Multiple-Instance Learning with Instance Selection via Dominant Sets

Aykut Erdem and Erkut Erdem

Hacettepe University, 06800 Beytepe, Ankara, Turkey
aykut.erdem@hacettepe.edu.tr, erkut@cs.hacettepe.edu.tr

Abstract. Multiple-instance learning (MIL) deals with learning under ambiguity, in which patterns to be classified are described by bags of instances. There has been a growing interest in the design and use of MIL algorithms as it provides a natural framework to solve a wide variety of pattern recognition problems. In this paper, we address MIL from a view that transforms the problem into a standard supervised learning problem via instance selection. The novelty of the proposed approach comes from its selection strategy to identify the most representative examples in the positive and negative training bags, which is based on an effective pairwise clustering algorithm referred to as dominant sets. Experimental results on both standard benchmark data sets and on multi-class image classification problems show that the proposed approach is not only highly competitive with state-of-the-art MIL algorithms but also very robust to outliers and noise.

1 Introduction

In recent years, multiple-instance learning (MIL) [7] has emerged as a major machine learning paradigm, which aims at classifying bags of instances with class label information available for the bags but not necessarily for the instances. In a typical MIL setting, a *negative bag* is composed of only negative instances, whereas a bag is considered *positive* if it contains at least one positive instance, leading to a learning problem with ambiguously labeled data. MIL paradigm provides a natural framework to handle many challenging problems in various domains, including drug-activity prediction [7], document classification [1], content-based image retrieval [25], object detection [21], image categorization [5,4], and visual tracking [2,10].

In general, MIL methods can be grouped into two main categories. The first class of approaches, including the APR [7], DD [16], EM-DD [24] methods, uses *generative* models to represent the target concept by a region in the instance feature space which covers all the true positive instances while remaining far from every instance in the negative bags. Alternatively, the second class of works employs *discriminative* learning paradigm to address the MIL problems. The methods in this group are mainly the generalizations of the standard single-instance learning (SIL) methods to the MIL setting, *e.g.* mi-SVM and MI-SVM [1], MI-Kernel [9], MIO [12], Citation KNN [22] and MILBoost-NOR [21].

Recently, a new group of SVM-based methods has been proposed for MIL, namely the DD-SVM [5], MILES [4], MILD_B [13] and MILIS [8] methods, which tackles multi-instance problems by transforming them into standard SIL problems. The basic

idea is to embed each bag into a feature space based on a representative set of instances selected from the training bags and to learn a classifier in this feature space. The major difference between these methods is how they select instance prototypes, which will be detailed in the next section. However, it should be noted here that a good set of prototypes is vital to the success of any method.

In this paper, a new instance selection mechanism is proposed for multiple-instance learning. The novelty comes from utilizing *dominant sets* [18], an effective pairwise clustering framework, to model the distributions of negative instances and accordingly to select a set of instance prototypes from the positive and negative training bags. Therefore, the proposed approach is named MILDS, Multiple-Instance Learning with instance selection via Dominant Sets. The main contributions are as follows: (i) The constructed feature space is usually of a lower dimension compared to those of other instance-selection based MIL approaches [5,4,8]. This is mainly due to the use of clustering performed on the instances from the negative training bags. (ii) The presented approach is highly insensitive to noise in the bag labels as the dominant sets framework is proven to be very robust against outliers that might exist in the data. (iii) The proposed binary MIL formulation can be easily generalized to solve multi-class problems in a natural way due to the proposed cluster-based representation of data. (iv) The experimental results demonstrate that the suggested approach is highly competitive with the state-of-the-art MIL approaches.

The remainder of the paper is organized as follows: Section 2 summarizes the previous work on instance-selection based MIL and provides background information on the dominant sets framework. Section 3 presents the proposed MILDS algorithm. Section 4 reports experimental results on some benchmark data sets and on multi-class image classification problems. Finally, Section 5 concludes the paper with a summary and possible directions for future work.

2 Background

2.1 Instance-Selection Based MIL

As mentioned in the introduction, the existing instance-selection based MIL methods, namely DD-SVM [5], MILES [4], MILD_B [13] and MILIS [8], can be differentiated mainly by the procedures they follow in identifying the set of instance prototypes used to map bags into a feature space. Below, we review these differences in detail.

In DD-SVM, a diverse density (DD) function [16] is used in identifying the instance prototypes. Within each training bag, the instance having the largest DD value is chosen as a prototype for the class of the bag. Then, a standard SVM in combination with radial basis function (RBF) is trained on the corresponding embedding space. The performance of DD-SVM is highly affected by the labeling noise since a negative bag close to a positive instance drastically reduces the DD value of the instance, thus its chance to be selected as a prototype.

In MILES, there is no explicit selection of instance prototypes. All the instances in the training bags are employed to build a very high-dimensional feature space, and then the instance selection is implicitly performed via learning a 1-norm SVM classifier. As

expected, the main drawback of MILES stems from its way of constructing the embedding space. Its computational load grows exponentially as the volume of the training data increases.

In [13], an instance-selection mechanism based on a conditional probability model is developed to identify the true positive instance in a positive bag. For each instance in a positive bag, a decision function is formulated whose accuracy on predicting the labels of the training bags is used to measure true positiveness of the corresponding instance. The authors of [13] use this instance selection mechanism to devise two MIL methods, MILD_I and MILD_B, for *instance-level* and for *bag-level* classification problems, respectively. Here, MILD_B is of our interest, which defines the instance-based feature space by the most positive instances chosen accordingly from each positive bag, and like DD-SVM, trains a standard SVM with the RBF kernel in that feature space.

In MILIS, instances in the negative bags are modeled as a probability distribution function based on kernel density estimation. Initially, the most positive (*i.e.* the least negative) instance and the most negative instance are selected respectively in each positive bag and each negative bag based on the distribution estimate. These instance prototypes form the feature space for the bag-level embedding in which a linear SVM is trained. To increase the robustness, once a classifier is learnt, MILIS employs an alternating optimization scheme for instance selection and classifier training to update the selected prototypes and the weights of the support vectors. As a final step, it includes an additional feature pruning step which removes all features with small weights.

2.2 Clustering with Dominant Sets

Our instance selection strategy makes use of a pairwise clustering approach known as *dominant sets* [18]. In a nut shell, the concept of a dominant set can be considered as a generalization of a maximal clique to edge-weighted graphs. Suppose the data to be clustered is represented in terms of their similarities by an undirected edge-weighted graph with no self-loops $G = (V, E, w)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and $w : E \rightarrow \mathbb{R}_+$ is the positive weight (similarity) function. Further, let $A = [a_{ij}]$ denote the $n \times n$ adjacency matrix of G where $a_{ij} = w(i, j)$ if $(i, j) \in E$ and is 0 otherwise. A dominant set is formulated based on a recursive characterization of the weight $\mathrm{w}_S(i)$ of element i w.r.t. to a set of elements S (A curious reader may refer to [18] for more details), as:

Definition 1. *A nonempty subset of vertices $S \subseteq V$ such that $\sum_{i \in T} \mathrm{w}_T(i) > 0$ for any nonempty $T \subseteq S$, is said to be* dominant *if:*

1. $\mathrm{w}_S(i) > 0$, *for all* $i \in S$,
2. $\mathrm{w}_{S \cup \{i\}}(i) < 0$, *for all* $i \notin S$.

The above definition of a dominant set also formalizes the notion of a *cluster* by expressing two basic properties: (i) elements within a cluster should be very similar (*high internal homogeneity*), (ii) elements from different clusters should be highly dissimilar (*high external inhomogeneity*).

Consider the following generalization of the Motzkin-Straus program [17] to an undirected edge-weighted graph $G{=}(V, E, w)$:

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \Delta \end{aligned} \tag{1}$$

where A is the weighted adjacency matrix of graph G, $\Delta{=}\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}{\geq}\mathbf{0} \text{ and } \mathbf{e}^T\mathbf{x}{=}1\}$ is the standard simplex in $\mathbb{R}^n$ with $\mathbf{e}$ being a vector of ones of appropriate dimension. The support of $\mathbf{x}$ is defined as the set of indices corresponding to its positive components, *i.e.* $\sigma(\mathbf{x}) = \{i{\in}V \mid x_i > 0\}$. The following theorem (from [18]) provides a one-to-one relation between dominant sets and strict local maximizers of (1).

Theorem 1. *If S is a dominant subset of vertices, then its weighted characteristic vector $\mathbf{x} \in \Delta$ defined as:*

$$x_i = \begin{cases} \frac{\mathrm{w}_S(i)}{\sum_{j \in S} \mathrm{w}_S(j)} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

is a strict local solution of (1). Conversely, if $\mathbf{x}$ is a strict local solution of (1), then its support $S = \sigma(\mathbf{x})$ is a dominant set, provided that $\mathrm{w}_{S\cup\{i\}}(i) \neq 0$ for all $i \notin S$.

The cohesiveness of a dominant set (cluster) S can be measured by the value of the objective function $\mathbf{x}^T A\mathbf{x}$. Moreover, the similarity of an element j to S can be directly computed by $(A\mathbf{x})_j$ where

$$(A\mathbf{x})_j \begin{cases} = \mathbf{x}^T A\mathbf{x} \text{ if } j{\in}\sigma(\mathbf{x}) \\ \leq \mathbf{x}^T A\mathbf{x} \text{ if } j{\notin}\sigma(\mathbf{x}) \end{cases}. \tag{3}$$

As a final remark, it should be noted that the spectral methods in [20,11] maximize the same quadratic function in Eq. (1). However, they differ from dominant sets in their choice of the feasible region. The solutions obtained with these methods are constrained to lie in the sphere defined by $\mathbf{x}^T\mathbf{x} = 1$ instead of the standard simplex Δ used in the dominant sets framework. This subtle difference is crucial for our practical purposes. First, the components of the weighted characteristic vector give us a measure of the participation of the corresponding data points in the cluster. Second, this constraint provides robustness against noise and outliers [18,15].

3 Proposed Method

In this section, we present a novel multiple-instance learning framework called MILDS, which transforms a MIL problem into a SIL problem via instance selecting. Unlike the similar approaches in [5,4,13,8], it makes use of the *dominant sets* clustering framework [18] for instance selection to build a more effective embedding space. We first restrict ourselves to the *two-class* case. However, as will be described later in Section 3.4, extension to *multi-class* MIL problems is quite straightforward.

3.1 Notations

Let $B_i = \{B_{i1}, \ldots, B_{ij}, \ldots, B_{in_i}\}$ denote a bag of instances where B_{ij} denotes the jth instance in the bag, and $y_i \in \{+1, -1\}$ denote the label of bag i. For the sake of simplicity, we will denote a positive bag as B_i^+ and a negative bag as B_i^-. Further, let $\mathcal{B} = \{B_1^+, \ldots, B_{m^+}^+, B_1^-, \ldots, B_{m^-}^-, \}$ denote the set of m^+ positive and m^- negative training bags. Note that each bag may contain different number of instances, and each instance may have a label which is not directly observable.

3.2 Instance Selection with Dominant Sets

Recall the two assumptions of the classical MIL formulation that a bag is positive if it contains at least one positive instance, and all negative bags contains only negative instances [7]. This means that positive bags may contain some instances from the negative class but there is no such ambiguity in the negative bags (provided that there is no labeling noise). Just like in [13,8], our instance selection strategy is heavily based on this observation. However, unlike those approaches, to select the representative set of instances we do not explicitly estimate either a probability density function or a conditional probability. Instead, we try to model the negative data by clustering the instances in the negative bags, and then making decisions according to the distances to the extracted clusters. As will be clear throughout the paper, the dominant sets framework provides a natural scheme to carry out these tasks in an efficient way.

Denote $\mathcal{N} = \{I_i \mid i = 1, \ldots, M\}$ as the collection of negative instances from all of the negative training bags, *i.e.* the set defined by $\left\{B_{ij}^- \in B_i^- \mid i = 1, \ldots, m^-\right\}$. Construct the matrix $A = [a_{ij}]$ composed of the similarities between the negative instances as:

$$a_{ij} = \begin{cases} exp\left(-\frac{d(I_i, I_j)^2}{2\sigma^2}\right) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $d(\cdot, \cdot)$ is a distance measure that depends on the application and σ is a scale parameter. In the experiments, the Euclidean distance was used.

To extract the clusters in $\mathcal{N}$, the iterative *peeling-off strategy* suggested in [18] is employed. In specific, at each iteration, a dominant set (a cluster) is found by solving the quadratic program in (1). Then, the instances in the cluster are removed from the similarity graph, and this process is reiterated on the remaining instances. In theory, the clustering process stops when all the instances are covered, but in dealing with large and noisy data sets, this is not very practical. Hence, in our experiments, an upper bound on the number of extracted clusters was introduced that at most m^- (*i.e.* the number of negative bags) most coherent dominant sets were selected according to internal coherency values measured by the corresponding values of the objective function. Notice that, in this way, *instance pruning* is carried our in an early stage. This is another fundamental point which distinguishes our work from the approaches in [4,8] as these two methods perform instance pruning implicitly in the SVM training step. Moreover, this provides robustness to noise and outliers.

Suppose $\mathcal{C} = \{C_1, \ldots, C_k\}$ denotes the set of clusters extracted from the collection of negative training instances $\mathcal{N}$. A representative set for $\mathcal{N}$ is found by selecting

one prototype from each cluster $C_i \in \mathcal{C}$. Recall that each cluster C_i is associated with a characteristic vector $\mathbf{x}^{C_i}$ whose components give us a measure of the participation of the corresponding instances in the cluster [18]. Hence, the instance prototype z_i^- representing the cluster C_i is identified based on the corresponding characteristic vector $\mathbf{x}^{C_i}$ as:

$$z_i^- = I_{j^*} \text{ with } j^* = \underset{j \in \sigma(\mathbf{x}^{C_i})}{\arg\max}\, x_j^{C_i} \quad . \tag{5}$$

In selecting the representative instances for the positive class, however, the suggested clustering-based selection strategy makes no sense on the collection of positive bags because the bags may contain some negative instances which may collectively form one or more clusters, thus if applied, the procedure may result in some instance prototypes belonging to the negative class. Hence, for selecting prototypes for the positive class, a different strategy is employed. In particular, the most positive instance in each positive bag is identified according to its relationship to the negative training data.

For a positive bag $B_i^+ = \left\{B_{i1}^+, \ldots, B_{in_i^+}^+\right\}$, let $A^\dagger$ be an $n_i^+ \times |\mathcal{N}|$ matrix composed of the similarities between the instances in B_i^+ and the negative training instances in $\mathcal{N}$, computed like in (4). The *true positive* (*i.e.* the *least negative*) instance in B_i^+, denoted with z_i^+, is picked as the instance which is the most distant from the extracted negative clusters in $\mathcal{C}$ as follows:

$$z_i^+ = B_{ij^*}^+ \text{ with } j^* = \underset{j=1,\ldots,n_i^+}{\arg\min} \frac{\sum_{\ell=1,\ldots,k} (A^\dagger \mathbf{x}^{C_\ell})_j \times |C_\ell|}{\sum_{\ell=1,\ldots,k} |C_\ell|} \tag{6}$$

where the term $(A^\dagger \mathbf{x}^{C_\ell})_j$ is the weighted similarity of the instance B_{ij}^+ to the cluster C_ℓ, and $|\cdot|$ denotes the cardinality of the set[1]. Intuitively, in (6), larger clusters have more significance in the final decision than the smaller ones.

To illustrate the proposed selection process, consider the two-dimensional synthetic data given in Fig. 1(a). It contains 8 positive bags and 8 negative bags, each having at least 8 and at most 10 instances. Each instance is randomly drawn from one of the five normal distributions: $\mathcal{N}([4,8]^T, I)$, $\mathcal{N}([0,4]^T, I)$, $\mathcal{N}([-1,12]^T, I)$, $\mathcal{N}([-4,-2]^T, I)$ and $\mathcal{N}([6,2]^T, I)$ with I denoting the identity matrix. A bag is labeled positive if it contains at least one instance from the first two distributions. In Fig. 1(a), positive and negative instances are respectively represented by crosses and circles, and drawn with colors showing the labels of the bags they belong: blue for positive and red for negative bags. The result of the proposed instance selection method is given in Fig. 1(b). The extracted negative clusters are shown in different colors, and the selected instance prototypes are indicated by squares. Notice that the dominant sets framework correctly captured the multi-modality of the negative class, and the prototypes selected from the extracted clusters are all close to the centers of the given negative distributions. Moreover, the true positive instances in the positive bags were successfully identified.

[1] Note that since the zero-components of $\mathbf{x}^{C_\ell}$ have no effect on estimating z_i^+s, in practice highly reduced versions of $A^\dagger$s are utilized in the computations.

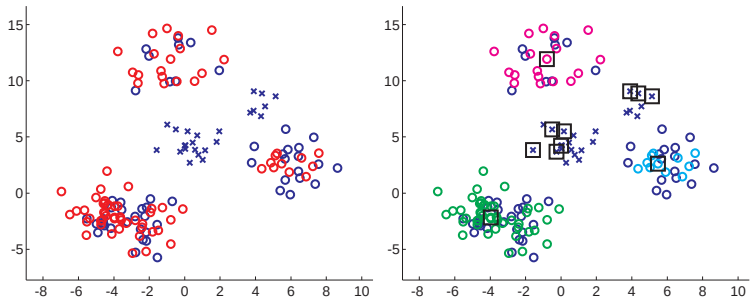

Fig. 1. Synthetic data set (best viewed in color). (a) Raw data. (b) The instance selection process.

3.3 Classification

We can now describe our classification scheme. Suppose $\mathcal{Z} = \left\{z_1^-, \ldots, z_k^-, z_1^+, \ldots, z_{m^+}^+\right\}$ denote the set of selected instance prototypes, where k is the number of extracted negative clusters, m^+ is the number of positive training bags[2]. A similarity measure $s(z, B_i)$ between a bag B_i and an instance prototype z is defined by

$$s(z, B_i) = \max_{B_{ij} \in B_i} exp\left(-\frac{d(z, B_{ij})^2}{2\sigma^2}\right) \tag{7}$$

which calculates the similarity between z and its nearest neighbor in B_i. Then, we define an embedding function φ which maps a bag B to a $(k+m^+)$-dimensional vector space by considering the similarities to the instance prototypes:

$$\varphi(B) = \left[s(z_1^-, B), \ldots, s(z_k^-, B), s(z_1^+, B), \ldots, s(z_{m^+}^+, B)\right]^T \tag{8}$$

For classification, the embedding in (8) can be used to convert the MIL problem into a SIL problem. In solving the SIL counterpart, we choose to train a standard linear SVM which has a single regularization parameter C needed to be tuned. In the end, we come up with a linear classifier to classify a test bag B as:

$$f(B; \mathbf{w}) = \mathbf{w}^T \varphi(B) + b \tag{9}$$

where $\mathbf{w} \in \mathbb{R}^{|\mathcal{Z}|}$ is the weight vector, b is the bias term. The label of a test bag B is simply estimated by:

$$y(B) = sign(f(B; \mathbf{w})) \tag{10}$$

The outline of the proposed MIL framework is summarized in Algorithm 1.

[2] Note that one can always select more than one instance from each cluster or each positive bag. A detailed analysis of this issue on the performance will be reported in a longer version.

Algorithm 1: Summary of the proposed MILDS framework.

Input : Training bags $\left\{B_1^+, \ldots, B_{m^+}^+, B_1^-, \ldots, B_{m^-}^-\right\}$

1 Apply dominant sets to cluster all the instances in the negative training bags
2 Select k ($\leq m^-$) instance prototypes from the extracted k negative clusters via Eq. (5)
3 Select m^+ instance prototypes from the positive bags via Eq. (6)
4 Form the instance-based embedding in Eq. (8) using the selected prototypes
5 Train a linear SVM classifier based on the constructed feature space

Output: The set of selected instance prototypes $\mathcal{Z}$ and the SVM classifier $f(B; \mathbf{w})$ with weight $\mathbf{w}$

3.4 Extension to Multi-class MIL

The proposed approach can be straightforwardly extended to solve multi-class MIL problems by employing a *one-vs-rest* strategy. In particular, one can train c binary classifiers, one for each class against all other classes. Then, a test bag can be classified according to the classifier with the highest decision value. Note that an implementation of this idea forms a different instance-based embedding for each binary subproblem. Here, we propose a second type of embedding which results from using a set of representative instances common for all classes, as:

$$\begin{aligned}\phi(B) = [&s(z_1^1, B), s(z_2^1, B), \ldots, s(z_{m_1}^1, B),\\ &s(z_1^2, B), s(z_2^2, B), \ldots, s(z_{m_2}^2, B),\\ &\vdots\\ &s(z_1^c, B), s(z_2^c, B), \ldots, s(z_{m_c}^c, B)\,]\end{aligned} \tag{11}$$

where z_i^k is the ith instance prototype selected from class k (note that the number of prototypes may differ from class to class). In this case, training data is kept the same for all binary subproblems, only the labels differ, and this makes the training phase much more efficient. This second approach is denoted with milDS to distinguish it with the naive multi-class extension of MILDS.

In milDS, instance selection is performed as follows. Let $\mathcal{I}^k = \left\{I_i^k \mid i = 1, \ldots, M_k\right\}$ denote the collection of instances in bags belonging to class k, *i.e.* the set defined by $\{B_{ij} \in B_i \mid \text{for all } B_i \in \mathcal{B} \text{ with } y(B_i) = k\}$. First, for each class k, the pairwise similarity matrix A_k of instances $\mathcal{I}^k$ is formed, and accordingly a set of clusters $\mathcal{C}^k = \left\{C_1^k, \ldots, C_{m_k}^k\right\}$ is extracted via dominant sets framework[3]. Then, an instance prototype from each cluster C_i^k is identified according to:

$$z_i^k = I_{j^*}^k \;\text{ with }\; j^* = \underset{j \in \sigma(\mathbf{x}^{C_i^k})}{\arg\max} \; x_j^{C_i^k} / \beta_{ik}(j) \tag{12}$$

where the function $\beta_{ik}(j)$ measures the similarity of jth instance in C_i^k to all the remaining classes. The basic idea is to select the most representative element in C_i^k which

[3] In the experiments, for each class k, we extract at most m_k clusters that is equal to the number of training bags belonging to class k.

is also quite dissimilar to the remaining training data from other classes. However, here we make a simplification and estimate $\beta_{ik}(j)$ by considering only the most closest class:

$$\beta_{ik}(j) = \max_{\substack{m=1,\ldots,c \\ m \neq k}} \frac{\sum_{C_\ell^m \in \mathcal{C}^m} (A_{km} \mathbf{x}^{C_\ell^m})_j \times |C_\ell^m|}{\sum_{C_\ell^m \in \mathcal{C}^m} |C_\ell^m|} \tag{13}$$

with A_{km} denoting the $M^k \times M^m$ matrix of similarities between the instances in $\mathcal{I}^k$ and the instances in $\mathcal{I}^m$.

The embedding procedure described above gives rise to a feature space whose dimensionality is at most $\sum_k m_k$, *i.e.* the sum of the total number of clusters extracted for each class.

3.5 Computational Complexity

From a computational point of view, the most time consuming step of the proposed MILDS method and its multi-class extensions is the calculation of pairwise distances, which is also the case for [4,13,8]. In addition, there is the cost of clustering negative data with dominant sets. In this matter, a dominant set can be computed in quadratic time using the approach in [19]. An important point here is that the size of the input graphs becomes smaller and smaller at each iteration of the employed peeling off strategy, and this further introduces an increase in the efficiency of the clustering step.

4 Experimental Results

In this section, we present two groups of experiments to evaluate the proposed MILDS algorithm. First, we carry out a thorough analysis on some standard MIL benchmark data sets. Following that, we investigate image classification by casting it as a multi-class MIL problem. In the experiments, LIBSVM [3] package was used for training linear SVMs. In addition to the SVM regularization parameter C, our algorithm has only a single scale parameter σ that needs to be tuned. The best values for C and σ are selected by using n-fold cross validation from the sets $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ and $linspace(0.05\mu, \mu, 20)$, respectively, with μ being the mean distance between pair of instances in the training data and $linspace(a, b, n)$ denoting the set n linearly spaced numbers between and including a and b.

4.1 Benchmark Data Sets

We evaluate our MILDS method on five popular MIL benchmark data sets used in many multiple-instance learning studies, namely *Musk1*, *Musk2*, *Elephant*, *Fox* and *Tiger*. In *Musk1* and *Musk2*, the task is to predict drug activity from structural information. Each drug molecule is considered as a bag in which the instances represents different structural configurations of the molecule. In *Elephant*, *Fox* and *Tiger*, the goal is to differentiate images containing elephants, tigers and foxes from those that do not, respectively. Each image is considered as a bag, and each region of interest within the image as an instance. The details of the data sets are given in Table 1.

data set	bags pos./neg.	avg. inst./bag	dim
Musk1	47/45	5.17	166
Musk2	39/63	64.69	166
Elephant	100/100	6.96	230
Fox	100/100	6.60	230
Tiger	100/100	6.10	230

Table 1. Information about the MIL benchmark data sets.

For experimental evaluation, we use the most common setting, 10 times 10-fold cross validation (CV). That is, we report the classification accuracies averaged over 10 runs where the parameter selection is carried our by using 10-fold cross validation. Our results are shown in Table 2 together with those of 12 other MIL algorithms in the literature [13,8,4,5,10,12,14,1,24]. All reported results are also based on 10-fold CV averaged over 10 runs[4], with the exception of MIForest, which is over 5 runs, and MILIS and MIO, which are over 15 runs. The results demonstrate that our proposed approach is competitive with and often better than the state-of-the-art MIL methods. In three out of five MIL benchmark data sets, it outperforms several MIL approaches. However, it is more important to note that it gives the best performance among the instance-selection based MIL approaches.

Algorithm	*Musk1*	*Musk2*	*Elephant*	*Fox*	*Tiger*
MILDS	**90.9**	86.1	**84.8**	**64.3**	81.5
MILD_B [13]	88.3	86.8	82.9	55.0	75.8
MILIS [8]	88.6	91.1	*n/a*	*n/a*	*n/a*
MILES [4]	83.3	**91.6**	84.1	63.0	80.7
DD-SVM [5]	85.8	91.3	83.5	56.6	77.2
MILD_I [13]	89.9	88.7	83.2	49.1	73.4
MIForest [10]	85.0	82.0	84.0	64.0	82.0
MIO [12]	88.3	87.7	*n/a*	*n/a*	*n/a*
Ins-KI-SVM [14]	84.0	84.4	83.5	63.4	82.9
Bag-KI-SVM [14]	88.0	82.0	84.5	60.5	**85.0**
mi-SVM [1]	87.4	83.6	82.2	58.2	78.9
MI-SVM [1]	77.9	84.3	81.4	59.4	84.0
EM-DD [24]	84.8	84.9	78.3	56.1	72.1

Table 2. Classification accuracies of various MIL algorithms on standard benchmark data sets. The best performances are indicated in bold typeface.

[4] Note that the results of MILD_B and MILD_I on *Musk1* and *Musk2* are different than reported in [13]. This is because, for a complete comparison, we downloaded the source codes of MILD_B and MILD_I available at the authors' webpage and repeated the experiments on all the five data sets with our setting of 10 times 10-fold CV.

In Table 3, for each instance-selection based MIL approach, we report the average dimensions of the corresponding embedding spaces. MILES has the highest dimension since it utilizes all the training instances in the mapping. On *Musk2* and *Fox*, our MILDS approach does not offer any advantage in terms of dimension reduction, but for the other data sets, it decreases the dimension $\sim 6-23\%$, as compared to MILIS and DD-SVM. Among all, MILD_B has the lowest dimension as it only uses positive instance prototypes in its embedding scheme. However, as can be seen in Table 2, neglecting the negative prototypes results in a poor performance compared to the other approaches.

Algorithm	*Musk1*	*Musk2*	*Elephant*	*Fox*	*Tiger*
MILDS	75.0	92.0	169.4	180.0	139.2
MILD_B	42.4	35.2	90.0	90.0	90.0
MILIS	83.0	92.0	180.0	180.0	180.0
MILES	429.4	5943.8	1251.9	1188.0	1098.0
DD-SVM	83.0	92.0	180.0	180.0	180.0

Table 3. The dimensions of the embedding spaces averaged over 10 runs of 10-fold CV.

4.2 Image Classification

The multi-class extensions of our approach have been investigated on image classification problems. In specific, we used the COREL data set which contains 2000 natural images from 20 diverse categories, each having 100 examples. Each image is considered as a bag of instances with instances corresponding to regions of interest obtained via segmentation. Each region is represented by a 9-dimensional feature vector describing shape and local image characteristics (refer to [5,4] for details). Some example images from the data set are given in Fig. 2.

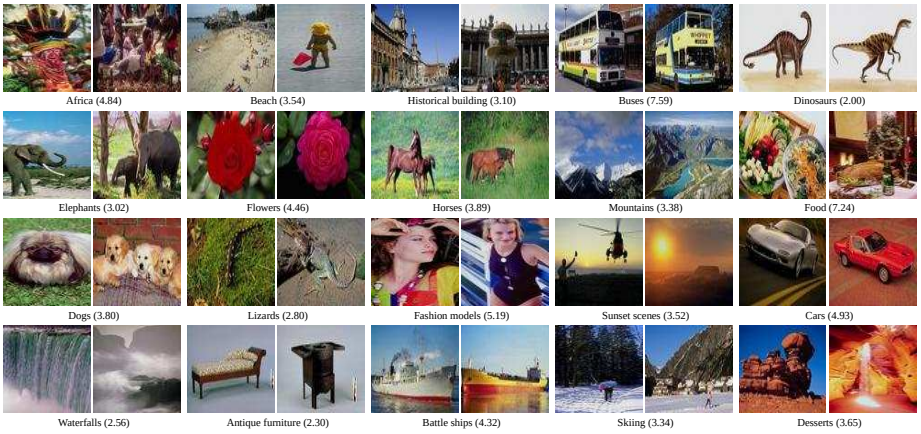

Fig. 2. Example images randomly drawn from the COREL data set. For each category, the average number of regions per image is given inside the parentheses.

In our evaluation, we used the same experimental setup described in [4], and performed two groups of experiments, which are referred to as *1000-Image* and *2000-Image*, respectively. In *1000-Image*, only the first ten categories are considered whereas in *2000-Image*, all the twenty categories in the data set are employed. On both experiments, five times two-fold CV is performed. The average categorization accuracies are presented in Table 4. As can be seen from the results, the performance of *MILDS* and *milDS* are competitive with the state-of-the-art MIL approaches. Especially for the larger *2000-Image* data set, our milDS method gives the best result.

Algorithm	*1000-Image*	*2000-Image*
milDS	82.2	**70.6**
MILDS	83.0	69.4
MILD_B [13]	79.6	67.7
MILIS [8]	**83.8**	70.1
MILES [4]	82.6	68.7
DD-SVM [5]	81.5	67.5
MIForest [10]	59.0	66.0
MissSVM [26]	78.0	65.2
mi-SVM [1]	76.4	53.7
MI-SVM [1]	74.7	54.6

Table 4. Classification accuracies of various MIL algorithms on COREL *1000-Image* and *2000-Image* data sets. The best performances are indicated in bold typeface.

Recall that in MILDS, each classifier trained for distinguishing a specific category from the rest is built upon a different embedding space, or in other words, the set of selected prototypes varies in every subproblem. For each subproblem in *1000-Image*, Fig. 3 shows the instance prototype identified in one of the training images from the target class. Notice that the prototypes are selected from the discriminative regions for that class. On the other hand, in milDS, the set of selected instance prototypes is the same for all the subproblems. This second selection strategy provides a rich way to include contextual relationships in representing visual categories. In some respects, it resembles the vocabulary generation step of the *bag-of-words* approach [6]. The subtle difference is that a similarity-based mapping is employed here instead of a frequency-based one. Fig. 4 shows five prototypes among the full set of representative instances selected for the *Horse* and *Battle ships* categories. Observe that for the *Horse* category, selected prototypes include not just horses but also the regions corresponding to grass regions. Likewise, for the *Battle ships* category, there are additional prototypes representing sky and sea regions.

4.3 Sensitivity to labeling noise

Lastly, we analyzed the sensitivity to labeling noise. For that purpose, we repeated the experiment in [4] which involves distinguishing *Historical building*s from *Horse*s in COREL data set. In this experiment, we compared our method with MILES, MILIS, MILD_B with varying degrees of noise levels where the results are based on five times

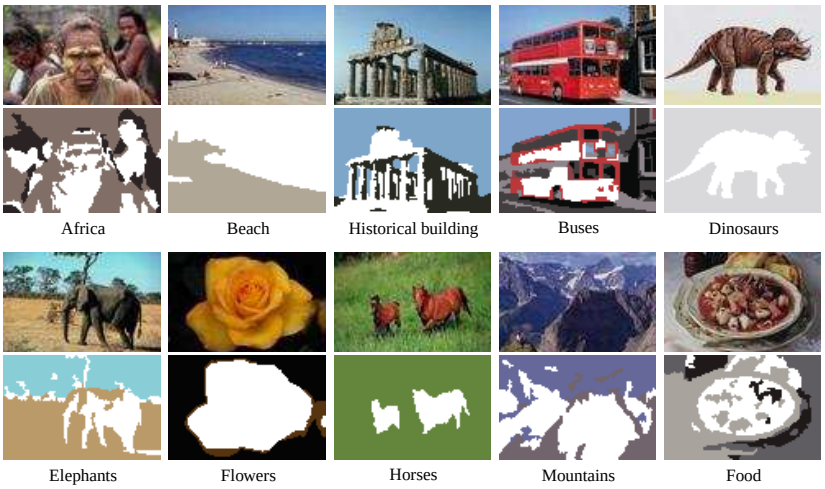

Fig. 3. Sample instance prototypes selected by the *MILDS* algorithm. For each image category, the first row shows a sample training image from that category, and the bottom row illustrates the selected prototype region (shown in white) on the corresponding segmentation map.

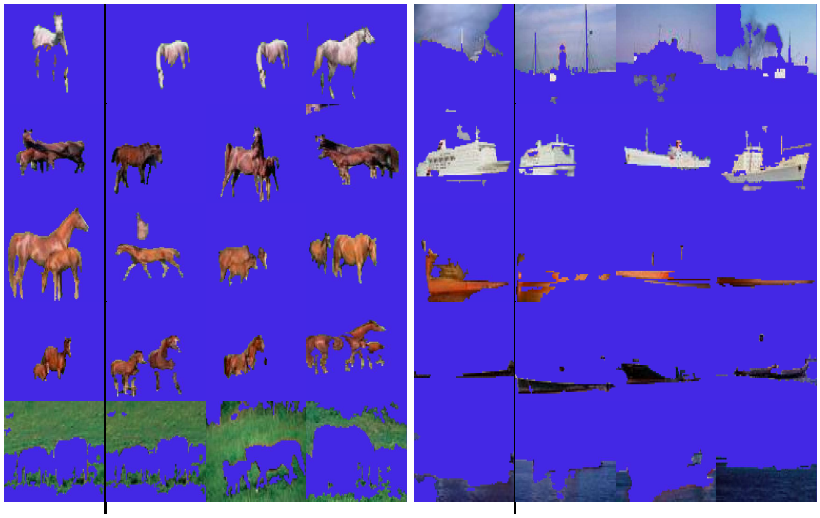

Fig. 4. Sample instance prototypes selected by the *milDS* algorithm for the *Horse* and the *Battle ships* categories. The leftmost columns are the prototypes. The rightmost three columns show other sample regions from the corresponding extracted clusters. The regions in each cluster share similar visual characteristics.

2-fold CV. For each noise level, $d\%$ of positive and $d\%$ of negative images are randomly selected from the training set, and then their labels are changed to form the noisy labels.

Fig. 5 shows the average classification accuracies. When the level of labeling noise is low ($d \leq 5\%$), there is no considerable difference in the performances. As the noise level increases, the performance of MILIS degrades. MILES gives comparable results to MILDS and MILD_B for the noise levels up to $d \leq 25\%$, but gives relatively poor outcomes afterwards. Overall, MILDS is the most robust MIL algorithm to labeling noise among all the tested MIL algorithms. Its performance remains almost the same over all levels of the labeling noise. This is expected, since dominant sets is known to be quite robust to outliers [18,15].

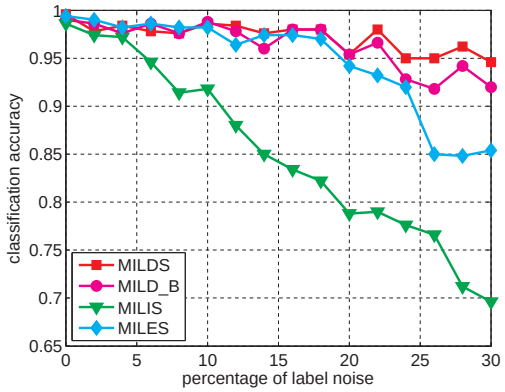

Fig. 5. Sensitivity of various MIL algorithms to labeling noise. MILDS produces the most robust results.

5 Summary and Future Work

In this paper, we proposed an effective MIL scheme, MILDS, which offers a new solution to select a set of instance prototypes, for transforming a given MIL problem into a standard SIL problem. This instance selection approach enables us to successfully identify the most representative examples in the positive and negative training bags. Its success lies in the use of dominant sets pairwise clustering framework. Our empirical results show that the proposed algorithm is competitive with state-of-the-art MIL methods and also robust to labeling noise. As a future work, we plan to extend our approach to multi-instance multi-label learning setting [27,23].

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS. pp. 1073–1080 (2003) 1, 10, 12
2. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. pp. 983–990 (2009) 1
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines (2001), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm 9

4. Chen, Y., Bi, J., Wang, J.Z.: MILES: Multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. 28(12), 1931–1947 (2006) 1, 2, 4, 5, 9, 10, 11, 12
5. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. J. Mach. Learn. Res. 5, 913–939 (2004) 1, 2, 4, 10, 11, 12
6. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Int. Workshop Stat. Learning in Comp. Vis. (2004) 12
7. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (1997) 1, 5
8. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: Multiple instance learning with instance selection. IEEE Trans. Pattern Anal. Mach. Intell. To appear 1, 2, 4, 5, 9, 10, 12
9. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: ICML. pp. 179–186 (2002) 1
10. Leistner, C., Saffari, A., Bischof, H.: MIForests: Multiple-instance learning with randomized trees. In: ECCV. pp. 29–42 (2010) 1, 10, 12
11. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV. vol. 2, pp. 1482–1489 (2005) 4
12. Li, M., Kwok, J., Lu, B.L.: Online multiple instance learning with no regret. In: CVPR. pp. 1395 –1401 (2010) 1, 10
13. Li, W.J., Yeung, D.Y.: MILD: Multiple-instance learning via disambiguation. IEEE Trans. on Knowl. and Data Eng. 22, 76–89 (2010) 1, 2, 3, 4, 5, 9, 10, 12
14. Li, Y.F., Kwok, J.T., Tsang, I.W., Zhou, Z.H.: A convex method for locating regions of interest with multi-instance learning. In: ECML PKDD, pp. 15–30 (2009) 10
15. Liu, H., Yan, S.: Common visual pattern discovery via spatially coherent correspondences. In: CVPR. pp. 1609–1616 (2010) 4, 14
16. Maron, O., Lozano-Pérez, T.: A framework for multiple instance learning. In: NIPS. pp. 570–576 (1998) 1, 2
17. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. Canad. J. Math. 17, 533–540 (1965) 4
18. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. IEEE Trans. on Pattern Anal. and Mach. Intell. 29(1), 167–172 (2007) 2, 3, 4, 5, 6, 14
19. Rota Bulò, S., Bomze, I., Pelillo, M.: Fast population game dynamics for dominant sets and other quadratic optimization problems. In: SSPR. pp. 275–285 (2010) 9
20. Sarkar, S., Boyer, K.L.: Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. Comput. Vis. Image Understand. 71(1), 110 – 136 (1998) 4
21. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS. pp. 1419–1426 (2006) 1
22. Wang, J., Zucker, Jean-Daniel: Solving multiple-instance problem: A lazy learning approach. In: ICML (2000) 1
23. Zha, Z.J., Hua, X.S., Mei, T., Wang, J., Qi, G.J., Wang, Z.: Joint multi-label multi-instance learning for image classification. In: CVPR (2008) 14
24. Zhang, Q., Goldman, S.A.: EM-DD: An improved multi-instance learning technique. In: NIPS. pp. 561–568 (2002) 1, 10
25. Zhang, Q., Goldman, S.A., Yu, W., Fritts, J.: Content-based image retrieval using multiple-instance learning. In: ICML. pp. 682–689 (2002) 1
26. Zhou, Z.H., Xu, J.M.: On the relation between multi-instance learning and semi-supervised learning. In: ICML. pp. 1167–1174 (2007) 12
27. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with applications to scene classification. In: NIPS. pp. 1609–1616 (2006) 14