

Multi-Instance Learning from Positive and Unlabeled Bags

Jia Wu^{1,3}, Xingquan Zhu², Chengqi Zhang¹, and Zhihua Cai³

¹ Centre for Quantum Computation & Intelligent Systems, FEIT,
University of Technology Sydney, NSW 2007, Australia

² Dept. of Computer & Electrical Engineering and Computer Science,
Florida Atlantic University, Boca Raton, FL 33431, USA

³ Dept. of Computer Science, China University of Geosciences, Wuhan, China
jia.wu@student.uts.edu.au, chengqi.zhang@uts.edu.au,
xzhu3@fau.edu, zhcai@cug.edu.cn

Abstract. Many methods exist to solve multi-instance learning by using different mechanisms, but all these methods require that both positive and negative bags are provided for learning. In reality, applications may only have positive samples to describe users learning interests and remaining samples are unlabeled (which may be positive, negative, or irrelevant to the underlying learning task). In this paper, we formulate this problem as positive and unlabeled multi-instance learning (puMIL). The main challenge of puMIL is to accurately identify negative bags for training discriminative classification models. To solve the challenge, we assign a weight value to each bag, and use an Artificial Immune System based self-adaptive process to select most reliable negative bags in each iteration. For each bag, a most positive instance (for a positive bag) or a least negative instance (for an identified negative bag) is selected to form a positive margin pool (PMP). A weighted kernel function is used to calculate pairwise distances between instances in the PMP, with the distance matrix being used to learn a support vector machines classifier. A test bag is classified as positive if one or multiple instances inside the bag are classified as positive, and negative otherwise. Experiments on real-world data demonstrate the algorithm performance.

Keywords: Multi-instance learning, unlabeled bags, classification.

1 Introduction

Multi-instance learning (MIL) [1] is a special type of learning task where each observation contains a bag of instances. A bag is labeled positive if one or multiple instances inside the bag are positive, and negative otherwise. The uniqueness of not requiring labels for individual instances makes multi-instance learning very suitable for applications without label information for individual instances. Because the genuine positive instance(s) inside each positive bag is unknown, the main challenge of multi-instance learning is to leverage bag labels and constraints to derive accurate classification models. Roughly, existing MIL methods

[2] can be separated into the following three categories: (1) instance-based modeling, which finds most positive and most negative instances from bags to derive MIL models; (2) bag-based modeling, which directly builds classification models at the bag level; and (3) hybrid approaches, which use instances and bags to confine the learning space to build classification models.

For all existing MIL methods, one prerequisite is that training data must contain both positive and negative bags to derive discriminative models. In reality, many applications may only have positive bags to indicate users learning interests and the remaining bags are unlabeled (which may be positive, negative, or irrelevant to the learning task). For example, during an image retrieval process [3], users may click one or multiple images which are interesting to them (the clicked images can be regarded as positive bags), but majority images remain unchecked, so we do not know whether those images do not contain users retrieval concepts (*i.e.* negative bags) or users simply overlook the images. In this case, there is no negative bag but only positive and unlabeled bags are available.

When only positive and unlabeled bags are available, a straightforward solution for MIL is to propagate a bag’s label to each instance inside the bag, so the problem can be solved by using standard Positive-Unlabeled learning [4]. Indeed, this simple solution is ineffective because not all instances inside a positive bag are positive, so some instances will be mislabeled and deteriorate the classification accuracy. A slightly more intelligent way is to use Positive and Unlabeled learning strategy [5] to first treat all unlabeled bags as negative bags and train an MIL classifier, and then iteratively refine identified negative bags by using trained MIL classifiers. This solution is still ineffective mainly because the identification of negative bags only relies on the MIL classifiers but does not take the unique MI bag constraints into consideration, so directly training MIL classifiers using positive bags and identified negative bags will severely deteriorate the classification accuracy.

The above observations motivate a very practical learning task where only positive and unlabeled bags are available for multi-instance learning. In this paper, we formulate this problem as positive and unlabeled multi-instance learning (puMIL), where the key challenge is twofold: (1) **MIL learning with unreliable bag labels**: Although it is always possible to identify some unlabeled bags as negative bags, the labels of identified negative bags are unreliable. Directly building MIL classifiers from positive and identified negative bags may result in low classification accuracy. This reality calls for new MIL learning frameworks capable of handling unreliable bag labels; and (2) **Tackling uncertainty inside positive bags**: For MIL, the genuine labels of instances inside a positive bag are unknown, although at least one instance has to be positive. Finding “most positive instances” plays a significant role for multi-instance learning. In a puMIL setting, this process is further complicated because no negative bags are available to help identify positive instance(s) in a positive bag.

In this paper, we propose a self-adaptive learning framework to tackle the above challenges for puMIL. More specifically, we assign a weight value to each bag, and use an Artificial Immune System based search process to update bag

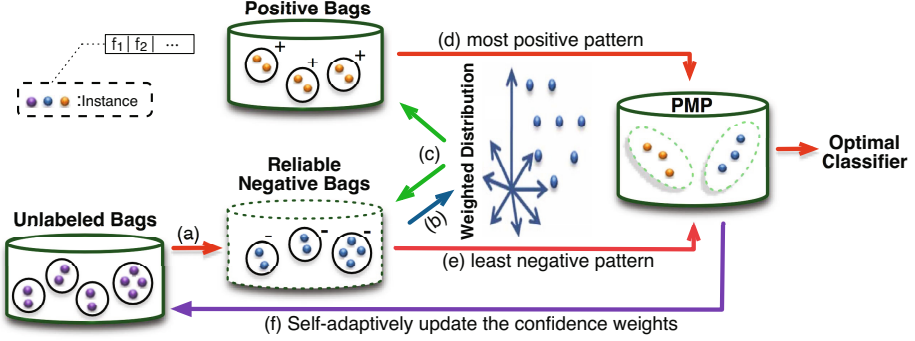


Fig. 1. A conceptual view of the proposed puMIL framework: Given a number of positive and unlabeled bags, puMIL starts from assigning a weight value for each bag. (a): the weight values help identify some unlabeled bags as reliable negative bags. (b): the obtained reliable negative bags help build a weighted distribution, which is used to measure instances in positive and reliable negative bags, respectively, (c). After that, the “most positive pattern” (d) for positive bags and the “least negative pattern” (e) for reliable negative bags are selected to form a Positive Margin Pool (PMP), which helps build a weighted SVM classifier for classification. Based on the constructed PMP, a self-adaptive strategy (discussed in Section 4.3) is used to update weight value of each bag (f). The iteration process continues, with the objective of improving the quality of reliable negative bags and the quality of PMP, and achieving optimal accuracy.

weight values and identify the most reliable negative bags in each iteration. For each bag, a “most positive instance” (for a positive bag) or a “least negative instance” (for an identified negative bag) is selected to form a positive margin pool (PMP). A weighted kernel function is used to calculate pairwise distances between instances in the PMP, with the distance matrix being used to learn a weighted support vector machines classifier. Experiments on real-world positive and unlabeled bags confirm the effectiveness of the proposed design.

The remainder of the paper is structured as follows. Preliminary and problem statement are addressed in Section 2, followed by the overall framework in Section 3. Section 4 introduces detailed algorithms, followed by experiments in Section 5. We conclude the paper in Section 6.

2 Preliminaries and Problem Statement

Denote $\mathcal{B} = \{B_1, \dots, B_n\}$ a bag set with n bags, and B_i is the i th bag in the set. The bag set contains both positive and unlabeled bags, with B_i^+ and B_j^u indicating a positive and an unlabeled bag, respectively (for ease of representation, we also use B_i to denote a bag). Let $Y = [y_1, \dots, y_n]$ where y_i is the label of B_i . In generic MIL settings, a positive and a negative bag’s label can be denoted by $y_i = +1$ and $y_i = -1$, respectively. In a puMIL setting, an unlabeled bag B_j^u ’s label is denoted by $y_j = 0$. The collections of positive and unlabeled bag sets can be denoted by \mathcal{B}^+ and \mathcal{B}^u , respectively. During the learning process,

the algorithm may identify a set of unlabeled bags as a negative bag set, which is denoted by \mathcal{B}^- . It is worth noting that \mathcal{B}^+ contains bags which are genuinely positive, whereas \mathcal{B}^- contains bags which are “identified” as negative bags.

For each bag B_i , the number of instances inside the bag is denoted by n_i , and $\mathbf{x}_{i,j}, j = 1, 2, \dots, n_i$ represents the j th instance in B_i . Each instance $\mathbf{x}_{i,j}$ also has a label but cannot be directly observed, because of the bag constraint. In order to tackle unreliable bag labels in puMIL setting (challenge # 1), we use a weight value w_i to indicate the label confidence of each bag B_i . So for a positive bag B_i^+ , its weight value w_i is 1 (because it is genuinely positive), whereas for an identified negative bag B_j , its weight value $w_j \in (0, 1]$, with a higher w_j value indicating that B_j is more likely being a negative bag.

Given a training set \mathcal{B} containing a handful of positive \mathcal{B}^+ bags and many unlabeled \mathcal{B}^u bags, puMIL learning **aims** to build a prediction model from \mathcal{B} to accurately predict previously unseen bags with maximum F-score.

3 Overall Framework

Figure 1 illustrates the conceptual view of the proposed puMIL framework, which includes two major steps to solve key challenges identified in Section 1.

Positive Margin Pool (PMP): In order to carry out multi-instance learning with positive and unlabeled bags only, we propose to use maximum margin idea [6] to build a positive margin pool (PMP) by modeling the distributions of the positive and unlabeled bags. More specifically, we utilize bag weight values to identify some unlabeled bags as most reliable negative bags. After that, we select “most positive patterns” from positive bags and “least negative patterns” from identified negative bags to form a positive margin pool. Because PMP contains signature patterns with respect to positive and identified negative bags, it will help differentiate decision boundaries to separate positive bags.

Self-Adaptive Bag Weight Updating: In order to properly identify most negative bags from unlabeled bags, we assign a weight value to each bag and will employ a self-adaptive iterative process to update bag weight values to find reliable negative bags. The proposed iterative bag weight updating process is based on clonal selection theory [7] in Artificial Immune Systems. It iteratively searches different weight values, including mutations and clones of weight values from the previous iteration, to find the one which optimizes the object function (*i.e.* the F-score in our case). As a result, the self-adaptive weight updating can ensure high quality negative bags are identified to form positive margin pool (PMP) for multi-instance learning without negative bags. The weight updating is detailed in Section 4.3.

4 Positive and Unlabeled Multi-Instance Learning

4.1 Optimization Framework with Unreliable Bag Labels

To handle unreliable bag labels from identified negative bags, we propose to build a Positive Margin Pool (PMP) which contains “most positive patterns” from

positive bags and “least negative patterns” from identified negative bags. Our multi-instance learning is then achieved by building an optimization framework based on the PMP, under the condition that negative bags are unreliable and therefore need to be carefully handled. In Eq. (1), we propose our objective function in handling unreliable bag labels.

$$\begin{aligned} \min_{\omega, \mathbf{x}^B} & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \ell(\mathbf{x}_i^B, w_i y_i; \omega) \\ \text{s.t. } & \ell(\mathbf{x}_i^B, w_i y_i; \omega) = \max(0, 1 - w_i y_i \omega^\top \mathbf{x}_i^B) \end{aligned} \quad (1)$$

where \mathbf{x}_i^B denotes a “most positive pattern” or a “least negative pattern” in PMP with m denoting the number of instances in PMP. w_i denotes weight confidence of the i th instance in PMP. In our design, because only one instance is selected from each bag, the weight confidence of the instance is equal to the weight confidence value of the bag. More specifically, if the instance is from a positive bag, its weight confidence $w_i = 1$. For instances from identified negative bags, their weight values are $w_j \in (0, 1]$, with a higher w_j value indicating that B_j is more likely being a negative bag.

The cost function in Eq. (1) is the linear SVM in the primal formulation, where $\ell(\mathbf{x}_i^B, w_i y_i; \omega)$ is the hinge loss function used by the SVM classifier. The above primal form can be formulated into a dual version as

$$\begin{aligned} \max_{\alpha, \mathbf{x}^B} & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i w_i y_i (\mathbf{x}_i^B)^\top \mathbf{x}_j^B \alpha_j w_j y_j \\ \text{s.t. } & \sum_i \alpha_i = 1 \text{ and } \alpha_i \geq 0 \end{aligned} \quad (2)$$

where α_i is the Lagrangian multiplier. Indeed, the above optimization problem defines a nonlinearly constrained nonconvex optimization task. It contains two sub-problems: (1) learning continuous variables α which is equivalent to finding hyper-plane ω , and (2) selection of instances \mathbf{x}^B for PMP which is equivalent to finding optimal weight values \mathbf{w} . To the best of our knowledge, no direct solution exists to find its global optimal. In this paper, we derive an self-adaptive approach to find optimal bag weights for unlabeled bags, and then use generic SVM optimization to solve the above optimization problem.

In the following, we first explain the construction of the PMP, and then introduce detailed self-adaptive process for finding optimal bag weight values.

4.2 PMP: Positive Margin Pool

The main purpose of building a Positive Margin Pool is to identify some “most positive patterns” (for positive bags) and “least negative patterns” (for possible negative bags), so PMP can help build classifiers to differentiate positive *vs.* negative instances for multi-instance classification. This also provides solutions to tackle unreliable bag labels obtained from unlabeled bags.

The construction of the PMP is motivated by the margin principle, which states that samples close to the decision boundary play critical roles in improving the performance of the underlying classifier. In our puMIL setting, we assign

a weight confidence value w_i to each bag, so it can help identify unlabeled bags which are most likely being negative as “reliable negative bags” and then extract most positive patterns from positive bags and least negative patterns from identified negative bags to form PMP.

According to MIL definition, a negative bag does not contain positive instances, and negative instances in negative bags can have very general distributions, we use a Weighted Kernel Density Estimator [8] (WKDE) to model the distributions of negative instances in reliable negative bags as follows.

$$p(\mathbf{x}|\mathcal{B}^-) = \frac{1}{\sum_i n_i^-} \sum_{i,j} K(w_x \mathbf{x}, w_j \mathbf{x}_{i,j}^-) \quad (3)$$

where w_x denotes weight of the bag to which the instance x belongs to. $\mathbf{x}_{i,j}^-$ denotes the j th instance of the i th reliable negative bag with its size to be n_i^- , and we use an isotropic Gaussian kernel K . Depending on whether the underlying bag is positive or is identified as negative, the “most positive pattern” or “least negative pattern” \mathbf{x}_i^B in PMP can be obtained by Eq.(4), with \mathcal{B}^- denoting the set of all identified reliable negative bags.

$$\mathbf{x}_i^B = \arg \min_{\mathbf{x}_{i,j} \in B_i, j=1, \dots, n_i} p(\mathbf{x}_{i,j}|\mathcal{B}^-) \quad (4)$$

After that, the optimization problem in Eq. (2) can be solved through standard linear SVM. In other words, once weight values are fixed, a weighted SVM classifier can be trained from PMP for classification.

4.3 Self-Adaptive Bag Weight Optimization

In the proposed puMIL learning framework, a weight confidence value w_i is assigned to each bag to determine whether an unlabeled bag is likely being a reliable negative bag or not. In this section, we propose a self-adaptive process to search optimal bag weight values. Because the aim of puMIL is to propose a learning framework to maximize the performance measure (F-score), a good weight combination should corresponds to a high F-score, which trades off the precision P and the recall R : $\text{F-score} = 2 \times P \times R / (P + R)$. Accordingly, we drive a self-adaptive strategy to obtain optimal weights based on a clone selection theory in Artificial Immune Systems.

When pathogens invade the body, antibodies that are produced from B-cells will respond for the detection of a foreign antigen. This response process can be explained by clonal selection theory. More specifically, immune individuals with high affinity will gradually increase during the clone and mutation process. Meanwhile, some immune individuals will polarize into memory individuals, which will be evolved towards final optimal. In our solution, antigen is simulated as training bag set \mathcal{B} , while the antibody, presented by confidence weight vector $\hat{\mathbf{w}}$ will experience a form of clone and mutation. The evolving optimization process will help discover optimal bag weight values \mathbf{w}^* which will lead to the best affinity (*i.e.* F-score). The details of the optimal weight search process are described as:

Algorithm 1. puMIL: Positive and Unlabeled Multi-Instance Learning**Input:**

Antibody Population (weight vectors): $\hat{\mathcal{W}} = \{\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_L\}$;
 Antigen Population (bags): $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^u$;
 Clone Factor c ; Threshold T ; Maximum Evolution Iterations M ;

Output:

The target class label y_s of a test bag B_s ;

// **Training Phase:**

```

1:  $\hat{\mathcal{W}} \leftarrow$  Initialize  $L$  weight vectors (Step (1) in Section 4.3).
2:  $t \leftarrow 1$ ;
3: while  $\{(t \leq M \text{ and } f[\hat{\mathbf{w}}_c^{t+1}] - f[\hat{\mathbf{w}}_c^t] \geq T) \text{ OR } (t=1)\}$  do
4:    $(\mathcal{B}_i^-)^t \leftarrow$  Use  $\hat{\mathbf{w}}_i^t$  to find unlabeled bags with the largest weight values to form reliable negative
     bag set (which has the same number of bags as  $\mathcal{B}^+$ ).
5:    $(\mathcal{D}_i^-)^t \leftarrow$  Generate a weighted distribution from the  $(\mathcal{B}_i^-)^t$  via Eq. (3).
6:    $PMP_i^t \leftarrow$  Apply  $(\mathcal{D}_i^-)^t$  to  $\mathcal{B}^+$  and  $(\mathcal{B}_i^-)^t$  and obtain positive margin pool via Eq. (4).
7:    $f[\hat{\mathbf{w}}_i^t] \leftarrow$  Apply  $PMP_i^t$  to  $\mathcal{B}^u$  and calculate the affinity score for  $\hat{\mathbf{w}}_i^t$  via Eq. (2).
8:    $\hat{\mathbf{w}}_c^t \leftarrow$  Apply  $f[\hat{\mathbf{w}}_i^t]$  to  $\hat{\mathcal{W}}^t$  and find weight vector  $\hat{\mathbf{w}}_c^t$  with the highest affinity score.
9:    $\hat{\mathcal{W}}^{t+1} \leftarrow$  Apply  $\hat{\mathbf{w}}_c^t$  to  $\hat{\mathcal{W}}^t$  via Steps (2-4) in Section 4.3 and update weight population.
10:   $\hat{\mathbf{w}}_c^{t+1} \leftarrow$  Apply  $f[\hat{\mathbf{w}}_i^{t+1}]$  to  $\hat{\mathcal{W}}^{t+1}$  and update the best weight vector.
11:   $t \leftarrow t + 1$ .
12: end while
13:  $\mathbf{w}_c^* \leftarrow \hat{\mathbf{w}}_c^t$  // The final optimal weight vector
// Testing Phase:
14:  $\mathcal{H} \leftarrow$  Apply the global optimal  $\mathbf{w}_c^*$  to all instances in the underlying PMP to build a weighted
   SVM classifier  $\mathcal{H}(\mathbf{x}^B) = \boldsymbol{\omega}^\top \mathbf{x}^B + b$  with  $\boldsymbol{\omega} = \sum_i \alpha_i \mathbf{w}_{c,i}^* y_i \mathbf{x}_i^B$ .
15:  $y_s \leftarrow$  Apply  $\mathcal{H}$  to each instance in test bag  $B_s$  to predict its bag label.
```

1) **Initialization:** Given an MI set \mathcal{B} with n bags, a set of weight vectors $\hat{\mathcal{W}} = \{\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_L\}$ are randomly generated, with each vector $\hat{\mathbf{w}}_i = \langle \hat{w}_{i,1}, \dots, \hat{w}_{i,k}, \dots, \hat{w}_{i,n} \rangle$ where $\hat{w}_{i,k}$ represents a weight value of bag $B_k \in \mathcal{B}$. If B_k is a positive bag, $\hat{w}_{i,k}$ is set to 1. If B_k is an unlabeled bag, $\hat{w}_{i,k}$ is a random value within range $(0, 1]$. It is worth noting that each weight vector $\hat{\mathbf{w}}_i$ contains initial weight value for all bags. Because there are L weight vectors, each bag will have L initial weight values.

2) **Antibody Clone:** For each weight vector $\hat{\mathbf{w}}_i^t$ in the t th generation, if we use their weight values as the bag weights as defined in Eq. (1), each $\hat{\mathbf{w}}_i^t$ will correspond to an SVM classifier with an affinity score (*i.e.* F-score) on the unlabeled bag set \mathcal{B}^u . This score allows us to assess the quality of weight vectors and use clone selection to find optimal weight vector. During antibody clone process, the individual $\hat{\mathbf{w}}_c^t \in \hat{\mathcal{W}}^t$ with the best affinity score will be selected as the memory antibody to be further cloned. To ensure a fixed population size in every generation, $\hat{\mathbf{w}}_c^t$ will be cloned under the clone factor c to replace weight vectors $\hat{\mathbf{w}}_j^t \in \hat{\mathcal{W}}^t$ with low affinity under the same rate c .

3) **Antibody Mutation:** In order to maintain the diversity of the weight vectors, the mutation operation should be applied. Specifically, for any $\hat{\mathbf{w}}_i^t$ from the t th generation, the new variation individual \mathbf{v}_i^{t+1} can be generated as follows:

$$\mathbf{v}_i^{t+1} = \hat{\mathbf{w}}_i^t + F * N(0, 1) * (\hat{\mathbf{w}}_c^t - \hat{\mathbf{w}}_i^t) \quad (5)$$

Among them, $N(0,1)$ is a normally distributed random variable within $[0,1]$. $F = 1 - f[\hat{\mathbf{w}}_i^t]$, as the variation factor, can be adaptively obtained according to different clones [9] where $f[\hat{\mathbf{w}}_i^t]$ denotes the affinity of $\hat{\mathbf{w}}_i^t$.

4) **Antibody Update:** This process determines whether the variation of weight vectors (from above steps (2) and (3)) can be used to replace a target weight vector in the next generation. In our algorithm, we adopt a greedy search strategy. Only if the affinity of \mathbf{v}_i^{t+1} is better than that of the target weight vector $\hat{\mathbf{w}}_i^t$, the new weight vector is then selected as the offspring.

4.4 puMIL Framework

Algorithm 1 reports the detailed process of the proposed puMIL framework, which combines the (1) PMP construction (Section 4.2); and (2) adaptive weight optimization (Section 4.3), to iteratively refine unlabeled bags for learning.

During the initial process, puMIL will initialize the bag weight vectors \mathcal{W} (line 1). During each *while* loop, puMIL will first select the “reliable negative bags” $(\mathcal{B}_i^-)^t$ (this set has the same size as the number of positive bags \mathcal{B}^+) for each weight vector $\hat{\mathbf{w}}_i^t$ in $\hat{\mathcal{W}}^t$. So $(\mathcal{B}_i^-)^t$ consists of unlabeled bags with the largest weight values in $\hat{\mathcal{W}}_i^t$ (line 4). After that, puMIL will form a positive margin pool for each weight vector (lines 5-6). The clone selection theory will be employed to update the weight vectors in order to find a better weight value in the next iteration (lines 7-10). The evolutionary process will repeat until (1) the algorithm surpasses the pre-set maximum number M , or (2) the same result is obtained for a number (*e.g.* T) of consecutive iterations.

After obtaining the best weight vector \mathbf{w}_c^* , puMIL uses the discovered optimal weight values to obtain PMP, and then builds a weighted SVM classifier \mathcal{H} . A test bag is classified as positive if one or multiple instances inside the bag are classified as positive, and negative otherwise.

5 Experiments

To evaluate the effectiveness of our puMIL framework, we use F-score as the evaluation metric (its definition is given in Section 4.3). For benchmark data sets used in the experiments, 30% of bags are randomly selected as testing set in each run, with the remaining bags being used as training set. All results are based on the average performance over 10 repetitions. Besides, the two parameters M and T in Algorithm 1 are set to 50 and 0.001, respectively.

Because there is no existing method for positive and unlabeled multi-instance learning, for comparison purposes we implement following baseline approaches from bag- and instance-level perspectives. The former directly employs Positive-Unlabeled learning [5] at the bag level, and the latter propagates bag label to instances and transfer MIL as a generic Positive-Unlabeled learning problem.

Bag-Level Approaches: (a) **MILR+MISVM-MI:** This method first labels all unlabeled bags as negative bags, and then uses MI learning approach MILR [10], which outputs probability estimation, to obtain a set of identified negative bags. After that, it runs MISVM [6] iteratively on the positive set and refined negative set until converges; and (b) **Spy+MISVM-MI:** The difference between Spy based PU learning [5] on bag-level and MILR+MISVM-MI is in its initialization.

Spy+MISVM-MI randomly samples a set of positive bags as “spies”, and marks them as unlabeled bags. Because spies are genuinely positive and behave similarly to unknown positive samples the unlabeled set, adding spies allows algorithm to infer characteristics of unknown positive bags in the unlabeled set. Previous studies [5] on text documents have demonstrated good performance of this PU learning strategy.

Instance-Level Approaches: **(a) NB+SVM-MI.** The variation of traditional PU setting in [5,11] is used for comparison. Specifically, a Naive Bayes classifier [12] is used to obtain identified negative instance set from unlabeled instance set. After that, an iterative process is used to train SVM classifier from positive instances and identified negative instances; and **(b) Spy+SVM-MI** Similar to the approaches used in [5,11], during the initialization process, Spy+SVM-MI randomly selects a set of positive instances as “spies”, and adds them into the unlabeled set. After that it follows the same procedure as NB+SVM-MI.

For ease of understanding, we also refer to MILR+MISVM-MI and NB+SVM-MI as “no-spy” based approaches, and Spy+MISVM-MI and Spy+SVM-MI are called “spy” based approaches.

5.1 Drug Activity Prediction

The objective of drug activity prediction is to predict whether a drug molecule can bind well to a target protein related to certain disease states, which are primarily determined by the shape of the molecule. Our drug activity prediction data set, MUSK1 [1], is a benchmark used for MI learning. It contains 476 instances grouped into 92 bags (45 inactive and 47 active), with each instance being described by a 166-dimensional feature vector. In our experiments, we use $r \times 100\%$ active bags as positive bags, and the remaining active bags and all inactive bags are treated as unlabeled bags. The results for the two types of baselines (bag-level and instance-level) and proposed puMIL with different r values, varying from 0.1 to 0.7, are reported in Figures 2(A) and 2(B), respectively.

Overall, “no-spy” based approaches achieve competitive F-score as “spy” based methods, which demonstrates that simply adding “spy” to unlabeled bags may not help differentiate positive *vs.* negative bags. Meanwhile, the proposed puMIL achieves better performances compared to other methods, especially for bag-level baselines. For small r values, such as $r = 0.2$ or less, puMIL demonstrates very significant performance gain, compared to other baselines. This further assets that when the number of positive bags is very limited, leveraging useful information from unlabeled bags, like puMIL does, can be very useful for positive and unlabeled multi-instance learning.

5.2 Scientific Publication Retrieval

The DBLP data set consists of bibliography data in computer science¹. Each record in DBLP contains a number of attributes such as abstract, authors, title, and references [13]. To build a puMIL task, we select papers published in

¹ <http://dblp.uni-trier.de/xml/>

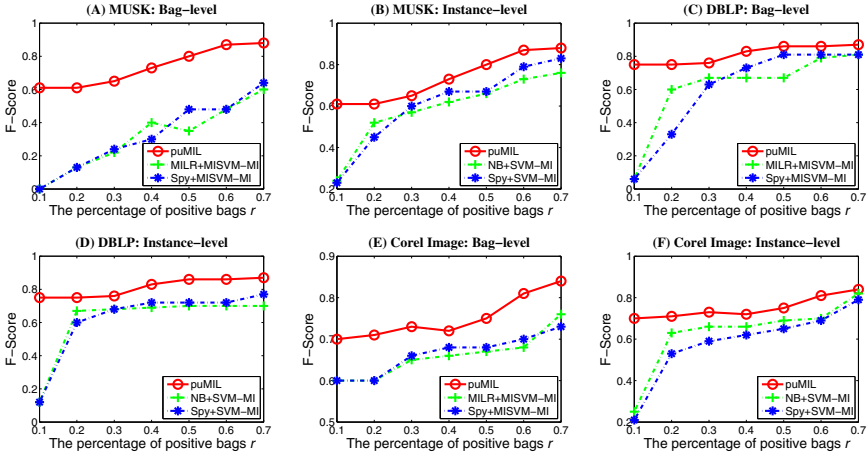


Fig. 2. F-score comparisons on **MUSK**, **DBLP**, and **Corel Image** data sets with respect to different r values. Where (A) and (B) represent MUSK data, (C) and (D) represent DBLP data, and (E) and (F) represent Corel Image data.

Artificial Intelligence field (AI: IJCAI, AAAI, NIPS, UAI, COLT, ACL, KR, ICML, ECML and IJCNN) as positive bags and randomly select papers from other fields, such as computer vision, multimedia, pattern recognition as unlabeled bags. A “bag-of-words” representation based on TFIDF [14] is adopted to convert the abstract of a paper to an instance. So each paper is a bag and each instance inside the bag denotes either the paper’s abstract or the abstract of a reference cited in the paper.

In our experiments, we choose 200 papers in total (which correspond to 200 bags), with each paper containing 1 to 10 references. For all 200 papers, the total number of references cited (*i.e.* instances) is 1136. Each instance is described by a 4497-dimensional feature vector. To vary the number of positive bags, we randomly select $r \times 100\%$ of AI bags (varying from 0.1 to 0.7) as positive bag set, and the remaining AI bags and all other bags are used as unlabeled bags. In Figures 2(C) and 2(D), we report the results with respect to different r values. The results show that “spy” based approaches can achieve a slightly better performance than ‘no-spy’ versions when r is greater than 0.4. This is possibly because that when a large number of positive bags are used, the “spies” selected from the positive bags will not reduce the role of positive bags. Meanwhile, puMIL clearly outperforms all baselines, especially when only a small portion of positive bags are labeled (*i.e.* $r=0.1$). This suggests that puMIL is effective over a wide range of percentage of labeled positive bags.

5.3 Region-Based Image Annotation

In the third experiment, we report puMIL’s performance for automatic image annotation tasks. The original data are color images from the Corel data set [15] that have been preprocessed and segmented using Blobworld system [16].

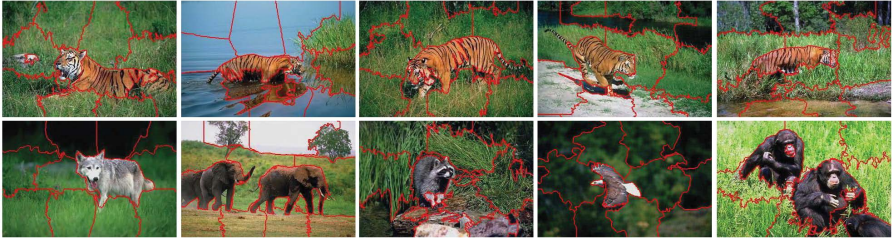


Fig. 3. Example images from the Corel image database used in the experiment. The first and the second rows show positive and negative objects, respectively. Each image is considered as a bag with each region denoting an instance.

An image contains a set of segments (or blobs), each characterized by color, texture, and shape descriptors. In this case, each image is considered as a bag, and each region inside the image denotes an instance. Some example images from the database are shown in Figure 3. In our experiment, we use category “tiger” as positive bags (100 bags with 544 instances) and randomly draw 100 photos of other animals to form unlabeled bags with 676 instances. Each instance is described by a 230-dimensional feature vector which represent color, texture, and shape of the region. To validate the performance of puMIL with respect to different number of positive bags, we randomly select $r \times 100\%$ “tiger” images (varying from 0.1 to 0.7) as positive bags, and combine remaining “tiger” images and all other images as unlabeled bag set.

In Figures 2(E) and 2(F), we compare the performance of puMIL with two types of baselines by using different portions of positive bags. The results show that bag-level baselines can have a high F-score than instance-level methods when the percentage of positive bag is very small (*e.g.* $r < 0.2$). Because instance-level approaches directly assign bag labels to all instances in positive bags, for a small number of positive bags, the mislabeled instances in positive bags will have severe impact on the classifiers trained from labeled data. As a result, the classifier may not be able to accurately differentiate positive vs. negative instances. However, as the r values become sufficiently large, instance-level methods demonstrate better performance than bag-level approaches. By properly utilizing information in unlabeled bags, puMIL consistently outperforms all baselines for different percentages of positive bags.

6 Conclusions

In this paper, we formulated a unique multi-instance learning task, which only has positive and unlabeled bags available for multi-instance learning. This problem setting is more general but significantly more challenging than traditional multi-instance learning because no negative bags exist for deriving discriminative classification models. To address the challenge, we proposed a puMIL framework which self-adaptively selects some reliable negative bags (from unlabeled bags) and further selects some representative patterns from the positive bags and identified negative bags to help train SVM classifiers. The classifiers will further help

refine the selection of negative bags and iteratively lead to updated classifiers for classification. Our main technical contribution, compared to existing research, is threefold: (1) a general framework to handle multi-instance learning with only positive and unlabeled bags; (2) an effective algorithm to identify reliable negative bags from unlabeled bags; and (3) an effective approach for utilizing unreliable labels derived for unlabeled bags.

Acknowledgments. The work was supported by the Key Project of the Natural Science Foundation of Hubei Province, China (Grant No. 2013CFA004), and the National Scholarship for Building High Level Universities, China Scholarship Council (No. 201206410056).

References

1. Dietterich, T., Lathrop, R., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* 89, 31–71 (1997)
2. Zhou, Z., Zhang, M., Huang, S., Li, Y.: Multi-instance multi-label learning. *Artificial Intelligence* 176, 2291–2320 (2012)
3. Qi, X., Han, Y.: Incorporating multiple SVMs for automatic image annotation. *Pattern Recogn.* 40, 728–741 (2007)
4. Zhang, B., Zuo, W.: Learning from positive and unlabeled examples: A survey. In: *ISIP*, pp. 650–654 (2008)
5. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: *ICDM*, Washington, DC, USA, p. 179 (2003)
6. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *NIPS*, pp. 561–568 (2003)
7. Shang, R., Jiao, L., Liu, F., Ma, W.: A novel immune clonal algorithm for mo problems. *IEEE Trans. Evol. Comput.* 16(1), 35–50 (2012)
8. Fu, Z., Robles-Kelly, A., Zhou, J.: Milis: Multiple instance learning with instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.* 33(5), 958–977 (2011)
9. Zhong, Y., Zhang, L.: An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* 50(3), 894–909 (2012)
10. Ray, S., Craven, M.: Supervised versus multiple instance learning: an empirical comparison. In: *ICML*, New York, NY, USA, pp. 697–704 (2005)
11. Zhao, Y., Kong, X., Yu, P.S.: Positive and unlabeled learning for graph classification. In: *ICDM*, pp. 962–971 (2011)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Mach. Learn.* 29(2-3), 131–163 (1997)
13. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: *KDD*, pp. 990–998 (2008)
14. He, J., Gu, H., Wang, Z.: Bayesian multi-instance multi-label learning using gaussian process prior. *Mach. Learn.* 88(1-2), 273–295 (2012)
15. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(6), 985–1002 (2008)
16. Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., Malik, J.: Blobworld: A system for region-based image indexing and retrieval. In: Huijsmans, D.P., Smeulders, A.W.M. (eds.) *VISUAL 1999*. LNCS, vol. 1614, pp. 509–517. Springer, Heidelberg (1999)