Multiple-Instance Learning from Similar and Dissimilar Bags

Lei Feng[1], Senlin Shu[2], Yuzhou Cao[3], Lue Tao[4], Hongxin Wei[5], Tao Xiang[1*], Bo An[5], Gang Niu[6]
[1]College of Computer Science, Chongqing University, China
[2]College of Computer and Information Science, Southwest University, China
[3]College of Science, China Agricultural University, China
[4]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
[5]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[6]RIKEN Center for Advanced Intelligence Project, Japan
{lfeng,txiang}@cqu.edu.cn,boan@ntu.edu.sg,gang.niu@riken.jp

ABSTRACT

Multiple-instance learning (MIL) is an important weakly supervised binary classification problem, where training instances are arranged in bags, and each bag is assigned a positive or negative label. Most of the previous studies for MIL assume that training bags are fully labeled. However, in some real-world scenarios, it could be difficult to collect fully labeled bags, due to the expensive time and labor consumption of the labeling task. Fortunately, it could be much easier for us to collect similar and dissimilar bags (indicating whether two bags share the same label or not), because we do not need to figure out the underlying label of each bag in this case. Therefore, in this paper, we for the first time investigate MIL from *only similar and dissimilar bags*. To solve this new MIL problem, we propose a convex formulation to train a *bag-level* classifier based on empirical risk minimization and theoretically derive a generalization error bound. In addition, we also propose a strong baseline for this new MIL problem, which aims to train an *instance-level* classifier by minimizing the instance-level empirical risk. Extensive experimental results clearly demonstrate that our proposed baseline works well, while our proposed convex formulation is even better.

CCS CONCEPTS

• **Computing methodologies → Learning settings**; **Machine learning algorithms**.

KEYWORDS

Multi-Instance Learning, Similar-Dissimilar Classification, Empirical Risk Minimization

ACM Reference Format:
Lei Feng, Senlin Shu, Yuzhou Cao, Lue Tao, Hongxin Wei, Tao Xiang, Bo An, Gang Niu. 2021. Multiple-Instance Learning from Similar and Dissimilar Bags. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467318

1 INTRODUCTION

Weakly supervised learning [52] covers a variety of studies [9, 17, 22, 28, 29, 32, 35, 39, 43, 54] that attempt to construct predictive models by learning with 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.

Multiple-instance learning (MIL) [1, 10, 13, 19] is an important weakly supervised learning problem, which aims to deal with the binary classification task where training instances are arranged in bags, and each bag is assigned a binary label (indicating whether the bag is positive or not). A training bag is labeled as a positive bag if at least one instance in the bag is positive, and a training bag is labeled as a negative bag if all the instances in the bag are negative. The goal of MIL is to predict the label of any unseen test bag. MIL is more difficult than ordinary binary classification because the labels of the instances in each bag are unavailable. MIL has been successfully applied to various real-world problems such as drug activity prediction [13], image retrieval [27, 33, 46], visual tracking [4], object detection [24, 37], text categorization [3], face detection [48], and medical diagnosis [16, 42].

In the past decades, a large number of methods have been developed to improve the performance of MIL, such as: citation kNN [44], EM-DD [49], MI-SVM [3], MIBoosting [47], MILES [11], miGraph [53], MIForests [26], and MI-ODM [50]. Although these methods have achieved satisfactory performance, all of them are demanding for fully labeled bags, in order to train an effective bag-level classifier. However, it may be difficult for us to collect a MIL dataset composed of fully labeled bags in some situations, due to the significant labeling costs. For example, a molecule (bag) can have many low-energy shapes (instances), a bag label that indicates whether the molecule can be used to make the drug depends on whether the molecule has some special shapes. It could be difficult for human experts to accurately figure out all the correct bag labels of all the molecules, due to high (time or money) costs. Fortunately, it would be much easier to judge whether two molecules share the same bag label, instead of knowing the correct bag label of each molecule. In this case, we refer to two bags that share the same bag label as a *similar* bag and two bags that do not share the same bag label as a *dissimilar* bag. Therefore, a natural question arises: *Can we still successfully learn an effective bag-level binary classifier from only similar and dissimilar bags when there are no labeled bags provided?*

[*] Corresponding author: Tao Xiang <txiang@cqu.edu.cn>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '21, August 14–18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467318

In this paper, we provide an affirmative answer to the above question. Our main contributions can be summarized as follows:

- We for the first time investigate MIL from *only similar and dissimilar bags*. To solve this new MIL problem, we propose a convex formulation to train a *bag-level* classifier based on empirical risk minimization and theoretically derive a generalization error bound.
- We also propose a strong baseline for this new MIL problem, which aims to train an *instance-level* classifier by minimizing the instance-level empirical risk.
- Extensive experimental results clearly demonstrate that our proposed baseline works well, while our proposed convex formulation is even better.

2 RELATED STUDIES AND PRELIMINARIES

In this section, we briefly review related studies and introduce preliminary knowledge.

In fully supervised binary classification, we normally require a vast amount of fully labeled data to train an effective binary classifier. However, it could be difficult to collect such fully labeled data, due to high labeling costs [54], privacy considerations [45], and social bias [36]. Therefore, many researchers have paid much attention to various weakly supervised binary classification problems [5, 12, 14, 15, 18, 21, 23, 25, 30, 31, 40, 41]. Because our goal in this paper is to learn a bag-level binary classifier (i.e., multiple-instance learning) from similar and dissimilar bags, two of the existing weakly supervised binary classification problems are highly related to our work, i.e., *multiple-instance learning* [10, 51] and *similar-dissimilar classification* [40], where similar-dissimilar classification aims to learn an instance-level binary classifier from only similar and dissimilar data.

In what follows, we will introduce ordinary binary classification, similar-dissimilar classification, and multi-instance learning.

2.1 Ordinary Binary Classification

In ordinary binary classification, let the feature space be $\mathcal{X} \in \mathbb{R}^d$ (with d dimensions) and the label space be $\mathcal{Y} = \{-1, +1\}$. Let us clearly define that $\boldsymbol{x}$ denotes an instance and and $(\boldsymbol{x}, y)$ denotes an example including an instance $\boldsymbol{x}$ and a label y (assigned to the instance $\boldsymbol{x}$). It is conventionally assumed that each training example $(\boldsymbol{x}, y)$ is independently sampled from an unknown data distribution with probability density $p(\boldsymbol{x}, y)$. The goal of ordinary binary classification is to construct an instance-level binary classifier f by minimizing the (expected) classification risk, which is defined as

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

where $\mathbb{E}_{p(\boldsymbol{x},y)}[\cdot]$ denotes the expected value over the joint probability density $p(\boldsymbol{x}, y)$ and $\ell : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}_+$ denotes a binary loss function. Because the joint probability density $p(\boldsymbol{x}, y)$ is unknown and we usually have training examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ that are independently drawn from $p(\boldsymbol{x}, y)$, a common strategy is to minimize the empirical risk $\widehat{R}(f) := \frac{1}{n}\sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$, which is called *empirical risk minimization.* As can be easily verified, $\mathbb{E}_{p(\boldsymbol{x},y)}[\widehat{R}(f)] = R(f)$. In this case, we refer to $\widehat{R}(f)$ as an unbiased estimator of the classification risk $R(f)$ (also known as *unbiased risk estimator*).

2.2 Similar-Dissimilar Classification

Recently, an interesting weakly supervised binary classification problem called similar-dissimilar classification [40] has been investigated, which aims to train an instance-level binary classifier from only similar and dissimilar data pairs that indicate whether two instances belong to the same class (similar) or not (dissimilar). Obviously, compared with fully labeled data, similar and dissimilar data could be easier to collect [5, 7, 40]. Here, we introduce the seminal work [40], which formally defines the generation process of similar and dissimilar data pairs and derive an unbiased risk estimator based on the data generation process.

Specifically, the densities of similar data and dissimilar data are formulated as

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

where $\pi = p(y = +1)$ denote the (positive) class prior, $\pi_{\mathrm{S}} = \pi^2 + (1-\pi)^2$ denotes the prior of similar data pairs, $\pi_{\mathrm{D}} = 2\pi(1-\pi)$ denotes the prior of dissimilar data pairs (hence $\pi_{\mathrm{S}} + \pi_{\mathrm{D}} = 1$), $p_+(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = +1)$ and $p_-(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = -1)$ denote the probability densities of positive and negative data respectively. Given the above data generation process of similar-dissimilar classification, Shimada et al. [40] showed that the following proposition holds.

Proposition 1 (Theorem 2 in Shimada et al. [40]). *The classification risk $R(f)$ in Eq. (1) can be equivalently represented as*

$$R(f) = R_{\mathrm{SD}}(f) = \pi_{\mathrm{S}}\mathbb{E}_{p_{\mathrm{S}}(\boldsymbol{x},\boldsymbol{x}')}\Big[\frac{\mathcal{L}(f(\boldsymbol{x}), +1) + \mathcal{L}(f(\boldsymbol{x}'), +1)}{2}\Big]$$
$$+ \pi_{\mathrm{D}}\mathbb{E}_{p_{\mathrm{D}}(\boldsymbol{x},\boldsymbol{x}')}\Big[\frac{\mathcal{L}(f(\boldsymbol{x}), -1) + \mathcal{L}(f(\boldsymbol{x}'), -1)}{2}\Big],$$

where $\mathcal{L}(f(\boldsymbol{x}), t)$ ($t \in \{+1, -1\}$) is a composite loss function defined as

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

As this proposition indicates, we can exactly recover the classification risk $R(f)$ using only similar data independently sampled from $p_{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}')$ and dissimilar data independently sampled from $p_{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{x}')$. This implies that we can learn an instance-level binary classifier from given similar and dissimilar data, by minimizing the empirical approximation of $R_{\mathrm{SD}}(f)$ since it is an unbiased risk estimator of $R(f)$.

2.3 Multiple-Instance Learning

In MIL, suppose the learner is given a training set with n bags, i.e., $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ where $X_i = \{\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ij}, \ldots, \boldsymbol{x}_{ib_i}\}$ is a bag with $\boldsymbol{x}_{ij} \in \mathcal{X}$ representing the j-th instance in the i-th bag and b_i denotes the number of instances in the bag X_i. If there exits at least one positive instance in X_i, then X_i is a positive bag (i.e., $Y_i = +1$), otherwise X_i is a negative bag (i.e., $Y_i = -1$). It is worth noting that only bag labels are available, while the specific labels of the instances in the bag are unknown. The goal of multi-instance learning is to learn a bag-level binary classifier, so that the label of any test bag could be correctly predicted.

For learning a bag-level binary classifier, the key issue is how to design a function that takes a bag (a set of instances) as the input and outputs a real value. In this work, we focus on constructing a bag-level linear-in-parameter classifier with a specially designed kernel for multi-instance learning. Specifically, the bag-level linear-in-parameter classifier[1] is formulated as follows:

$$g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X), \tag{2}$$

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

$$\boldsymbol{\phi}(X) = \begin{bmatrix} \widetilde{\mathcal{K}}(X, X_1) \\ \vdots \\ \widetilde{\mathcal{K}}(X, X_n) \end{bmatrix}. \tag{3}$$

Here, the problem becomes how to properly design a special kernel $\widetilde{\mathcal{K}}$ for bags in MIL. Gärtner et al. [20] proposed multiple-instance kernels, which map 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 (e.g., Gaussian kernel or polynomial kernel) and $\boldsymbol{s}(X)$ is a statistic with respect to the bag X. For example, the minimax statistic is a common choice:

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

where $x^{(i)}$ denotes the i-th element of the instance $\boldsymbol{x}$ in the bag X. As shown in the experimental results from Gärtner et al. [20], the statistical kernel $\widetilde{\mathcal{K}}$ associated with the minimax statistic $\boldsymbol{s}_{\text{minimax}}(X)$ and the polynomial kernel $\mathcal{K}$ achieves satisfactory performance:

$$\widetilde{\mathcal{K}}_{\text{minimax}}(X, X') := \left(\boldsymbol{s}_{\text{minimax}}(X)^\top \boldsymbol{s}_{\text{minimax}}(X') + 1\right)^p, \tag{4}$$

where p denotes the degree of the polynomial kernel.

In summary, we aim to learn a bag-level classifier $g(X)$ (defined by Eqs. (2), (3), and (4)) for MIL from similar and dissimilar bags.

3 MULTIPLE-INSTANCE LEARNING FROM SIMILAR AND DISSIMILAR BAGS

In this section, we propose a convex formulation for MIL from similar and dissimilar bags. We first define the generation process of similar and dissimilar bags and then derive an empirical risk estimator based on the data generation process.

3.1 Generation Process of Similar and Dissimilar Bags

Following Shimada et al. [40], we adopt an analogous generation process of similar and dissimilar bags. Let us denote the collected training set comprised of similar and dissimilar bags as $\mathcal{D}_{\text{SD}} = \{(X_i, X'_i, Z_i)\}_{i=1}^{N_{\text{SD}}}$ where $Z_i = +1$ if $Y_i = Y'_i$, otherwise $Z_i = -1$. Here, N_{SD} bag pairs in $\mathcal{D}_{\text{SD}}$ can be decomposed into N_{S} similar bag pairs and N_{D} dissimilar bag pairs:

$$\mathcal{D}_{\text{S}} := \{(X_{\text{S}.i}, X'_{\text{S}.i})\}_{i=1}^{N_{\text{S}}} = \{(X, X') \mid (X, X', Z = +1) \in \mathcal{D}_{\text{SD}}\},$$
$$\mathcal{D}_{\text{D}} := \{(X_{\text{D}.j}, X'_{\text{D}.j})\}_{j=1}^{N_{\text{D}}} = \{(X, X') \mid (X, X', Z = -1) \in \mathcal{D}_{\text{SD}}\}.$$

Then, we can consider the generation process of similar and dissimilar pairs as: $\mathcal{D}_{\text{S}} \overset{\text{i.i.d.}}{\sim} p_{\text{S}}(X, X')$ and $\mathcal{D}_{\text{D}} \overset{\text{i.i.d.}}{\sim} p_{\text{D}}(X, X')$. For convenience, we introduce the following notations representing the similar and dissimilar priors and conditional densities:

$$\theta_{\text{S}} := p(Y = Y'), \quad p_{\text{S}}(X, X') := p(X, X' \mid Y = Y'),$$
$$\theta_{\text{D}} := p(Y \neq Y'), \quad p_{\text{D}}(X, X') := p(X, X' \mid Y \neq Y').$$

It is noteworthy that we assume each data point in a data pair is independently generated. By further denoting the bag-level (positive) class priors as $p(Y = 1) = \theta$, we have

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

Given the above generation process of similar and dissimilar bags, we can derive a convex formulation.

3.2 Convex Formulation for Multiple-Instance Learning from Similar and Dissimilar Bags

When collected similar and dissimilar bags satisfy the above data generation process, motivated by Proposition 1, we propose to train a bag-level classifier by minimizing the following empirical risk:

$$\begin{aligned}\widehat{R}_{\text{SDbag}}(g) = {} & \frac{\theta_{\text{S}}}{2N_{\text{S}}} \sum\nolimits_{i=1}^{N_{\text{S}}} \left(\mathcal{L}(g(X_i), +1) + \mathcal{L}(g(X'_i), +1)\right) \\ & + \frac{\theta_{\text{D}}}{2N_{\text{D}}} \sum\nolimits_{j=1}^{N_{\text{D}}} \left(\mathcal{L}(g(X_j), -1) + \mathcal{L}(g(X'_j), -1)\right),\end{aligned} \tag{5}$$

It is worth noting that Eq. (5) may not be convex even if a convex loss function ℓ (e.g., the hinge loss) is used. Fortunately, as verified by previous studies [5, 40], if the used binary loss function ℓ in the composition loss function $\mathcal{L}$ satisfies the condition: $\ell(g(X), +1) - \ell(g(X), -1) = -g(X)$, then the objective function Eq. (5) is convex. When we choose ℓ that satisfies the above condition, $\mathcal{L}(g(X), t)$ can be equivalently expressed as

$$\mathcal{L}(g(X), t) = \ell(g(X), t) - \frac{1-\theta}{2\theta - 1} t \cdot g(X). \tag{6}$$

It is also worth noting that in Eq. (5), the two bags X and X' in the same similar or dissimilar 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 Eq. (6) into Eq. (5), we can rewrite Eq. (5) as

$$\begin{aligned}\widehat{R}_{\text{SDbag}}(g) = {} & \frac{\theta_{\text{S}}}{2N_{\text{S}}} \sum\nolimits_{i=1}^{2N_{\text{S}}} \left(\ell(g(X_{\text{S}.i}), +1) - \frac{1-\theta}{2\theta-1} g(X_{\text{S}.i})\right) \\ & + \frac{\theta_{\text{D}}}{2N_{\text{D}}} \sum\nolimits_{j=1}^{2N_{\text{D}}} \left(\ell(g(X_{\text{D}.j}), -1) + \frac{1-\theta}{2\theta-1} g(X_{\text{D}.j})\right).\end{aligned} \tag{7}$$

Here, because only similar and dissimilar bags are available, the vector of basis function $\boldsymbol{\phi}$ (from $g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X)$) becomes the

[1] It is worth noting that for the formulation $g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X)$, if we set $\widetilde{\boldsymbol{w}} := [\boldsymbol{w}\ b]^\top$ and $\widetilde{\boldsymbol{\phi}}(X) := [\boldsymbol{\phi}(X)\ 1]^\top$, then we can recover $g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X) + b$.

following form:

$$\boldsymbol{\phi}(X) = \begin{bmatrix} \widetilde{\mathcal{K}}_{\text{minimax}}(X, X_{\text{S},1}) \\ \vdots \\ \widetilde{\mathcal{K}}_{\text{minimax}}(X, X_{\text{S},2N_\text{S}}) \\ \widetilde{\mathcal{K}}_{\text{minimax}}(X, X_{\text{D},1}) \\ \vdots \\ \widetilde{\mathcal{K}}_{\text{minimax}}(X, X_{\text{D},2N_\text{D}}) \end{bmatrix}, \tag{8}$$

where $\widetilde{\mathcal{K}}_{\text{minimax}}$ is defined in Eq. (4).

Now we need to consider a convex loss ℓ in Eq. (7) that satisfies the condition $\ell(g(X), +1) - \ell(g(X), -1) = -g(X)$ for practical implementation. In this paper, we consider the squared loss and the double hinge loss [15].

3.3 Practical Implementation

Let us introduce the following symbols for notational convenience:

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

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

Then, we can insert the squared loss and the double hinge loss into Eq. (7) for practical implementation. Given the empirical risk in Eq. (7), we adopt the widely used L_2 regularization to restore stability and ensure generalization. Next, we present the technical details when we use the squared loss and the double hinge loss.

3.3.1 *Squared Loss.*

We use the squared loss defined as $\ell_{\text{SQ}}(z, t) = \frac{1}{4}(tz - 1)^2$. By inserting it into Eq. (7), we have the following objective function:

$$\begin{aligned}
&J_{\text{SQ}}(\boldsymbol{w}) \\
&= \frac{\theta_\text{S}}{2N_\text{S}} \sum_{i=1}^{2N_\text{S}} \Big(\frac{1}{4}(\boldsymbol{x}_{\text{S},i}^\top \boldsymbol{w} - 1)^2 - \frac{1-\theta}{2\theta-1} \boldsymbol{x}_{\text{S},i}^\top \boldsymbol{w}\Big) \\
&+ \frac{\theta_\text{D}}{2N_\text{D}} \sum_{j=1}^{2N_\text{D}} \Big(\frac{1}{4}(-\boldsymbol{x}_{\text{D},j}^\top \boldsymbol{w} - 1)^2 + \frac{1-\theta}{2\theta-1} \boldsymbol{x}_{\text{D},j}^\top \boldsymbol{w}\Big) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 \\
&= \frac{\theta_\text{S}}{2N_\text{S}} \sum_{i=1}^{2N_\text{S}} \Big(\frac{1}{4}\boldsymbol{w}^\top \boldsymbol{x}_{\text{S},i} \boldsymbol{x}_{\text{S},i}^\top \boldsymbol{w} + \frac{1}{4} - (\frac{1}{2} + \frac{1-\theta}{2\theta-1}) \boldsymbol{x}_{\text{S},i}^\top \boldsymbol{w}\Big) \\
&+ \frac{\theta_\text{D}}{2N_\text{D}} \sum_{j=1}^{2N_\text{D}} \Big(\frac{1}{4}\boldsymbol{w}^\top \boldsymbol{x}_{\text{D},j} \boldsymbol{x}_{\text{D},j}^\top \boldsymbol{w} + \frac{1}{4} + (\frac{1}{2} + \frac{1-\theta}{2\theta-1}) \boldsymbol{x}_{\text{D},j}^\top \boldsymbol{w}\Big) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 \\
&= \boldsymbol{w}^\top \Big(\frac{\theta_\text{S}}{8N_\text{S}} \boldsymbol{X}_\text{S}^\top \boldsymbol{X}_\text{S} + \frac{\theta_\text{D}}{8N_\text{D}} \boldsymbol{X}_\text{D}^\top \boldsymbol{X}_\text{D} + \frac{\lambda}{2} \boldsymbol{I}_{d \times d}\Big) \boldsymbol{w} + \frac{\theta_\text{S} + \theta_\text{D}}{4} \\
&\qquad + (\frac{1}{2} + \frac{1-\theta}{2\theta-1}) \Big(- \frac{\theta_\text{S}}{2N_\text{S}} \boldsymbol{1}_{N_\text{S}}^\top \boldsymbol{X}_\text{S} + \frac{\theta_\text{D}}{2N_\text{D}} \boldsymbol{1}_{N_\text{D}}^\top \boldsymbol{X}_\text{D}\Big) \boldsymbol{w},
\end{aligned}$$

where $\boldsymbol{I}_{d \times d}$ denotes the $d \times d$ identity matrix, d denotes $2N_S + 2N_D$ (i.e., $d = 2N_S + 2N_D$), and $\boldsymbol{x}_{\text{S},i}$ denotes $\boldsymbol{\phi}(X_{\text{S},i})$. By taking the derivative with respect to $\boldsymbol{w}$ and set to zero, we obtain an analytical solution:

$$\begin{aligned}
\boldsymbol{w} = (\frac{1}{2} + \frac{1-\theta}{2\theta-1}) \Big(\frac{\theta_\text{S}}{4N_\text{S}} \boldsymbol{X}_\text{S}^\top \boldsymbol{X}_\text{S} \qquad\qquad (9) \\
+ \frac{\theta_\text{D}}{4N_\text{D}} \boldsymbol{X}_\text{D}^\top \boldsymbol{X}_\text{D} + \lambda \boldsymbol{I}_{d \times d} \Big)^{-1} \Big(\frac{\theta_\text{S}}{2N_\text{S}} \boldsymbol{X}_\text{S}^\top \boldsymbol{1}_{N_\text{S}} - \frac{\theta_\text{D}}{2N_\text{D}} \boldsymbol{X}_\text{D}^\top \boldsymbol{1}_{N_\text{D}} \Big),
\end{aligned}$$

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

3.3.2 *Double-Hinge Loss.*

We use the double-hinge loss [15] defined as $\ell_{\text{DH}}(z, t) = \max(-tz, \max(0, \frac{1}{2} - \frac{1}{2}tz))$. By inserting it into Eq. (7), we have the following objective function:

$$\begin{aligned}
J_{\text{DH}}(\boldsymbol{w}) = \frac{\theta_\text{S}}{2N_\text{S}} \Big(\boldsymbol{1}_{2N_\text{S}}^\top \boldsymbol{\xi} - \frac{1-\theta}{2\theta-1} \boldsymbol{1}_{2N_\text{S}}^\top \boldsymbol{X}_\text{S} \boldsymbol{w}\Big) \\
+ \frac{\theta_\text{D}}{2N_\text{D}} \Big(\boldsymbol{1}_{2N_\text{D}}^\top \boldsymbol{\eta} + \frac{1-\theta}{2\theta-1} \boldsymbol{1}_{2N_\text{D}}^\top \boldsymbol{X}_\text{D} \boldsymbol{w}\Big) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 \\
\text{s.t.} \quad \boldsymbol{\xi} \geq \boldsymbol{0}_{2N_\text{S}}, \quad \boldsymbol{\xi} \geq \frac{1}{2}(\boldsymbol{1}_{2N_\text{S}} - \boldsymbol{X}_\text{S} \boldsymbol{w}), \quad \boldsymbol{\xi} \geq -\boldsymbol{X}_\text{S} \boldsymbol{w}, \\
\boldsymbol{\eta} \geq \boldsymbol{0}_{2N_\text{D}}, \quad \boldsymbol{\eta} \geq \frac{1}{2}(\boldsymbol{1}_{2N_\text{D}} + \boldsymbol{X}_\text{D} \boldsymbol{w}), \quad \boldsymbol{\eta} \geq \boldsymbol{X}_\text{D} \boldsymbol{w},
\end{aligned}$$

where $\geq$ for vectors denotes the element-wise inequality.

Below, we rewrite the optimization problem into the standard quadratic programming form. Let $\boldsymbol{\gamma} = [\boldsymbol{w}^\top \boldsymbol{\xi}^\top \boldsymbol{\eta}^\top]^\top \in \mathbb{R}^{d+2N_\text{S}+2N_\text{D}}$ be a objective variable and we introduce the following notations:

$$\boldsymbol{P} = \begin{bmatrix} \lambda \boldsymbol{I}_{d \times d} & \boldsymbol{0}_{d \times 2N_\text{S}} & \boldsymbol{0}_{d \times 2N_\text{D}} \\ \boldsymbol{0}_{2N_\text{S} \times d} & \boldsymbol{0}_{2N_\text{S} \times 2N_\text{S}} & \boldsymbol{0}_{2N_\text{S} \times 2N_\text{D}} \\ \boldsymbol{0}_{2N_\text{D} \times d} & \boldsymbol{0}_{2N_\text{D} \times 2N_\text{S}} & \boldsymbol{0}_{2N_\text{D} \times 2N_\text{D}} \end{bmatrix},$$

$$\boldsymbol{q} = \begin{bmatrix} \frac{1-\theta}{2\theta-1}\big(- \frac{\theta_\text{S}}{2N_\text{S}} \boldsymbol{X}_\text{S}^\top \boldsymbol{1}_{2N_\text{S}} + \frac{\theta_\text{D}}{2N_\text{D}} \boldsymbol{X}_\text{D}^\top \boldsymbol{1}_{2N_\text{D}}\big) \\ \frac{\theta_\text{S}}{2N_\text{S}} \boldsymbol{1}_{2N_\text{S}} \\ \frac{\theta_\text{D}}{2N_\text{D}} \boldsymbol{1}_{2N_\text{D}} \end{bmatrix},$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{0}_{2N_\text{S} \times d} & -\boldsymbol{I}_{2N_\text{S} \times 2N_\text{S}} & \boldsymbol{0}_{2N_\text{S} \times 2N_\text{D}} \\ -\frac{1}{2}\boldsymbol{X}_\text{S} & -\boldsymbol{I}_{2N_\text{S} \times 2N_\text{S}} & \boldsymbol{0}_{2N_\text{S} \times 2N_\text{D}} \\ -\boldsymbol{X}_\text{S} & -\boldsymbol{I}_{2N_\text{S} \times 2N_\text{S}} & \boldsymbol{0}_{2N_\text{S} \times 2N_\text{D}} \\ \boldsymbol{0}_{2N_\text{D} \times d} & \boldsymbol{0}_{2N_\text{D} \times 2N_\text{S}} & -\boldsymbol{I}_{2N_\text{D} \times 2N_\text{D}} \\ \frac{1}{2}\boldsymbol{X}_\text{D} & \boldsymbol{0}_{2N_\text{D} \times 2N_\text{S}} & -\boldsymbol{I}_{2N_\text{D} \times 2N_\text{S}} \\ \boldsymbol{X}_\text{D} & \boldsymbol{0}_{2N_\text{D} \times 2N_\text{S}} & -\boldsymbol{I}_{2N_\text{D} \times 2N_\text{D}} \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{0}_{2N_\text{S}} \\ -\frac{1}{2}\boldsymbol{1}_{2N_\text{S}} \\ \boldsymbol{0}_{2N_\text{S}} \\ \boldsymbol{0}_{2N_\text{D}} \\ -\frac{1}{2}\boldsymbol{1}_{2N_\text{D}} \\ \boldsymbol{0}_{2N_\text{D}} \end{bmatrix}.$$

Then, the optimization objective becomes:

$$\min_{\boldsymbol{\gamma}} \frac{1}{2} \boldsymbol{\gamma}^\top \boldsymbol{P} \boldsymbol{\gamma} + \boldsymbol{q}^\top \boldsymbol{\gamma} \quad \text{s.t.} \quad \boldsymbol{G}\boldsymbol{\gamma} \leq \boldsymbol{h}, \tag{10}$$

which is the standard quadratic programming problem, which can be solved by any off-the-shelf quadratic programming tools.

3.4 Analysis of Generalization Error Bound

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

$$\mathcal{G} := \{g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X) \mid \|\boldsymbol{w}\| \leq C_{\boldsymbol{w}}, \sup_{X \in \mathcal{X}} \|\boldsymbol{\phi}(X)\| \leq C_{\boldsymbol{\phi}}\}$$

be a given function class, where $\boldsymbol{\phi}$ is a vector of basis functions defined in Eq. (8). Throughout this section, we simply adopt the double hinge loss as the used loss function ℓ for the analysis because it is 1-Lipschitz, and this loss function is also used in our experiments. In contrast to the empirical risk $\widehat{R}_{\text{SDbag}}(g)$ in Eq. (5), we denote the expected risk of a bag-level classifier g (in terms of similar and dissimilar bags) as

$$\begin{aligned}
R_{\text{SDbag}}(g) = \theta_\text{S} \mathbb{E}_{p_\text{S}(X,X')} \Big[\frac{\mathcal{L}(g(X, +1) + \mathcal{L}(g(X'), +1)}{2} \Big] \\
+ \theta_\text{D} \mathbb{E}_{p_\text{D}(X,X')} \Big[\frac{\mathcal{L}(g(X), -1) + \mathcal{L}(g(X'), -1)}{2} \Big]. \quad (11)
\end{aligned}$$

Then, we analyze the generalization error bound based on the widely used *Rademacher complexity* [8].

Definition 1. *Let n be a positive integer, $X_1, \dots, X_n$ be independent and identically distributed random variables drawn from a probability distribution with density μ, $\mathcal{G} = g : \mathcal{X} \mapsto \mathbb{R}$ be a class of measurable functions, and $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$ be Rademacher variables that take value only from $\{+1, -1\}$ with even probabilities. Then, the (expected) Rademacher complexity of $\mathcal{G}$ is defined as*

$$\mathfrak{R}_n(\mathcal{G}) := \mathbb{E}_{X_1,\dots,X_n \overset{\text{i.i.d.}}{\sim} \mu} \mathbb{E}_{\boldsymbol{\sigma}} \Big[\sup_{g \in \mathcal{G}} \frac{1}{n} \sum\nolimits_{i=1}^{n} \sigma_i h(X_i) \Big].$$

For the function class $\mathcal{G}$ and any probability density μ, $\mathfrak{R}_n(\mathcal{G})$ can be normally bounded by $\mathfrak{R}_n(\mathcal{G}) \leq C_{\mathcal{G}}/\sqrt{n}$, where $C_{\mathcal{G}}$ is a positive constant. This condition holds for many model classes including the used model class $\mathcal{G} = \{g(X) = \boldsymbol{w}^\top \boldsymbol{\phi}(X)\}$.

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

$$\sup_{g \in \mathcal{G}} \Big| R_{\text{SDbag}}(g) - \widehat{R}_{\text{SDbag}}(g) \Big| \leq \frac{\theta}{2\theta{-}1} \frac{C_{\mathcal{G}}}{\sqrt{2N_{\text{S}}}} + \frac{\theta_{\text{S}} C_{\boldsymbol{w}} C_{\boldsymbol{\phi}}}{2\theta{-}1} \sqrt{\frac{\log \frac{4}{\delta}}{4N_{\text{S}}}}$$
$$+ \frac{\theta}{2\theta{-}1} \frac{C_{\mathcal{G}}}{\sqrt{2N_{\text{D}}}} + \frac{\theta_{\text{D}} C_{\boldsymbol{w}} C_{\boldsymbol{\phi}}}{2\theta{-}1} \sqrt{\frac{\log \frac{4}{\delta}}{4N_{\text{D}}}}.$$

This theorem shows that the generalization error decreases with order $1/\sqrt{N_{\text{S}}}$ and $1/\sqrt{N_{\text{D}}}$. Therefore, it is clear that increasing the number of similar and dissimilar bags can decrease the generalization error. It is also noteworthy that this order is the optimal parametric rate for empirical risk minimization without additional assumptions [34].

4 A STRONG BASELINE: LEARNING AN INSTANCE-LEVEL BINARY CLASSIFIER

The proposed convex formulation in the previous section is a bag-level learning method, which is able to directly classify test bags, instead of aggregating instance-level classification results. In this section, we propose a strong baseline that trains an instance-level binary classifier for MIL from similar and dissimilar bags.

Our motivation stems from *unlabeled-unlabeled learning* [30, 31], which aims to train an instance-level binary classifier from two sets of unlabeled data with different class priors. In our problem setting, we can consider all the instances in similar bags as an unlabeled set and all the instances in dissimilar bags as another unlabeled set. Thus, we can find that the (instance-level) class priors of the two unlabeled sets are different. In this way, we can learn an instance-level binary classifier for MIL from similar and dissimilar bags by employing the unlabeled-unlabeled learning method.

Therefore, the key issue is how to figure out the class priors of the two unlabeled sets. Firstly, we can easily know that the proportion of positive bags in similar bag pairs is $\theta^2/\theta_{\text{S}}$ and the proportion of positive bags in dissimilar bag pairs is $1/2$. Secondly, following Bao et al. [6], we assume that instances in positive bags are drawn from the instance-level marginal distribution $p(\boldsymbol{x})$ for every instance $\boldsymbol{x}$ in positive bags, 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 positive class prior. Besides, we also assume that instances in negative bags are drawn from the instance-level negative class-conditional distribution $p(\boldsymbol{x} \mid y = -1)$, for every instance $\boldsymbol{x}$ in negative bags. In this way, we can calculate the class prior in the unlabeled set of similar bags as $\pi\theta^2/\theta_{\text{S}}$ and the class prior in the unlabeled set of dissimilar bags as $\pi/2$. Since there are two unlabeled sets with different class priors, we can train a binary classifier by minizing the empirical approaximation of the risk estimator provided in the following proposition.

Proposition 2 (Theorem 4 in Lu et al. [30]). *Let η and η' be different class priors of two unlabeled datasets (with $\eta > \eta'$), and $p_{\text{tr}}(\boldsymbol{x})$ and $p_{\text{tr}'}(\boldsymbol{x}')$ be the densities of two datasets of unlabeled data, respectively. The classification risk $R(f)$ in Eq. (1) can be equivalently represented as*

$$\begin{aligned} R(f) = R_{\text{UU}}(f) = \mathbb{E}_{p_{\text{tr}}(\boldsymbol{x})} \Big[& \frac{(1 - \eta')\pi}{\eta - \eta'} \ell(f(\boldsymbol{x}), +1) \\ & - \frac{\eta'(1 - \pi)}{\eta - \eta'} \ell(f(\boldsymbol{x}), -1) \Big] \\ + \mathbb{E}_{p_{\text{tr}'}(\boldsymbol{x}')} \Big[& \frac{\eta(1 - \pi)}{\eta - \eta'} \ell(f(\boldsymbol{x}'), -1) \\ & - \frac{(1 - \eta)\pi}{\eta - \eta'} \ell(f(\boldsymbol{x}'), +1) \Big]. \end{aligned} \tag{12}$$

As we have analyzed, in our problem of training an instance-level binary classifier for MIL from similar and dissimilar bags, we can obtain $\eta = \pi\theta^2/\theta_{\text{S}}$ and $\eta' = \pi/2$. Let us denote all the instances in similar bags as $\{\boldsymbol{x}_i\}_{i=1}^{n_{\text{tr}}}$ and all the instances in dissimilar bags as $\{\boldsymbol{x}'_j\}_{j=1}^{n_{\text{tr}'}}$. In this way, we can minimizing the following empirical risk for training an instance-level classifier:

$$\begin{aligned} \widehat{R}_{\text{UU}}(f) = \sum_{i=1}^{n_{\text{tr}}} \Big(& \frac{(2 - \pi)\theta_{\text{S}}}{2\theta - 1} \ell(f(\boldsymbol{x}_i), +1) - \frac{(1 - \pi)\theta_{\text{S}}}{2\theta - 1} \ell(f(\boldsymbol{x}), -1) \Big) \\ + \sum_{i=1}^{n_{\text{tr}'}} \Big(& \frac{2(1 - \pi)\theta^2}{2\theta - 1} \ell(f(\boldsymbol{x}'), -1) - \frac{2(\theta_{\text{S}} - \theta^2\pi)}{2\theta - 1} \ell(f(\boldsymbol{x}'), +1) \Big). \end{aligned} \tag{13}$$

After training the instance-level binary classifier, we can predict the labels of all the instances in the test bag, so that we can predict the bag label of the test bag. By inserting different binary loss functions into Eq. (13), we can obtain various compared methods (each method could be considered as a strong baseline) for MIL from similar and dissimilar bags.

It is worth noting that our goal in this paper is to predict only bag-level labels and we do not need to know instance-level labels. Therefore, learning an instance-level classifier could be considered as a more complex solution to our problem than our proposed convex formulation that directly trains a bag-level classifier. According to *Ockham's Razor* that the simplest is usually the right one, we can expect that our proposed convex formulation in the previous section is superior to the baseline proposed in this section.

5 EXPERIMENTS

In this section, we conduct extensive experiments on both benchmark datasets and text categorization datasets. We compare our proposed convex formulation (in Eq. (7)) including two bag-level methods: **CVX-SQ** (using the squared loss) and **CVX-DH** (using the double hinge loss) with the proposed baseline (in Eq. (13)) including six instance-level methods: **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

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: Classification accuracy of each method on the benchmark datasets. The best performance is highlighted in bold.

Datasets	θ	Convex Formulation		Baseline					
		CVX-SQ	CVX-DH	BL-SQ	BL-DH	BL-RP	BL-LG	BL-HG	BL-SG
Musk1	0.6	**0.802** **(0.032)**	0.779 (0.071)	0.663 (0.090)	0.726 (0.056)	0.769 (0.058)	0.726 (0.056)	0.743 (0.060)	0.761 (0.053)
	0.7	**0.894** **(0.021)**	0.854 (0.052)	0.678 (0.043)	0.798 (0.061)	0.840 (0.049)	0.800 (0.056)	0.815 (0.044)	0.846 (0.047)
	0.8	**0.940** **(0.024)**	0.922 (0.026)	0.649 (0.061)	0.875 (0.035)	0.888 (0.021)	0.878 (0.034)	0.887 (0.029)	0.892 (0.021)
Musk2	0.6	**0.822** **(0.052)**	0.790 (0.076)	0.560 (0.103)	0.657 (0.104)	0.687 (0.086)	0.662 (0.104)	0.659 (0.111)	0.670 (0.089)
	0.7	**0.887** **(0.030)**	0.876 (0.028)	0.639 (0.134)	0.750 (0.068)	0.763 (0.057)	0.754 (0.068)	0.750 (0.067)	0.751 (0.049)
	0.8	**0.930** **(0.025)**	0.896 (0.025)	0.746 (0.064)	0.828 (0.028)	0.822 (0.024)	0.829 (0.030)	0.828 (0.029)	0.822 (0.030)
Fox	0.6	0.625 (0.038)	0.607 (0.011)	0.613 (0.030)	0.627 (0.044)	0.630 (0.033)	0.623 (0.042)	**0.632** **(0.031)**	0.625 (0.027)
	0.7	**0.745** **(0.044)**	0.708 (0.021)	0.702 (0.021)	0.721 (0.025)	0.725 (0.033)	0.715 (0.031)	0.729 (0.026)	0.723 (0.030)
	0.8	**0.832** **(0.056)**	0.809 (0.008)	0.802 (0.030)	0.814 (0.015)	0.820 (0.016)	0.810 (0.014)	0.815 (0.016)	0.816 (0.021)
Elephant	0.6	**0.747** **(0.059)**	0.736 (0.077)	0.646 (0.050)	0.698 (0.034)	0.722 (0.030)	0.710 (0.027)	0.711 (0.025)	0.719 (0.029)
	0.7	**0.829** **(0.052)**	0.821 (0.056)	0.751 (0.025)	0.792 (0.035)	0.793 (0.028)	0.788 (0.034)	0.795 (0.028)	0.784 (0.029)
	0.8	**0.886** **(0.050)**	0.868 (0.017)	0.834 (0.028)	0.873 (0.018)	0.855 (0.020)	0.872 (0.019)	0.872 (0.024)	0.850 (0.016)
Tiger	0.6	0.697 (0.050)	0.682 (0.073)	0.578 (0.054)	0.698 (0.021)	0.705 (0.032)	0.694 (0.023)	0.698 (0.023)	**0.719** **(0.031)**
	0.7	**0.814** **(0.028)**	0.744 (0.041)	0.710 (0.044)	0.792 (0.028)	0.800 (0.021)	0.799 (0.028)	0.800 (0.028)	0.791 (0.022)
	0.8	**0.869** **(0.021)**	0.825 (0.029)	0.767 (0.065)	0.861 (0.029)	0.856 (0.025)	0.860 (0.025)	0.859 (0.026)	0.855 (0.019)

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 derive the analytical solution in Eq. (9). For CVX-DH, we solve the standard quadratic programming problem in Eq. (10) using CVXOPT [2]. For other compared baselines, we implement them using PyTorch [38]. 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 $\{10^{-3}, 10^{-2}, 10^{-1}\}$.

We evaluate the performance of our proposed methods under different bag-level class priors ($\theta \in \{0.6, 0.7, 0.8\}$). It is noteworthy

Table 3: 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 4: Classification accuracy of each method on text categorization datasets. The best performance is highlighted in bold.

Datasets	θ	Convex Formulation		Baseline					
		CVX-SQ	CVX-DH	BL-SQ	BL-DH	BL-RP	BL-LG	BL-HG	BL-SG
Component	0.6	0.830 (0.039)	**0.833** **(0.033)**	0.415 (0.052)	0.497 (0.032)	0.728 (0.055)	0.539 (0.061)	0.580 (0.074)	0.742 (0.063)
	0.7	**0.835** **(0.031)**	**0.835** **(0.029)**	0.462 (0.085)	0.630 (0.134)	0.800 (0.036)	0.695 (0.108)	0.744 (0.059)	0.803 (0.020)
	0.8	0.861 (0.020)	**0.867** **(0.021)**	0.666 (0.088)	0.840 (0.058)	0.838 (0.012)	0.845 (0.029)	0.849 (0.036)	0.828 (0.013)
Function	0.6	**0.869** **(0.028)**	0.868 (0.026)	0.455 (0.024)	0.519 (0.045)	0.764 (0.068)	0.535 (0.045)	0.584 (0.062)	0.775 (0.060)
	0.7	**0.871** **(0.026)**	0.867 (0.027)	0.416 (0.089)	0.645 (0.093)	0.795 (0.027)	0.695 (0.084)	0.773 (0.054)	0.812 (0.033)
	0.8	0.890 (0.032)	**0.893** **(0.029)**	0.764 (0.124)	0.851 (0.070)	0.862 (0.027)	0.869 (0.031)	0.864 (0.033)	0.848 (0.024)
Process	0.6	**0.881** **(0.012)**	**0.881** **(0.015)**	0.454 (0.014)	0.542 (0.041)	0.788 (0.065)	0.644 (0.051)	0.553 (0.065)	0.800 (0.058)
	0.7	0.885 (0.019)	**0.886** **(0.015)**	0.516 (0.111)	0.717 (0.063)	0.839 (0.036)	0.772 (0.085)	0.810 (0.041)	0.845 (0.031)
	0.8	0.899 (0.013)	**0.901** **(0.013)**	0.810 (0.067)	0.880 (0.035)	0.871 (0.022)	0.873 (0.024)	0.878 (0.031)	0.847 (0.015)

that it seems that we need to know the value of the bag-level class prior θ in advance. However, we show that we are able to empirically estimate θ according to our introduced data generation process of similar and dissimilar bags. Specifically, we can exactly estimate θ_{S} by counting the proportion of the collected similar bag pairs in all the bag pairs. Since $\theta_{\mathrm{S}} = \theta^2 + (1-\theta)^2$, we have $2\theta_{\mathrm{S}} - 1 = \theta_{\mathrm{S}} - \theta_{\mathrm{D}} = (2\theta - 1)^2 \geq 0$, thus we obtain $\theta = (\sqrt{2\theta_{\mathrm{S}} - 1} + 1)/2$. Since $\sqrt{\theta_{\mathrm{S}} - 1} \geq 0$, we obtain $\theta \geq 0.5$. This implies that our only assumption of θ is that θ should be larger than 0.5. This is also why we select θ from $\{0.6, 0.7, 0.8\}$ for performance evaluation. For all the used datasets, we sample 600 similar and dissimilar bag pairs for $\theta = 0.6$, 500 bag pairs for $\theta = 0.7$, and 400 bag pairs for $\theta = 0.8$, following the data generation process introduced in Section 3.1. We repeat the sampling-and-training process 10 times and record mean classification accuracy with standard deviation.

5.1 Experiments on Benchmark Datasets

We use five commonly used benchmark datasets in MIL studies [3, 13], including Musk1, Musk2, Elephant, Fox, and Tiger. For these datasets, Musk1 has 47 positive bags and 45 negative bags. Musk2 consists of 39 positive bags and 63 negative bags. The other three datasets contain 100 positive bags and 100 negative bags. It is worth noting that these datasets are too small to evaluate the task of MIL from similar and dissimilar bags, we follow Bao et al. [6] to augment them for increasing the number of bags. Specifically, bags chosen randomly from the original datasets were duplicated and then Gaussian noise with mean zero and variance 0.01 was added to each dimension. In this way, we increased the number of samples in the Musk datasets (Musk1 and Musk2) 10 times and the Corel datasets (Elephant, Fox, and Tiger) 5 times. Table 1 reports the characteristics of these datasets[2] after preprocessing. Table 2 reports the classification accuracy with standard deviation of each learning method on the five benchmark datasets. As can be seen from Table 2, the baseline (including various instance-level methods) achieves decent performance, while our proposed bag-level methods CVX-SQ and CVX-DH are even better. Besides, CVX-SQ achieves the best performance in most cases.

5.2 Experiments on Text Categorization

We use three datasets[3] for the task of *biocreative* text categorization. In this task, we aim to decide whether a given <protein, document> pair should be annotated with some Gene Ontology (GO) code. We

[2] http://www.cs.columbia.edu/~andrews/mil/datasets.html

[3] https://veronikach.com/research/data-code/

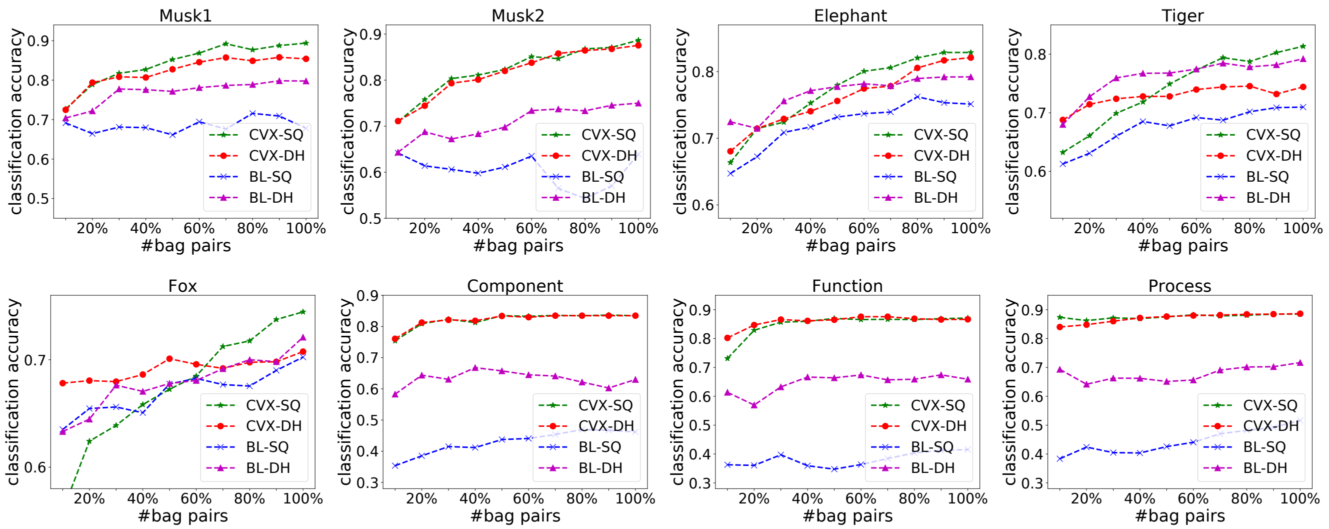

Figure 1: Classification accuracy of each method when the number of bag pairs increases.

have some documents (bags) comprised of paragraphs (instances), and each paragraph is represented by a feature vector. The used features 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. A <protein, document> pair is labeled with a GO code if the document contains some paragraphs that link the protein to the component, process, or function described by the GO code. Thus, we have three datasets in this biocreative text categorization task: Component, Function, and Process. Table 3 reports the detailed information of the three datasets. Table 4 reports the classification accuracy of each method on these three datasets. We can also observe that our proposed bag-level methods CVX-SQ and CVX-DH are clearly superior to other compared instance-level methods, and CVX-SQ achieves similar performance as CVX-DH.

5.3 Further Analysis

5.3.1 Performance of Increasing Bag Pairs. As shown by Theorem 1, the performance of our proposed convex formulation is expected to be improved if more similar and dissimilar bags are provided. To empirically validate such a theoretical finding, we further conduct experiments on the above datasets with $\theta = 0.7$, by changing the number of total bag pairs (100% means that we use all the generated similar and dissimilar bag pairs in the training process). As shown in Figure 1, the classification accuracy of our bag-level methods generally increases given more bag pairs. This observation is clearly in accordance with our derived generalization error bound in Theorem 1, because the generalization error decreases as the number of bag pairs increases. However, such a trend is not very clear for our proposed baseline (e.g., BL-SQ) since its performance is not theoretically guaranteed given more training data. In addition, the bag-level methods (CVX-SQ and CVX-DH) generally outperform the instance-level methods (BL-SQ and BL-DH) given different number of bag pairs.

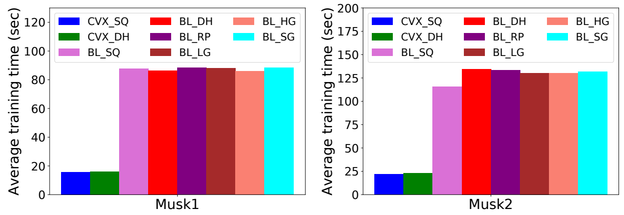

Figure 2: Average training time of each method on the benchmark datasets Musk1 and Musk2.

5.3.2 Training Efficiency Analysis. To show the advantage of our bag-level methods (convex formulation) over instance-level methods (baseline) in terms in terms of training efficiency, we perform MIL from similar and dissimilar bags using each method on Musk1 and Musk2 with $\theta = 0.7$. We show the average training time in Figure 2. As can be seen from Figure 2, the average training time of our 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.

6 CONCLUSION

In this paper, we investigated a novel weakly supervised binary classification called multiple-instance learning from similar and dissimilar bags, where we aim to train a bag-level binary classifier from only similar and dissimilar bags (indicating whether two bags share the same label or not). To the best of our knowledge, this paper provided the first attempt to study this problem. To solve this new MIL problem, we proposed a convex formulation to train a *bag-level* classifier based on empirical risk minimization and theoretically derived a generalization error bound. In addition, we also proposed a strong baseline for this new MIL problem, which aims to train an *instance-level* classifier by minimizing the instance-level empirical risk. Extensive experimental results clearly demonstrated that our proposed baseline works well, while our proposed convex formulation is even better. In future work, we will investigate multiple-instance learning with other types of weak supervision.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grants 62072062 and U20A20176, and the Natural Science Foundation of Chongqing, China, under Grant cstc2019jcyjjqX0026. This work was also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-0013), National Satellite of Excellence in Trustworthy Software Systems (Award No: NSOE-TSS2019-01), and NTU. Gang Niu was supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, Japan.

REFERENCES

[1] Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *AIJ* 201 (2013), 81–105.

[2] Martin S Andersen, Joachim Dahl, and Lieven Vandenberghe. 2013. CVXOPT: Python software for convex optimization. *URL https://cvxopt. org* 64 (2013).

[3] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2002. Support vector machines for multiple-instance learning. In *NeurIPS*. 577–584.

[4] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. 2009. Visual tracking with online multiple instance learning. In *CVPR*. 983–990.

[5] Han Bao, Gang Niu, and Masashi Sugiyama. 2018. Classification from Pairwise Similarity and Unlabeled Data. In *ICML*. 452–461.

[6] Han Bao, Tomoya Sakai, Issei Sato, and Masashi Sugiyama. 2018. Convex formulation of multiple instance learning from positive and unlabeled bags. *Neural Networks* 105 (2018), 132–141.

[7] Han Bao, Takuya Shimada, Liyuan Xu, Issei Sato, and Masashi Sugiyama. 2020. Similarity-based Classification: Connecting Similarity Learning to Binary Classification. *arXiv preprint arXiv:2006.06207* (2020).

[8] Peter L Bartlett and Shahar Mendelson. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR* 3, 11 (2002), 463–482.

[9] Yuzhou Cao, Lei Feng, Yitian Xu, Bo An, Gang Niu, and Masashi Sugiyama. 2021. Larning from Similarity-Confidence Data. In *ICML*.

[10] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition* 77 (2018), 329–353.

[11] Yixin Chen, Jinbo Bi, and James Ze Wang. 2006. MILES: Multiple-instance learning via embedded instance selection. *TPAMI* 28, 12 (2006), 1931–1947.

[12] Zhenghang Cui, Nontawat Charoenphakdee, Issei Sato, and Masashi Sugiyama. 2020. Classification from triplet comparison data. *Neural Computation* 32, 3 (2020), 659–681.

[13] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 1-2 (1997), 31–71.

[14] M. C. du Plessis, G. Niu, and M. Sugiyama. 2014. Analysis of learning from positive and unlabeled data. In *NeurIPS*. 703–711.

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

[16] M Murat Dundar, Glenn Fung, Balaji Krishnapuram, and R Bharat Rao. 2008. Multiple-instance learning algorithms for computer-aided detection. *IEEE Transactions on Biomedical Engineering* 55, 3 (2008), 1015–1021.

[17] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. 2020. Provably consistent partial-label learning. In *NeurIPS*.

[18] Lei Feng, Senlin Shu, Nan Lu, Bo Han, Miao Xu, Gang Niu, Bo An, and Masashi Sugiyama. 2021. Pointwise Binary Classification with Pairwise Confidence Comparisons. In *ICML*.

[19] James Richard Foulds and Eibe Frank. 2010. A review of multi-instance learning assumptions. *The Knowledge Engineering Review* 25 (2010), 1–25.

[20] Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola. 2002. Multi-instance kernels. In *ICML*. 179–186.

[21] Chen Gong, Hong Shi, Tong-Liang Liu, Chuang Zhang, Jian Yang, and Da-Cheng Tao. 2019. Loss decomposition and centroid estimation for positive and unlabeled learning. *TPAMI* (2019).

[22] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872* (2018).

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

[24] Asako Kanezaki, Tatsuya Harada, and Yasuo Kuniyoshi. 2011. Scale and rotation invariant color features for weakly-supervised object learning in 3D space. In *ICCV Workshops*. 617–624.

[25] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*. 1675–1685.

[26] Christian Leistner, Amir Saffari, and Horst Bischof. 2010. MIForests: Multiple-instance learning with randomized trees. In *ECCV*. 29–42.

[27] Weixin Li and Nuno Vasconcelos. 2015. Multiple instance learning for soft bags via top instances. In *CVPR*. 4277–4285.

[28] Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. 2019. Towards safe weakly supervised learning. *TPAMI* 43, 1 (2019), 334–346.

[29] Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *TPAMI* 38, 3 (2015), 447–461.

[30] Nan Lu, Gang Niu, Aditya K. Menon, and Masashi Sugiyama. 2019. On the Minimal Supervision for Training Any Binary Classifier from Only Unlabeled Data. In *ICLR*.

[31] Nan Lu, Tianyi Zhang, Gang Niu, and Masashi Sugiyama. 2020. Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach. In *AISTATS*. 1115–1125.

[32] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. 2020. Progressive identification of true labels for partial-label learning. In *ICML*. 6500–6510.

[33] Oded Maron and Tomás Lozano-Pérez. 1998. A framework for multiple-instance learning. In *NeurIPS*. 570–576.

[34] Shahar Mendelson. 2008. Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory* 54, 8 (2008), 3797–3803.

[35] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI* 41, 8 (2018), 1979–1993.

[36] Anton J Nederhof. 1985. Methods of coping with social desirability bias: A review. *European Journal of Social Psychology* 15, 3 (1985), 263–280.

[37] Megha Pandey and Svetlana Lazebnik. 2011. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*. 1307–1314.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

[39] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*. 1944–1952.

[40] Takuya Shimada, Han Bao, Issei Sato, and Masashi Sugiyama. 2020. Classification from pairwise similarities/dissimilarities and unlabeled data via empirical risk minimization. *Neural Computation* (2020).

[41] Kazuhiko Shinoda, Hirotaka Kaji, and Masashi Sugiyama. 2020. Binary Classification from Positive Data with Skewed Confidence. In *IJCAI*. 3328–3334.

[42] Tong Tong, Robin Wolz, Qinquan Gao, Ricardo Guerrero, Joseph V Hajnal, Daniel Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. 2014. Multiple instance learning for classification of dementia in brain MRI. *Medical Image Analysis* 18, 5 (2014), 808–818.

[43] Deng-Bao Wang, Li Li, and Min-Ling Zhang. 2019. Adaptive graph guided disambiguation for partial label learning. In *KDD*. 83–91.

[44] Jun Wang and Jean-Daniel Zucker. 2000. Solving multiple-instance problem: A lazy learning approach. In *ICML*. 1119–1125.

[45] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[46] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. 2015. Deep multiple instance learning for image classification and auto-annotation. In *CVPR*. 3460–3469.

[47] Xin Xu and Eibe Frank. 2004. Logistic regression and boosting for labeled bags of instances. In *PAKDD*. Springer, 272–281.

[48] Cha Zhang and Paul Viola. 2007. Multiple-instance pruning for learning efficient cascade detectors. *NeurIPS* 20, 1681–1688.

[49] Qi Zhang and Sally A Goldman. 2001. EM-DD: An improved multiple-instance learning technique. In *NeurIPS*. 1073–1080.

[50] Teng Zhang and Hai Jin. 2020. Optimal Margin Distribution Machine for Multi-Instance Learning. In *IJCAI*. 2383–2389.

[51] Ya-Lin Zhang and Zhi-Hua Zhou. 2017. Multi-Instance Learning with Key Instance Shift.. In *IJCAI*. 3441–3447.

[52] Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National Science Review* 5, 1 (2018), 44–53.

[53] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *ICML*. 1249–1256.

[54] Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3, 1 (2009), 1–130.