Multi-Instance Embedding Learning through High-Level Instance Selection

Mei Yang[1][0000−0003−1347−1963], Wen-Xi Zeng[1][0000−0001−5652−3087], and Fan Min[1,2][0000−0002−3290−1036]

[1] School of Computer Science, Southwest Petroleum University, Chengdu, China
[2] Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu, China
yangmei@swpu.edu.cn, zwx971206@163.com, minfan@swpu.edu.cn

Abstract. Multi-instance learning (MIL) handles complex structured data represented by bags and their instances. MIL embedded algorithms based on representative instance selection transform bags into a single-instance space. However, they may select weak representative instances due to the ignorance of the internal bag structure. In this paper, we propose the multi-instance embedding learning through high-level instance selection (MIHI) algorithm with two techniques. The fast bag-inside instance selection technique obtains instance prototypes of each bag. It fully utilizes the bag information using our new density and affinity metrics. Based on the instance prototypes, the high-level instance selection technique chooses instances using the peak density metric. It obtains high-level instances with higher representative power than the instance prototypes. Experiments were conducted on six learning tasks and nine comparison algorithms. The results confirmed that MIHI achieved better performance in terms of efficiency and classification accuracy. This method, in particular, has a substantial advantage in image retrieval and web data sets.

Keywords: Embedding · High-level instance · Instance selection · Multi-instance learning

1 Introduction

Compared with traditional single-instance learning (SIL), multi-instance learning (MIL) is the study of bags containing multiple instances. Taking the drug activity prediction as an example, molecules and their isomers are viewed as bags and instances, respectively. The task is to predict whether the new molecule is suitable for making drugs. A molecule is positive if at least one of its isomers can be used to make drugs, otherwise it is negative. Furthermore, multi-instance problems are common in real-world application scenarios, such as image retrieval [2], text classification [21], and web index recommendation [14].

In recent years, many embedded MIL algorithms based on instance prototypes have been widely proposed. Their common strategies tend to perform clustering in the entire instance space to select instance prototypes [10,15]. MILFM

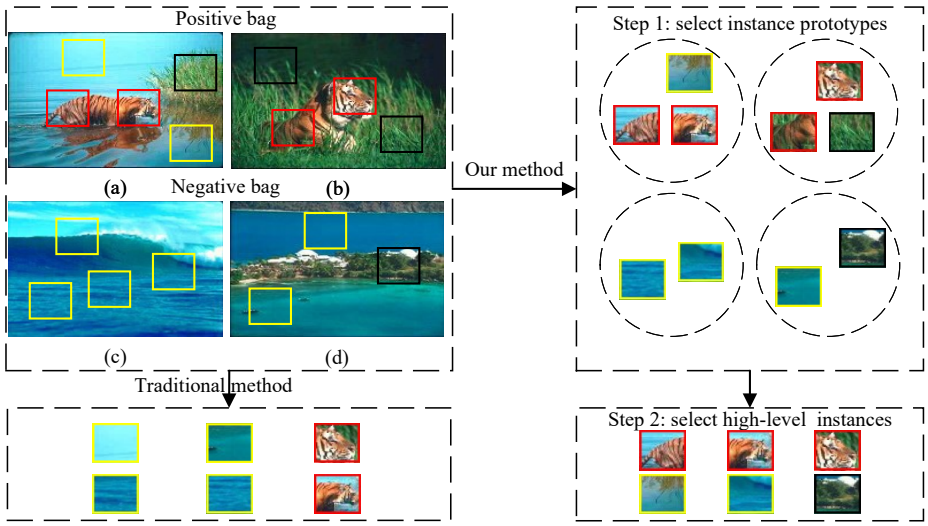

Fig. 1. The main framework of MIHI is compared with traditional methods. Traditional methods usually use clustering algorithms to select instance prototypes in the entire data space. Our method first selects the instance prototypes from the bag, and then selects the high-level instances from the instance prototypes.

[10] first selects instance prototypes in the entire instance space, and selects cluster instances from the negative bags. CMIL [9] only divides the instances of the positive bag into multiple clusters, and selects the instances with the high score in each cluster as the instance prototypes. However, two dilemmas will be encountered: 1) The cardinality of the instance space is much larger than that of the bag space; and 2) The number of negative instances is far greater than that of positive instances. As a result, the classification effectiveness may be reduced. Figure 1 shows an example of tiger image classification task. In subgraphs (a) and (b), there are tigers, grass and water. In subgraphs (c) and (d), there are only grass and water. Obviously, grass and water occupy a large proportion of the entire feature space. The instances prototypes chosen by traditional methods are more likely to be grass and water than tigers. However, the selected instances have weak representativeness due to ignoring the internal structure of the bag. Therefore, the selection of highly representative instance prototypes is the key to the embedded MIL algorithms.

In this paper, we propose the multi-instance embedding learning through high-level instance selection (MIHI) algorithm to handle these issues with two techniques. Figure 1 shows the main framework of MIHI. The goal is to select high-level instances with strong representativeness. In Step 1, the fast bag-inside instance selection technique is designed to select instance prototypes from each bag. This technique takes into account the density and affinity of instances in the bag. The instance prototypes highlight the bag's internal structure information. Accordingly, the high-level instance selection technique chooses global represen-

tative instances. For each instance prototype, we calculate its local density and the minimum distance from higher-density prototypes. Then the instance prototypes with peak density are identified as the high-level instances. Experiments on six learning tasks confirmed the effectiveness of MIHI in terms of efficiency and classification accuracy. The main contributions of our work are:

- We propose a fast bag-inside instance selection technique, which can effectively exploit the structure information of the bag. By using new density and affinity metrics, the instance prototypes of the bag are found.
- We propose a high-level instance selection technique based on instance prototypes. Through peak density metric, the high-level instances have more representative power than other prototypes.

2 Related work

MIL was first proposed in the study of drug activity prediction [7]. After that, many MIL algorithms have been proposed. They are mainly divided into two categories: 1) Basic methods predict the bag label based on the structural characteristics of bag [21] or instance [8] spaces; and 2) Embedding methods transform MIL into SIL based on representative samples [3,17].

The basic methods mainly handle MIL problems by designing a bag-level kernel. mi-SVM and MI-SVM [2] treat bags as samples and use support vector machines to handle problems. mi-SVM tries to identify the maximum edge hyperplane for the instances. Its constraint is that at least one instance of each positive bag is located in the positive half space. MI-SVM treats the edge of the most positive instance as the edge of the bag. The purpose is to identify the maximum edge hyperplane of the bag. miGraph [21] proposes an effective bag-level kernel through the affinity matrix. However, it only focuses on the relationship between bags and fails to extract instance-level information.

The bag embedding methods deal with MIL problems by transforming the space. DD-SVM [4] learns a set of instance prototypes by using Diverse Density. Then the bags are embedded into the new feature space based on the instance prototypes. MILES [3] uses a joint strategy based on all instances to implement bag embedding. Bamic [22] selects the representative bags through unsupervised learning. MIKI [19] first trains a weighted multi-class model to select instance prototypes with high positiveness. Then the bag is converted into a vector with instance prototype information. To narrow the gap between the training and testing distribution, the weights of the instance prototypes are combined into the converted bag vector. However, these algorithms directly select instance prototypes in the entire feature space, ignoring the internal structure of the bag. As a result, they may choose weakly representative instances and affect classification performance. MIHI provides a solution for selecting high-level instances through two techniques.

3 The proposed algorithm

In this section, we first give the basic symbol definition of MIL. Then we describe the proposed MIHI algorithm process. Furthermore, two key techniques of MIHI are described in detail.

3.1 Algorithm description

Algorithm 1 reports the detailed process of the proposed MIHI. Let $\mathcal{T} = \{\boldsymbol{B}_i\}_{i=1}^{N}$ be the MIL data set with N bags, where $\boldsymbol{B}_i = \{\boldsymbol{x}_{ij}\}_{j=1}^{n_i}$ is a bag containing n_i instances. Let $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ be the label vector corresponding to $\mathcal{T}$, where $y_i \in \{-1, +1\}$ is the label of $\boldsymbol{B}_i$. Lines 2–11 use two techniques to obtain high-level instance set $\boldsymbol{H}$. By analyzing the internal structure of each bag, at least one instance can be selected to construct the instance prototype set $\boldsymbol{C}$. Specifically, Lines 4–5 calculate the representativeness of the instances in each bag $\boldsymbol{B}_i \in \mathcal{T}$. Lines 6–7 select the top-ranked instances as the instance prototypes (IP). Next, our goal is to generate the high-level instance set $\boldsymbol{H}$ by identifying $\boldsymbol{C}$. Lines 9–11 select instances with peak density from $\boldsymbol{C}$ to construct $\boldsymbol{H}$. We design an embedding function to transform each bag into a single instance in the new feature space. Lines 13–17 embed each bag $\boldsymbol{B}_i$ into a new feature vector $\boldsymbol{V}_i$ through $\boldsymbol{H}$. Finally, Line 18 trains the SIL classifier $\mathcal{F}(\cdot)$ through the new data set $\{(\boldsymbol{V}_i, y_i)\}_{i=1}^{N}$.

3.2 The fast bag-inside instance selection technique

The common method for instance prototype (IP) generation is to select cluster centers [15] or causal instances [18] in the entire feature space. However, these methods have the following two problems: a) High time complexity; and b) The selected instances have no bag structure information. The fast bag-inside instance selection technique chooses instance prototypes of each bag through using its internal structure. The density ρ_{ij} and affinity l_{ij} metric of the instance $\boldsymbol{x}_{ij}$ are computed as follows.

The density of instance. For each instance $\boldsymbol{x}_{ij} \in \boldsymbol{B}_i$, the density ρ_{ij} is defined as

$$\rho_{ij} = \sum_{k \neq j}^{n_i} \exp -(\frac{d_{jk}}{d_c})^2, \tag{1}$$

where d_c is a cutoff distance and d_{jk} is the distance between $\boldsymbol{x}_{ij}$ and $\boldsymbol{x}_{ik}$. High-density instances mean that there are more adjacent instances within a given neighborhood radius. Therefore, high-density instances can reflect the local feature distribution of the bag.

In addition, the instances in the bag are not completely independent and distributed [21]. It is not enough to determine the representativeness only based on the density of the instance. Therefore, we use cosine similarity to represent the affinity between instances. The closer the cosine similarity of the two instances is to 1, the more similar they are.

Algorithm 1 Multi-instance embedding learning through high-level instance selection.

Input:
The data set $\mathcal{T}$;
The label vector $\mathcal{Y} = \{y_i\}_{i=1}^N$;
The proportion of instance prototypes r_c;
The number of high-level instances n_h;
Output:
The SIL classifier $\mathcal{F}(\cdot)$;
The high-level instance set $\boldsymbol{H}$;

// Step 1. Select the high-level instances.
$\boldsymbol{C} = \emptyset$; // Initialize instance prototype set.
for ($\boldsymbol{B}_i \in \mathcal{T}$) **do**
  $k = \lceil r_c \times n_i \rceil$; // The number of instance prototypes of each bag.
  Compute the score s_{ij} of $\boldsymbol{x}_{ij} \in \boldsymbol{B}_i$ according to Eq. (3);
  $\boldsymbol{C}'$ = the top-k score instances;
  $\boldsymbol{C} = \boldsymbol{C} \cup \boldsymbol{C}'$;
end for
$\boldsymbol{H} = \emptyset$; // Initialize high-level instance set.
Compute the score λ_i for each prototype $\boldsymbol{c}_i \in \boldsymbol{C}$ according to Eq. (5);
$\boldsymbol{H}$ = the set of top-n_h score prototypes;
// Step 2. Bag embedding.
for ($\boldsymbol{B}_i \in \mathcal{T}$) **do**
  Compute the embedding vector $\boldsymbol{V}_i$ according to Eq. (7) or (8) with $\boldsymbol{B}_i$;
  $V_{il} \leftarrow sign(V_{il})\sqrt{|V_{il}|}$, where V_{il} represents the l-th attribute of $\boldsymbol{V}_i$;
  $\boldsymbol{V}_i \leftarrow \boldsymbol{V}_i / \parallel \boldsymbol{V}_i \parallel_2$;
end for
Train the classifier $\mathcal{F}(\cdot)$ with the new data set $\{(\boldsymbol{V}_i, y_i)\}_{i=1}^N$;
Output $\mathcal{F}(\cdot)$ and $\boldsymbol{H}$;

The affinity of instance. For each instance $\boldsymbol{x}_{ij} \in \boldsymbol{B}_i$, the affinity l_{ij} is defined as

$$l_{ij} = \sum_{1 \le k \le n_i} \frac{\boldsymbol{x}_{ij} \cdot \boldsymbol{x}_{ik}}{\|\boldsymbol{x}_{ij}\| \|\boldsymbol{x}_{ik}\|}, \tag{2}$$

where $j, k \in [1..n_i]$.

After obtaining the density and affinity of each instance in the bag, the representativeness score s_{ij} of the instance can be computed as

$$s_{ij} = \rho_{ij} \times l_{ij}. \tag{3}$$

According to the MIL assumption, the proportion of positive and negative instances in each bag is different (e.g., tiger, grass and water in Figure 1). Therefore, we can chose the low/high score instances from the positive/negative bag as the IP. Finally, we can obtain the instance prototype set $\boldsymbol{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{n_c}\}$, where n_c is the cardinality of $\boldsymbol{C}$.

By considering the solution interval of the optimization objective, we design three types of instance prototypes selection modes as follows:

- **Global (G)** selects $\lceil r_c \times n_i \rceil$ instance prototypes from each bag.
- **Positive (P)** only selects from all positive bags.
- **Negative (N)** only selects from all negative bags.

The time complexity of the fast bag-inside instance selection technique is $O(dn)$, where d is the dimension and n is the number of all instances. The time complexity of instance selection based on the entire space is $O(dn^2)$. In contrast, our complexity is only linearly related to the number of instances rather than square related.

3.3 High-level instance selection technique

In order to explore the characteristics of the instance space, high-level instance selection technique is proposed. Based on $\boldsymbol{C} = \{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_{n_c}\}$, we can obtain high-level instances (HI).

For each $\boldsymbol{c}_i$, we calculate two quantities: its local density δ_i and its minimum distance β_i from the higher density prototypes. The local density δ_i is computed by Eq. (1). The difference is that the calculation interval is migrated from each bag to $\boldsymbol{C}$. The distance β_i is measured by computing the minimum distance between the $\boldsymbol{c}_i$ and any other IP with higher density:

$$\beta_i = \min_{j:\delta_j > \delta_i} (d_{ij}). \tag{4}$$

Particularly, for the IP with highest density, its distance is $\beta_i = \max_j(d_{ij})$. Finally the score λ_i of IP is calculated as

$$\lambda_i = \delta_i \times \beta_i. \tag{5}$$

With the scores of all IP calculated by Eq. (5), we select the top-n_h IP as the HI. Finally, we can obtain $\boldsymbol{H} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{n_h}\}$, where n_h the cardinality of $\boldsymbol{H}$.

3.4 Embedding technique via HI

After getting $\boldsymbol{H}$, we design the following method to embed the bags into a new feature space. Firstly, each instance $\boldsymbol{x}_{ij} \in \boldsymbol{B}_i$ is assigned to its nearest $\boldsymbol{h}_k$, denoted by $NH(\boldsymbol{x}_{ij}) = \boldsymbol{h}_k$. Then, each bag $\boldsymbol{B}_i$ can be expressed by n_h local vectors $\boldsymbol{v}_{ik}$:

$$\boldsymbol{v}_{ik} = \sum_{\boldsymbol{x}_{ij} \in \Omega} \boldsymbol{x}_{ij} - \boldsymbol{h}_k, \tag{6}$$

where $\Omega = \{\boldsymbol{x}_{ij} | NH(\boldsymbol{x}_{ij}) = \boldsymbol{h}_k\}$. Finally, the embedding vector $\boldsymbol{V}_i$ of bag $\boldsymbol{B}_i$ is a D-dimensional vector composed of concatenated local vectors [15]:

$$\boldsymbol{V}_i = \Big\Vert_{k=1}^{n_h} \boldsymbol{v}_{ik}, \tag{7}$$

where $D = n_h \times d$ and d is the dimension of instance $\boldsymbol{x}_{ij}$. However, the above embedding method will embed each bag into a high-dimensional space. Therefore, we design the second embedding method, which superimposes all the local vectors to get the embedding vector:

$$\boldsymbol{V}_i = \sum_{k=1}^{n_h} \boldsymbol{v}_{ik}. \tag{8}$$

Furthermore, each element of $\boldsymbol{V}_i$ is processed by $V_{il} \leftarrow sign(V_{il})\sqrt{|V_{il}|}$, and then the embedding vector is normalized by $\boldsymbol{V}_i \leftarrow \boldsymbol{V}_i / \parallel \boldsymbol{V}_i \parallel_2$ [11]. After getting the $\boldsymbol{V}_i$ for each $\boldsymbol{B}_i$, we can predict the bag label by processing $\boldsymbol{V}_i$ with any single-instance classifier $\mathcal{F}(\cdot)$ (e.g., SVM).

4 Experiments

In this section, we conducted experiments on MIHI and 9 comparison algorithms for six learning tasks. To ensure the validity of the experiment, we used 10 times 10-fold cross-validation to calculate the average accuracy. The averaged results (mean) and standard deviation (std) of each algorithm is reported.

4.1 Comparison algorithms

We compared MIHI with 9 state-of-the-art algorithms: a) MILES [3] embeds bags based on the bag-instance similarity measure and all instances; b) BAMIC [22] embeds bags by employing bag-level k-means, with the parameters including average Hausdorff distance and the number of clustering centers ($r \times \min\{N, 100\}$, where r is enumerated in $\{0.1, \cdots, 1.0\}$); c) MILFM [10] uses AdaBoost to select the bag features embedded by instance prototypes, with the parameters including the number of cluster centers (40); d) Simple-MI [1] uses the arithmetic mean of instances in the bag as the representation of the bag itself. e) miFV [15] extracts the instance information with the Gaussian mixture model (GMM), with the parameters including the number of components for GMM (enumerate in $\{1, 2, 3\}$); f) miVLAD [15] embeds bags based on the instance-level k-means, with the parameters including the number of clustering centers (enumerate in $\{1, 2\}$); g) MILDM [16] selects the discriminative instances via instance evaluation criteria, with the parameters including the size of discriminative instance pool (the number of bags); h) StabelMIL [18] embeds bags based on causal instances, with the parameters including the scale variable (0.25); and i) ELDB [17] selects more representative bags with the discriminative analysis and reinforcement technique, and finally obtains more distinguishable single vectors.

4.2 Experimental data sets

Six fields of learning tasks across 26 data sets are used to validate MIHI. We briefly introduce the domain knowledge of these data sets: 1) **Image retrieval:**

Content-based image retrieval problems include identifying the expected target object in the image [2]. In our experiments, elephant, fox, and tiger data sets are used; 2) **Mutagenicity prediction:** Mutagenesis is a drug activity prediction problem. There are two versions, easy (1) and hard (2), of the data set [13]; 3) **Medical image:** Messidor is a medical classification problem data set, which consists of $1,200$ fundus images from 546 healthy and 654 diabetic patients [5]; 4) **Newsgroups:** The newsgroups is a text categorization data set [21]. Posts from different newsgroups form a bag. Each category has 50 positive bags and 50 negative bags; 5) **Web recommendation:** The question is whether to classify web pages as interesting web pages [20]. There are a total of 9 users who rate the web page this way, so there are 9 different data sets; A web page is a bag, and the links on the web page are instances; and 6) **Biocreative:** Biocreative is a large-scale text classification data set [12]. The task is to decide whether some genetic ontology (GO) code should be used to annotate a given pair.

4.3 Performance comparison

Table 1. Accuracy ($\%, mean_{\pm std}$) with standard deviations on 26 MIL data sets. The highest average accuracy is marked with •.

Datasets	(d)	MILES	BAMIC	MILFM	Simple-MI	miFV	miVLAD	MILDM	StableMIL	ELDB	**MIHI**
Elephant ♣	(230)	$81.2_{\pm 2.36}$	$75.8_{\pm 0.12}$	$81.7_{\pm 0.23}$	$80.2_{\pm 0.08}$	$84.2_{\pm 0.09}$	$84.1_{\pm 0.13}$	$76.1_{\pm 0.29}$	$84.2_{\pm 4.23}$	$75.8_{\pm 3.21}$	$90.4_{\pm 0.93}$•
Fox ♣	(230)	$58.5_{\pm 3.63}$	$51.9_{\pm 0.33}$	$45.4_{\pm 0.24}$	$62.6_{\pm 0.13}$	$61.9_{\pm 0.12}$	$62.3_{\pm 0.16}$	$58.8_{\pm 0.41}$	$55.3_{\pm 2.83}$	$60.7_{\pm 2.02}$	$65.5_{\pm 2.46}$•
Tiger ♣	(230)	$77.1_{\pm 2.65}$	$69.2_{\pm 0.14}$	$73.3_{\pm 3.09}$	$79.3_{\pm 0.07}$	$77.2_{\pm 0.06}$	$84.8_{\pm 0.12}$	$64.3_{\pm 0.14}$	$60.7_{\pm 2.24}$	$72.2_{\pm 2.00}$	$85.7_{\pm 1.10}$•
Mutagenesis1 ◊	(7)	$88.3_{\pm 2.11}$	$75.8_{\pm 0.09}$	$84.8_{\pm 0.21}$	$66.6_{\pm 0.06}$	$79.2_{\pm 0.08}$	$77.6_{\pm 0.13}$	$81.1_{\pm 0.28}$	$88.3_{\pm 2.11}$•	$84.9_{\pm 1.71}$	$70.0_{\pm 1.91}$
Mutagenesis2 ◊	(7)	$84.2_{\pm 1.60}$	$82.8_{\pm 0.17}$	$83.5_{\pm 0.12}$	$68.8_{\pm 0.17}$	$79.0_{\pm 0.17}$	$78.8_{\pm 0.32}$	$81.7_{\pm 0.34}$	$85.3_{\pm 0.02}$•	$82.8_{\pm 0.83}$	$82.0_{\pm 1.50}$
Messidor ♦	(687)	$50.3_{\pm 3.33}$	$62.0_{\pm 0.05}$	$54.5_{\pm 0.00}$	$55.9_{\pm 0.03}$	$71.5_{\pm 0.05}$•	$67.5_{\pm 0.05}$	$54.5_{\pm 0.24}$	$54.5_{\pm 0.01}$	$63.8_{\pm 0.45}$	$68.6_{\pm 0.39}$
alt.atheism ▲	(200)	$50.9_{\pm 0.30}$	$84.9_{\pm 0.05}$	$52.9_{\pm 0.07}$	$83.4_{\pm 0.11}$	$82.4_{\pm 0.17}$	$85.6_{\pm 0.18}$	$53.9_{\pm 0.50}$	$52.5_{\pm 5.37}$	$85.6_{\pm 2.01}$	$88.5_{\pm 1.22}$•
comp.graphics ▲	(200)	$49.4_{\pm 1.28}$	$80.7_{\pm 0.10}$	$52.7_{\pm 0.15}$	$77.3_{\pm 0.05}$	$80.1_{\pm 0.11}$	$78.8_{\pm 0.12}$	$52.0_{\pm 0.49}$	$51.4_{\pm 2.97}$	$81.1_{\pm 1.02}$	$83.5_{\pm 1.76}$•
comp.os.ms ▲	(200)	$51.9_{\pm 1.64}$	$72.1_{\pm 0.16}$	$46.6_{\pm 0.26}$	$53.2_{\pm 0.29}$	$72.5_{\pm 0.12}$	$68.8_{\pm 0.26}$	$47.7_{\pm 0.29}$	$47.8_{\pm 3.34}$	$73.7_{\pm 1.33}$•	$73.0_{\pm 4.24}$
comp.sys.mac ▲	(200)	$51.0_{\pm 4.45}$	$80.0_{\pm 0.13}$	$52.3_{\pm 0.46}$	$77.6_{\pm 0.09}$	$77.3_{\pm 0.11}$	$78.2_{\pm 0.15}$	$51.5_{\pm 0.43}$	$51.2_{\pm 3.79}$	$81.1_{\pm 1.71}$•	$80.5_{\pm 1.73}$
comp.window.x ▲	(200)	$64.3_{\pm 4.12}$	$77.9_{\pm 0.08}$	$53.0_{\pm 0.10}$	$66.0_{\pm 0.11}$	$85.4_{\pm 0.11}$•	$82.1_{\pm 0.14}$	$58.2_{\pm 0.55}$	$53.4_{\pm 2.91}$	$79.7_{\pm 1.41}$	$83.9_{\pm 0.94}$
misc.forsale ▲	(200)	$50.3_{\pm 1.49}$	$67.3_{\pm 0.11}$	$51.2_{\pm 0.19}$	$56.2_{\pm 0.36}$	$72.5_{\pm 0.25}$•	$71.8_{\pm 0.23}$	$45.5_{\pm 0.53}$	$49.3_{\pm 5.51}$	$70.2_{\pm 0.63}$	$68.5_{\pm 1.86}$
rec.motorcycles ▲	(200)	$50.7_{\pm 0.46}$	$78.4_{\pm 0.10}$	$52.5_{\pm 0.45}$	$45.6_{\pm 0.22}$	$86.7_{\pm 0.13}$•	$81.2_{\pm 0.12}$	$53.8_{\pm 0.41}$	$55.4_{\pm 3.99}$	$79.7_{\pm 2.40}$	$83.3_{\pm 1.49}$
rec.sport ▲	(200)	$52.9_{\pm 4.09}$	$83.1_{\pm 0.05}$	$50.0_{\pm 0.00}$	$74.8_{\pm 0.12}$	$85.1_{\pm 0.10}$	$82.9_{\pm 0.16}$	$48.5_{\pm 0.50}$	$49.5_{\pm 3.93}$	$82.2_{\pm 1.01}$	$90.2_{\pm 1.17}$•
sci.crypt ▲	(200)	$51.4_{\pm 0.66}$	$76.8_{\pm 0.07}$	$51.1_{\pm 0.10}$	$73.4_{\pm 0.08}$	$75.6_{\pm 0.14}$	$81.1_{\pm 0.16}$	$47.7_{\pm 0.35}$	$50.7_{\pm 5.24}$	$77.1_{\pm 1.02}$	$89.6_{\pm 1.36}$•
sci.med ▲	(200)	$53.7_{\pm 3.82}$	$82.5_{\pm 0.05}$	$55.0_{\pm 0.57}$	$71.1_{\pm 0.09}$	$83.1_{\pm 0.08}$	$82.2_{\pm 0.15}$	$50.9_{\pm 0.36}$	$50.4_{\pm 3.85}$	$82.7_{\pm 0.83}$	$89.9_{\pm 0.83}$•
web1 ▼	$(5,863)$	$82.1_{\pm 2.71}$	$81.2_{\pm 0.06}$	$81.5_{\pm 0.04}$	$79.0_{\pm 0.11}$	$81.5_{\pm 0.06}$	$79.9_{\pm 0.09}$	$82.5_{\pm 0.09}$•	$82.4_{\pm 1.15}$	$82.5_{\pm 2.04}$	$81.2_{\pm 1.22}$
web2 ▼	$(6,519)$	$81.5_{\pm 0.58}$	$81.4_{\pm 0.06}$	$82.4_{\pm 0.15}$	$79.4_{\pm 0.12}$	$81.5_{\pm 0.06}$	$80.2_{\pm 0.07}$	$83.1_{\pm 0.08}$•	$80.5_{\pm 2.16}$	$82.9_{\pm 2.27}$	$81.5_{\pm 0.60}$
web3 ▼	$(6,306)$	$82.1_{\pm 2.19}$	$81.2_{\pm 0.04}$	$83.2_{\pm 0.15}$•	$79.5_{\pm 0.17}$	$81.6_{\pm 0.08}$	$81.2_{\pm 0.08}$	$82.9_{\pm 0.04}$	$81.2_{\pm 0.82}$	$81.4_{\pm 0.68}$	$81.7_{\pm 1.43}$
web4 ▼	$(6,059)$	$78.9_{\pm 2.75}$	$77.7_{\pm 0.07}$	$79.5_{\pm 0.17}$	$78.1_{\pm 0.08}$	$77.7_{\pm 0.06}$	$81.7_{\pm 0.14}$	$79.3_{\pm 0.16}$	$77.6_{\pm 0.45}$	$79.8_{\pm 1.31}$	$83.9_{\pm 0.91}$•
web5 ▼	$(6,407)$	$78.8_{\pm 0.71}$	$79.3_{\pm 0.05}$	$78.8_{\pm 0.12}$	$77.2_{\pm 0.09}$	$77.1_{\pm 0.08}$	$82.1_{\pm 0.11}$	$78.6_{\pm 0.27}$	$78.1_{\pm 0.61}$	$78.1_{\pm 1.22}$	$82.5_{\pm 0.68}$•
web6 ▼	$(6,417)$	$81.7_{\pm 2.71}$	$77.3_{\pm 0.15}$	$81.8_{\pm 0.23}$	$79.6_{\pm 0.07}$	$77.7_{\pm 0.06}$	$82.5_{\pm 0.14}$	$83.6_{\pm 0.20}$	$73.3_{\pm 0.34}$	$80.9_{\pm 2.33}$	$84.1_{\pm 1.30}$•
web7 ▼	$(6,450)$	$56.4_{\pm 1.55}$	$42.9_{\pm 0.31}$	$61.5_{\pm 0.16}$	$58.4_{\pm 0.47}$	$68.5_{\pm 0.23}$	$73.5_{\pm 0.26}$	$63.6_{\pm 0.34}$	$62.0_{\pm 2.75}$	$52.8_{\pm 4.56}$	$75.7_{\pm 2.03}$•
web8 ▼	$(5,999)$	$56.4_{\pm 2.86}$	$48.5_{\pm 0.44}$	$61.5_{\pm 0.37}$	$58.0_{\pm 0.52}$	$71.0_{\pm 0.33}$	$73.8_{\pm 0.28}$	$57.0_{\pm 0.37}$	$59.0_{\pm 3.19}$	$50.5_{\pm 3.57}$	$78.4_{\pm 1.56}$•
web9 ▼	$(6,279)$	$59.5_{\pm 2.61}$	$41.5_{\pm 0.41}$	$59.8_{\pm 0.28}$	$58.1_{\pm 0.28}$	$71.5_{\pm 0.37}$	$76.1_{\pm 0.11}$	$56.5_{\pm 0.32}$	$54.9_{\pm 3.73}$	$49.3_{\pm 4.42}$	$78.5_{\pm 1.99}$•
component ★	(200)	N/A	$92.2_{\pm 0.01}$	N/A	$69.6_{\pm 0.04}$	$91.5_{\pm 0.01}$	$92.9_{\pm 0.01}$	N/A	N/A	N/A	$93.4_{\pm 0.04}$•
function ★	(200)	N/A	$95.6_{\pm 0.01}$	N/A	$71.7_{\pm 0.04}$	$94.9_{\pm 0.01}$	$95.8_{\pm 0.01}$	N/A	N/A	N/A	$96.6_{\pm 0.05}$•
process ★	(200)	N/A	$96.0_{\pm 0.00}$	N/A	$66.7_{\pm 0.04}$	$95.8_{\pm 0.00}$	$96.8_{\pm 0.00}$	N/A	N/A	N/A	$97.1_{\pm 0.01}$•
Mean rank		6.43	6.57	5.89	7.21	4.79	4.43	6.21	6.64	4.11	2.71•

♣ image retrieval, ◊ mutagenicity prediction, ♦ medical image, ▲ newsgroups, ▼ web recommendation, ★ biocreative.

Table 1 shows the experimental results of MIHI and comparison algorithms. The best performance value for each data set is highlighted with a small black bullet. Mean rank represents the ranking of the average performance of the cur-

rent algorithm on each data set [6]. The symbol "N/A" means that the algorithm cannot get experimental results.

The experimental results show that the MIHI algorithm has achieved the best experimental results on more than 70% of the data sets. And its mean rank is 2.71, which is superior to 9 traditional algorithms. Specifically, the accuracy of MIHI on some data sets is about 10% higher than other algorithms, such as elephant, rec.sport.hockey and web4. The reason may be that the our instance selection techniques can effectively select the instance with the largest amount of information from each bag. On image retrieval data, MIHI performed well on the three image data sets. However, MIHI performed poorly on the two mutagenicity prediction data sets, which may be caused by the low dimensionality of mutagenicity. StableMIL performs very well on mutagenicity. The reason may be that StableMIL can obtain the most informative causal instance from the super low-dimensional positive bag. From the performance of newsgroups, web recommendation and large-scale data sets, MIHI can get better results whether it is low-dimensional or high-dimensional data. We only compare MIHI with the four algorithms on large-scale data sets, because the time complexity of MILES, MILFM, MILDM, StableMIL and ELDB is relatively high.

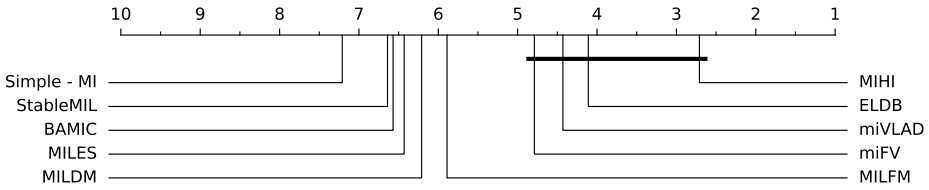

Fig. 2. Comparison of MIHI with 9 comparison algorithms with Bonferroni-Dunn test. Algorithms not connected to MIHI in the CD plot were considered to have significant performance of the control algorithm (CD = 2.24, significance level 0.05).

We also applied the post hoc Bonferroni-Dunn test [6] to test whether MIHI achieves competitive performance among the 9 compared algorithms. Figure 2 reports the critical difference (CD) plot at the 0.05 significance level. The mean accuracy ranks for each algorithm are marked along the axis (lower grades on the left). In addition, algorithms with an mean ranking within one CD of MIHI are connected by thick lines. Otherwise, any MIHI-independent algorithm is considered significantly different.

4.4 Parameter analysis

Figure 3 shows the experimental results of parameter analysis on elephant data set. The symbols "S" and "C" respectively represent the two modes of bag embedding: superimpose and concatenate; "G", "P" and "N" respectively represent three instance selection modes. For all these subgraphs, the abscissa indicates the

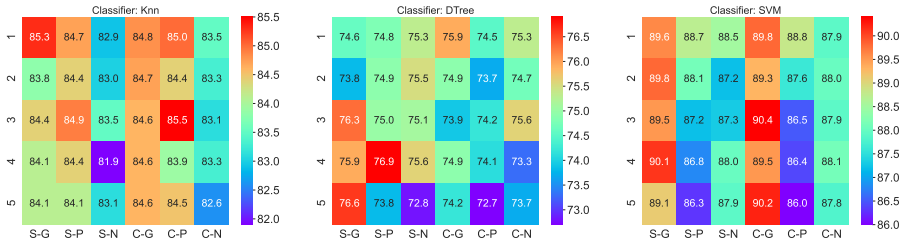

Fig. 3. Parameter analysis of MIHI with the number of instance prototype, three instance selection modes, two bag embedding modes and three classifiers for elephant data set. The best parameter settings of elephant are: 3 instance prototypes, instance selection mode "G" and bag embedding mode "C".

mode selected by the instance prototypes, and the ordinate indicates the number of instance prototypes. The three subgraphs show the classification accuracy on the classifier Knn, Decision Tree (DTree) and SVM respectively. The darkest colored table of the heat map indicates the highest accuracy. The following summarizes the impact of parameters on MIHI:

- **Bag embedding modes:** The classification performance of the two bag embedding modes is equivalent. However, since mode "C" embeds each bag in a high-dimensional space, we choose mode "S" for bag embedding in subsequent experiments.
- **Instance selection modes:** The results of the elephant in the classifier SVM show that the classification performance of mode "G" is better than the other two modes. However, in the other two classifiers, it is the best in mode "P".
- **The number of instance prototypes:** MIHI can achieve the best performance in most cases when the number of instance prototypes is 3-5.
- **Classifier:** SVM is more suitable for these data sets than DTree and Knn.

4.5 Efficiency comparison

Table 2. The CPU runtime (in seconds) of one 10CV of the comparison algorithm on the 4 MIL classification data set.

Data sets	$(d/n/N)$	MILES	BAMIC	MILFM	Simple-MI	miFV	miVLAD	MILDM	StableMIL	ELDB	**MIHI**
Time Complexity		$O(dn^2)$	$O(dN^2)$	$O(dn^2)$	$O(dN)$	$O(dn)$	$O(dn)$	$O(dn^2)$	$O(dn^2)$	$O(dn^2)$	$O(dn)$
Fox	$(230/1,320/200)$	2.378	1.512	8.514	0.151	4.202	1.284	5.425	13.959	3.712	1.034
alt.atheism	$(200/5,443/100)$	24.870	20.200	46.422	0.124	4.694	1.827	30.625	48.483	39.465	15.417
comp.graphics	$(200/3,094/100)$	8.939	6.651	25.217	0.104	3.520	1.533	11.635	43.796	13.543	5.526
web4	$(6,059/3,423/113)$	27.216	25.228	163.253	1.205	406.751	20.005	39.843	610.798	52.981	8.134
Mean rank		5.50	4.50	8.75	1.00	5.50	2.50	7.00	10.00	7.25	3.00

Table 2 shows the time complexity and runtime of MIHI compared with 9 competing algorithms. For MIHI, the construction of the high-level instances cost $O(dn)$, where d is the dimension and n is the cardinality of instance space. Table 2 compares the CPU running time of these methods on four data sets. The mean rank shows that the speed of MIHI is slightly lower than that of Simple-MI and miVLAD. This may be because Simple-MI does not need to consume a lot of time to calculate the distance of instances. However, Simple-MI does not perform well on these data sets. The k-means algorithm used by miVLAD has low time complexity. Besides, even on the small scale data set, the runtime of MILFM and StableMIL are relatively large.

5 Conclusion

In this paper, we proposed the MIHI algorithm to select high-level instances. MIHI fully utilizes the structure information of the bag-inside and effectively explores the characteristics of the instance space. The experiments were conducted on 26 MIL data sets. According to Table 1, the MIHI algorithm has achieved the best accuracy on more than 70% of the data sets. Its mean rank is 2.71, which is superior to 9 traditional algorithms. In addition, MIHI has linear time complexity, and its efficiency is slightly lower than that of Simple-MI and miVLAD.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62006200), Natural Science Foundation of Sichuan Province (2019YJ0314), Sichuan Province Youth Science and Technology Innovation Team (2019JDTD0017), and Central Government Funds of Guiding Local Scientific and Technological Development (2021ZYD0003).

References

1. Amores, J.: Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence **201**(4), 81–105 (2013)
2. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS. pp. 561–568 (2002)
3. Chen, Y.X., Bi, J.B., Wang, J.Z.: MILES: Multiple-instance learning via embedded instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(12), 1931–1947 (2006)
4. Chen, Y.X., Wang, J.Z.: Image categorization by learning and reasoning with regions. The Journal of Machine Learning Research **5**, 913–939 (2004)
5. Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., Charton, B., Klein, J.C.: Feedback on a publicly distributed image database: The messidor database. Image Analysis & Stereology **33**(3), 231–234 (2014)

6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006)
7. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence **89**(1-2), 31–71 (1997)
8. Faria, A.W., Coelho, F.G.F., Silva, A., Rocha, H.P., Almeida, G., Lemos, A.P., Braga, A.P.: MILKDE: A new approach for multiple instance learning based on positive instance selection and kernel density estimation. Engineering Applications of Artificial Intelligence **59**, 196–204 (2017)
9. He, C.K., Shao, J., Zhang, J.S., Zhou, X.M.: Clustering-based multiple instance learning with multi-view feature. Expert Systems with Applications **162**, 113027 (2020)
10. Hong, R.C., Wang, M., Gao, Y., Tao, D.C., Li, X.L., Wu, X.D.: Image annotation by multiple-instance learning with discriminative feature mapping and selection. IEEE Transactions on cybernetics **44**(5), 669–680 (2014)
11. Jorge, S., Florent, P., Thomas, M., Jakob, V.: Image classification with the fisher vector: Theory and practice. International journal of computer vision **105**(3), 222–245 (2013)
12. Ray, S., Craven, M.: Learning statistical models for annotating proteins with function information using biomedical text. BMC Bioinformatics **6**(1), 1–9 (2005)
13. Srinivasan, A., Muggleton, S., King, R.: Comparing the use of background knowledge by inductive logic programming systems. In: ILP. pp. 199–230 (1995)
14. Tarragó, D.S., Cornelis, C., Bello, R., Herrera, F.: A multi-instance learning wrapper based on the Rocchio classifier for web index recommendation. Knowledge-Based Systems **59**(0950-7051), 173–181 (2014)
15. Wei, X.S., Wu, J.X., Zhou, Z.H.: Scalable algorithms for multi-instance learning. IEEE Transactions on Neural Networks and Learning Systems **28**(4), 975–987 (2017)
16. Wu, J., Pan, S.R., Zhu, X.Q., Zhang, C.Q., Wu, X.D.: Multi-instance learning with discriminative bag mapping. IEEE Transactions on Knowledge and Data Engineering **30**(6), 1065–1080 (2018)
17. Yang, M., Zhang, Y.X., Wang, X.Z., Min, F.: Multi-instance ensemble learning with discriminative bags. IEEE transactions on systems, man, and cybernetics: Systems pp. 1–12 (2021)
18. Zhang, W., Li, J., Liu, L.: Robust multi-instance learning with stable instances (2019)
19. Zhang, Y.L., Zhou, Z.H.: Multi-instance learning with key instance shift. In: IJCAI. pp. 3441–3447 (2017)
20. Zhou, Z.H., Jiang, K., Li, M.: Multi-instance learning based web mining. Applied Intelligence **22**(2), 135–147 (2005)
21. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-I.I.D. samples. In: ICML. pp. 1249–1256 (2009)
22. Zhou, Z.H., Zhang, M.L.: Multi-instance clustering with applications to multi-instance prediction. Applied Intelligence **31**(1), 47–68 (2009)