mathbb{R}^{N_{\text{U}}\times(2N_{\text{S}}+2N_{\text{D}}+N_{\text{U}})}.
\end{aligned}
$$

Then, we can insert the squared loss and the double hinge loss into Eq. (8) for practical implementation. By adopting the widely used L_2 regularization to restore stability and ensure generalization, we have the following objective function:

$$
J^{\text{bag}}_{\text{SDU}}(\boldsymbol{w}) := \widehat{R}^{\text{bag}}_{\text{SDU}}(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2, \tag{9}
$$

where $\widehat{R}^{\text{bag}}_{\text{SDU}}(\boldsymbol{w})$ is defined in Eq. (8) and $\lambda > 0$ is a hyper-parameter that controls the importance of the L_2 regularization. In what follows, we give the final formulation when we use the squared loss and the double hinge loss.

Squared Loss. We use the squared loss [55] defined as $\ell_{\text{SQ}}(z,t) = \frac{1}{4}(tz-1)^2$. By inserting it into $J^{\text{bag}}_{\text{SDU}}(\boldsymbol{w})$ (i.e., Eq. (9)), we have

$$
\begin{aligned}
J^{\text{bag}}_{\text{SDU}}(\boldsymbol{w}) = \boldsymbol{w}^\top\Big[&\frac{\beta_1}{8}(\frac{\theta_{\text{S}}}{N_{\text{S}}}\boldsymbol{X}_{\text{S}}^\top X_{\text{S}} + \frac{\theta_{\text{D}}}{N_{\text{D}}}X_{\text{D}}^\top\boldsymbol{X}_{\text{D}}) + \frac{\beta_2+\beta_3}{4N_{\text{U}}}X_{\text{U}}^\top X_{\text{U}} \\
&+ \frac{\lambda}{2}I_{d\times d}\Big]\boldsymbol{w} + \frac{1}{2\theta-1}\Big[-\frac{\theta_{\text{S}}}{2N_{\text{S}}}(\frac{\beta_1}{2}+\beta_2)\mathbf{1}_{2N_{\text{S}}}^\top X_{\text{S}} \\
&+ \frac{\theta_{\text{D}}}{2N_{\text{D}}}(\frac{\beta_1}{2}+\beta_3)\mathbf{1}_{2N_{\text{D}}}^\top\boldsymbol{X}_{\text{D}} + \frac{1}{2N_{\text{U}}}(\beta_2-\beta_3)\mathbf{1}_{N_{\text{U}}}^\top\boldsymbol{X}_{\text{U}}\Big]\boldsymbol{w} + \text{const},
\end{aligned}
$$

where $\boldsymbol{I}_{d\times d}$ denotes the $d\times d$ identity matrix with $d = 2N_{\text{S}} + 2N_{\text{D}} + N_{\text{U}}$. By setting the derivative with respect to $\boldsymbol{w}$ to zero, an analytical solution can be obtained:

$$
\begin{aligned}
\boldsymbol{w} = \frac{1}{2\theta-1}\Big[&\frac{\beta_1}{4}(\frac{\theta_{\text{S}}}{N_{\text{S}}}\boldsymbol{X}_{\text{S}}^\top\boldsymbol{X}_{\text{S}} + \frac{\theta_{\text{D}}}{N_{\text{D}}}\boldsymbol{X}_{\text{D}}^\top\boldsymbol{X}_{\text{D}}) + \lambda\boldsymbol{I}_{d\times d} \\
&+ \frac{\beta_2+\beta_3}{2N_{\text{U}}}\boldsymbol{X}_{\text{U}}^\top\boldsymbol{X}_{\text{U}}\Big]^{-1}\Big[\frac{\theta_{\text{S}}}{2N_{\text{S}}}(\frac{\beta_1}{2}+\beta_2)\boldsymbol{X}_{\text{S}}^\top\mathbf{1}_{2N_{\text{S}}} \\
&- \frac{\theta_{\text{D}}}{2N_{\text{D}}}(\frac{\beta_1}{2}+\beta_3)\boldsymbol{X}_{\text{D}}^\top\mathbf{1}_{2N_{\text{D}}} - \frac{1}{2N_{\text{U}}}(\beta_2-\beta_3)\boldsymbol{X}_{\text{U}}^\top\mathbf{1}_{N_{\text{U}}}\Big].
\end{aligned} \tag{10}
$$

where $\mathbf{1}_{2N_{\text{S}}}$ denotes the $2N_{\text{S}}\times 1$ vector whose elements are all ones.

Double Hinge Loss. We use the double hinge loss [55] defined as $\ell_{\text{DH}}(z,t) = \max(-tz, \max(0, \frac{1}{2} - \frac{1}{2}tz))$. By inserting it into $J^{\text{bag}}_{\text{SDU}}(\boldsymbol{w})$ (i.e., Eq. (9)), we can simply the optimization objective as

$$
\min_{\boldsymbol{\gamma}} \frac{1}{2}\boldsymbol{\gamma}^\top\boldsymbol{P}\boldsymbol{\gamma} + \boldsymbol{q}^\top\boldsymbol{\gamma} \quad \text{s.t.} \quad \boldsymbol{G}\boldsymbol{\gamma} \le \boldsymbol{h}, \tag{11}
$$

where $\boldsymbol{\gamma}$ is the defined optimization variable, $\boldsymbol{P}$, $\boldsymbol{q}$, $\boldsymbol{G}$, and $\boldsymbol{h}$ are introduced notations (please refer to the supplementary materials for the details), and $\le$ for vectors denotes the element-wise inequality. As we can easily verify, Eq. (11) is a standard quadratic programming problem, which can be easily solved by any off-the-shelf quadratic programming tools.

3.3.2 Analysis of Generalization Error Bound

Here, we analyze the generalization error for our proposed convex formulation. Let $\mathscr{X}$ be the bag-level domain set and

$$
\mathcal{G} := \{g(X) = \boldsymbol{w}^\top\boldsymbol{\phi}(X) \mid \|\boldsymbol{w}\| \le C_{\boldsymbol{w}}, \sup\nolimits_{X\in\mathscr{X}}\|\boldsymbol{\phi}(X)\| \le C_{\boldsymbol{\phi}}\}
$$

be a given function class, where $\boldsymbol{\phi}$ is a vector of basis functions defined in Eq. (5). In this part of theoretical analysis, we simply adopt the double hinge loss as the used loss function ℓ because it is 1-Lipschitz, and this loss function is also used in our experiments. In contrast to the empirical risk $\widehat{R}^{\text{bag}}_{\text{SDU}}(g)$ in Eq. (8), we denote by $R^{\text{bag}}_{\text{SDU}}(g)$ its expected version. Then, we analyze the generalization error bound based on the widely used *Rademacher complexity* [56] and the Rademacher complexity of $\mathcal{G}$ (i.e., $\mathfrak{R}_n(\mathcal{G})$) can be normally bounded by $\mathfrak{R}_n(\mathcal{G}) \le C_{\mathcal{G}}/\sqrt{n}$.

Theorem 1. *With conditions above, for any $\delta > 0$, with probability at least $1 - 3\delta$, we have the following generalization error bound:*

$$
\begin{aligned}
&\sup_{g\in\mathcal{G}}\Big|R^{\text{bag}}_{\text{SDU}}(g) - \widehat{R}^{\text{bag}}_{\text{SDU}}(g)\Big| \\
&\le \Big[\frac{\theta\beta_1+\beta_2}{\sqrt{2N_{\text{S}}}} + \frac{\theta\beta_1+\beta_3}{\sqrt{2N_{\text{D}}}} + \frac{\theta\beta_2+\theta\beta_3}{\sqrt{N_{\text{U}}}}\Big]\frac{C_{\mathcal{G}}}{2\theta-1} \\
&+ \Big[\frac{(\beta_1+2\beta_2)\theta_{\text{S}}}{\sqrt{4N_{\text{S}}}} + \frac{(\beta_1+2\beta_3)\theta_{\text{D}}}{\sqrt{4N_{\text{D}}}} + \frac{\beta_2+\beta_3}{\sqrt{2N_{\text{U}}}}\Big]\frac{C_{\boldsymbol{w}}C_{\boldsymbol{\phi}}}{2\theta-1}\sqrt{\log\frac{4}{\delta}}.
\end{aligned}
$$

TABLE 1: The characteristics of the used benchmark datasets.

Dataset	# Features	# Positive bags	# Negative bags	# Avg. Pos. Ins. per bag	# Avg. Neg. Ins. per bag
Musk1	166	475	445	2.2±2.5	2.9±7.0
Musk2	166	413	607	8.9±22.7	49.9±169.7
Elephat	230	504	496	3.9±4.2	3.2±3.6
Fox	230	498	502	3.2±3.6	3.4±3.8
Tiger	230	506	494	2.8±3.1	3.4±3.9

TABLE 2: The characteristics of the used datasets for the biocreative text categorization task.

Dataset	#Features	#Positive bags	#Negative bags	#Avg. Pos. Ins. per bag	#Avg. Neg. Ins. per bag
Component	200	423	2707	2.9±8.7	8.9±7.6
Function	200	443	4799	1.8±6.8	8.8±7.0
Process	200	757	10961	1.4±6.0	8.7±6.9

TABLE 3: Classification accuracy of bag-level methods (i.e., convex formulation) using the double hinge loss. The best performance is highlighted in bold.

Datasets	CVX-SDU	CVX-SD	CVX-SU	CVX-DU
Musk1	**0.799** **(0.055)**	0.778 (0.079)	0.724 (0.080)	0.748 (0.056)
Musk2	**0.785** **(0.049)**	0.773 (0.042)	0.742 (0.073)	0.737 (0.075)
Elephant	**0.773** **(0.073)**	0.772 (0.058)	0.689 (0.084)	0.699 (0.066)
Fox	**0.701** **(0.016)**	0.708 (0.013)	0.691 (0.022)	0.701 (0.013)
Tiger	**0.728** **(0.059)**	0.725 (0.072)	0.660 (0.085)	0.685 (0.068)
Component	**0.836** **(0.021)**	0.811 (0.046)	0.778 (0.069)	0.827 (0.034)
Function	**0.875** **(0.025)**	0.841 (0.041)	0.830 (0.069)	0.855 (0.036)
Process	**0.883** **(0.020)**	0.877 (0.011)	0.869 (0.035)	0.863 (0.060)

TABLE 4: Classification accuracy of instance-level methods (i.e., strong baseline) using the double hinge loss. The best performance is highlighted in bold.

Datasets	BL-SDU	BL-SD	BL-SU	BL-DU
Musk1	**0.767** **(0.061)**	0.766 (0.053)	0.705 (0.105)	0.738 (0.044)
Musk2	**0.710** **(0.099)**	0.705 (0.084)	0.640 (0.079)	0.709 (0.076)
Elephant	**0.786** **(0.034)**	0.777 (0.037)	0.715 (0.043)	0.760 (0.044)
Fox	**0.665** **(0.047)**	0.655 (0.042)	0.618 (0.059)	0.652 (0.072)
Tiger	**0.728** **(0.076)**	0.730 (0.077)	0.648 (0.083)	0.731 (0.050)
Component	**0.723** **(0.134)**	0.618 (0.150)	0.390 (0.112)	0.418 (0.082)
Function	**0.742** **(0.076)**	0.640 (0.132)	0.413 (0.150)	0.441 (0.052)
Process	**0.813** **(0.070)**	0.701 (0.117)	0.503 (0.149)	0.549 (0.091)

This theorem shows that the generalization error decreases with the order $\mathcal{O}(1/\sqrt{N_{\mathrm{S}}} + 1/\sqrt{N_{\mathrm{D}}} + 1/\sqrt{N_{\mathrm{U}}})$. This is also the optimal parametric rate for empirical risk minimization without additional assumptions [57]. From Theorem 1, we can find that the generalization error would be decreased if we increase the number of similar, dissimilar, and unlabeled bags.

4 Experiments

In this section, we conduct extensive experiments on both benchmark datasets and text categorization datasets.

4.1 Experimental settings

Compared Methods. As this is the first work on MIL from similar and dissimilar bags, there are no previous methods for solving the new MIL problem. Although some previous methods tried to exploit unlabeled data for algorithm development (e.g., semi-supervised or unsupervised domain adaptation [58], [59]), they cannot be used to solve our problem. We compare our proposed convex formulation (in Eq. (8)) including two bag-level methods: **CVX-SQ** (using the squared loss) and **CVX-DH** (using the double hinge loss) with the proposed baseline methods (in Eq. (7)) including: **BL-SQ** (using the squared loss), **BL-DH** (using the double hinge loss), **BL-HG** (using the hinge loss $\ell(z,t) = \max(0, 1-tz)$), **BL-LG** (using the logistic loss $\ell(z,t) = \log(1+\exp(-tz))$), **BL-RP** (using the ramp loss $\ell(z,t) = \frac{1}{2}\max(0, \min(2, 1-tz))$), and **BL-SG** (using the sigmoid loss $\ell(z,t) = 1/1+\exp(tz)$). For CVX-SQ, we directly use the analytical solution in Eq. (10). For CVX-DH, we solve the standard quadratic programming problem in Eq. (11) using CVXOPT [60]. For other compared baseline methods, we implement them using PyTorch [61].

Hyper-parameter Settings. For CVX-SQ and CVX-DH, the degree of the polynomial kernel is simply fixed at 1, and the regularization parameter λ is selected from $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. For other compared methods, the number of training epochs is set to 1,000 with full batch size, the learning rate is set to 10^{-3}, and the weight decay is selected from $\{10^{-4}, 10^{-3}, 10^{-2}\}$. For the above methods, it seems that the bag-level class prior θ needs to be known in advance. Actually, we showed that θ can be empirically estimated according to the data generation process in Section

TABLE 5: Classification accuracy of compared methods (with various loss functions) on the benchmark datasets. The best performance is highlighted in bold.

Datasets	Convex Formulation		Baseline					
	CVX-SQ	CVX-DH	BL-SQ	BL-DH	BL-RP	BL-LG	BL-HG	BL-SG
Musk1	**0.831** **(0.044)**	0.799 (0.055)	0.723 (0.054)	0.767 (0.061)	0.795 (0.066)	0.767 (0.065)	0.779 (0.066)	0.797 (0.040)
Musk2	**0.795** **(0.050)**	0.785 (0.049)	0.680 (0.082)	0.710 (0.099)	0.751 (0.068)	0.712 (0.106)	0.712 (0.096)	0.759 (0.070)
Elephant	0.773 (0.062)	0.773 (0.073)	0.740 (0.035)	**0.786** **(0.034)**	0.769 (0.026)	0.785 (0.032)	0.783 (0.028)	0.773 (0.041)
Fox	0.695 (0.021)	**0.701** **(0.016)**	0.672 (0.032)	0.665 (0.047)	0.697 (0.050)	0.665 (0.052)	0.679 (0.063)	0.697 (0.050)
Tiger	0.720 (0.058)	0.730 (0.058)	0.651 (0.039)	0.728 (0.076)	0.731 (0.029)	0.731 (0.071)	**0.741** **(0.068)**	0.730 (0.046)

TABLE 6: Classification accuracy of compared methods (with various loss functions) on text categorization datasets. The best performance is highlighted in bold.

Datasets	Convex Formulation		Baseline					
	CVX-SQ	CVX-DH	BL-SQ	BL-DH	BL-RP	BL-LG	BL-HG	BL-SG
Component	**0.838** **(0.023)**	0.836 (0.021)	0.468 (0.088)	0.723 (0.134)	0.747 (0.051)	0.736 (0.148)	0.745 (0.087)	0.768 (0.063)
Function	0.873 (0.022)	**0.875** **(0.025)**	0.411 (0.033)	0.742 (0.076)	0.794 (0.034)	0.771 (0.062)	0.790 (0.040)	0.801 (0.034)
Process	**0.883** **(0.018)**	**0.883** **(0.020)**	0.462 (0.055)	0.813 (0.070)	0.836 (0.046)	0.840 (0.052)	0.818 (0.059)	0.848 (0.038)

3.1. Specifically, we can first estimate θ_{S} by the proportion of the collected similar bag pairs in all the bag pairs. Since $\theta_{\mathrm{S}} = \theta^2 + (1-\theta)^2$, we have $2\theta_{\mathrm{S}} - 1 = \theta_{\mathrm{S}} - \theta_{\mathrm{D}} = (2\theta - 1)^2 \geq 0$, then we obtain $\theta = (\sqrt{2\theta_{\mathrm{S}} - 1} + 1)/2$. Since $\sqrt{\theta_{\mathrm{S}} - 1} \geq 0$, we can know $\theta \geq 0.5$. This implies that θ should be assumed to be larger than 0.5. We simply fix θ at 0.7 on all the datasets for performance evaluation. We repeat the sampling-and-training process 10 times and record mean classification accuracy with standard deviation.

Benchmark Datasets. We do experiments on five widely used MIL benchmark datasets [26], [31], including Musk1, Musk2, Elephant, Fox, and Tiger. There are 47 positive bags and 45 negative bags in Musk1, 39 positive bags and 63 negative bags in Musk2, and 100 positive bags and 100 negative bags in Elephant, Fox, and Tiger. It should be noted that these datasets are too small to evaluate the performance of MIL from similar-dissimilar-unlabeled bags. Therefore, we follow Bao et al. [54] increasing the number of bags for them. Specifically, we randomly select bags from the original datasets and duplicate them by adding Gaussian noise with mean zero and variance 0.01 to each feature. As a result, we increased the number of bags 10 times for Musk1 and Musk2, and 5 times for Elephant, Fox, and Tiger. In Table 1, we report the characteristics of the five benchmark datasets[1] after preprocessing.

Text Categorization Datasets. We also use three datasets[2] for the task of biocreative text categorization in the experiments. In this task, we need to decide whether a given ⟨protein, document⟩ pair should be annotated with some Gene Ontology (GO) code. Documents are represented by bags, and the paragraphs in the documents are represented by instances. The used features of instances are word occurrence frequencies and some statistics about the nature of the protein-GO code interaction for each paragraph. The GO consists of three hierarchical domains of standardized biological terms referring to cellular components, biological processes, and molecular functions. The hypothesis is that a document should be annotated with some GO code if it contains a paragraph that supports this annotation. Conversely, if no paragraph supports such an annotation, the document should not be annotated. In the biocreative text categorization task, we have three datasets: Component, Function, and Process. Table 2 reports the characteristics of the three datasets.

4.2 The Respective Effectiveness of Similar, Dissimilar, and Unlabeled Bags

To validate the respective effectiveness of similar, dissimilar, and unlabeled bags, we do experiments using the bag-level convex formulation and the instance-level strong baseline with the double hinge loss by learning from the following types of training bags: **CVX-SDU** and **BL-SDU** (learning from similar-dissimilar-unlabeled bags), **CVX-SD** and **BL-SD** (learning from similar-dissimilar bags), **CVX-SU** and **BL-SU** (learning from similar-unlabeled bags), **CVX-DU** and **BL-DU** (learning from dissimilar-unlabeled bags). Table 3 and Table 4 report the classification accuracy with standard deviation of the above methods on all the eight datasets using bag-level methods and instance-level methods, respectively. As shown in Table 3 and Table 4, CVX-SDU outperforms its three degenerated versions (by only keeping β_1, β_1, and β_3 in Eq. (8), respectively) CVX-SD, CVX-SU, and CVX-DU; BL-SDU outperforms its three degenerated versions (by only keeping α_1, α_2, α_3 in Eq. (7),

[1] http://www.cs.columbia.edu/~andrews/mil/datasets.html
[2] https://veronikach.com/research/data-code/

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3232141

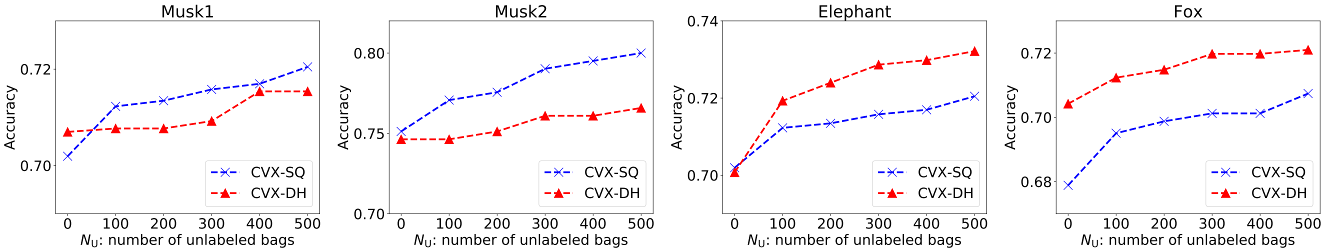

Fig. 1: Classification accuracy of convex methods when the number of unlabeled bags increases.

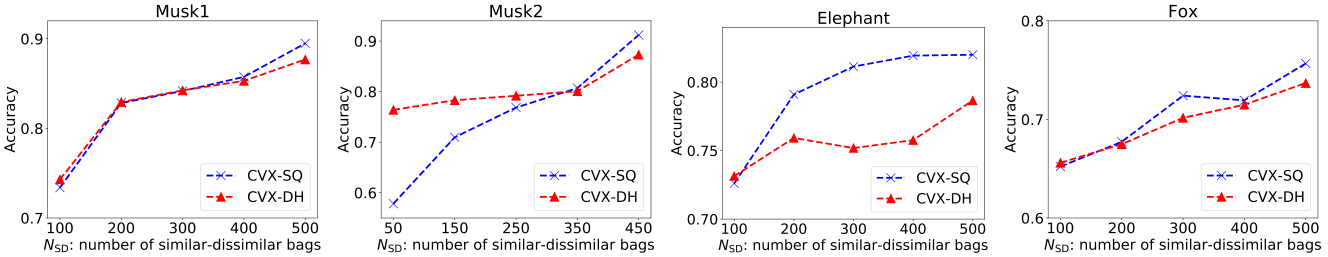

Fig. 2: Classification accuracy of convex methods when the number of similar-dissimilar bag pairs increases.

respectively) BL-SD, BL-SU, and BL-DU. From these observations, the respective effectiveness of similar, dissimilar, and unlabeled bags can be verified. Therefore, learning from similar-dissimilar-unlabeled data is advantageous.

4.3 Performance Comparison Between Convex Formulation and Strong Baseline

Table 5 and Table 6 report the classification accuracy with standard deviation of each learning method (with different loss functions) on the benchmark datasets and the text categorization datasets, respectively. As can be seen from the two tables, the instance-level baseline methods achieve decent performance, while they are generally inferior to the bag-level convex methods.

4.4 Further Analysis

Performance of Increasing Unlabeled Bags and Similar-Dissimilar Bag Pairs. We conduct experiments on Musk1, Musk2, Elephant, and Fox. To investigate the performance of increasing unlabeled bags, we set the total number of similar and dissimilar bag pairs to 50 and set the number of unlabeled bags to $\{0, 100, 200, 300, 400, 500\}$. From the results shown in Fig. 1, we can find that the performance of our proposed convex formulation becomes better when more unlabeled bags are provided. To investigate the performance of increasing similar-dissimilar bag pairs, we set the number of unlabeled bags to 100 and set the number of similar-dissimilar bag pairs to $\{100, 200, 300, 400, 500\}$. From the results shown in Fig. 2, we can observe that the performance of our proposed convex formulation becomes better when more similar-dissimilar bag pairs are provided. These observations are evidently in accordance with our derived generalization error bound in Theorem 1, because the generalization error decreases as the number of similar-dissimilar-unlabeled bags increases.

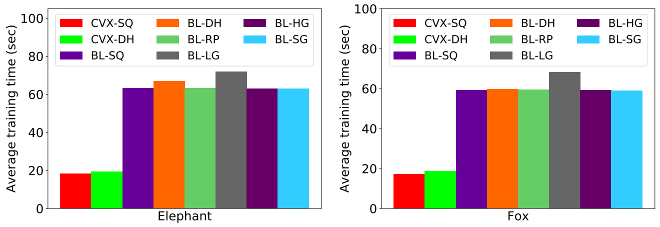

Fig. 3: Average training time of each method on the benchmark datasets Elephant and Fox.

Training Efficiency Analysis. To further demonstrate the efficiency advantage of the bag-level convex formulation over the instance-level strong baseline, we perform MIL from similar-dissimilar-unlabeled bags using various learning methods on Elephant and Fox. We show the average training time of each method in Fig. 3. As can be seen from Fig. 3, the average training time of bag-level methods is significantly smaller that of instance-level methods. Therefore, our experimental results clearly demonstrate that our proposed bag-level methods are not only more effective but also more efficient than the instance-level methods.

5 Conclusion

In this paper, we provided an extended study of our earlier research [62] on multiple-instance learning from similar and dissimilar bags. We studied an extended problem setting called multiple-instance learning from similar-dissimilar-unlabeled bags, where we aim to learn a classifier from only similar-dissimilar-unlabeled bags, instead of fully labeled bags. To the best of our knowledge, we provided the first attempt to investigate this problem. To solve this new multiple-instance learning problem, we first proposed a strong baseline that trains an instance-level classifier by employing the unlabeled-unlabeled learning strategy. Then, we proposed a convex formulation that trains a bag-level classifier based on bag-level empirical risk minimization. Comprehensive experimental results clearly demonstrated that the instance-level methods work well, while the bag-level methods are even better.

The main limitation of our proposed empirical risk minimization methods lies in that they rely on the assumed data distribution (as shown in Section 3.1). When the collected training data do not satisfy the assumed distribution, the performance of our methods would be degraded. Therefore, it is interesting to further explore effective methods that can be robust to the mismatch of the data distribution. We leave this for future work. Besides, it would be also interesting to investigate multiple-instance learning with other types of weak supervision information in future work.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China under Grant 2022YFB3103500. Lei Feng is supported by the National Natural Science Foundation of China (Grant No. 62106028), Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program, and CAAI-Huawei MindSpore Open Fund. Bo An is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

[2] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *ICML*, 2006, pp. 161–168.

[3] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.

[4] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[5] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[6] W. He and Z. Jiang, "Semi-supervised learning with the em algorithm: A comparative study between unstructured and structured prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[7] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NeurIPS*, 2018.

[8] H.-X. Wei, L. Feng, X.-Y. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *CVPR*, 2020, pp. 13 726–13 735.

[9] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, "Learning from incomplete and inaccurate supervision," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[10] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2155–2167, 2017.

[11] G. Lyu, S. Feng, T. Wang, C. Lang, and Y. Li, "Gm-pll: graph matching based partial label learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 521–535, 2019.

[12] L. Feng, J. Lv, B. Han, M. Xu, G. Niu, X. Geng, B. An, and M. Sugiyama, "Provably consistent partial-label learning," in *NeurIPS*, 2020.

[13] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, and J. Zhao, "Pico: Contrastive label disambiguation for partial label learning," in *ICLR*, 2022.

[14] R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *NeurIPS*, 2017, pp. 1675–1685.

[15] C. Gong, H. Shi, T. Liu, C. Zhang, J. Yang, and D. Tao, "Loss decomposition and centroid estimation for positive and unlabeled learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 918–932, 2021.

[16] C. Li, X. Li, L. Feng, and J. Ouyang, "Who is your right mixup partner in positive and unlabeled learning," in *ICLR*, 2022.

[17] T. Ishida, G. Niu, and M. Sugiyama, "Binary classification for positive-confidence data." in *NeurIPS*, 2018, pp. 5917–5928.

[18] K. Shinoda, H. Kaji, and M. Sugiyama, "Binary classification from positive data with skewed confidence," in *IJCAI*, 2021, pp. 3328–3334.

[19] T. Shimada, H. Bao, I. Sato, and M. Sugiyama, "Classification from pairwise similarities/dissimilarities and unlabeled data via empirical risk minimization," *Neural Computation*, 2020.

[20] H. Bao, T. Shimada, L. Xu, I. Sato, and M. Sugiyama, "Pairwise supervision can provably elicit a decision boundary," in *AISTATS*, 2022, pp. 2618–2640.

[21] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama, "On the minimal supervision for training any binary classifier from only unlabeled data," in *ICLR*, 2019.

[22] N. Lu, T. Zhang, G. Niu, and M. Sugiyama, "Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach," in *AISTATS*, 2020, pp. 1115–1125.

[23] Z. Cui, N. Charoenphakdee, I. Sato, and M. Sugiyama, "Classification from triplet comparison data," *Neural Computation*, vol. 32, no. 3, pp. 659–681, 2020.

[24] L. Feng, S. Shu, N. Lu, B. Han, M. Xu, G. Niu, B. An, and M. Sugiyama, "Pointwise binary classification with pairwise confidence comparisons," in *ICML*, 2021, pp. 3252–3262.

[25] Y. Cao, L. Feng, Y. Xu, B. An, G. Niu, and M. Sugiyama, "Larning from similarity-confidence data," in *ICML*, 2021.

[26] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[27] W.-J. Li *et al.*, "Mild: Multiple-instance learning via disambiguation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, pp. 76–89, 2009.

[28] J. R. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, pp. 1–25, 2010.

[29] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.

[30] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognition*, vol. 77, pp. 329–353, 2018.

[31] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NeurIPS*, 2002, pp. 577–584.

[32] C. Zhang and P. Viola, "Multiple-instance pruning for learning efficient cascade detectors," in *NeurIPS*, 2007, pp. 1681–1688.

[33] M. M. Dundar, G. Fung, B. Krishnapuram, and R. B. Rao, "Multiple-instance learning algorithms for computer-aided detection," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1015–1021, 2008.

[34] T. Tong, R. Wolz, Q. Gao, R. Guerrero, J. V. Hajnal, D. Rueckert, A. D. N. Initiative *et al.*, "Multiple instance learning for classification of dementia in brain mri," *Medical Image Analysis*, vol. 18, no. 5, pp. 808–818, 2014.

[35] W. Li and N. Vasconcelos, "Multiple instance learning for soft bags via top instances," in *CVPR*, 2015, pp. 4277–4285.

[36] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *NeurIPS*, 1998, pp. 570–576.

[37] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in *CVPR*, 2015, pp. 3460–3469.

[38] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009, pp. 983–990.

[39] A. Kanezaki, T. Harada, and Y. Kuniyoshi, "Scale and rotation invariant color features for weakly-supervised object learning in 3d space," in *ICCV Workshops*, 2011, pp. 617–624.

[40] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011, pp. 1307–1314.

[41] J. Wang and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach," in *ICML*, 2000, pp. 1119–1125.

[42] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *NeurIPS*, 2001, pp. 1073–1080.

[43] X. Xu and E. Frank, "Logistic regression and boosting for labeled bags of instances," in *PAKDD*, 2004, pp. 272–281.

[44] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

[45] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-iid samples," in *ICML*, 2009, pp. 1249–1256.

[46] C. Leistner, A. Saffari, and H. Bischof, "Miforests: Multiple-instance learning with randomized trees," in *ECCV*, 2010, pp. 29–42.

[47] T. Zhang and H. Jin, "Optimal margin distribution machine for multi-instance learning," in *IJCAI*, 2020, pp. 2383–2389.

[48] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[49] A. J. Nederhof, "Methods of coping with social desirability bias: A review," *European Journal of Social Psychology*, vol. 15, no. 3, pp. 263–280, 1985.

[50] H. Bao, G. Niu, and M. Sugiyama, "Classification from pairwise similarity and unlabeled data," in *ICML*, 2018, pp. 452–461.

[51] Z.-H. Zhou and M.-L. Zhang, "Solving multi-instance problems with classifier ensemble based on constructive clustering," *Knowledge and Information Systems*, vol. 11, no. 2, pp. 155–170, 2007.

[52] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable algorithms for multi-instance learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 975–987, 2016.

[53] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *ICML*, 2002, pp. 179–186.

[54] H. Bao, T. Sakai, I. Sato, and M. Sugiyama, "Convex formulation of multiple instance learning from positive and unlabeled bags," *Neural Networks*, vol. 105, pp. 132–141, 2018.

[55] M. C. du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *ICML*, 2015, pp. 1386–1394.

[56] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. 11, pp. 463–482, 2002.

[57] S. Mendelson, "Lower bounds for the empirical minimization algorithm," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3797–3803, 2008.

[58] Z. Fang, J. Lu, F. Liu, and G. Zhang, "Semi-supervised heterogeneous domain adaptation: Theory and algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[59] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[60] M. S. Andersen, J. Dahl, and L. Vandenberghe, "Cvxopt: Python software for convex optimization," *URL https://cvxopt. org*, vol. 64, 2013.

[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[62] L. Feng, S. Shu, Y. Cao, L. Tao, H. Wei, T. Xiang, B. An, and G. Niu, "Multiple-instance learning from similar and dissimilar bags," in *KDD*, 2021, pp. 374–382.

Lei Feng received his Ph.D. degree in Computer Science from Nanyang Technological University, Singapore, in 2021. He is currently a Professor at the College of Computer Science, Chongqing University, China. He was named to Forbes 30 under 30 Asia 2022 and Forbes 30 Under 30 China 2021. He received ICLR 2022 outstanding paper award honourable mention. His main research interests include weakly supervised learning, robust deep learning, and data mining. He has published over 30 papers at top conferences and journals, such as ICML, NeurIPS, ICLR, KDD, CVPR, ICCV, AAAI, and IJCAI. He has also served as a senior program committee member for IJCAI 2021 and AAAI 2022, and a program committee member (or reviewer) for other top conferences and journals.

Senlin Shu received his M.E. degree in Technology of Computer Application from Southwest University, China, in 2021. He is currently working towards a Ph.D. degree in the College of Computer Science, Chongqing University, China. His main research interests include weakly supervised learning and data mining.

Yuzhou Cao received his bachelor's degree in mathematics and applied mathematics from the College of Science, China Agricultural University, Beijing, China. He is currently pursing his Ph.D. degree at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include machine learning, optimization, and data mining.

Lue Tao received his BSc and MSc degrees in Computer Science from Nanjing University of Aeronautics and Astronautics in 2019 and 2022. Currently, he is pursing the PhD degree with the National Key Lab for Novel Software Technology in Nanjing University. His research interest is mainly on machine learning and data mining.

Hongxin Wei received his B.E. degree in Software Engineering from Huazhong University of Science and Technology, China, in 2016. He is currently pursing his Ph.D. degree at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His main research interests include weakly supervised learning, adversarial robustness, OOD detection, and data mining. He has also served as program committee member (or reviewer) for NeurIPS, ICLR, KDD, CVPR, and ICML.

Tao Xiang received the BEng, MS and PhD degrees in computer science from Chongqing University, China, in 2003, 2005, and 2008, respectively. He is currently a Professor of the College of Computer Science at Chongqing University. Prof. Xiang's research interests include multimedia security, cloud security, data privacy and cryptography. He has published over 100 papers on international journals and conferences. He also served as a referee for numerous international journals and conferences.

Bo An is a President's Council Chair Professor in Computer Science and Engineering and Co-Director of Artificial Intelligence Research Institute (AI.R) at Nanyang Technological University, Singapore. He received the Ph.D degree in Computer Science from the University of Massachusetts, Amherst, in 2010. His current research interests include artificial intelligence, multiagent systems, computational game theory, reinforcement learning, and optimization. He has published over 100 referred papers at AAMAS, IJCAI, AAAI, ICAPS, KDD, UAI, EC, WWW, ICLR, NeurIPS, ICML, JAAMAS, AIJ and ACM/IEEE Transactions. Dr. An was the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, an Operational Excellence Award from the Commander, First Coast Guard District of the United States, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and 2018 Nanyang Research Award (Young Investigator). His publications won the Best Innovative Application paper Award at AAMAS'12 and the Innovative Application Award at IAAI'16. He was invited to give Early Career Spotlight talk at IJCAI'17. He led the team HogRider which won the 2017 Microsoft Collaborative AI Challenge. He was named to IEEE Intelligent Systems' "AI's 10 to Watch" list for 2018. He is PC Co-Chair of AAMAS'20. He is a member of the editorial board of JAIR and the Associate Editor of JAAMAS, IEEE Intelligent Systems, and ACM TIST. He was elected to the board of directors of IFAAMAS, senior member of AAAI, and Distinguished member of ACM.

Gang Niu is currently an indefinite-term research scientist at RIKEN Center for Advanced Intelligence Project. He received the PhD degree in computer science from Tokyo Institute of Technology in 2013. Before joining RIKEN as a research scientist, he was a senior software engineer at Baidu and then an assistant professor at the University of Tokyo. He has published more than 90 journal articles and conference papers, including 31 ICML, 17 NeurIPS (1 oral and 3 spotlights), and 11 ICLR (1 outstanding paper honorable mention, 2 orals, and 1 spotlight) papers. He has co-authored the book "Machine Learning from Weak Supervision: An Empirical Risk Minimization Approach" (the MIT Press). On the other hand, he has served as an area chair 18 times, including ICLR 2021–2022, ICML 2019–2022, and NeurIPS 2019–2022. He also serves/has served as an action editor of TMLR and a guest editor of a special issue at MLJ. Moreover, he has served as a publication chair for ICML 2022, and has co-organized 9 workshops, 1 competition, and 2 tutorials.