Exploiting Conjugate Label Information for Multi-Instance Partial-Label Learning

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 Lab. of Computer Network and Information Integration (Southeast University), MoE, China
$^3$School of Information and Physical Sciences, The University of Newcastle, NSW 2308, Australia
tangw@seu.edu.cn, weijia.zhang@newcastle.edu.au, zhangml@seu.edu.cn

Abstract

Multi-instance partial-label learning (MIPL) addresses scenarios where each training sample is represented as a multi-instance bag associated with a candidate label set containing one true label and several false positives. Existing MIPL algorithms have primarily focused on mapping multi-instance bags to candidate label sets for disambiguation, disregarding the intrinsic properties of the label space and the supervised information provided by non-candidate label sets. In this paper, we propose an algorithm named ELIMIPL, i.e., Exploiting conjugate Label Information for Multi-Instance Partial-Label learning, which exploits the conjugate label information to improve the disambiguation performance. To achieve this, we extract the label information embedded in both candidate and non-candidate label sets, incorporating the intrinsic properties of the label space. Experimental results obtained from benchmark and real-world datasets demonstrate the superiority of the proposed ELIMIPL over existing MIPL algorithms and other well-established partial-label learning algorithms.

1 Introduction

Weakly supervised learning has emerged as a powerful strategy in scenarios with limited annotated data. Based on label quality and quantity, weak supervision can be broadly categorized into three types: inaccurate, inexact, and incomplete supervision [Zhou, 2018]. Inexact supervision refers to a coarse correspondence between instances and labels. To work with inexact supervision, these are two prevalent learning paradigms, i.e., multi-instance learning (MIL) [Amores, 2013; Carbonneau et al., 2018; Ilse et al., 2018; Zhang et al., 2022c; Zhang et al., 2022b] and partial-label learning (PLL) [Cour et al., 2011; Lyu et al., 2020; Zhang et al., 2022a; He et al., 2022; Gong et al., 2022; Li et al., 2023]. In MIL, a sample is represented as a bag of instances and associated with a single bag-level label, while the instance-level labels are inaccessible to the learner. In PLL, a sample is represented as a single instance and linked to a candidate label set, including one true label and multiple false positives. Therefore, MIL and PLL can be perceived as two sides of the same coin: inexact supervision within MIL manifests in the instance space, whereas inexact supervision appears in the label space within PLL.

However, many tasks exhibit a phenomenon of dual inexact supervision, where ambiguity arises in both instance and label spaces. To work with the dual inexact supervision, Tang et al. [2024] introduced a learning paradigm known as multi-instance partial-label learning (MIPL) and developed a Gaussian Processes-based algorithm (MIPLGP), which derives a bag-level predictor by aggregating predictions of all instances within the same bag. To capture global representations for multi-instance bags, an algorithm named DeMIPL equipped with an attention mechanism is introduced [Tang et al., 2023]. The existing algorithms mainly operate in the instance space and only utilize the candidate label information.

The non-candidate label set holds crucial roles in MIPL. In histopathological image classification, images are commonly segmented into patches [Campanella et al., 2019], and their labels may come from crowd-sourced annotators rather than expert pathologists [Grote et al., 2019]. Figure 1(a) illustrates that crowd-sourced annotators treat an image as a multi-
instance bag \( X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,G}\} \) and provide a candidate label set \( S_i = \{k_1, k_2, k_3, k_4\} \), whose candidate label matrix can be written as \( S_i = [1, 1, 0, 0, 1, 0, 1] \). Similarly, the non-candidate label set \( \tilde{S}_i = \{k_3, k_4, k_5\} \) corresponds to the non-candidate label matrix \( \tilde{S}_i = [0, 0, 1, 1, 0, 1, 0] \), indicating that \( X_i \) must not belong to categories \( k_3, k_4, \) or \( k_5 \). Therefore, we can extract exact supervision from the non-candidate label set. As depicted in Figure 1(b), we decompose a complete label matrix \( Y \) into a candidate label matrix \( S \) and a non-candidate label matrix \( \tilde{S} \). Subsequently, \( S \) may be further disintegrated into a false positive label matrix \( Y_T \) and a true label matrix \( Y_T \), i.e., \( Y = S + \tilde{S} = Y_T + Y_T + \tilde{S} \). Notably, \( Y_T \) is sparse, as each row must have one and only one non-zero element. However, the current MIPL algorithms have predominantly concentrated on the mappings from multi-instance bags to \( S \), neglecting the sparsity of \( Y_T \) and the information from \( \tilde{S} \).

![Figure 2: Predicted probabilities of DEMIPL (left) and EL1MIPL (right) on the sample in CRC-MIPL-Row dataset.](image)

Consequently, Figure 2 illustrates the predicted probabilities on the true label, along with the average predicted probabilities on each candidate label and non-candidate label. The left side depicts the probabilities of the DEMIPL, revealing proximity in the average predicted probabilities on candidate and non-candidate labels. This observation indicates that DEMIPL encounters difficulty in effectively discerning between candidate and non-candidate labels. To address this challenge, we introduce the concept of conjugate label information (CLI), encapsulating information from both candidate and non-candidate label sets, along with the sparsity of the true label matrix. The right side in Figure 2 shows the predicted probabilities when exploiting the CLI. It is evident that (a) the predicted probabilities on the true label exhibit a noticeable increase, (b) the average predicted probabilities on the non-candidate label are reduced, and (c) the average probabilities on each candidate label and non-candidate label are distinctly separated. This suggests that the CLI conduce to train a more discriminative MIPL classifier.

In this paper, we present an algorithm named EL1MIPL, i.e., *Exploiting conjugate Label Information for Multi-Instance Partial-Label learning*. Firstly, we introduce a scaled additive attention mechanism to aggregate each multi-instance bag into a bag-level feature representation. Secondly, to enhance the utilization of candidate label information, we leverage the mappings from the bag-level features to the candidate label sets, coupled with the sparsity of the candidate label matrix. Lastly, to incorporate the non-candidate label information, we propose an inhibition loss to diminish the model’s predictions on the non-candidate labels. To the best of our knowledge, we are the first to introduce the scaled additive attention mechanism and the CLI in MIPL. Extensive experimental results demonstrate that EL1MIPL outperforms the state-of-the-art MIPL algorithms and the PLL algorithms.

The remainder is organized as follows. Firstly, we review related work in Section 2. Secondly, we present the proposed EL1MIPL in Section 3 and report the experimental results in Section 4. Lastly, we conclude this paper in Section 5.

## 2 Related Work

### 2.1 Multi-Instance Learning

Originating from drug activity prediction [Dietterich et al., 1997], MIL has found extensive adoption in diverse applications, including text classification [Zhou et al., 2009; Zhang, 2021] and image annotation [Wang et al., 2018]. Contemporary deep MIL approaches predominantly rely on attention mechanisms [Wang et al., 2022b; Chen et al., 2022; Tan et al., 2023]. Ise et al. [2018] introduced attention mechanisms to aggregate each multi-instance bag into a feature vector. For multi-classification tasks, Shi et al. [2020] proposed a loss-based attention mechanism to learn instance-level weights, predictions, and bag-level predictions. Furthermore, researchers have explored the intrinsic attributes of attention mechanisms to improve performance [Cui et al., 2023; Xiang et al., 2023]. While these approaches achieve promising results in cases with exact bag-level labels, they face challenges in learning from ambiguous bag-level labels.

### 2.2 Partial-Label Learning

Recent PLL approaches heavily rely on deep learning techniques. Yao et al. [2020] employed deep convolutional neural networks for feature extraction and utilized the exponential moving average technique to uncover latent true labels. Building on the empirical risk minimization principle, Lv et al. [2020] devised a classifier-consistent risk estimator that progressively identifies true labels. Similarly, Feng et al. [2020] delved into the generation process of partial-labeled data, proposing both a risk-consistent approach and a classifier-consistent approach. Taking a more generalized stance, Wen et al. [2021] presented a weighted loss function capable of accommodating various methods through distinct weight assignments. Furthermore, Wu et al. [2022] proposed a supervised loss to constrain outputs on non-candidate labels, coupled with consistency regularization on candidate labels. While the supervised loss bears resemblance to our inhibition loss, our proposed CLI loss incorporates additional components, namely the mapping loss and the sparse loss. Although these methods effectively learn from partial-labeled data, they lack the capability to manage multi-instance bags.

### 2.3 Multi-Instance Partial-Label Learning

In contrast to the inherent limitations of addressing only unilateral inexact supervision in MIL and PLL, MIPL possesses the capability to work with dual inexact supervision. To the best of our knowledge, there are only two viable MIPL algorithms. Tang et al. [2024] is the first to introduce the framework of MIPL along with a Gaussian processes-based algorithm (MIPLGp), which follows an instance-space paradigm. MIPLGp begins by augmenting a negative class...
for each candidate label set, subsequently treating the candidate label set of each multi-instance bag as that of each instance within the bag. Finally, it employs the Dirichlet disambiguation strategy and the Gaussian processes regression model for disambiguation. Differing from MIPLGP, DEMIPL follows the embedded-space paradigm and aggregates each multi-instance bag into a feature representation and employs a momentum-based disambiguation strategy to find true labels from candidate label sets [Tang et al., 2023]. However, both methods primarily depend on mapping from instances or multi-instance bags to candidate label sets for disambiguation, without considering the proposed CLI in this paper.

3 Methodology

3.1 Preliminaries

In this study, we define a MIPL training dataset as \( D = \{(X_i, S_i) \mid 1 \leq i \leq m\} \), comprising \( m \) multi-instance bags and their corresponding candidate label sets. Specifically, a candidate label set \( S_i \) consists of one true label and multiple false positive labels, but the true label is unknown. It is crucial to note that a bag contains at least one instance pertaining to the true label, while excluding any instances corresponding to false positive labels. The instance space is denoted as \( \mathcal{X} \subset \mathbb{R}^d \), while the label space \( \mathcal{Y} = \{1, 2, \cdots, k\} \) encompasses \( k \) class labels. The \( i \)-th bag \( X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}\} \) comprises \( n_i \) instances of dimension \( d \). Both the candidate label set \( S_i \) and the non-candidate label set \( \overline{S_i} \) are proper subsets of the label space \( \mathcal{Y} \), satisfying the conditions \( |S_i| + |\overline{S_i}| = |\mathcal{Y}| = k \), where \( |\cdot| \) denotes the cardinality of a set.

The pipeline of the proposed E\textsc{LI}\textsc{MIPL} is depicted in Figure 3, which contains three main components: an instance-level feature extractor, a scaled additive attention mechanism, and a classifier. When presented with a multi-instance bag \( X_i \) along with its associated candidate label set \( S_i \) and non-candidate label set \( \overline{S_i} \), we initially employ a feature extractor to procure instance-level feature representations. Subsequently, the scaled additive attention mechanism is applied to aggregate a bag of instances into a unified bag-level feature representation. Finally, the classifier is invoked to estimate the class probabilities based on the bag-level features. To utilize the CLI, we introduce a mapping loss \( \mathcal{L}_{ma} \) and a sparsity loss \( \mathcal{L}_{sp} \) to disambiguate the candidate label sets, along with an inhibition loss \( \mathcal{L}_{in} \) to suppress the model’s prediction over the non-candidate label sets.

3.2 Instance-Level Feature Extractor

For a given multi-instance bag \( X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}\} \) with \( n_i \) instances, instance-level feature representations \( H_i \) are learned using a feature extractor \( \psi(\cdot) \) as follows:

\[
H_i = \psi(X_i) = \{h_{i,1}, h_{i,2}, \cdots, h_{i,n_i}\},
\]

where \( h_{i,j} \in \mathbb{R}^l \) indicates the feature representation of the \( j \)-th instance within the \( i \)-th multi-instance bag, and \( \psi(\cdot) \) is a neural network comprised of two components, i.e.,

\[
\psi_1(X_i) = \psi_2(\psi_1(X_i)).
\]

Here, \( \psi_1(\cdot) \) is a feature extractor that can be tailored to the specific characteristics of the datasets, and \( \psi_2(\cdot) \) is composed of fully connected layers that map instance-level features to an embedded space of dimension \( l \).

3.3 Scaled Additive Attention Mechanism

To aggregate instance-level features into bag-level representations, we introduce a scaled additive attention mechanism specifically designed for MIPL. The existing attention mechanism for MIPL utilizes the sigmoid function for calculating attention scores, followed by normalization [Tang et al., 2023]. The attention scores derived through the sigmoid function are constrained within the range \((0, 1)\), leading to a limited distinction between instances. Therefore, we introduce an additive attention mechanism calculating attention scores by the softmax function to distinguish instances, equipped with a scaling factor to prevent vanishing gradients [Vaswani et al., 2017]. Specifically, we first denote the output of the additive attention mechanism as \( \xi(h_{i,j}) \), quantifying the impact of the \( j \)-th instance on the \( i \)-th bag as follows:

\[
\xi(h_{i,j}) = W^T \tanh(W_i^T h_{i,j} + b_i) \odot \text{sigm}(W_s^T h_{i,j} + b_s),
\]

where \( W^T, W_i^T, W_s^T, b_i, \) and \( b_s \) are learnable parameters, \( \tanh(\cdot) \) and \( \text{sigm}(\cdot) \) are the hyperbolic tangent and sigmoid functions, respectively. The operator \( \odot \) denotes element-wise multiplication. Then, we normalize \( \xi(h_{i,j}) \) using softmax.
with a scaling factor $1/\sqrt{t}$ to derive the attention score:

$$a_{i,j} = \frac{\exp \left( \xi (h_{i,j}) / \sqrt{t} \right)}{\sum_{j'=1}^{n_i} \exp \left( \xi (h_{i,j'}) / \sqrt{t} \right)},$$  \hspace{1cm} (3)

where $a_{i,j}$ represents the attention score of the $j$-th instance in the $i$-th bag. Finally, we consolidate the instance-level features into a bag-level representation, as demonstrated below:

$$z_i = \sum_{j=1}^{n_i} a_{i,j} h_{i,j},$$  \hspace{1cm} (4)

where $z_i$ represents the bag-level representation of the $i$-th multi-instance bag. The bag-level representations of all multi-instance bags in the training dataset are denoted by $Z$.

### 3.4 Conjugate Label Information

**Candidate Label Information.** Once the bag-level feature representations are acquired, the subsequent task is to disambiguate the candidate label set. The disambiguation entails establishing the mapping relationship from the bag-level features to their corresponding candidate label set. The goal of precise mapping is to guide the classifier to assign higher class probabilities to true labels and lower probabilities to false positive labels. To attain this objective, we employ a weighted mapping loss function:

$$L_{ma}(Z,S) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c \in S_i} w_{i,c}^{(t)} \log(f_c(z_i)),$$  \hspace{1cm} (5)

where $f$ is the classifier, and $f_c(\cdot)$ represents the classifier’s prediction probability for the candidate label $c$. $w_{i,c}^{(t)}$ denotes the weight assigned to the prediction of the $c$-th class at the $t$-th epoch, using the features of the $i$-th bag as input for the classifier. For candidate labels, we initialize $w_{i,c}^{(0)} = \frac{1}{|S_i|}$ through an averaging approach. During training, we update $w_{i,c}^{(t)}$ by computing a weighted sum of the classifier’s outputs at both the previous epoch and current epoch as follows:

$$w_{i,c}^{(t)} = \rho^{(t)} w_{i,c}^{(t-1)} + (1 - \rho^{(t)}) \frac{f_c(z_i)}{\sum_{c' \in S_i} f_{c'}(z_i)},$$  \hspace{1cm} (6)

where $\rho^{(t)} = (T-t)/T$ is dynamically adjusted across epochs, and $T$ is the maximum of the training epochs.

While the mapping loss can assess the relative labeling probabilities of candidate labels, it fails to capture the mutually exclusive relationships among the candidate labels. To address this issue in PLL, Feng et al. [2019] introduced the maximum infinity norm on the predicted probabilities of all classes and alternately optimize the maximum infinity norm by solving $k$ independent quadratic programming problems. However, as depicted in Figure 1(b), we observe that each row of the true label matrix exhibits sparsity. Although the true labels remain inaccessible during the training process, we encourage the classifier to generate sparse prediction probabilities for the candidate labels. Specifically, the goal is to push the prediction probability of the unknown true label toward 1 while simultaneously driving the prediction probabilities of other candidate labels toward 0. Therefore, we directly capture the mutually exclusive relationships among the candidate labels by implementing the sparsity loss, as detailed below:

$$L_{sp}(S) = \frac{1}{m} \sum_{i=1}^{m} \left\| P_i \odot S_i \right\|_0,$$  \hspace{1cm} (7)

where $P_i$ and $S_i$ is the prediction probabilities and the candidate label set matrix of the $i$-th bag, respectively. $\odot$ denotes element-wise multiplication. Since minimizing the $\ell_0$ norm is NP-hard, we employ the $\ell_1$ norm as a surrogate for the $\ell_0$ norm, promoting sparsity while allowing for efficient optimization [Wright and Ma, 2022].

**Non-candidate Label Information.** For a multi-instance bag $X_i$, linked to a candidate label set $S_i$, the non-candidate label set $S_i'$ complements the candidate label set $S_i$ within the label space $Y$. As the label space has a fixed size, an antagonistic relationship arises between the non-candidate and candidate label sets. To enhance the classifier’s prediction probabilities for the candidate label set, a natural strategy is to diminish the classifier’s prediction probabilities for the non-candidate label set. Motivated by this insight, we introduce an inhibition loss as follows:

$$L_{in}(Z,S) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{c \in S_i'} \log(1 - f_c(z_i)),$$  \hspace{1cm} (8)

where $f_c(\cdot)$ denotes the classifier’s prediction probability over the non-candidate label $\bar{c}$.

**CLI Loss.** During the training, CLI is formed by a loss function named CLI loss that is a weighted fusion of the mapping loss, sparsity loss, and inhibition loss, as shown below:

$$L = L_{ma}(Z,S) + \mu L_{sp}(S) + \gamma L_{in}(Z,S),$$  \hspace{1cm} (9)

where $\mu$ and $\gamma$ represent the weighting coefficients for the sparsity loss and the inhibition loss, respectively.
Algorithm 1 summarizes the training procedure of ELIMPL. Firstly, the algorithm initializes the weights for the mapping loss uniformly (Step 1). Subsequently, instance-level features are extracted and aggregated into bag-level features within each mini-batch (Steps 5-8). The algorithm then updates the weights for the mapping loss and calculates the total loss function (Steps 9-11). Finally, the model is optimized using gradient descent (Steps 12 and 13).

4 Experiments

In this section, we begin by introducing the experimental configurations, including the datasets, comparative algorithms, and the parameters used in the experiments. Subsequently, we present the experimental results on both benchmark and real-world datasets. Finally, we conduct further analysis to gain deeper insights into the impact of CLI.

4.1 Experimental Configurations

Datasets. We employ four benchmark MIPL datasets [Tang et al., 2024; Tang et al., 2023]: MNIST-MIPL, FMNIST-MIPL, Birdsong-MIPL, and SIVAL-MIPL, spanning diverse domains such as image analysis and biology [LeCun et al., 1998; Xiao et al., 2017; Briggs et al., 2012; Settles et al., 2007]. The characteristics of the datasets are presented in Table 1, where the abbreviations within parentheses in the first column represent the abbreviated names of the MIPL datasets. The dataset includes quantities of multi-instance bags and total instances, denoted as \#bag and \#ins, respectively. Additionally, we use max. \#ins, min. \#ins, and avg. \#ins to indicate the maximum, minimum, and average instance count within all bags. The dimensionality of the instance-level feature is represented by \#dim. Labeling details are elucidated using \#class and avg. \#CLS, signifying the length of the label space and the average length of candidate label sets, respectively. For a comprehensive performance assessment, we vary the count of false positive labels, denoted as \( r \) (\(|S| = r + 1\)).

CRC-MIPL dataset is a real-world MIPL dataset for colorectal cancer classification. We utilize multi-instance features generated by four image bag generators [Wei and Zhou, 2016]: Row [Maron and Ratan, 1998], single blob with neighbors (SBN) [Maron and Ratan, 1998], k-means segmentation (KMeansSeg) [Zhang et al., 2002], and scale-invariant feature transform (SIFT) [Lowe, 2004].

The appendix contains detailed information about the datasets and the four image bag generators.

Comparative Algorithms. We conduct a comprehensive comparison involving ELIMPL along with two established MIPL algorithms: MIPLGP [Tang et al., 2024] and DEMIPL [Tang et al., 2023]. These represent the entirety of available MIPL methods. Furthermore, we include four PLL algorithms: PRODEN [Lv et al., 2020], RC [Feng et al., 2020], LWS [Wen et al., 2021], and PL-AGGD [Wang et al., 2022a]. The first three algorithms can be equipped with diverse backbone networks, such as linear models and MLP. Due to spatial constraints, we present the results obtained from the linear models in the main body, while the results with MLP are shown in the appendix. Parameters for all algorithms are selected based on recommendations from original literature or refined through our search for enhanced outcomes.

Since PLL algorithms are not directly tailored for MIPL data, two common strategies, known as the Mean strategy and the MaxMin strategy, are employed to adapt MIPL data for PLL algorithms [Tang et al., 2024]. The Mean strategy involves calculating average feature values across all instances within a bag, resulting in a bag-level feature representation. In contrast, the MaxMin strategy identifies both the maximum and minimum feature values for each dimension among instances within a bag, and then concatenates these values to form a bag-level feature representation.

Implementation. We implement ELIMPL using PyTorch and execute it on a single NVIDIA Tesla V100 GPU. We utilize the stochastic gradient descent (SGD) optimizer with a momentum value of 0.9 and a weight decay of 0.0001. The initial learning rate is selected from the set \{0.01, 0.05\} and accompanied by a cosine annealing technique. We set the number of epochs uniformly to 100 for all datasets. For the MNIST-MIPL and FMNIST-MIPL datasets, \( \mu \) is set to 1 or 0.1; \( \gamma \) is chosen from \{0.1, 0.5\}, and the feature extraction network \( \psi_1(\cdot) \) is a two-layer convolutional neural network. For the remaining datasets, we set both \( \mu \) and \( \gamma \) to 10, and \( \psi_1(\cdot) \) is an identity transformation. The feature transformation network \( \psi_2(\cdot) \) is implemented by a fully connected network, with the dimension \( l \) set to 512 for the CRC-MIPL dataset and 128 for the other datasets. The way of dataset partitioning is consistent with that of DEMIPL. We conduct ten random train/test splits with a ratio of 7 : 3. We report the mean accuracies and standard deviations obtained from the ten runs, with the highest accuracy highlighted in bold. The code of ELIMPL can be found at https://github.com/tang-w-seu/ELIMPL.

4.2 Results on the Benchmark Datasets

Table 2 presents the results of ELIMPL and the comparative algorithms on benchmark datasets, considering varying numbers of false positive labels (\( r \in \{1, 2, 3\} \)). Compared to MIPL algorithms, ELIMPL consistently achieves higher

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Bag</th>
<th>max. #ins</th>
<th>min. #ins</th>
<th>avg. #ins</th>
<th>#dim</th>
<th>#class</th>
<th>avg. #CLS</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST-MIPL (MNIST)</td>
<td>500</td>
<td>204664</td>
<td>48</td>
<td>35</td>
<td>41.31</td>
<td>784</td>
<td>5</td>
<td>2, 3, 4 image</td>
</tr>
<tr>
<td>FMNIST-MIPL (FMNIST)</td>
<td>500</td>
<td>20810</td>
<td>48</td>
<td>36</td>
<td>41.62</td>
<td>784</td>
<td>5</td>
<td>2, 3, 4 image</td>
</tr>
<tr>
<td>Birdsong-MIPL (Birdsong)</td>
<td>1300</td>
<td>48425</td>
<td>76</td>
<td>25</td>
<td>37.25</td>
<td>38</td>
<td>13</td>
<td>2, 3, 4 biology</td>
</tr>
<tr>
<td>SIVAL-MIPL (SIVAL)</td>
<td>1500</td>
<td>47414</td>
<td>32</td>
<td>31</td>
<td>31.61</td>
<td>30</td>
<td>25</td>
<td>2, 3, 4 image</td>
</tr>
<tr>
<td>CRC-MIPL-Row (C-Row)</td>
<td>7000</td>
<td>56000</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>2.08</td>
<td>image</td>
</tr>
<tr>
<td>CRC-MIPL-SBN (C-SBN)</td>
<td>7000</td>
<td>65000</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>15</td>
<td>7</td>
<td>2.08 image</td>
</tr>
<tr>
<td>CRC-MIPL-KMeansSeg (C-KMeans)</td>
<td>7000</td>
<td>30178</td>
<td>6</td>
<td>3</td>
<td>4.311</td>
<td>6</td>
<td>7</td>
<td>2.08 image</td>
</tr>
<tr>
<td>CRC-MIPL-SIFT (C-SIFT)</td>
<td>7000</td>
<td>175000</td>
<td>25</td>
<td>25</td>
<td>128</td>
<td>7</td>
<td>2.08</td>
<td>image</td>
</tr>
</tbody>
</table>
comparative algorithms on the benchmark datasets with varying
IPL
highlighting that they cannot be reduced to PLL problems.

on the Birdsong-
MIPL
LI
tably, D
E
consistently achieves the highest average accuracies in both fewer and
dataset spans
055
. The SIV AL-
P
passes D
E
13
comprises
IPL
higher than D
E
5
FMNIST-
MIPL
LI
achieves an average accuracy at least
M
achieves an average accuracy at least
P
at least
M
achieves an average accuracy at least
IPL
across diverse label space but also emphasizes the limitations of addressing MIPL problems using PLL algorithms. The results underscore the importance of algorithmic designs specifically tailored to MIPL tasks.

4.3 Results on the Real-World Datasets

Table 3: The classification accuracies (mean±std) of ELIMIPL and comparative algorithms on the real-world datasets.

Table 2: The classification accuracies (mean±std) of ELIMIPL and comparative algorithms on the benchmark datasets with varying numbers of false positive candidate labels (r ∈ {1, 2, 3}).
Figure 4 presents the experimental results of ELIMPI, DEIMPI, and MIPLGP with $r \in \{1, 2, \cdots, 10\}$. Particularly, ELIMPI and DEIMPI adhere to the embedded-space paradigm, while MIPLGP follows the instance-space paradigm. Three distinct phenomena are observed: (a) ELIMPI consistently exhibits higher average accuracy compared to DEIMPI and MIPLGP. (b) For $r < 7$, DEIMPI outperforms MIPLGP. However, when $r \geq 7$, MIPLGP surpasses DEIMPI. (c) The gaps between ELIMPI and DEIMPI are greater when $r \in \{6, 7, 8, 9, 10\}$ than when $r \in \{1, 2, 3, 4, 5\}$. The widening gaps signify the growing significance of the scaled additive attention mechanism and CLI. Therefore, Figure 4 clearly demonstrates that ELIMPI outperforms both MIPLGP and DEIMPI in disambiguation, even when confronted with challenging scenarios.

Comparison of CLI and Cross-Entropy Loss. For a comparative analysis between the CLI loss and the cross-entropy (CE) loss, we substitute the mapping loss and the CLI loss with the CE loss, resulting in variants CE-SP-IN (which utilizes CE loss, sparsity loss, and inhibition loss) and CE (which only utilizes CE loss), respectively. Table 5 illustrates that, in all cases, accuracies obtained with the CLI loss surpass those achieved with CE-SP-IN and CE. Notably, the incorporation of inhibition loss and sparsity loss enhances the performance of the CE loss, underscoring the importance of considering the intrinsic properties of the label space and the information from non-candidate label sets.

Interpretability of the Attention Mechanism. Figure 5 displays the attention scores of a test multi-instance bag in the MNIST-MIPL dataset ($r = 1$). The bag contains positive instances represented by the digit 6, while negative instances are drawn from digits $\{1, 3, 5, 7, 9\}$. Additionally, we visualize the attention scores of all three positive instances and the negative instances. Figure 5 illustrates that ELIMPI can accurately identify all positive instances by assigning significantly higher attention scores to them, and the attention scores can be directly utilized for interpretability.

5 Conclusion

This paper investigates a multi-instance partial-label learning algorithm that introduces a scaled additive attention mechanism and exploits conjugate label information. This information includes both candidate label information and non-candidate label information, along with the sparsity of the true label matrix. Experimental results demonstrate the superiority of our proposed ELIMPI algorithm. The utilization of CLI proves significantly more effective than relying on incomplete label information, especially in challenging disambiguation scenarios. In the future, we will explore the instance-depend MIPL algorithm and conduct theoretical analyses to develop more effective algorithms.
Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China (62176055), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX23_0317), and the Big Data Computing Center of Southeast University.

References


[Shi et al., 2020] Xiaoshuang Shi, Fuyong Xing, Yuanpu Xie, Zizhao Zhang, Lei Cui, and Lin Yang. Loss-based


