SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2024, Vol. 67, Iss. 3, 132103:1–132103:14
https://doi.org/10.1007/s11432-023-3771-6

Multi-instance partial-label learning: towards exploiting dual inexact supervision

Wei TANG[1,2], Weijia ZHANG[3] & Min-Ling ZHANG[1,2*]

[1]*School of Computer Science and Engineering, Southeast University, Nanjing* 210096*, China;*
[2]*Key Laboratory of Computer Network and Information Integration* (*Southeast University*),
Ministry of Education, Nanjing 211189*, China;*
[3]*School of Information and Physical Sciences, The University of Newcastle, Callaghan* NSW 2308*, Australia*

Received 27 January 2023/Revised 27 February 2023/Accepted 6 April 2023/Published online 19 February 2024

Abstract Weakly supervised machine learning algorithms are able to learn from ambiguous samples or labels, e.g., multi-instance learning or partial-label learning. However, in some real-world tasks, each training sample is associated with not only multiple instances but also a candidate label set that contains one ground-truth label and some false positive labels. Specifically, at least one instance pertains to the ground-truth label while no instance belongs to the false positive labels. In this paper, we formalize such problems as multi-instance partial-label learning (MIPL). Existing multi-instance learning algorithms and partial-label learning algorithms are suboptimal for solving MIPL problems since the former fails to disambiguate a candidate label set, and the latter cannot handle a multi-instance bag. To address these issues, a tailored algorithm named MIPLGP, i.e., multi-instance partial-label learning with Gaussian processes, is proposed. MIPLGP first assigns each instance with a candidate label set in an augmented label space, then transforms the candidate label set into a logarithmic space to yield the disambiguated and continuous labels via an exclusive disambiguation strategy, and last induces a model based on the Gaussian processes. Experimental results on various datasets validate that MIPLGP is superior to well-established multi-instance learning and partial-label learning algorithms for solving MIPL problems.

Keywords machine learning, multi-instance partial-label learning, multi-instance learning, partial-label learning, Gaussian processes

1 Introduction

In standard supervised learning, each training sample is represented by a single instance associated with a class label. In recent years, supervised learning has achieved fruitful progress when a large amount of supervision is available. However, annotating large amounts of high-quality labels is time-consuming and costly, especially in fields that require expert knowledge. To overcome these issues, several weakly supervised learning paradigms have been proposed and have attracted significant research attention.

According to the quality and quantity of labeling information, weak supervision can be roughly divided into three categories, i.e., incomplete, inexact, and inaccurate supervision [1]. The inexact supervision refers to coarse-grained labels and contains two popular learning frameworks, i.e., multi-instance learning (MIL) and partial-label learning (PLL). As illustrated in Figure 1(a), in MIL, multiple training instances are arranged in a bag and we only know the binary bag-level label rather than the instance-level labels [2, 3]. A negative bag constitutes only negative instances, while a positive bag contains at least one positive instance and should contain several negative instances. Therefore, the exact positive instance is unknown; i.e., the inexact supervision exists in the instance space. The framework of PLL is shown in Figure 1(b), where each training sample is represented by a single instance coupled with a candidate label set, which consists of a ground-truth label and several false positive labels [4]. Therefore, the exact ground-truth label is unknown; i.e., the inexact supervision exists in the label space. In a sense, multi-instance learning and partial-label learning are dual frameworks to each other in which the inexact supervision exists in the instance space and the label space, respectively.

* Corresponding author (email: zhangml@seu.edu.cn)

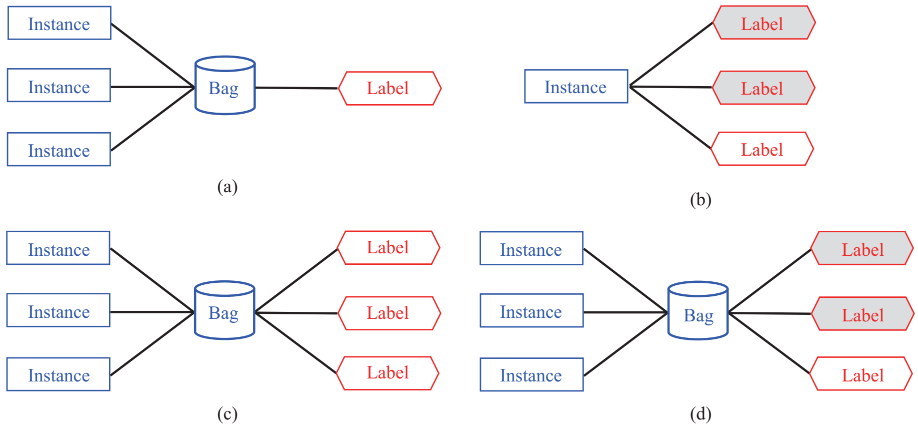

Figure 1 (Color online) Different weakly supervised learning frameworks, where the grey polygons refer to the false positive labels. (a) Multi-instance learning; (b) partial-label learning; (c) multi-instance multi-label learning; (d) multi-instance partial-label learning.

However, the inexact supervision can exist simultaneously in the instance space and the label space. For example, in histopathological image classification (as illustrated in Figure 2(a)), each image can be treated as a multi-instance bag, while ground-truth labels are provided by annotators with professional domain knowledge [5, 6]. To reduce the labeling cost, we can assign each multi-instance bag with a candidate label set rather than an exact label, and train a model to learn from the partially labeled multi-instance bags. In video classification (as illustrated in Figure 2(b)), each video consists of multiple frames represented as a set of instances, and the labels from social media contain noises that need to be corrected manually [7, 8]. The labeling cost can be significantly reduced if the video classification algorithm can learn from samples represented as sets of instances associated with candidate label sets.

Motivated by the potential applications, we formalize a novel framework named multi-instance partial-label learning (MIPL), which can learn from data with dual inexact supervision; i.e., the inexact supervision exists both in the instance space and the label space. In Figure 1(d), each training sample is represented by a multi-instance bag associated with a bag-level candidate label set, which consists of only one ground-truth label and some false positive labels. Moreover, the bag contains at least one instance that belongs to the ground-truth label while no instance pertains to the false positive labels. It is noteworthy that MIPL is different from multi-instance multi-label learning (MIML) presented in Figure 1(c), where each multi-instance bag is also associated with a label set [9]. The differences between MIPL and MIML lie in that the label set in MIML only contains ground-truth labels, while the label set in MIPL consists of one ground-truth label and some false positive labels.

To solve the MIPL problems, we propose a tailored algorithm named MiplGp, i.e., multi-instance partial-label learning with Gaussian processes. First, in order to assign each instance with a candidate label set containing the ground-truth label, we propose a label augmentation strategy to augment each candidate label set with a negative class label. Second, to infer the ground-truth labels from the candidate label sets, and render the MIPL problems amenable to be solved by a multi-output Gaussian processes regression model, we propose the Dirichlet disambiguation strategy. Last, to infer the parameters of the Dirichlet disambiguation strategy accurately, MiplGp induces a multi-output Gaussian processes regression model with GPU accelerations.

Empirical evaluation of MiplGp is conducted on five MIPL datasets. The experimental results indicate the following. (1) The MIPL is an exclusive problem that is difficult to be solved by either multi-instance learning approaches or partial-label learning approaches. (2) MiplGp achieves superior results against well-established multi-instance learning and partial-label learning approaches. (3) The proposed label augmentation and Dirichlet disambiguation strategies are both important for solving the MIPL problems.

The rest of the paper is organized as follows. First, related work is briefly reviewed. Second, we present the proposed MiplGp and report the experimental setting and results. Last, we conclude this paper.

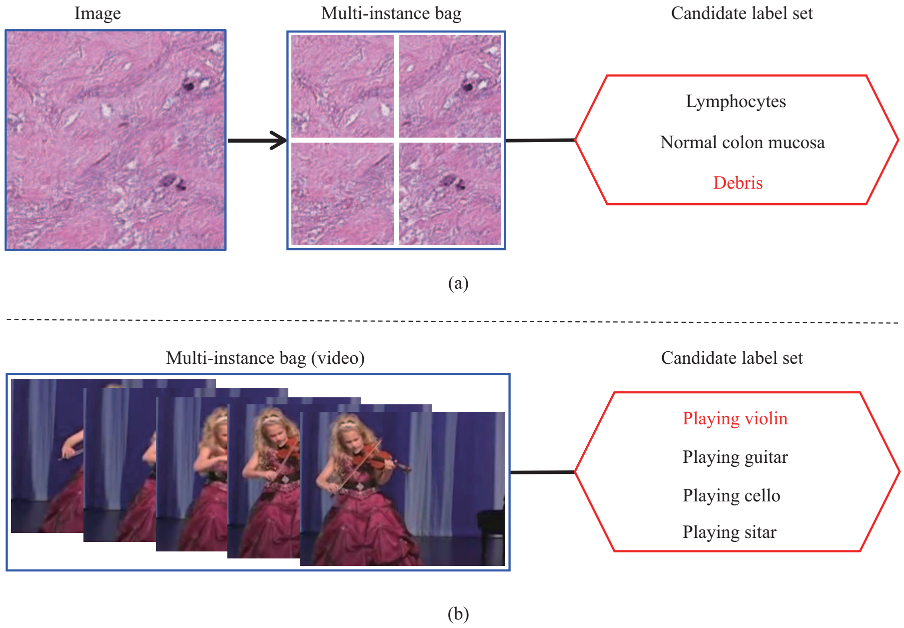

Figure 2 (Color online) Potential applications of MIPL, where the ground-truth label is shown in red. (a) Histopathological image classification; (b) video classification.

2 Related work

2.1 Multi-instance learning

Multi-instance learning algorithms can be roughly divided into two groups, i.e., instance-level algorithms and bag-level algorithms [2]. The former predicts bag-level labels by aggregating instance-level predictions, e.g., averaging the probabilities of all instances in a bag. The latter induces classifiers by treating each bag as a whole entity, which includes the bag-space paradigm and embedded-space paradigm.

In general, probabilistic multi-instance learning methods create a model that characterizes the distribution of instance-level labels and yields aggregated bag-level labels. Kim and Torre [10] proposed a nonparametric model to capture the underlying generative process by integrating a special bag class likelihood into the Gaussian processes. Along this line, Haußmann et al. [11] modified the standard bag likelihood and inferred an instance-label Gaussian processes classifier using variational Bayes. To model the dependencies among the instances, the variational autoencoder is employed to predict both the instance-level and bag-level labels [12, 13]. A recent tendency to address multi-instance learning problems is combining neural networks with attention mechanisms [14], where the attention scores indicate the importance of the instances to the bag [6, 15]. To our knowledge, these multi-instance learning algorithms are designed for binary classification problems, which cannot be directly adopted to solve MIPL problems. Although there are some multi-instance learning algorithms that can handle multi-classification problems [16, 17], they cannot tackle the challenge of false positive labels in the candidate label set.

2.2 Partial-label learning

Partial-label learning algorithms utilize the averaging-based or identification-based disambiguation strategies to disambiguate the candidate label sets. The averaging-based disambiguation strategy treats all labels in the candidate label set equally, and comprehensively considers the outputs of the learned model on each candidate label [18, 19]. The identification-based disambiguation strategy considers the potential ground-truth label as a latent variable, and disambiguates the ambiguous labels by optimizing the objective function related to the latent variable [20, 21].

Based on the graphic model, Jin and Ghahramani [4] minimized the relative entropy between the estimated label distribution and the prior distribution of the class labels. To capture underlying structures of the data, Liu and Dietterich [22] mapped training instances to mixture components and sampled a label for each mixture component. Based on Gaussian processes, Zhou et al. [23] defined a non-Gaussian likelihood to disambiguate the candidate label sets and computed the posterior distribution using Laplace approximation. Recently, some deep learning-based disambiguation methods have been investigated for partial-label learning [24, 25]. However, we note that all of them cannot handle multi-instance bags.

2.3 Multi-class Gaussian processes classification

Let $\mathcal{X} = \mathbb{R}^d$ denote the instance space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with q class labels. $\{(\boldsymbol{X}, \boldsymbol{Y}) \mid 1 \leqslant i \leqslant m\}$ is a supervised dataset, where $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ indicate m training instances, and $\boldsymbol{Y} = (y_1, y_2, \ldots, y_m)$ are the target class labels1). The goal of multi-class Gaussian processes classification (MCGPC) models is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$. For MCGPC models, the latent functions at m instances for q classes are $\boldsymbol{F} = [\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^q]^{\mathrm{T}} = [f_1^1, \ldots, f_m^1, f_1^2, \ldots, f_m^2, \ldots, f_1^q, \ldots, f_m^q]^{\mathrm{T}}$. Following Gaussian processes, the distribution of $\boldsymbol{F}$ is Gaussian, i.e., $\boldsymbol{f}^c \sim \mathcal{GP}(\boldsymbol{0}, k^c(\boldsymbol{x}, \boldsymbol{x}'))$, where $c \in \{1, 2, \ldots, q\}$, and $k^c(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function to measure the correlation at any a pair of instances $\boldsymbol{x}$ and $\boldsymbol{x}'$ for the c-th class. For an instance $\boldsymbol{x}_i$, the largest value of $\boldsymbol{f}_i = [f_i^1, f_i^2, \ldots, f_i^q]$ can be considered the class label, i.e., $Y_i = \arg\max_c f_i^c$. A common form of the likelihood is $P(Y_i = c \mid \boldsymbol{f}_i) = \exp(f_i^c)/\sum_{j=1}^q \exp(f_i^j)$. Consequently, the posterior distribution $P(\boldsymbol{F} \mid \boldsymbol{Y}, \boldsymbol{X}) \propto P(\boldsymbol{F} \mid \boldsymbol{X}) P(\boldsymbol{Y} \mid \boldsymbol{F})$ and the data likelihood $P(\boldsymbol{Y} \mid \boldsymbol{X}) = \int_{\boldsymbol{F}} P(\boldsymbol{F} \mid \boldsymbol{X}) P(\boldsymbol{Y} \mid \boldsymbol{F})$ are non-Gaussian, which is intractable and requires approximations.

Several algorithms have been proposed for MCGPC. Villacampa-Calvo and Hernández-Lobato [26] proposed to use expectation propagation to scale MCGPC to millions of data points. Théo et al. [27] introduced a logistic softmax likelihood and employed stochastic variational inference to yield fast and stable solutions. Without being limited to specific likelihoods, Liu et al. [28] proposed a framework for multiple likelihoods. Motivated by an astrophysics dataset that contains noisy inputs, Villacampa-Calvo et al. [29] trained MCGPC models with variational inference. However, these models cannot directly solve the MIPL problems.

3 Methodology

In this section, we propose an MIPL algorithm based on Gaussian processes, i.e., MiplGp. To the best of our knowledge, this is the first algorithm to address the MIPL problems. First, we introduce the notations and define an augmented label space, which equips each instance among a bag with a suitable candidate label set. Then, we propose a novel Dirichlet disambiguation strategy that effectively disambiguates the candidate label sets. Last, we present a multi-output Gaussian processes model.

Let $\mathcal{X} = \mathbb{R}^d$ denote the instance space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with q class labels. The goal of MIPL is to learn a classifier $h : 2^{\mathcal{X}} \to \mathcal{Y}$ from a training dataset $\{(\boldsymbol{X}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant m\}$ with m bags and corresponding candidate label sets. Specifically, a multi-instance partial-label sample is defined as $(\boldsymbol{X}_i, \boldsymbol{y}_i)$, where $\boldsymbol{X}_i = [\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^{z_i}]^{\mathrm{T}}$ is a bag of z_i instances, $\boldsymbol{x}_i^j \in \mathcal{X}$ for $\forall j \in \{1, 2, \ldots, z_i\}$, and $\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^q]^{\mathrm{T}} \in \{0, 1\}^q$ is the candidate label set of $\boldsymbol{X}_i$ where $y_i^c = 1$ means that the c-th label is one of the candidate labels of $\boldsymbol{X}_i$ and $y_i^c = 0$ otherwise.

3.1 Label augmentation

In multi-instance learning, only the bag-level labels are available, while the instance-level labels are unknown. To tackle this problem, a straightforward approach is to propagate the bag label to be the dummy label of all instances in the bag. But an obvious problem is that it will incorrectly assign the negative instances in a positive bag with positive labels. Analogously, in MIPL, if the bag-level candidate label set is directly applied to all the instances in the bag, the ground-truth labels of a substantial amount of instances will not exist in their candidate label sets, which violates the settings of partial-label learning.

To address the issue, we propose to utilize an augmented label space $\widetilde{\mathcal{Y}} = \{l_1, l_2, \ldots, l_q, l_{\mathrm{neg}}\}$ with $\tilde{q} = q + 1$ class labels, which augments a negative class label to the original label space $\mathcal{Y}$. Specifically, we assign instances that do not pertain to label space $\mathcal{Y}$ with the augmented negative class l_{neg}. For example, given a multi-instance bag $\boldsymbol{X}_i$ associated with a candidate label set $\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^q]^{\mathrm{T}}$, each instance in $\boldsymbol{X}_i$ is endowed with an augmented candidate label set $\widetilde{\boldsymbol{y}}_i = [y_i^1, y_i^2, \ldots, y_i^q, y_i^{\mathrm{neg}}]^{\mathrm{T}}$ where $y_i^{\mathrm{neg}} = 1$.

Consequently, we can derive the instance-level features and semantic information based on the augmented label space. Let $\boldsymbol{X} = [\boldsymbol{x}_1^1, \boldsymbol{x}_1^2, \ldots, \boldsymbol{x}_1^{z_1}, \boldsymbol{x}_2^1, \boldsymbol{x}_2^2, \ldots, \boldsymbol{x}_m^1, \ldots, \boldsymbol{x}_m^{z_m}]^{\mathrm{T}} \in \mathbb{R}^{n \times d}$ denote the feature matrix of n instances spread over m bags and $\widetilde{\boldsymbol{Y}} = [\widetilde{\boldsymbol{y}}_1^1, \widetilde{\boldsymbol{y}}_1^2, \ldots, \widetilde{\boldsymbol{y}}_1^{z_1}, \widetilde{\boldsymbol{y}}_2^1, \widetilde{\boldsymbol{y}}_2^2, \ldots, \widetilde{\boldsymbol{y}}_m^1, \ldots, \widetilde{\boldsymbol{y}}_m^{z_m}]^{\mathrm{T}} \in \mathbb{R}^{n \times \tilde{q}}$ denote the partial-label matrix of the n instances, where $n = \sum_{i=1}^m z_i$ is the total number of the instances

1) We use symbols in bold to denote matrices and vectors, and use regular symbols to denote scalars.

in the dataset and $\widetilde{\boldsymbol{y}}_i$ is the augmented candidate label set of $\boldsymbol{X}_i$ as well as each instance in bag $\boldsymbol{X}_i$, i.e., $\widetilde{\boldsymbol{y}}_i = \widetilde{\boldsymbol{y}}_i^1 = \widetilde{\boldsymbol{y}}_i^2 = \cdots = \widetilde{\boldsymbol{y}}_i^{z_i}$. Notably, the label augmentation occurs before the training phase.

3.2 Dirichlet disambiguation

To find the ground-truth labels from the candidate label sets, we conceive a novel disambiguation strategy for MIPL, which is named Dirichlet disambiguation. Following the disambiguation process, the likelihood is expressed as a logarithmic Gaussian function, which renders the MIPL problems amenable to be solved by a multi-output Gaussian processes regression model.

Given an augmented MIPL training dataset $(\boldsymbol{X}, \widetilde{\boldsymbol{Y}})$ with m bags totaling n instances, one instance and its candidate label set can be written as $(\boldsymbol{x}_i^j, \widetilde{\boldsymbol{y}}_i^j)$ for $\forall i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, z_i\}$. When the context is clear, we omit the instance index j to $(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i)$ for brevity. It is intuitive to use a categorical distribution $\mathrm{Cat}(\boldsymbol{\theta}_i)$ to infer the ground-truth label of the instance, where the class probability $\boldsymbol{\theta}_i = [\theta_i^1, \theta_i^2, \ldots, \theta_i^q, \theta_i^{\mathrm{neg}}]^{\mathrm{T}}$ is a multivariate continuous random variable constrained in a q dimensional probability simplex, i.e., $\sum_{c=1}^{\tilde{q}} \theta_i^c = 1$ and $\theta_i^c \geqslant 0$ $(\forall c \in \{1, 2, \ldots, q, \mathrm{neg}\})$. It is worth noting that the simplex promotes mutual exclusion among the candidate labels. To establish the class probability of the categorical distribution, we utilize the Dirichlet distribution, which is the conjugate prior to the categorical distribution, to reduce computational difficulty. Accordingly, the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}_i)$ with concentration parameters $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^q, \alpha_i^{\mathrm{neg}}]^{\mathrm{T}}$ is adopted to measure $\boldsymbol{\theta}_i$. Concretely, the likelihood model is given by

$$p(\widetilde{\boldsymbol{y}}_i \mid \boldsymbol{\alpha}_i) = \mathrm{Cat}(\boldsymbol{\theta}_i), \;\; \boldsymbol{\theta}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i). \tag{1}$$

To draw the coefficient $\boldsymbol{\theta}_i$ from the Dirichlet distribution, the accurate values of $\boldsymbol{\alpha}_i$ become pivotal. In supervised learning, each instance is associated with a unique ground-truth label, and thus a constant weight w can be directly added to the index corresponding to the ground-truth. For example, given an observation $\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^q]^{\mathrm{T}}$ that satisfies $y_i^c = 1$ and $y_i^j = 0$ $(\forall j \neq c)$, we have $\alpha_i^c = w + \alpha_\epsilon$ and $\alpha_i^j = \alpha_\epsilon$ $(\forall j \neq c)$, where α_ϵ is the Dirichlet prior such that $0 < \alpha_\epsilon \ll 1$. However, it is not appropriate to keep the weights unchanged during the training process, since the candidate label set is contaminated by the false positive labels. To overcome this limitation, we propose to synergize the Dirichlet distribution with an iterative disambiguation strategy to identify the ground-truth label from the contaminated candidate label set. To achieve the disambiguation strategy, $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^q, \alpha_i^{\mathrm{neg}}]$ are initialized to constant weights for $c \in \{1, 2, \ldots, q, \mathrm{neg}\}$:

$$\alpha_i^c = \begin{cases} \frac{1}{|\widetilde{\boldsymbol{y}}_i|} + \alpha_\epsilon, & \text{if } y_i^c = 1, \\ \alpha_\epsilon, & \text{otherwise}, \end{cases} \tag{2}$$

where $0 < \alpha_\epsilon \ll 1$ and $|\widetilde{\boldsymbol{y}}_i|$ is the cardinality which measures the number of non-zero elements in the augmented candidate label set $\widetilde{\boldsymbol{y}}_i$. The softmax values of the classifier output $\widetilde{\boldsymbol{h}}_i = \widetilde{\boldsymbol{h}}(\boldsymbol{x}_i) = [h_i^1, h_i^2, \ldots, h_i^q, h_i^{\mathrm{neg}}]^{\mathrm{T}}$ on the candidate label set indicate the probabilities that each candidate label is a ground-truth label. Therefore, we utilize the softmax values to gradually eliminate the false positive labels and identify the ground-truth label in each iteration:

$$\alpha_i^c = \begin{cases} \frac{\exp(h_i^c)}{\sum_{y_i^t=1} \exp(h_i^t)} + \alpha_\epsilon, & \text{if } y_i^c = 1, \\ \alpha_\epsilon, & \text{otherwise}. \end{cases} \tag{3}$$

In a nutshell, $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^q, \alpha_i^{\mathrm{neg}}]^{\mathrm{T}}$ are first initialized to constant weights as defined in (2) and then gradually updated according to (3).

Next, the problem becomes how to sample from the Dirichlet distribution. Considering both generation quality and computation cost, we design a two-step process to generate the Dirichlet samples from $\tilde{q}$ independent Gamma-distributed random variables. First, we generate $\tilde{q}$ Gamma-distributed random variables $\{\gamma_i^1, \gamma_i^2, \ldots, \gamma_i^q, \gamma_i^{\mathrm{neg}}\}$ from the Gamma distribution $\mathrm{Gamma}(\alpha_i^c, 1)$ for $c = 1, 2, \ldots, q, \mathrm{neg}$, respectively. Then, we normalize the $\tilde{q}$ Gamma-distributed random variables to obtain the realizations. The formulations of the generation process are as follows:

$$\theta_i^c = \frac{\gamma_i^c}{\sum_{j=1}^{\tilde{q}} \gamma_i^j}, \;\; \gamma_i^c \sim \mathrm{Gamma}(\alpha_i^c, 1). \tag{4}$$

The probability density function of $\mathrm{Gamma}(\alpha_i^c, 1)$ is $\frac{\gamma^{(\alpha_i^c-1)}\exp(-\gamma)}{\Gamma(\alpha_i^c)}$, where $\alpha_i^c > 0$ is called the shape parameter and $\Gamma(\cdot)$ is the Gamma function.

To use an exact Gaussian processes model to infer α_i^c accurately, we employ the random variables $\dot{x}_i^c$ drawn from a logarithmic normal distribution $\mathrm{LogNormal}(\dot{y}_i^c, \dot{\sigma}_i^c)$ to approximate the Gamma-distributed random variables γ_i^c by moment matching, i.e., $\mathbb{E}[\gamma_i^c] = \mathbb{E}[\dot{x}_i^c]$ and $\mathbb{V}[\gamma_i^c] = \mathbb{V}[\dot{x}_i^c]$:

$$\alpha_i^c = \exp\left(\dot{y}_i^c + \frac{\dot{\sigma}_i^c}{2}\right), \quad \alpha_i^c = \left(\exp\left(\dot{\sigma}_i^c\right) - 1\right)\exp\left(2\dot{y}_i^c + \dot{\sigma}_i^c\right). \tag{5}$$

As Milios et al. [30] showed, it is reasonable to estimate γ_i^c by $\mathrm{LogNormal}(\dot{y}_i^c, \dot{\sigma}_i^c)$. The parameters of $\mathrm{LogNormal}(\dot{y}_i^c, \dot{\sigma}_i^c)$ are derived by solving (5):

$$\dot{\sigma}_i^c = \log\left(\frac{1}{\alpha_i^c} + 1\right), \quad \dot{y}_i^c = \log\alpha_i^c - \frac{\dot{\sigma}_i^c}{2} = \frac{3}{2}\log\alpha_i^c - \frac{1}{2}\log(\alpha_i^c + 1), \tag{6}$$

where $\dot{y}_i^c$ is a continuous label in the logarithmic space and $\dot{\sigma}_i^c$ is a variance related to $\dot{y}_i^c$. Based on the above Dirichlet disambiguation strategy, the original sample $(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i)$ is transformed into $(\boldsymbol{x}_i, \dot{\boldsymbol{y}}_i)$, where $\dot{\boldsymbol{y}}_i$ is a candidate label set with continuous labels. Meanwhile, a Gaussian likelihood is constructed in the logarithmic space. Given an MIPL training dataset $(\boldsymbol{X}, \widetilde{\boldsymbol{Y}})$, we reshape the transformed candidate label matrix to yield a row-wise concatenation $\dot{\boldsymbol{Y}} = [\dot{\boldsymbol{y}}_1^1; \dot{\boldsymbol{y}}_1^2; \cdots; \dot{\boldsymbol{y}}_1^{z_1}; \dot{\boldsymbol{y}}_2^1; \dot{\boldsymbol{y}}_2^2; \cdots; \dot{\boldsymbol{y}}_m^1; \cdots; \dot{\boldsymbol{y}}_m^{z_m}] \in \mathbb{R}^{\tilde{q}n}$.

3.3 Gaussian processes regression model

Based on the continuous candidate label set matrix $\dot{\boldsymbol{Y}}$, we can transform the MIPL problems from multi-class classification problems to Gaussian processes regression problems with $\tilde{q}$ outputs. To estimate $\boldsymbol{\alpha}_i$ accurately, we develop a multi-output Gaussian processes regression model based on the Gaussian likelihood in the logarithmic space.

For the multi-output Gaussian processes regression model, the vectors of $\tilde{q}$ latent functions $\{f^1(\cdot), f^2(\cdot), \ldots, f^q(\cdot), f^{\mathrm{neg}}(\cdot)\}$ at all n training instances are first introduced: $\boldsymbol{F} = [\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^{\mathrm{neg}}]^{\mathrm{T}} = [f_1^1, \ldots, f_n^1, f_1^2, \ldots, f_n^2, \ldots, f_1^{\mathrm{neg}}, \ldots, f_n^{\mathrm{neg}}]^{\mathrm{T}}$, where the latent variate $\boldsymbol{F}$ has length $\tilde{q}n$. The distribution of $\boldsymbol{F}$ is defined by a prior mean function $\boldsymbol{\mu} = \boldsymbol{0}$ and a covariance function, i.e., a prior kernel $k(\cdot,\cdot): \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which is chosen to be a Matérn kernel [31] in this paper. For $\forall c, c' \in \{1, 2, \ldots, q, \mathrm{neg}\}$, the correlation of the outputs at any a pair of instances $\boldsymbol{x}$ and $\boldsymbol{x}'$ can be represented as

$$\mathrm{Cov}[f^c(\boldsymbol{x}), f^{c'}(\boldsymbol{x}')] = k^c(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}d}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}d}{\ell}\right), & \text{if } c = c', \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Generally speaking, ν is a smoothness parameter and takes the value from $\{0.5, 1.5, 2.5\}$. Here, ℓ is a positive parameter, d is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, and K_ν is a modified Bessel function. Finally, the covariance matrix $\boldsymbol{K} \in \mathbb{R}^{\tilde{q}n \times \tilde{q}n}$ of the $\tilde{q}$ latent processes is block diagonal by the matrices $\boldsymbol{K}^1, \boldsymbol{K}^2, \ldots, \boldsymbol{K}^{\tilde{q}}$ of shape $n \times n$. Gaussian processes place a Gaussian prior over a latent variable, i.e., $P(\boldsymbol{F} \mid \boldsymbol{X}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$, and the Gaussian likelihood in the logarithmic space is $P(\dot{\boldsymbol{Y}} \mid \boldsymbol{F}) = \mathcal{N}(\boldsymbol{F}, \Sigma)$, where Σ is the matrix form of $\dot{\sigma}_i^c$ in $\mathrm{LogNormal}(\dot{y}_i^c, \dot{\sigma}_i^c)$. Following the Bayes' rule, the posterior distribution $P(\boldsymbol{F} \mid \boldsymbol{X}, \dot{\boldsymbol{Y}}) \propto P(\boldsymbol{F} \mid \boldsymbol{X})P(\dot{\boldsymbol{Y}} \mid \boldsymbol{F})$ and likelihood $P(\dot{\boldsymbol{Y}} \mid \boldsymbol{X}) = \int_{\boldsymbol{F}} P(\boldsymbol{F} \mid \boldsymbol{X})P(\dot{\boldsymbol{Y}} \mid \boldsymbol{F})$ are both Gaussian.

The above likelihoods are based on the instances-level features and labels. In multi-instance learning, however, a ubiquitous issue is how to aggregate the instance labels to generate a bag label. This problem also takes place in MIPL and is more difficult due to the ambiguous multi-class classification. A feasible solution is to set the bag label with the class label corresponding to the maximum value among the class probabilities of all instances in the bag. Let $\boldsymbol{\Theta}_i = [\boldsymbol{\theta}_i^1, \boldsymbol{\theta}_i^2, \ldots, \boldsymbol{\theta}_i^{z_i}]^{\mathrm{T}} \in \mathbb{R}^{z_i \times \tilde{q}}$ ($\boldsymbol{\theta}_i^j = [\theta_i^1, \theta_i^2, \ldots, \theta_i^q, \theta_i^{\mathrm{neg}}]^{\mathrm{T}}$ for $j = 1, 2, \ldots, z_i$) denote the class probabilities of the multi-instance bag $\boldsymbol{X}_i$ among $\tilde{q}$ outputs. The aggregated bag label is as follows:

$$Y_i = \underset{j \neq \tilde{q}}{\arg\max}\ \Theta_i^{(z,j)}, \tag{8}$$

where $\Theta_i^{(z,j)}$ is the element at the z-th row and the j-th column of $\boldsymbol{\Theta}_i$. The plate diagram of MiplGp is illustrated in Figure 3, where the grey circles represent the observed variables, i.e., features and bag labels, and the white circles represent the latent variables.

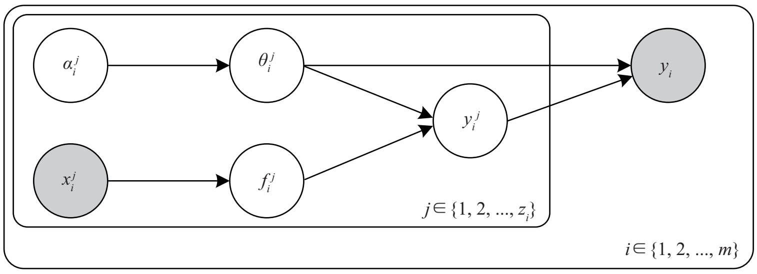

Figure 3 Plate diagram for MIPLGP, where the grey circles are observed variables and white ones are latent variables.

During the training, we minimize the negative log marginal likelihood:

$$\mathcal{L} = -\log P(\dot{\boldsymbol{Y}} \mid \boldsymbol{X}, \Phi) \propto \log |\boldsymbol{K}| + \dot{\boldsymbol{Y}}^\top \boldsymbol{K}^{-1} \dot{\boldsymbol{Y}}, \tag{9}$$

where Φ is the model parameter. The derivative is as follows:

$$\frac{\partial \mathcal{L}}{\partial \Phi} \propto \operatorname{Tr}\left(\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial \Phi}\right) - \dot{\boldsymbol{Y}}^\top \boldsymbol{K}\frac{\partial \boldsymbol{K}^{-1}}{\partial \Phi}\boldsymbol{K}\dot{\boldsymbol{Y}}, \tag{10}$$

where $\operatorname{Tr}(\cdot)$ is the trace operation. Using the Cholesky decomposition, the naive multi-output Gaussian processes regression algorithms computer $\boldsymbol{K}^{-1}\dot{\boldsymbol{Y}}$ with $\mathcal{O}(\tilde{q}n^3)$ time. However, we follow the preconditioned conjugate gradients (PCG) algorithm in [32], which formulizes $\boldsymbol{K}^{-1}\dot{\boldsymbol{Y}}$ as the solution to an optimization problem $\boldsymbol{K}^{-1}\dot{\boldsymbol{Y}} = \arg\min_{\boldsymbol{v}}(\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{v} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{Y})$, and solves the problem iteratively. Consequently, the asymptotic complexity of our multi-output Gaussian processes regression model is $\mathcal{O}(\tilde{q}n^2)$ [33].

Given an unseen multi-instance bag $\boldsymbol{X}_* = [\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{z_*}]$ with z_* instances, the GP model generates the predictive distribution $P(\boldsymbol{F}^*|\boldsymbol{X}, \dot{\boldsymbol{Y}}, \boldsymbol{x}^{i_*})$ $(i_* = 1, 2, \ldots, z_*)$ for the corresponding latent variables $\boldsymbol{F}^* = [f^{*1}, f^{*2}, \ldots, f^{*\tilde{q}}]^{\mathrm{T}}$. The Dirichlet posterior of the class label is procured from the predictive distribution using the Monte Carlo method so as to calculate the expectation of class probability:

$$\mathbb{E}[\theta_{i_*}^c \mid \boldsymbol{X}, \dot{\boldsymbol{Y}}, \boldsymbol{x}^{i_*}] = \int_{\boldsymbol{F}^*} \frac{\exp\left(f^{*c}(\boldsymbol{x}^{i_*})\right)}{\sum_{j=1}^{\tilde{q}} \exp\left(f^{*j}(\boldsymbol{x}^{i_*})\right)} P(f^{*c}(\boldsymbol{x}^{i_*}) \mid \boldsymbol{X}, \dot{\boldsymbol{Y}}, \boldsymbol{x}^{i_*}), \tag{11}$$

where $P(f^{*c}(\boldsymbol{x}^{i_*}) \mid \boldsymbol{X}, \dot{\boldsymbol{Y}}, \boldsymbol{x}^{i_*})$ is the predictive distribution for $f^{*c}(\cdot,\cdot)$. We write the class probability of $\boldsymbol{x}^{i_*}$ as $\boldsymbol{\theta}_*^{i_*} = [\theta_{i_*}^1, \theta_{i_*}^2, \ldots, \theta_{i_*}^q, \theta_{i_*}^{\mathrm{neg}}]$, and let $\boldsymbol{\Theta}_* = [\boldsymbol{\theta}_*^1, \boldsymbol{\theta}_*^2, \ldots, \boldsymbol{\theta}_*^{z_*}]^{\mathrm{T}} \in \mathbb{R}^{z_* \times \tilde{q}}$ denote the class probabilities of all instances in the test bag $\boldsymbol{X}_*$. A bag label is predicted by

$$Y_* = \underset{j \neq \tilde{q}}{\arg\max}\ \Theta_*^{(z,j)}, \tag{12}$$

where $\Theta_*^{(z,j)}$ is the element at the z-th row and the j-th column of $\boldsymbol{\Theta}_*$.

Algorithm 1 summarizes the complete procedure of MiplGp. First, the algorithm propagates the augmented candidate label set of each bag to all instances in the bag (Steps 1–7). After initializing the shape parameter of the Dirichlet distribution (Step 8), the Gaussian processes model is induced based on the transformed labels (Steps 9–20). Last, the label of an unseen multi-instance bag is returned by querying the predicted class probabilities (Step 21).

4 Experiments

4.1 Experimental setup

4.1.1 *Datasets*

To the best of our knowledge, there are no ready-made datasets for benchmarking MIPL algorithms. To overcome this limitation, we synthesize five MIPL datasets stemming from relevant literature [34–38], i.e., MNIST-mipl, FMNIST-mipl, Newsgroups-mipl, Birdsong-mipl, and SIVAL-mipl, from domains of image, text, and biology to compare MiplGp with other algorithms[2)].

2) The datasets are publicly available at http://palm.seu.edu.cn/zhangml/Resources.htm#MIPL_data.

Algorithm 1 $Y_* = \text{MIPLGP}(\mathcal{D}, \alpha_\epsilon, T, \boldsymbol{X}_*)$

Inputs: $\mathcal{D}$: the multi-instance partial-label training set $\{(\boldsymbol{X}_i, \boldsymbol{y}_i) \mid 1 \leqslant i \leqslant m\}$ ($\boldsymbol{X}_i = [\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^{z_i}]^{\mathrm{T}}$, $\boldsymbol{x}_i^j \in \mathcal{X}$, $\boldsymbol{X}_i \subseteq \mathcal{X}$, $\mathcal{X} = \mathbb{R}^d$, $\boldsymbol{y}_i = [y_i^1, y_i^2, \ldots, y_i^q]^{\mathrm{T}}$, $y_i^c \in \mathcal{Y}$, $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$); α_ϵ: the Dirichlet prior; T: the number of iterations; $\boldsymbol{X}_*$: the unseen multi-instance bags with z_* instances;
Outputs: $\boldsymbol{Y}_*$: the predicted bag label for $\boldsymbol{X}_*$;
Process:

```
Augment 𝒴 to 𝒴̃ = {l_1, l_2, ..., l_q, l_neg};
for i = 1 to m do
    Augment y_i to ỹ_i = [y_i^1, y_i^2, ..., y_i^q, y_i^neg]^T;
    for j = 1 to z_i do
        Propagate ỹ_i to be the label of each instance x_i^j;
    end for
end for
Initialize α_i as defined in (2);
for t = 1 to T do
    for i = 1 to m do
        for j = 1 to z_i do
            Model the likelihood p(ỹ_i | α_i) = Cat(θ_i), θ_i ~ Dir(α_i) according to (1);
            Generate Dirichlet samples from the Gamma distribution according to (4);
            Derive the continuous candidate label set ẏ_i as stated by (6);
        end for
    end for
    Calculate ℒ according to (9) and ∂ℒ/∂Φ as defined in (10);
    Update Φ by the optimizer;
    Update α_i according to (3);
end for
Return Y_* according to (12).
```

Table 1 Characteristics of the MIPL datasets

Dataset	#bags	#ins	#max	#min	#dims	#l-o	#l-t	#l-r	Percentage (%)	Domain
MNIST-MIPL	500	20664	48	35	784	10	5	5	8.0	Image
FMNIST-MIPL	500	20810	48	36	784	10	5	5	8.0	Image
Newsgroups-MIPL	1000	43122	86	11	200	20	10	10	8.0	Text
Birdsong-MIPL	1300	48425	76	25	38	13	13	1	8.3	Biology
SIVAL-MIPL	1500	47414	32	31	30	25	25	0	25.6	Image

The characteristics of the synthetic datasets are summarized in Table 1. Let #bags, #ins, #max, #min, #dims, #l-o, #l-t, #l-r, and percentage to denote the number of bags, number of instances, maximum number of instances in a bag, minimum number of instances in a bag, dimension of each instance, number of targeted class labels in the corresponding literature, number of targeted class labels in MIPL, number of reserved class labels, and percentage of the number of positive instances in each dataset.

To synthesize the multi-instance bag with a candidate label set, we take positive instances of the corresponding label from the targeted class labels and negative instances in the whole reserved class labels. Furthermore, we sample the false positive labels from the target classes without replacement. For comprehensive performance evaluation, the number of false positive labels depends on the controlling parameter r ($|\boldsymbol{y}_i| = r + 1$). The instances and false positive labels are randomly sampled.

We further explain the process for generating MIPL datasets using the popular 10-class classification dataset MNIST. To obtain an MNIST-MIPL sample, we use a two-step generating procedure. First, a multi-instance bag is procured. Specifically, we extract positive instances from one of the five target classes $\{0, 2, 4, 6, 8\}$ and draw all negative ones from the reserved classes $\{1, 3, 5, 7, 9\}$ randomly. Second, the candidate label set of the multi-instance bag is generated by randomly choosing r false positive labels from the target classes as false positive labels of the multi-instance bag. In our experiments, r is equal to 1, 2, and 3, respectively. More detailed information on the MIPL datasets is provided in Appendix A.

4.1.2 *Comparative algorithms*

MIPLGP is compared against four well-established multi-instances learning algorithms including three Gaussian processes-based algorithms, VWSGP [39], VGPMIL [11], and LM-VGPMIL [11], as well as variational autoencoder-based algorithm MIVAE [12]. In addition, we employ six partial-label learning algorithms containing two averaging-based algorithms PL-kNN [40] and CLPL [18], two identification-based algorithms LSB-CMM [22] and SURE [21], the graph matching based disambiguation algorithm GM-PLL [41], and the feature-aware disambiguation algorithm Pl-aggd [42]. The parameter configurations of the algorithms are suggested in the respective literature.

To verify the effectiveness of the Dirichlet disambiguation and the label augmentation, we evaluate two variants of MiplGp, i.e., MiplGp-uniform and MiplGp-naive. The former utilizes the uniform weights as defined in (2) throughout the iterations, and the latter handles the candidate label sets only in the original label space $\mathcal{Y}$, i.e., without label augmentation.

4.1.3 *Implementation*

We implement MiplGp using GPyTorch, which is a modular Gaussian process library in PyTorch [33]. For MiplGp and its variants, we use the Adam optimizer [43] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is 0.1 which is decayed via a cosine annealing method [44]. We set the number of iterations to 500 for MNIST-mipl and FMNIST-mipl datasets and 50 for the remaining three datasets. For Newsgroups-mipl dataset, the smoothness parameter ν in Matérn kernel is 0.5, while $\nu = 2.5$ for the rest. We uniform that ℓ in Matérn kernel is equal to 1, $\alpha_\epsilon = 0.0001$, the size of the preconditioner is 100, and the number of Monte Carlo sampling points is 512 for all datasets. We perform ten runs of 50%/50% random train/test splits on all datasets, and record the mean accuracies and standard deviations for each algorithm. Specifically, we conduct the pairwise t-test at a 0.05 significance level based on the results of ten runs. Experiments are conducted with Nvidia Tesla V100 GPUs.

4.2 Experimental results

To overcome the inability of multi-instance learning and partial-label learning to handle the dual inexact supervision, we carry out experiments with multi-instance learning algorithms and partial-label learning algorithms using different versions of degenerated MIPL datasets. It is worth noting that MiplGp and its variants can learn directly from the standard MIPL datasets.

4.2.1 *Comparison with partial-label learning algorithms*

Existing partial-label learning algorithms are incapable of handling the multi-instance bag. Therefore, an aggregated feature representation of the bag is a prerequisite for tackling the MIPL problems with partial-label learning algorithms. In multi-instance learning, the idea of the embedded-space paradigm is to explicitly distill the whole bag by defining a mapping function from the bag to a feature vector. Inspired by the idea, we utilize two schemes to map the bag to a holistic feature vector, respectively.

• Mean scheme: For each bag, the average value of all instances in the corresponding feature dimension is calculated as the final feature value in that dimension. The dimension of the holistic feature vector is the same as that of each instance.

• MaxMin scheme: We choose the maximum values of all instances in each feature dimension and concatenate them with the minimum values of all instances in each feature dimension. Finally, the holistic feature representation of a bag is distilled with the length of $2d$.

The classification results with the varying number of false positive labels r are reported in Table 2, and Table 3 summarizes the win/tie/loss counts between MiplGp and each comparing algorithm. MiplGp achieves superior or competitive performance against the comparing algorithms. Out of the 180 statistical tests, we yield the following observations:

• MiplGp is statistically superior to the partial-label learning algorithms in 99.444% of the cases.

• Regardless of the Mean scheme or MaxMin scheme, MiplGp consistently outperforms the comparing partial-label learning algorithms by a notable margin, e.g., more than 5.3%, in almost all cases.

• Compared with MiplGp-uniform and MiplGp-naive, MiplGp achieves statistically favorable performance in 86.667% and 93.333% of the cases, respectively.

• In most cases, MiplGp-uniform is superior to MiplGp-naive, which means that the label augmentation strategy plays an important role in MiplGp. As the average accuracy of MiplGp-uniform decreases faster than that of MiplGp-naive as the number of false positive labels increases, the results demonstrate that the Dirichlet disambiguation is indispensable especially when there are a lot of false positive labels.

Based on the above observations, we speculate that there are two main reasons why the partial-label learning algorithms are inferior to MiplGp. First, as partial label learning algorithms cannot handle multi-instance bags, their learning procedures can only access the aggregated features from multi-instance bags using the Mean and MaxMin schemes, which are not as effective as the original features of multi-instance

Table 2 Classification accuracy (mean ± std) of each comparing algorithm in terms of the different number of false positive candidate labels [$r \in \{1, 2, 3\}$][a)]

Algorithm	r	MNIST-MIPL	FMNIST-MIPL	Newsgroups-MIPL	Birdsong-MIPL	SIVAL-MIPL
MiplGp	1	0.921 ± 0.018	0.806 ± 0.031	0.432 ± 0.018	0.628 ± 0.012	0.599 ± 0.020
	2	0.712 ± 0.045	0.778 ± 0.042	0.424 ± 0.019	0.589 ± 0.020	0.535 ± 0.020
	3	0.521 ± 0.084	0.592 ± 0.076	0.373 ± 0.023	0.538 ± 0.014	0.497 ± 0.023
MiplGp-uniform	1	0.834 ± 0.023•	0.778 ± 0.031•	0.417 ± 0.019•	0.623 ± 0.013•	0.595 ± 0.023
	2	0.531 ± 0.070•	0.746 ± 0.042•	0.401 ± 0.025•	0.581 ± 0.022•	0.530 ± 0.019•
	3	0.206 ± 0.009•	0.226 ± 0.039•	0.365 ± 0.013	0.526 ± 0.016•	0.489 ± 0.026•
MiplGp-naive	1	0.522 ± 0.025•	0.570 ± 0.016•	0.422 ± 0.019•	0.551 ± 0.010•	0.585 ± 0.019•
	2	0.438 ± 0.049•	0.468 ± 0.065•	0.407 ± 0.025•	0.511 ± 0.026•	0.523 ± 0.018•
	3	0.309 ± 0.072•	0.258 ± 0.045•	0.373 ± 0.017	0.464 ± 0.019•	0.480 ± 0.021•
				Mean		
Pl-kNN	1	0.397 ± 0.021•	0.419 ± 0.032•	0.133 ± 0.009•	0.213 ± 0.011•	0.155 ± 0.009•
	2	0.337 ± 0.020•	0.360 ± 0.030•	0.148 ± 0.006•	0.197 ± 0.012•	0.138 ± 0.008•
	3	0.284 ± 0.023•	0.264 ± 0.032•	0.142 ± 0.010•	0.182 ± 0.009•	0.123 ± 0.009•
Clpl	1	0.644 ± 0.023•	0.734 ± 0.031•	0.131 ± 0.027•	0.330 ± 0.013•	0.239 ± 0.009•
	2	0.528 ± 0.033•	0.671 ± 0.023•	0.112 ± 0.015•	0.295 ± 0.012•	0.221 ± 0.012•
	3	0.377 ± 0.042•	0.524 ± 0.046	0.111 ± 0.012•	0.282 ± 0.011•	0.200 ± 0.017•
Lsb-cmm	1	0.631 ± 0.045•	0.709 ± 0.025•	0.100 ± 0.000•	0.260 ± 0.013•	0.144 ± 0.012•
	2	0.416 ± 0.047•	0.560 ± 0.059•	0.100 ± 0.000•	0.242 ± 0.013•	0.116 ± 0.014•
	3	0.277 ± 0.038•	0.295 ± 0.032•	0.100 ± 0.000•	0.218 ± 0.011•	0.095 ± 0.015•
Sure	1	0.666 ± 0.027•	0.753 ± 0.019•	0.358 ± 0.019•	0.345 ± 0.008•	0.313 ± 0.022•
	2	0.512 ± 0.031•	0.685 ± 0.013•	0.300 ± 0.013•	0.319 ± 0.008•	0.284 ± 0.019•
	3	0.344 ± 0.075•	0.441 ± 0.063•	0.251 ± 0.017•	0.308 ± 0.013•	0.256 ± 0.013•
Gm-pll	1	0.248 ± 0.014•	0.268 ± 0.030•	0.183 ± 0.012•	0.146 ± 0.018•	0.175 ± 0.013•
	2	0.260 ± 0.022•	0.251 ± 0.019•	0.180 ± 0.015•	0.106 ± 0.012•	0.160 ± 0.014•
	3	0.235 ± 0.028•	0.246 ± 0.021•	0.157 ± 0.015•	0.092 ± 0.012•	0.136 ± 0.013•
Pl-aggd	1	0.643 ± 0.021•	0.714 ± 0.017•	0.325 ± 0.009•	0.332 ± 0.010•	0.312 ± 0.023•
	2	0.535 ± 0.034•	0.642 ± 0.026•	0.256 ± 0.015•	0.304 ± 0.013•	0.277 ± 0.019•
	3	0.363 ± 0.039•	0.429 ± 0.040•	0.213 ± 0.015•	0.292 ± 0.015•	0.244 ± 0.011•
				MaxMin		
Pl-kNN	1	0.388 ± 0.023•	0.309 ± 0.029•	0.119 ± 0.004•	0.274 ± 0.009•	0.177 ± 0.007•
	2	0.330 ± 0.016•	0.288 ± 0.019•	0.135 ± 0.008•	0.270 ± 0.006•	0.153 ± 0.009•
	3	0.266 ± 0.025•	0.239 ± 0.021•	0.138 ± 0.007•	0.250 ± 0.011•	0.136 ± 0.011•
Clpl	1	0.481 ± 0.020•	0.364 ± 0.026•	0.246 ± 0.009•	0.361 ± 0.016•	0.266 ± 0.011•
	2	0.396 ± 0.028•	0.332 ± 0.021•	0.200 ± 0.009•	0.328 ± 0.014•	0.223 ± 0.010•
	3	0.334 ± 0.039•	0.332 ± 0.028•	0.166 ± 0.018•	0.300 ± 0.015•	0.199 ± 0.014•
Lsb-cmm	1	0.372 ± 0.099•	0.238 ± 0.073•	0.221 ± 0.018•	0.319 ± 0.011•	0.248 ± 0.015•
	2	0.324 ± 0.038•	0.284 ± 0.039•	0.146 ± 0.039•	0.292 ± 0.014•	0.200 ± 0.017•
	3	0.220 ± 0.017•	0.210 ± 0.017•	0.113 ± 0.021•	0.272 ± 0.020•	0.157 ± 0.017•
Sure	1	0.528 ± 0.021•	0.404 ± 0.023•	0.316 ± 0.015•	0.381 ± 0.013•	0.372 ± 0.022•
	2	0.415 ± 0.028•	0.351 ± 0.025•	0.274 ± 0.013•	0.371 ± 0.015•	0.324 ± 0.013•
	3	0.321 ± 0.030•	0.304 ± 0.029•	0.245 ± 0.014•	0.341 ± 0.014•	0.288 ± 0.011•
Gm-pll	1	0.386 ± 0.024•	0.195 ± 0.016•	0.209 ± 0.020•	0.180 ± 0.019•	0.143 ± 0.013•
	2	0.346 ± 0.030•	0.225 ± 0.019•	0.181 ± 0.018•	0.139 ± 0.023•	0.121 ± 0.014•
	3	0.294 ± 0.024•	0.221 ± 0.013•	0.163 ± 0.019•	0.121 ± 0.020•	0.104 ± 0.012•
Pl-aggd	1	0.514 ± 0.024•	0.392 ± 0.016•	0.289 ± 0.014•	0.370 ± 0.013•	0.361 ± 0.021•
	2	0.428 ± 0.035•	0.346 ± 0.019•	0.249 ± 0.013•	0.353 ± 0.014•	0.310 ± 0.013•
	3	0.333 ± 0.039•	0.324 ± 0.025•	0.212 ± 0.011•	0.328 ± 0.015•	0.277 ± 0.013•

a) •/◦ indicates whether the performance of MiplGp is statistically superior/inferior to the comparing algorithm on each dataset (pairwise t-test at 0.05 significance level).

bags learned by MiplGp. Second, our proposed label augmentation strategy ensures that the ground-truth label for each instance in a multi-instance bag exists in the candidate label set, which keeps the Dirichlet disambiguation strategy from being misled by the instances with only false positive labels in the candidate label set. However, the partial-label learning algorithms do not have such a strategy and

Table 3 Win/tie/loss counts on the classification performance of MiplGp against the comparing PLL algorithms

	MiplGp against						In total
	Pl-kNN	Clpl	Lsb-cmm	Sure	Gm-pll	Pl-aggd	
$r = 1$	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	60/0/0
$r = 2$	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	60/0/0
$r = 3$	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	10/0/0	59/1/0
In total	30/0/0	29/1/0	30/0/0	30/0/0	30/0/0	30/0/0	179/1/0

Table 4 Classification accuracy (mean ± std) of each comparing algorithm (with one false positive candidate label [$r = 1$])[a)]

Algorithm	MNIST-mipl	FMNIST-mipl	Newsgroups-mipl	Birdsong-mipl	SIVAL-mipl
MiplGp	0.921 ± 0.018	0.806 ± 0.031	0.432 ± 0.018	0.628 ± 0.012	0.599 ± 0.020
Vwsgp	0.402 ± 0.026•	0.422 ± 0.028•	0.098 ± 0.013•	0.250 ± 0.047•	0.050 ± 0.009•
Vgpmil	0.469 ± 0.047•	0.455 ± 0.034•	0.097 ± 0.010•	0.080 ± 0.034•	0.041 ± 0.006•
Lm-vgpmil	0.471 ± 0.021•	0.486 ± 0.036•	0.101 ± 0.008•	0.081 ± 0.042•	0.045 ± 0.008•
Mivae	0.793 ± 0.019•	0.638 ± 0.213	0.135 ± 0.245•	0.067 ± 0.091•	0.068 ± 0.119•

a) •/◦ indicates whether the performance of MiplGp is statistically superior/inferior to the comparing algorithm on each dataset (pairwise t-test at 0.05 significance level).

thus the violation may confuse their disambiguation strategy.

4.2.2 *Comparison with multi-instance learning algorithms*

Most of the existing multi-instance learning algorithms are only designed to solve binary classification problems, and thus are not directly applicable to the MIPL problems.

To make multi-instance learning algorithms fit the MIPL problems, we employ the One vs. Rest (OvR) decomposition strategy. Specifically, given a multi-instance bag $\boldsymbol{X}_i$ associated with a candidate label set $\boldsymbol{y}_i$, we assign each label in the candidate label set to the bag in turn and yield $|\boldsymbol{y}_i|$ multi-instance bags with a single bag-level label. For $c = 1, 2, \ldots, q$, we recompose the label c to 1, i.e., positive, and other labels to 0, i.e., negative. After recomposing all multi-instance bags for the label c, we train and test the c-th classifier. For an unseen multi-instance bag, we can obtain q predictions from the q classifiers. If only one of the predictions is positive, the corresponding class label of the positive prediction is regarded as the classification result of the bag. If the number of positive predictions among the q predictions is greater than one, the class label corresponding to the classifier with the largest prediction confidence is selected as the classification result of the bag. If the predictions of all q classifiers are negative, the classification result is the class label with the lowest prediction confidence.

The computational cost of multi-instance learning algorithms rises with the increase of false positive labels, and the classification accuracy decreases accordingly. We present the classification accuracy of multi-instance learning algorithms with one false positive label in Table 4, which reveals the following.

• It is obvious that MiplGp achieves significantly better performance against the comparing multi-instance learning algorithms in almost all cases.

• Due to the noisy bag-level labels in the degenerated datasets, the comparing multi-instance learning algorithms can learn well in multi-instance learning but cannot effectively work on the MIPL datasets, such as Newsgroups-mipl, Birdsong-mipl, and SIVAL-mipl. This phenomenon indicates that it is necessary to propose tailored algorithms for solving the MIPL problems effectively.

4.3 Further analyses

Exploration of Dirichlet prior α_ϵ. As defined in (2), (3), and (6), the transformed labels are affected by the Dirichlet prior α_ϵ. When α_ϵ approaches 0, the transformed labels and variances of non-candidate labels become negative infinity and positive infinity, respectively, which makes the Gaussian processes regression impossible. To avoid this issue, the Dirichlet prior plays a role in restricting the transformed labels of the non-candidate labels finite. At the same time, the consequential labels and variances of candidate labels are negative values and positive ones that come near to 0. During the iterations, the transformed results of the ground-truth labels are closer to 0 than those of the false positive labels, and the differences between the ground-truth labels and the false positive ones become larger.

The classification accuracy of MiplGp with the varying number of false positive labels $r \in \{1, 2, 3, 4, 5\}$ and the Dirichlet prior $\alpha_\epsilon \in \{0.0001, 0.001, 0.01\}$ is shown in Figure 4. There are several observations:

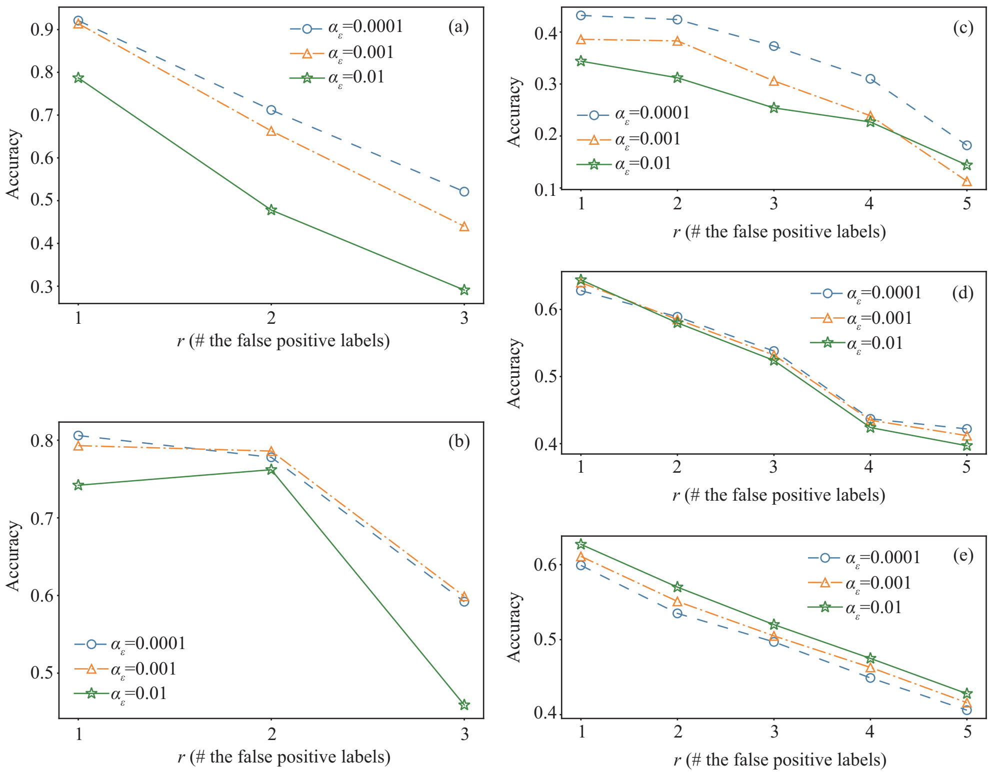

Figure 4 (Color online) Classification accuracy of MIPLGP on the MIPL datasets with varying r and α_ϵ. (a) MNIST-MIPL; (b) FMNIST-MIPL; (c) Newsgroups-MIPL; (d) Birdsong-MIPL; (e) SIVAL-MIPL.

• On MNIST-MIPL and Newsgroups-MIPL datasets, the smaller α_ϵ can achieve better results, while the results are reversed on SIVAL-MIPL dataset.

• On Birdsong-MIPL datasets, the differences between the varying α_ϵ are slight. Similarly, there is no obvious difference between $\alpha_\epsilon = 0.0001$ and $\alpha_\epsilon = 0.001$ on FMNIST-MIPL dataset.

• Different datasets have diverse optimums of the Dirichlet prior, which depend mainly on the characteristics of the MIPL datasets themselves.

In our experiments on MiplGp and its variants, $\alpha_\epsilon = 0.0001$ performs satisfactorily on all datasets.

5 Conclusion

In this paper, we formalize a novel learning framework named multi-instance partial-label learning (MIPL), where each training sample is associated with not only multiple instances but also multiple candidate labels that contain only one ground-truth label. Although the MIPL problems widely exist in many real-world applications, to the best of our knowledge, we are the first to establish a formal MIPL framework and propose the tailored MiplGp algorithm for learning from MIPL data. Specifically, MiplGp transforms the candidate label sets from the augmented label space into a logarithmic space, yielding a Gaussian likelihood and transforming the classification problem into a regression problem. To solve the regression problem, MiplGp induces an efficient Gaussian processes model with GPU accelerations. Extensive comparative studies validate that existing multi-instance and partial-label algorithms are not able to handle the MIPL problems, and MiplGp performs significantly better than other algorithms under the MIPL setting. In the future, there are many directions to explore, for example, exploiting the instance-candidate label dependencies or exploring the theoretical properties of MIPL.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62225602, 62206047). We thank the Big Data Center of Southeast University for providing the facility support on the numerical calculations in this paper. The authors wish to thank anonymous reviewers for their helpful comments and suggestions.

References

1 Zhou Z H. A brief introduction to weakly supervised learning. Natl Sci Rev, 2018, 5: 44–53
2 Amores J. Multiple instance classification: review, taxonomy and comparative study. Artif Intell, 2013, 201: 81–105

3 Carbonneau M A, Cheplygina V, Granger E, et al. Multiple instance learning: a survey of problem characteristics and applications. Pattern Recognition, 2018, 77: 329–353
4 Jin R, Ghahramani Z B. Learning with multiple labels. In: Proceedings of the Advances in Neural Information Processing Systems, Vancouver, 2002. 897–904
5 Li B, Li Y, Eliceiri K W. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In: Proceedings of the 34th IEEE Conference on Computer Vision and Pattern Recognition, Virtual Event, 2021. 14318–14328
6 Zhang H R, Meng Y D, Zhao Y T, et al. DTFD-MIL: double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In: Proceedings of the 35th IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, 2022. 18802–18812
7 Ghadiyaram D, Tran D, Mahajan D. Large-scale weakly-supervised pre-training for video action recognition. In: Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 12046–12055
8 Yang W J, Li C Q, Jiang L X. Learning from crowds with robust support vector machines. Sci China Inf Sci, 2023, 66: 132103
9 Zhou Z H, Zhang M L, Huang S J, et al. Multi-instance multi-label learning. Artif Intell, 2012, 176: 2291–2320
10 Kim Y, Torre F D L. Gaussian processes multiple instance learning. In: Proceedings of the 27th International Conference on Machine Learning, Haifa, 2010. 535–542
11 Haußmann M, Hamprecht F A, Kandemir M. Variational bayesian multiple instance learning with Gaussian processes. In: Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 810–819
12 Zhang W J. Non-I.I.D. multi-instance learning for predicting instance and bag labels with variational auto-encoder. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence, Virtual Event/Montréal, 2021. 3377–3383
13 Zhang W J, Zhang X H, Deng H W, et al. Multi-instance causal representation learning for instance label prediction and out-of-distribution generalization. In: Proceedings of the Advances in Neural Information Processing Systems, Los Angeles, 2022. 1–13
14 Wang J, Li Y H, Pan Y W, et al. Contextual and selective attention networks for image captioning. Sci China Inf Sci, 2022, 65: 222103
15 Ilse M, Tomczak J M, Welling M. Attention-based deep multiple instance learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 2132–2141
16 Shao Z C, Bian H, Chen Y, et al. TransMIL: transformer based correlated multiple instance learning for whole slide image classification. In: Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 2021. 2136–2147
17 Brand L, Baker L Z, Ellefsen C, et al. A linear primal-dual multi-instance SVM for big data classifications. In: Proceedings of the 21st IEEE International Conference on Data Mining, Auckland, 2021. 21–30
18 Cour T, Sapp B, Taskar B. Learning from partial labels. J Mach Learn Res, 2011, 12: 1501–1536
19 Gong C, Liu T L, Tang Y Y, et al. A regularization approach for instance-based superset label learning. IEEE Trans Cybern, 2018, 48: 967–978
20 Yu F, Zhang M L. Maximum margin partial label learning. Mach Learn, 2017, 106: 573–593
21 Feng L, An B. Partial label learning with self-guided retraining. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3542–3549
22 Liu L P, Dietterich T G. A conditional multinomial mixture model for superset label learning. In: Proceedings of the Advances in Neural Information Processing Systems, Cambridge, 2012. 548–556
23 Zhou Y, He J J, Gu H. Partial label learning via Gaussian processes. IEEE Trans Cybern, 2017, 47: 4443–4450
24 Lv J Q, Xu M, Feng L, et al. Progressive identification of true labels for partial-label learning. In: Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 2020. 6500–6510
25 Wang H B, Xiao R X, Li Y X, et al. PiCO: contrastive label disambiguation for partial label learning. In: Proceedings of the 10th International Conference on Learning Representations, Virtual Event, 2022. 1–18
26 Villacampa-Calvo C, Hernández-Lobato D. Scalable multi-class Gaussian process classification using expectation propagation. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 3550–3559
27 Théo G F, Wenzel F, Donner C, et al. Multi-class Gaussian process classification made conjugate: efficient inference via data augmentation. In: Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence, Tel Aviv, 2019. 755–765
28 Liu H T, Ong Y S, Yu Z W, et al. Scalable Gaussian process classification with additive noise for non-Gaussian likelihoods. IEEE Trans Cybern, 2021, 52: 5842–5854
29 Villacampa-Calvo C, Zaldívar B, Garrido-Merchán E C, et al. Multi-class Gaussian process classification with noisy inputs. J Mach Learn Res, 2021, 22: 1696–1747
30 Milios D, Camoriano R, Michiardi P, et al. Dirichlet-based Gaussian processes for large-scale calibrated classification. In: Proceedings of the Advances in Neural Information Processing Systems, Montréal, 2018. 6008–6018
31 Rasmussen C E, Williams C K I. Gaussian Processes for Machine Learning. Cambridge: MIT Press, 2006
32 Wang K A, Pleiss G, Gardner J R, et al. Exact Gaussian processes on a million data points. In: Proceedings of the Advances in Neural Information Processing Systems, Vancouver, 2019. 14622–14632
33 Gardner J R, Pleiss G, Weinberger K Q. Gpytorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration. In: Proceedings of the Advances in Neural Information Processing Systems, Montréal, 2018. 7587–7597
34 LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proc IEEE, 1998, 86: 2278–2324
35 Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017. ArXiv:1708.07747
36 Lang K. NewsWeeder: learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning, Tahoe City, 1995. 331–339
37 Briggs F, Fern X L Z, Raich R. Rank-loss support instance machines for MIML instance annotation. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012. 534–542
38 Settles B, Craven M, Ray S. Multiple-instance active learning. In: Proceedings of the Advances in Neural Information Processing Systems, Vancouver, 2007. 1289–1296
39 Kandemir M, Haußmann M, Diego F, et al. Variational weakly supervised Gaussian processes. In: Proceedings of the 27th British Machine Vision Conference, York, 2016. 1–12
40 Hüllermeier E, Beringer J. Learning from ambiguously labeled examples. Intell Data Analysis, 2006, 10: 419–439
41 Lv G Y, Feng S H, Wang T, et al. GM-PLL: graph matching based partial label learning. IEEE Trans Knowl Data Eng, 2021, 33: 521–535

42 Wang D B, Zhang M L, Li L. Adaptive graph guided disambiguation for partial label learning. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 8796–8811
43 Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, 2015. 1–15
44 Loshchilov I, Hutter F. SGDR: stochastic gradient descent with warm restarts. In: Proceedings of the 5th International Conference on Learning Representations, Toulon, 2017. 1–16

Appendix A The MIPL datasets

We provide the details of the MIPL datasets, i.e., MNIST-MIPL, FMNIST-MIPL, Newsgroups-MIPL, Birdsong-MIPL, and SIVAL-MIPL.

For MNIST-MIPL, FMNIST-MIPL, and Newsgroups-MIPL datasets, we need to choose the targeted class labels and the reserved class labels to provide each multi-instance bag with positive instances and negative ones. For MNIST-MIPL dataset, we extract $\{0, 2, 4, 6, 8\}$ as five target classes for providing the positive instances according to the corresponding class and draw all negative ones from the reserved classes $\{1, 3, 5, 7, 9\}$ randomly. For FMNIST-MIPL dataset, the targeted class labels and the reserved class labels are {T-shirt, Trouser, Coat, Sneaker, Bag} and {Pullover, Dress, Sandal, Shirt, Ankle boot}, respectively. Newsgroups-MIPL, the dataset is widely used in binary multi-instance learning, where each instance is represented by the top 200 TF-IDF features, and each positive bag contains 3% positive instances drawn from the target class. Similarly, we represent each instance by the top 200 TF-IDF features in Newsgroups-MIPL, and Table A1 summarizes the targeted class labels and the reserved class labels of Newsgroups-MIPL dataset.

Table A1 The targeted class labels and reserved class labels of Newsgroups-MIPL dataset

Targeted class labels	Reserved class labels	Targeted class labels	Reserved class labels
alt.atheism	comp.graphics	rec.sport.hockey	sci.crypt
comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware	sci.med	sci.electronics
comp.sys.mac.hardware	comp.windows.x	sci.space	talk.politics.guns
misc.forsale	rec.motorcycles	soc.religion.christian	talk.politics.misc
rec.autos	rec.sport.baseball	talk.politics.mideast	talk.religion.misc

The Birdsong dataset is proposed in multi-instance multi-label learning, which contains 548 multi-instance bags totaling 10232 instances. Each instance is represented by a 38-dimensional feature vector and associated with a single label, which is chosen from 13 targeted class labels or 1 negative class labels. In Birdsong-MIPL, the 13 targeted class labels and the negative class label are regarded as the targeted class labels and the reserved class label, respectively.

SIVAL is a multi-instance learning dataset for content-based image retrieval with 1500 images. Each image is a multi-instance bag, which is associated with one of the 25 class labels and consists of 31 or 32 instances. In addition, each instance is represented by a 30-dimensional feature vector. To yield the SIVAL-MIPL, we only need to generate the false positive labels for each image. Specifically, we treat the 25 class labels as the targeted class labels and sample r false positive labels from the targeted class labels excluding the ground-truth label randomly.