MILES: Multiple-Instance Learning via Embedded Instance Selection

Yixin Chen, *Member*, *IEEE*, Jinbo Bi, *Member*, *IEEE*, and James Z. Wang, *Senior Member*, *IEEE*

Abstract—Multiple-instance problems arise from the situations where training class labels are attached to sets of samples (named *bags*), instead of individual samples within each bag (called *instances*). Most previous multiple-instance learning (MIL) algorithms are developed based on the assumption that a bag is positive if and only if at least one of its instances is positive. Although the assumption works well in a drug activity prediction problem, it is rather restrictive for other applications, especially those in the computer vision area. We propose a learning method, MILES (Multiple-Instance Learning via Embedded instance Selection), which converts the multiple-instance learning problem to a standard supervised learning problem that does not impose the assumption relating instance labels to bag labels. MILES maps each bag into a feature space defined by the instances in the training bags via an instance similarity measure. This feature mapping often provides a large number of redundant or irrelevant features. Hence, 1-norm SVM is applied to select important features as well as construct classifiers simultaneously. We have performed extensive experiments. In comparison with other methods, MILES demonstrates competitive classification accuracy, high computation efficiency, and robustness to labeling uncertainty.

Index Terms—Multiple-instance learning, feature subset selection, 1-norm support vector machine, image categorization, object recognition, drug activity prediction.

1 Introduction

Multiple-instance learning (MIL) was introduced by Dietterich et al. [19] in the context of drug activity prediction. It provides a framework to handle the scenarios where training class labels are naturally associated with sets of samples, instead of individual samples. In recent years, a variety of learning problems (e.g., drug activity prediction [19], [35], stock market prediction [36], data mining applications [46], image retrieval [56], [60], natural scene classification [37], text categorization [2], and image categorization [16]) have been tackled as multiple-instance problems.

1.1 Multiple-Instance Problems

A *multiple-instance problem* involves ambiguous training examples: A single example is represented by several feature vectors (instances), some of which may be responsible for the observed classification of the example; yet, the training label is only attached to the example instead of the instances. The multiple-instance problem was formally introduced in the context of drug activity prediction [19], while a similar type of learning scenario was first proposed in [13].

The goal of drug activity prediction is to foretell the potency of candidate drug molecules by analyzing a collection of previously synthesized molecules whose potencies are already known. The potency of a drug molecule is determined by the degree to which it binds to a site of medical interest on the target protein. The three-dimensional structure of a drug molecule decides principally the binding strength: A drug molecule binds to the target protein if the shape of the molecule conforms closely to the structure of the binding site. Unfortunately, a molecule can adopt a wide range of shapes by rotating some of its internal bonds. Therefore, without the knowledge of the three-dimensional structure of the target protein, which is in general not available, knowing that a previously-synthesized molecule binds to the target protein does not directly provide the binding information on the shapes of the molecule. For the multiple-instance problem in [19], training examples are molecules (named bags [35]). A bag contains several shapes, called instances, of the molecule. A label of *binding* or *not binding* is attached to each training bag instead of its instances. The goal is to learn the concept of binding, hence, to predict whether a new bag, i.e., a new drug molecule, binds to the target protein. Note that a bag is a multiset in that the order of the instances is ignored; yet, the multiplicity is explicitly significant, i.e., a bag may contain identical instances.

Far beyond the drug activity prediction problem, the multiple-instance problem emerges naturally in a variety of challenging learning problems in computer vision, including natural scene classification [37], content-based image retrieval [34], [56], [60], [15], image categorization [6], [32], [16], and object detection and recognition [40], [1], [21], [20], [45], [40], [42], [5]. Generally speaking, the goal of all these problems is to learn visual concepts from labeled images. The multiple-instance situation arises in these learning problems because of the choice of image representation. A single image in [37], [56], [60] is represented by a collection of fixed-size blocks. In [16], an image is characterized by regions obtained from image segmentation. In [1], [21], [20], [45], [17], [42], [5], an image is described by a set of patches invariant to certain geometric transformations. These image patches are typically generated by salient region detectors

- *Y. Chen is with the Department of Computer and Information Science, University of Mississippi, 201 Weir Hall, University, MS 38677. E-mail: ychen@cs.olemiss.edu.*
- *J. Bi is with Computer Aided Diagnosis and Therapy Solutions, Siemens Medical Solutions, Inc., MS51, 51 Valley Stream Parkway, Malvern, PA 19355. E-mail: jinbo.bi@siemens.com.*
- *J.Z. Wang is with the College of Information Sciences and Technology and the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: jwang@ist.psu.edu.*

Manuscript received 6 Oct. 2005; revised 1 Mar. 2006; accepted 13 Apr. 2006; published online 12 Oct. 2006.
Recommended for acceptance by Z. Ghahramani.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0544-1005.

[27], interest point detectors [38], and edge-based detectors [52]. In all of the previously mentioned learning problems, training labels are attached to images instead of the blocks, regions, or patches contained in the images. These learning problems match squarely with the bag-of-instances model: An image is a bag that contains instances corresponding to blocks, regions, or patches; instance labels are only indirectly accessible through the labels attached to bags.

1.2 Previous Work Related to Multiple-Instance Learning

One of the earliest algorithms for learning from multiple-instance examples was developed by Dietterich et al. [19] for drug activity prediction. The key assumption of their MIL formulation is that: A bag is positive if at least one of its instances is a positive example; otherwise, the bag is negative. Their algorithm, named the axis-parallel rectangle (APR) method [19], attempts to find an APR by expanding or shrinking a hyperrectangle in the instance feature space to maximize the number of instances from different positive bags enclosed by the rectangle while minimizing the number of instances from negative bags within the rectangle. A bag is classified as positive if at least one of its instances falls within the APR; otherwise, the bag is classified as negative. A number of theoretical results have been obtained following the APR framework [4], [10], [33]. De Raedt showed the connection between MIL and inductive logic programming [18].

The basic idea of APR was extended in different ways that led to several MIL algorithms. Maron and Lozano-Pérez [35] proposed a general framework, named *diverse density* (DD), which has been tested on applications including stock market prediction [36], natural scene classification [37], and content-based image retrieval [56], [60]. Diverse density measures a co-occurrence of similar instances from different positive bags. The desired concept is learned by maximizing the DD function. Zhang and Goldman [61] combined the idea of expectation-maximization (EM) with DD and developed an algorithm, EM-DD, to search for the most likely concept. Andrews et al. [2] formulated MIL as a mixed integer quadratic program. Integer variables are selector variables that select a positive instance from each positive bag. Their algorithm, which is called MI-SVM, has an outer loop and an inner loop. The outer loop sets the values of the selector variables. The inner loop then trains a standard support vector machine, SVM, in which the selected positive instances replace the positive bags. Later, Andrews and Hofmann [3] developed an algorithm based on a generalization of linear programming boosting. Ramon and De Raedt presented a neural networks framework for MIL [43]. A similar method has been derived in [59]. Wang and Zucker [53] presented two variants of the k-nearest neighbor algorithm (Bayesian-kNN and Citation-kNN) using Hausdorff distance. Zhou and Zhang [63] proposed ensembles of multi-instance learners, which achieved competitive test results on the benchmark data sets of drug activity prediction.

All the above MIL formulations explicitly or implicitly encode the assumption that a bag is positive if and only if at least one of its instances is positive. The assumption is valid in drug activity prediction. Typically, there is a single shape that allows binding; hence, it is natural to relate the instance labels to the bag label through a disjunction (or a soft disjunction) function. However, for applications such as object recognition, a negative bag (e.g., a background image without the object of interest) may also contain instances that look similar to parts of the object. In other words, if a positive instance label indicates that the instance appears to be part of the object, a negative bag may contain positive instances as well. Several algorithms have been proposed to tackle this type of situation. Scott et al. [47] developed a method in which a bag is positive if and only if it contains a collection of instances, each near one of a set of target points. In [60], Zhang et al. observed that the average prediction of a collection of classifiers, constructed from the EM-DD algorithm with different starting points, is usually more accurate than the prediction given by any individual classifier in a content-based image retrieval experiment. Chen and Wang [16] proposed a MIL framework, DD-SVM, which, instead of taking the average prediction of classifiers built from the EM-DD, trains an SVM in a feature space constructed from a mapping defined by the local maximizers and minimizers of the DD function. Through the feature mapping, DD-SVM essentially converts MIL to a standard supervised learning problem.

There is an abundance of prior work that uses standard supervised learning techniques to solve multiple-instance problems. Zucker and Chevaleyre [65] applied decision trees and decision rules to a generalized multiple-instance problem. Gärtner et al. [22] designed a kernel for multiple-instance data. Hence, SVMs can be learned directly from the training bags. Xu and Frank [58] proposed logistic regression and boosting approaches for MIL with the assumption that all instances contribute equally and independently to a bag's label. Weidmann et al. [55] introduced a two-level learning framework in which a bag is transformed into a meta-instance that can be learned by a propositional method. In a recent work [44], Ray and Craven compared several MIL methods with their supervised counterparts. They observed that although some MIL methods consistently outperformed their supervised counterparts, in several domains, a supervised algorithm gave the best performance among all the MIL algorithms they tested.

In the area of computer vision, several standard supervised learning techniques have been adapted for object detection and recognition where the underlying learning problems are essentially the multiple-instance problem [1], [21], [17], [42], [5]. Fergus et al. [21] presented a generative approach to learning and recognizing an object class from images. Each image (bag) is represented as a collection of image patches (instances) produced by a scale invariant detector [26]. An object is modeled as a group of patches, which are assumed to be generated by a probabilistic model. The model is estimated by maximizing the likelihood function using the EM technique. Agarwal and Roth [1] described a discriminative approach for object detection. A vocabulary of object parts is constructed from a set of sample images. The multiple-instance representation of each image is then transformed to a binary feature vector based on the vocabulary. Each binary feature indicates whether or not a part or relation occurs in the image. The classifier is learned using the Sparse Network of Winnows (SNoW) framework. Csurka et al. [17] proposed an object categorization method using the bag-of-keypoints model, which is identical to the bag-of-instances model. An image is represented by a collection of affine invariant patches. Vector quantization is applied to the descriptors of all image patches to generate a predetermined number of clusters. Each image is then transformed to an integer-valued feature vector indicating the number of patches assigned to each cluster. Classifiers are implemented as Naive Bayes and SVM. Opelt et al. [42] described a boosting approach to object detection where, at each iteration, the weak hypothesis finder selects one local image patch and one type of local descriptor. Bar-Hillel et al. [5] proposed a boosting technique that learns a generative object model in a discriminative manner.

1.3 An Overview of Our Approach

The approach we take to tackle the multiple-instance problem further extends ideas from the diverse density framework [35], [16] and the wrapper model in feature selection [29]. Our approach identifies instances that are relevant to the observed classification by embedding bags into an instance-based feature space and selecting the most important features. We define a similarity measure between a bag and an instance. The coordinates of a given bag in the feature space represent the bag's similarities to various instances in the training set. The embedding produces a possibly high-dimensional space when the number of instances in the training set is large. In addition, many features may be redundant or irrelevant because some of the instances might not be responsible for the observed classification of the bags, or might be similar to each other. It is essential and indispensable to select a subset of mapped features that is most relevant to the classification problem of interest. Although any feature selection approaches could be applied for this purpose, we choose a joint approach that constructs classifiers and selects important features simultaneously. Specifically, we use the 1-norm SVM method because of its excellent performance in many applications [8], [64]. Since each feature is defined by an instance, feature selection is essentially instance selection. Therefore, we name our approach MILES, Multiple-Instance Learning via Embedded instance Selection.

The proposed approach has the following characteristics:

- Broad adaptability: It provides a learning framework that converts a multiple-instance problem to a supervised learning problem. It demonstrates highly competitive classification accuracy in our empirical studies using benchmark data sets from different application areas. Moreover, in comparison with the DD-SVM algorithm [16], the proposed approach is less sensitive to the class label uncertainties.
- Low complexity: It is efficient in computational complexity, therefore, can potentially be tailored to tasks that have stringent time or resource limits. According to our empirical studies, the learning process of the proposed approach is on average one order of magnitude faster than that of the algorithms described in [21] and [16].
- Prediction capability: In some multiple-instance problems, classification of instances is at least as important as the classification of bags. The proposed approach supports *instance naming*, i.e., predicting instance labels. This is in contrast to the methods described in [53], [16], [17], [42].

1.4 Outline of the Paper

The remainder of the paper is organized as follows: Section 2 describes an instance-based embedding of bags that transforms every bag to a uniform representation. Section 3 is dedicated to the description of a joint feature selection and classification method using a 1-norm SVM. A concrete 1-norm SVM formulation is presented. Section 4 provides an algorithmic view of the approach. In Section 5, we explain the extensive experimental studies conducted and demonstrate the results. We conclude and discuss possible future work in Section 6.

2 Instance-Based Embedding of Bags

We introduce the instance-based embedding in this section. Before discussing the mathematical definition of the mapping, we first give a brief review of the diverse density framework based on [36], [35] which forms the conceptual basis of the instance-based feature mapping. We denote positive bags as $\mathbf{B}_i^+$ and the jth instance in that bag as $\mathbf{x}_{ij}^+$. The bag $\mathbf{B}_i^+$ consists of n_i^+ instances $\mathbf{x}_{ij}^+$, $j = 1, \cdots, n_i^+$. Similarly, $\mathbf{B}_i^-$, $\mathbf{x}_{ij}^-$, and n_i^- represent a negative bag, the jth instance in the bag, and the number of instances in the bag, respectively. When the label on the bag does not matter, it will be referred to as $\mathbf{B}_i$ with instances as $\mathbf{x}_{ij}$. All instances belong to the feature space $\mathbb{X}$. The number of positive (negative) bags is denoted as ℓ^+ (ℓ^-). For the sake of convenience, when we line up all instances in all bags together, we reindex these instances as $\mathbf{x}^k$, $k = 1, \cdots, n$, where $n = \sum_{i=1}^{\ell^+} n_i^+ + \sum_{i=1}^{\ell^-} n_i^-$.

2.1 Review of Diverse Density

The diverse density framework was derived in [36], [35] based on the assumption that there exists a single target concept, which can be used to label individual instances correctly. If we denote a given concept class as $\mathcal{C}$, the diverse density of a concept $t \in \mathcal{C}$ is defined as the probability[1] that the concept t is the target concept given the training bags [36]:

$$DD(t) = \Pr(t|\mathbf{B}_1^+, \cdots, \mathbf{B}_{\ell^+}^+, \mathbf{B}_1^-, \cdots, \mathbf{B}_{\ell^-}^-). \quad (1)$$

The target concept that is most likely to agree with the data is then determined by maximizing this probability. Applying Bayes' rule to (1) and further assuming that all bags are conditionally independent given the true target concept, we can write (1) as

$$\begin{aligned} DD(t) &= \frac{\Pr(t)\prod_{i=1}^{\ell^+}\Pr(\mathbf{B}_i^+|t)\prod_{i=1}^{\ell^-}\Pr(\mathbf{B}_i^-|t)}{\Pr(\mathbf{B}_1^+, \cdots, \mathbf{B}_{\ell^+}^+, \mathbf{B}_1^-, \cdots, \mathbf{B}_{\ell^-}^-)} \\ &= \left[\frac{\prod_{i=1}^{\ell^+}\Pr(\mathbf{B}_i^+)\prod_{i=1}^{\ell^-}\Pr(\mathbf{B}_i^-)}{\Pr(\mathbf{B}_1^+, \cdots, \mathbf{B}_{\ell^+}^+, \mathbf{B}_1^-, \cdots, \mathbf{B}_{\ell^-}^-)\Pr(t)^{\ell^+ + \ell^- - 1}}\right] \quad (2) \\ &\quad \left[\prod_{i=1}^{\ell^+}\Pr(t|\mathbf{B}_i^+)\prod_{i=1}^{\ell^-}\Pr(t|\mathbf{B}_i^-)\right]. \end{aligned}$$

If we assume a uniform prior on t, for given training bags, maximizing $DD(t)$ is equivalent to maximizing the second factor in (2), i.e., $\prod_{i=1}^{\ell^+}\Pr(t|\mathbf{B}_i^+)\prod_{i=1}^{\ell^-}\Pr(t|\mathbf{B}_i^-)$. Maron [36] proposes several ways to estimate $\Pr(t|\mathbf{B}_i)$ for various concept classes. For example, if $\mathcal{C}$ is *a single point concept class*, where every concept corresponds to a single point in $\mathbb{X}$, the *most-likely-cause estimator* [36] is then defined as:

1. Here, $\mathcal{C}$ is assumed to be a countable set. Otherwise, the diverse density should be interpreted as probability density instead of probability. To simplify the notations, we use $\Pr()$ to represent either probability or probability density.

$$\Pr(t|\mathbf{B}_i^+) \propto \max_j \exp\left(-\frac{\| \mathbf{x}_{ij}^+ - t \|^2}{\sigma^2}\right) \tag{3}$$

$$\Pr(t|\mathbf{B}_i^-) \propto 1 - \max_j \exp\left(-\frac{\| \mathbf{x}_{ij}^- - t \|^2}{\sigma^2}\right), \tag{4}$$

where σ is a predefined scaling factor.

An optimization algorithm, such as a gradient descent approach [35] or EM-DD [61], is used to search for a target concept that achieves maximal DD. Neither the gradient descent algorithm nor EM-DD can guarantee the global optimality and, hence, they may get stuck at local solutions. Typically, multiple runs with different starting search points are necessary. Therefore, the process of maximization is often very time-consuming.

2.2 Instance-Based Feature Mapping

The DD framework can be interpreted from a feature selection point of view as follows: For a given concept class $\mathcal{C}$, each concept $t \in \mathcal{C}$ is viewed as an attribute or a feature (which is denoted as h_t) for the bags; and the value of the feature for bag $\mathbf{B}_i$ is defined as

$$h_t(\mathbf{B}_i) = \Pr(t|\mathbf{B}_i). \tag{5}$$

If $\mathcal{C}$ is a countable set, i.e., $\mathcal{C} = \{t_1, t_2, \cdots, t_j, \cdots\cdots\}$, then

$$[h_{t_1}(\mathbf{B}_i), h_{t_2}(\mathbf{B}_i), \cdots, h_{t_j}(\mathbf{B}_i), \cdots]^T = \\ \left[\Pr(t_1|\mathbf{B}_i), \Pr(t_2|\mathbf{B}_i), \cdots, \Pr(t_j\mathbf{B}_i), \cdots\right]^T$$

determines the values of all the attributes (or features) for bag $\mathbf{B}_i$. We denote as $\mathbb{F}_\mathcal{C}$ the space defined by $h_{t_1}, h_{t_2}, \cdots, h_{t_j}, \cdots\cdots$. As a result, each bag can be viewed as a point in $\mathbb{F}_\mathcal{C}$, and $[\Pr(t_j|\mathbf{B}_1^+), \cdots, \Pr(t_j|\mathbf{B}_{\ell^+}^+), \ \Pr(t_j|\mathbf{B}_1^-), \cdots, \Pr(t_j|\mathbf{B}_{\ell^-}^-)]^T$ realizes the feature h_{t_j} for all the bags.

Under the uniform prior assumption on concepts, it is not difficult to see that finding a concept to maximize the DD function in (2) is equivalent to selecting one feature, h_{t_j}, to maximize the following measure:

$$f(h_{t_j}) = \prod_{i=1}^{\ell^+} h_{t_j}(\mathbf{B}_i^+) \prod_{i=1}^{\ell^-} h_{t_j}(\mathbf{B}_i^-) = \prod_{i=1}^{\ell^+} \Pr(t_j|\mathbf{B}_i^+) \prod_{i=1}^{\ell^-} \Pr(t_j|\mathbf{B}_i^-).$$

From the perspective of feature selection, the DD framework appears to be rather restrictive because it always seeks for one and only one feature. Can we improve the performance by searching for multiple features? This is the basic motivation of our approach. In particular, our approach further extends the idea from the DD framework in constructing the features, which is described below. It then applies 1-norm SVM to build classifiers and select features simultaneously. The details of the joint feature selection and classification approach will be discussed in Section 3.

The new feature mapping is derived based on (5). First, we need to specify a concept class. We choose to use the single point concept class. In addition, we assume that there may exist more than one target concept (the exact number needs to be determined by the feature selection process) and a target concept can be well approximated by an instance in the training bags. In other words, each instance in the training bags is a candidate for target concepts. Therefore, each instance corresponds to a concept, i.e.,

$$\mathcal{C} = \{\mathbf{x}^k : k = 1, \cdots, n\}, \tag{6}$$

where $\mathbf{x}^k$ are those reindexed instances as defined at the beginning of Section 2. Furthermore, we assume that a target concept can be related to either positive bags or negative bags, whereas, in the DD framework, the target concept is defined for positive bags only. Under this symmetric assumption and our choice of concept class in (6), the most-likely-cause estimator in (3) and (4) can be written, independent of the bag label, as

$$\Pr(\mathbf{x}^k|\mathbf{B}_i) \propto s(\mathbf{x}^k, \mathbf{B}_i) = \max_j \exp\left(-\frac{\| \mathbf{x}_{ij} - \mathbf{x}^k \|^2}{\sigma^2}\right).$$

$s(\mathbf{x}^k, \mathbf{B}_i)$ can also be interpreted as a measure of similarity between the concept $\mathbf{x}^k$ and the bag $\mathbf{B}_i$: The similarity between a concept and a bag is determined by the concept and the closest instance in the bag. A bag $\mathbf{B}_i$ is then embedded in $\mathbb{F}_\mathcal{C}$ with coordinates $\mathbf{m}(\mathbf{B}_i)$ defined as

$$\mathbf{m}(\mathbf{B}_i) = [s(\mathbf{x}^1, \mathbf{B}_i), s(\mathbf{x}^2, \mathbf{B}_i), \cdots, s(\mathbf{x}^n, \mathbf{B}_i)]^T. \tag{7}$$

For a given training set of ℓ^+ positive bags and ℓ^- negative bags, applying the mapping (7) yields the following matrix representation of all training bags in $\mathbb{F}_\mathcal{C}$:

$$\begin{aligned}
&[\mathbf{m}_1^+, \cdots, \mathbf{m}_{\ell^+}^+, \mathbf{m}_1^-, \cdots, \mathbf{m}_{\ell^-}^-] \\
&= [\mathbf{m}(\mathbf{B}_1^+), \cdots, \mathbf{m}(\mathbf{B}_{\ell^+}^+), \mathbf{m}(\mathbf{B}_1^+), \cdots, \mathbf{m}(\mathbf{B}_{\ell^-}^-)] \\
&= \begin{bmatrix} s(\mathbf{x}^1, \mathbf{B}_1^+) & \cdots & (\mathbf{x}^1, \mathbf{B}_{\ell^-}^-) \\ s(\mathbf{x}^2, \mathbf{B}_1^+) & \cdots & (\mathbf{x}^2, \mathbf{B}_{\ell^-}^-) \\ \vdots & \ddots & \vdots \\ s(\mathbf{x}^n, \mathbf{B}_1^+) & \cdots & s(\mathbf{x}^n, \mathbf{B}_{\ell^-}^-) \end{bmatrix},
\end{aligned}$$

where each column represents a bag, and the kth feature in $\mathbb{F}_\mathcal{C}$ realizes the kth row of the matrix, i.e.,

$$s(\mathbf{x}^k, \cdot) = [s(\mathbf{x}^k, \mathbf{B}_1^+), \cdots, s(\mathbf{x}^k, \mathbf{B}_{\ell^+}^+), s(\mathbf{x}^k, \mathbf{B}_1^-), \cdots, s(\mathbf{x}^k, \mathbf{B}_{\ell^-}^-)].$$

Intuitively, if $\mathbf{x}^k$ achieves high similarity to some positive bags and low similarity to some negative bags, the feature induced by $\mathbf{x}^k$, $s(\mathbf{x}^k, \cdot)$, provides some "useful" information in separating the positive and negative bags. Next, we present a simple example to illustrate the efficiency of the mapping (7).

2.3 An Example

We formulate a multiple-instance problem where each instance is generated by one of the following two-dimensional probability distributions: $N_1 \sim \mathcal{N}([5,5]^T, I)$, $N_2 \sim \mathcal{N}([5,-5]^T, I)$, $N_3 \sim \mathcal{N}([-5,5]^T, I)$, $N_4 \sim \mathcal{N}([-5,-5]^T, I)$, and $N_5 \sim \mathcal{N}([0,0]^T, I)$, where $\mathcal{N}([5,5]^T, I)$ denotes the normal distribution with mean $[5,5]^T$ and identity covariance matrix. Each bag comprises at most eight instances. A bag is labeled positive if it contains instances from at least two different distributions among N_1, N_2, and N_3. Otherwise, the bag is negative.

Using this model, we generated 20 positive bags and 20 negative bags with a total of 219 instances. Fig. 1a depicts all the instances on a two-dimensional plane. Note that instances from negative bags mingle with those from positive bags because a negative bag may include instances from any one, but only one, of the distributions N_1, N_2, and N_3.

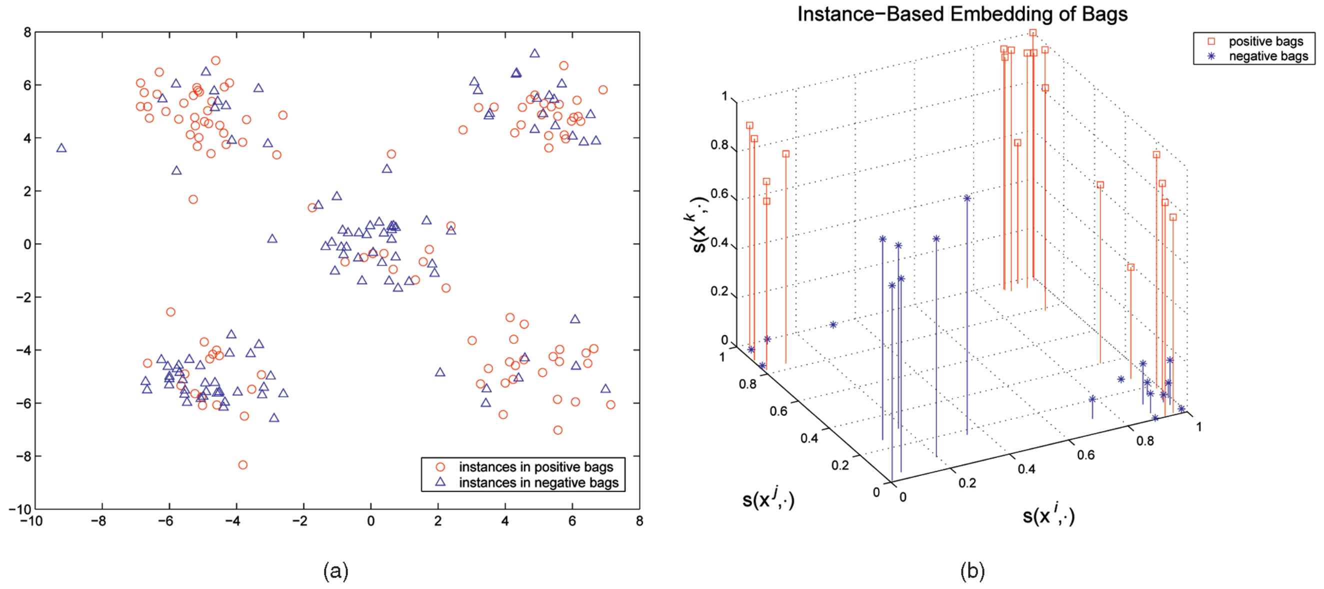


Fig. 1. Visualization of instances and bags. (a) Instances on a two-dimensional plane. (b) Bags embedded in a features space induced by three manually selected instances. Straight lines in (b) illustrate the projection of bags onto the two-dimensional space defined by features $s(\mathbf{x}^i, \cdot)$ and $s(\mathbf{x}^j, \cdot)$.

After applying mapping (7), each bag was embedded in a 219-dimensional feature space (σ was chosen to be 4.5). Intuitively, the mean vectors of N_1, N_2, and N_3 might be helpful in bag classification. So, we manually selected three instances, $\mathbf{x}^i = [4.3, 5.2]^T$, $\mathbf{x}^j = [5.4, -3.9]^T$, and $\mathbf{x}^k = [-6.0, 4.8]^T$, which were close to the mean vectors of N_1, N_2, and N_3. Fig. 1b shows the embedding of all 40 bags in a subspace defined by the three instances. It is not difficult to observe that the bags can be separated by a plane in this three-dimensional space, which corresponds to a simple bag classifier. This suggests that the mapping (7) can indeed induce useful features for bag classification, yet automatic feature subset selection and the subsequent classifier design are the remaining questions. These questions are addressed in the next section.

3 Joint Feature Selection and Classification for Multiple-Instance Learning

Feature subset selection is a well-studied research problem in the areas of statistics, machine learning, and pattern recognition [41], [29], [25], [30], [39], [12], [23], [28], [31], [50], [57]. Existing feature selection approaches generally fall in two categories: filter and wrapper [29], [57]. Some filter methods, such as ranking through correlation coefficients or through Fisher scores, tend to select intercorrelated features and do not guarantee an acquisition of a good classifier. On the contrary, wrappers include the desired classifier as part of their performance evaluation. They tend to produce better generalization but may require an expensive computational cost. Our method is essentially a wrapper where the 1-norm SVM is used to construct classifiers and select important features simultaneously. The 1-norm SVM can be formulated as a linear program, which, in general, can be solved efficiently from an optimization point of view. The computational cost will not be an issue.

3.1 1-Norm Support Vector Machines

Denote the class label variable by y, which takes values of $+1$ and -1. We consider the classification problem of finding a linear classifier

$$y = \text{sign}(\mathbf{w}^T\mathbf{m} + b)$$

in the feature space $\mathbb{F}_\mathcal{C}$ to distinguish between positive bags and negative bags, where $\mathbf{w}$ and b are model parameters and $\mathbf{m} \in \mathbb{F}_\mathcal{C}$ corresponds to a bag. The SVM approach constructs classifiers based on hyperplanes by minimizing a regularized training error, $\mathcal{E}_{\text{training}}$,

$$\lambda P[\cdot] + \mathcal{E}_{\text{training}},$$

where $P[\cdot]$ is a regularizer, λ is called the regularization parameter, and $\mathcal{E}_{\text{training}}$ is commonly defined as a total of the loss that each bag introduces through a hinge loss function

$$\xi = \max\{1 - y(\mathbf{w}^T\mathbf{m} + b), 0\}.$$

When an optimal solution $\mathbf{w}$ is obtained, the magnitude of its component w_k indicates the significance of the effect of the kth feature in $\mathbb{F}_\mathcal{C}$ on the classifier. Those features corresponding to a nonzero w_k are selected and used in the classifier.

The regularizer in standard SVMs is the squared 2-norm of the weight vector $\parallel \mathbf{w} \parallel$, which formulates SVMs as quadratic programs (QP). Solving QPs is typically computationally more expensive than solving linear programs (LPs). There exist alternative LP formulations of SVMs [7], [49], [64] which involve regularization with a norm favoring sparsity, e.g., the 1-norm of $\mathbf{w}$,

$$\parallel \mathbf{w} \parallel_1 = \sum_k |w_k|.$$

The 1-norm penalty was adopted in other approaches similarly, such as basis pursuit [14] and LASSO [51] for driving more components of $\mathbf{w}$ to zero. Thus, 1-norm SVM is also referred to as sparse SVM and has been applied to other practical problems such as drug discovery [8].

Another characteristic of the MIL problem is that training sets in many MIL problems are very imbalanced between classes, e.g., the number of negative bags can be much larger than the number of positive bags. To tackle this imbalanced issue and make classifiers biased toward the minor class, a simple strategy we used is to penalize differently on errors produced, respectively, by positive bags and by negative bags. Hence, the 1-norm SVM is formulated as follows:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,\eta} \quad & \lambda\sum_{k=1}^{n}|w_k| + C_1\sum_{i=1}^{\ell^+}\xi_i + C_2\sum_{j=1}^{\ell^-}\eta_j \\
\text{s.t.} \quad & (\mathbf{w}^T\mathbf{m}_i^+ + b) + \xi_i \geq 1, i = 1,\cdots,\ell^+, \\
& -(\mathbf{w}^T\mathbf{m}_j^- + b) + \eta_j \geq 1, j = 1,\cdots,\ell^-, \\
& \xi_i,\eta_j \geq 0, i = 1,\cdots,\ell^+, j = 1,\cdots,\ell^-,
\end{aligned} \tag{8}$$

where $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ are hinge losses. Choosing unequal values for parameters C_1 and C_2 will penalize differently on false negatives and false positives. Usually, C_1 and C_2 are chosen so that the training error is determined by a convex combination of the training errors occurred on positive bags and on negative bags. In other words, let $C_1 = \mu$ and $C_2 = 1-\mu$, where $0 < \mu < 1$.

To form an LP for the 1-norm SVM, we rewrite $w_k = u_k - v_k$, where $u_k, v_k \geq 0$. If either u_k or v_k has to equal to 0, we have $|w_k| = u_k + v_k$. The LP is then formulated in variables $\mathbf{u}$, $\mathbf{v}$, b, $\boldsymbol{\xi}$, and $\boldsymbol{\eta}$ as

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{v},b,\boldsymbol{\xi},\boldsymbol{\eta}} \quad & \lambda\sum_{k=1}^{n}(u_k+v_k) + \mu\sum_{i=1}^{\ell^+}\xi_i + (1-\mu)\sum_{j=1}^{\ell^-}\eta_j \\
\text{s.t.} \quad & [(\mathbf{u}-\mathbf{v})^T\mathbf{m}_i^+ + b] + \xi_i \geq 1, i = 1,\cdots,\ell^+, \\
& -[(\mathbf{u}-\mathbf{v})^T\mathbf{m}_j^- + b] + \eta_j \geq 1, j = 1,\cdots,\ell^-, \\
& u_k, v_k \geq 0, k = 1,\cdots,n, \\
& \xi_i,\eta_j \geq 0, i = 1,\cdots,\ell^+, j = 1,\cdots,\ell^-.
\end{aligned} \tag{9}$$

Solving LP (9) yields solutions equivalent to those obtained by the 1-norm SVM (8) because any optimal solution to (9) has at least one of the two variables u_k and v_k equal to 0 for all $k = 1,\cdots,n$. Otherwise, assume $u_k \geq v_k > 0$ without loss of generality, and we can find a better solution by setting $u_k = u_k - v_k$ and $v_k = 0$, which contradicts the optimality of $(\mathbf{u},\mathbf{v})$.

Let $\mathbf{w}^* = \mathbf{u}^* - \mathbf{v}^*$ and b^* be the optimal solution of (9). The magnitude of w_k^* determines the influence of the kth feature on the classifier. The set of selected features is given as $\{s(\mathbf{x}^k,\cdot) : k \in \mathcal{I}\}$, where

$$\mathcal{I} = \{k : |w_k^*| > 0\}$$

is the index set for nonzero entries in $\mathbf{w}^*$. The classification of bag $\mathbf{B}_i$ is computed as

$$y = \operatorname{sign}\left(\sum_{k\in\mathcal{I}} w_k^* s(\mathbf{x}^k,\mathbf{B}_i) + b^*\right). \tag{10}$$

3.2 Instance Classification

In some multiple-instance problems, classification of instances is at least as important as the classification of bags. For one example, an object detection algorithm needs not only to identify whether or not an image contains a certain object, but also to locate the object (or part of the object) from the image if it contains the object. Under the multiple-instance formulation, this requires the classification of the bags (*containing an object* versus *not containing any such object*) as well as the instances in a bag that correspond to the object. The classifier (10) predicts a label for a bag. Next, we introduce a way to classify instances based on a bag classifier.

The basic idea is to classify instances according to their contributions to the classification of the bag. Instances in a bag can be grouped into three classes: *positive class*, *negative class*, and *void class*. An instance in bag $\mathbf{B}_i$ is assigned to the positive class (negative class) if its contribution to $\sum_{k\in\mathcal{I}} w_k^* s(\mathbf{x}^k,\mathbf{B}_i)$ is greater than or equal to (or less than) a threshold. An instance is assigned to the void class if it makes no contribution to the classification of the bag.

Given a bag $\mathbf{B}_i$ with instances $\mathbf{x}_{ij}$, $j = 1,\cdots,n_i$, we define an index set $\mathcal{U}$ as

$$\mathcal{U} = \left\{j^* : j^* = \arg\max_j \exp\left(-\frac{\|\mathbf{x}_{ij}-\mathbf{x}^k\|^2}{\sigma^2}\right), k \in \mathcal{I}\right\}.$$

It is not difficult to verify that the evaluation of (10) only needs the knowledge of the instances $\mathbf{x}_{ij^*}$, $j^* \in \mathcal{U}$. In this sense, $\mathcal{U}$ defines a minimal set of instances responsible for the classification of $\mathbf{B}_i$. Hence, removing an instance $\mathbf{x}_{ij^*}$, $j^* \notin \mathcal{U}$ from the bag will not affect the value of $\sum_{k\in\mathcal{I}} w_k^* s(\mathbf{x}^k,\mathbf{B}_i)$ in (10), and $\{1,\cdots,n_i\}$ minus the set $\mathcal{U}$ specifies the instances in the void class. Since there can be more than one instance in the bag $\mathbf{B}_i$ that maximizes $\exp(-\frac{\|\mathbf{x}_{ij}-\mathbf{x}^k\|^2}{\sigma^2})$ for a given $\mathbf{x}^k$, $k \in \mathcal{I}$, we denote the number of maximizers for $\mathbf{x}^k$ by m_k. Also, an instance $\mathbf{x}_{ij^*}$, $j^* \in \mathcal{U}$ can be a maximizer for different $\mathbf{x}^k$s, $k \in \mathcal{I}$. Hence, for each $j^* \in \mathcal{U}$, we define

$$\mathcal{I}_{j^*} = \left\{k : k \in \mathcal{I}, j^* = \arg\max_j \exp\left(-\frac{\|\mathbf{x}_{ij}-\mathbf{x}^k\|^2}{\sigma^2}\right)\right\}.$$

All the features $s(\mathbf{x}^k,\mathbf{B}_i)$, $k \in \mathcal{I}_{j^*}$ can be computed from $\mathbf{x}_{ij^*}$, i.e., $s(\mathbf{x}^k,\mathbf{B}_i) = s(\mathbf{x}^k,\{\mathbf{x}_{ij^*}\})$ for $k \in \mathcal{I}_{j^*}$.

It is straightforward to show that $\mathcal{I} = \bigcup_{j^*\in\mathcal{U}} \mathcal{I}_{j^*}$ and, in general, $\mathcal{I}_{j_1^*} \bigcap \mathcal{I}_{j_2^*} \neq \emptyset$ for arbitrary $j_1^* \neq j_2^* \in \mathcal{U}$. In fact, the number of appearances of k in $\mathcal{I}_{j_1^*},\cdots,\mathcal{I}_{j_{|\mathcal{U}|}^*}$ is m_k. We then rewrite the bag classifier (10) in terms of the instances indexed by $\mathcal{U}$:

$$y = \operatorname{sign}\left(\sum_{j^*\in\mathcal{U}} g(\mathbf{x}_{ij^*}) + b^*\right),$$

where

$$g(\mathbf{x}_{ij^*}) = \sum_{k\in\mathcal{I}_{j^*}} \frac{w_k^* s(\mathbf{x}^k,\mathbf{x}_{ij^*})}{m_k} \tag{11}$$

determines the contribution of $\mathbf{x}_{ij^*}$ to the classification of the bag $\mathbf{B}_i$. The instances can be classified according to (11): If $g(\mathbf{x}_{ij^*}) > \tau$, $\mathbf{x}_{ij^*}$ belongs to the positive class; otherwise, $\mathbf{x}_{ij^*}$ belongs to the negative class. The choice of τ is application specific and is an interesting research problem for its own sake. In our experiments, the parameter τ is chosen to be bag dependent as $-\frac{b^*}{|\mathcal{U}|}$.

Next, we present one simple example to illustrate the major steps in the above instance classification process.

Example. We assume that five features are selected by 1-norm SVM and denote these features as $s(\mathbf{x}^1,\cdot),\cdots,s(\mathbf{x}^5,\cdot)$. Therefore, $\mathcal{I} = \{1,2,3,4,5\}$. A bag $\mathbf{B}_i$, containing four instances ($\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{x}_{i3}$, and $\mathbf{x}_{i4}$), is classified as positive. The key information needed in computing $s(\mathbf{x}^k, \mathbf{B}_i)$ is the nearest neighbors of $\mathbf{x}^k$ in the instances of $\mathbf{B}_i$. Suppose that

$$\begin{array}{rll} \mathbf{x}_{i2}, \mathbf{x}_{i3} & \text{are the nearest neighbors of} & \mathbf{x}^1; \\ \mathbf{x}_{i1} & \text{is the nearest neighbor of} & \mathbf{x}^2; \\ \mathbf{x}_{i3} & \text{is the nearest neighbor of} & \mathbf{x}^3; \\ \mathbf{x}_{i1} & \text{is the nearest neighbor of} & \mathbf{x}^4; \\ \mathbf{x}_{i3} & \text{is the nearest neighbor of} & \mathbf{x}^5. \end{array} \quad (12)$$

Instances $\mathbf{x}_{i1}$, $\mathbf{x}_{i2}$, and $\mathbf{x}_{i3}$ are the nearest neighbors of at least one of $\mathbf{x}^k$s, so $\mathcal{U} = \{1, 2, 3\}$. Since $\mathbf{x}_{i4}$ is not the nearest neighbor for any of $\mathbf{x}^k$s, $\mathbf{x}_{i4}$ is assigned to the void class. Because $\mathbf{x}^1$ has two nearest neighbors, it yields $m_1 = 2$. Similarly, we have $m_2 = m_3 = m_4 = m_5 = 1$. From (12), we find that $\mathbf{x}_{i1}$ is the nearest neighbor of $\mathbf{x}^2$ and $\mathbf{x}^4$. So, $\mathbf{x}_{i1}$ determines the values of features $s(\mathbf{x}^2, \cdot)$ and $s(\mathbf{x}^4, \cdot)$. Therefore, $\mathcal{I}_1 = \{2, 4\}$. Similarly, we derive $\mathcal{I}_2 = \{1\}$ and $\mathcal{I}_3 = \{1, 3, 5\}$. In other words, the instance $\mathbf{x}_{i1}$ contributes to the classification of $\mathbf{B}_i$ via features $s(\mathbf{x}^2, \cdot)$ and $s(\mathbf{x}^4)$, the instance $\mathbf{x}_{i2}$ contributes to the classification via $s(\mathbf{x}^1, \cdot)$, and the instance $\mathbf{x}_{i3}$ contributes to the classification via $s(\mathbf{x}^1, \cdot)$, $s(\mathbf{x}^3, \cdot)$, and $s(\mathbf{x}^5, \cdot)$. The labels of instances $\mathbf{x}_{i1}$, $\mathbf{x}_{i2}$, and $\mathbf{x}_{i3}$ can be predicted using (11).

4 An Algorithmic View

We summarize the above discussion in pseudo code. The input is a set of labeled bags $\mathcal{D}$, parameters σ^2, λ, and μ. The collection of instances from all the bags is denoted as $\mathcal{C} = \{\mathbf{x}^k : k = 1, \cdots, n\}$. The following pseudo code learns a bag classifier defined by $\mathbf{w}^*$ and b^*).

Algorithm 4.1: Learning Bag Classifier

```
1 FOR (every bag $\mathbf{B}_i = \{\mathbf{x}_{ij} : j = 1, \cdots, n_i\}$ in $\mathcal{D}$)
2   FOR (every instance $\mathbf{x}^k$ in $\mathcal{C}$)
3     $d = \min_j \| \mathbf{x}_{ij} - \mathbf{x}^k \|$
4     the $k$th element of $\mathbf{m}(\mathbf{B}_i)$ is $s(\mathbf{x}^k, \mathbf{B}_i) = e^{-\frac{d^2}{\sigma^2}}$
5   END
6 END
7 solve the LP in (9)
8 OUTPUT ($\mathbf{w}^*$ and $b^*$)
```

The pseudocode for instance classification is given below. The input is a bag $\mathbf{B}_i = \{\mathbf{x}_{ij} : j = 1, \cdots, n_i\}$, which is classified as positive by a bag classifier $(\mathbf{w}^*, b^*)$. The output is a list of positive instances along with their contributions to the classification of the bag.

Algorithm 4.2: Instance Classification of Bag $\mathbf{B}_i$

```
 1 let $\mathcal{I} = \{k : |w_k^*| > 0\}$
 2 let $\mathcal{U} = \{j^* : j^* = \arg\min_j \| \mathbf{x}_{ij} - \mathbf{x}^k \|, k \in \mathcal{I}\}$
 3 initialize $m_k = 0$ for every $k$ in $\mathcal{I}$
 4 FOR (every $j^*$ in $\mathcal{U}$)
 5   $\mathcal{I}_{j^*} = \{k : k \in \mathcal{I}, j^* = \arg\min_j \| \mathbf{x}_{ij} - \mathbf{x}^k \|\}$
 6   $m_k \leftarrow m_k + 1$ for every $k$ in $\mathcal{I}_{j^*}$
 7 END
 8 FOR (every $\mathbf{x}_{ij^*}$ with $j^*$ in $\mathcal{U}$)
 9   compute $g(\mathbf{x}_{ij^*})$ using (11)
10 END
11 OUTPUT (all $\mathbf{x}_{ij^*}$ satisfying $g(\mathbf{x}_{ij^*}) > -\frac{b^*}{|\mathcal{U}|}$)
```

These outputs $\mathbf{x}_{ij^*}$ correspond to positive instances.

5 Experimental Results

We present systematic evaluations of the proposed MIL framework, MILES, based on three publicly available benchmark data sets. In Section 5.1, we compare MILES with other MIL methods using the benchmark data sets in MIL, MUSK data sets [19]. In Section 5.2, we test the performance of MILES on a region-based image categorization problem using the same data set as in [16], and compare MILES with other techniques. In Section 5.3, We apply MILES to an object class recognition problem [21]. MILES is compared to several techniques specifically designed for the recognition task in terms of its performance. Computational issues are discussed in Section 5.4. A Matlab implementation of MILES is available at http://www.cs.olemiss.edu/~ychen/MILES.html/.

5.1 Drug Activity Prediction

5.1.1 Experimental Setup

The MUSK data sets, MUSK1 and MUSK2, are benchmark data sets for MIL. Both data sets are publicly available from the UCI Machine Learning Repository [9]. The data sets consist of descriptions of molecules. Specifically, a bag represents a molecule. Instances in a bag represent low-energy shapes of the molecule. To capture the shape of molecules, a molecule is placed in a standard position and orientation and then a set of 162 rays emitting from the origin is constructed to sample the molecular surface approximately uniformly [19]. There are also four features that describe the position of an oxygen atom on the molecular surface. Therefore, each instance in the bags is represented by 166 features. The data were preprocessed by dividing each feature value by 100 (this is identical to the data preprocessing step in [36]). MUSK1 has 92 molecules (bags), of which 47 are labeled positive, with an average of 5.17 shapes (instances) per molecule. MUSK2 has 102 molecules, of which 39 are positive, with an average of 64.69 shapes per molecule.

Three parameters σ^2 (in the most-likely-cause estimator), λ, and μ (in the LP) need to be specified for MILES. We fixed $\mu = 0.5$ to penalize equally on errors in the positive class and the negative class. The parameters λ and σ^2 were selected according to a twofold cross-validation on the training set. We chose λ from 0.1 to 0.6 with step size 0.01, and σ^2 from 10^5 to 9×10^5 with step size 5×10^4. We found that $\lambda = 0.45$ and $\sigma^2 = 5 \times 10^5$ gave the minimum twofold cross-validation error on MUSK1, and $\lambda = 0.37$ and $\sigma^2 = 8 \times 10^5$ gave the best twofold cross-validation performance on MUSK2. These values were fixed for the subsequent experiments. The linear program of 1-norm SVM was solved using CPLEX version 9.0 [24].

5.1.2 Classification Results

Table 1 shows the prediction accuracy. We observed variations on the average accuracy of tenfold cross-validation for different random runs. The prediction accuracy in 15 runs varied from 84.1-90.2 percent for MUSK1, and from 84.5-91.5 percent for MUSK2. Therefore, we reported the mean and 95 percent confidence interval of the results of 15 runs of tenfold cross-validation for MILES. We also included the results using the leave-one-out test for comparison with certain algorithms.

Table 1 summarizes the performance of twelve MIL algorithms in the literature: APR [19], DD [35], DD-SVM

TABLE 1
Comparing the Prediction Accuracy (in Percent) Obtained Using MILES with Those of Other Methods on the MUSK Data Sets

Algorithms	MUSK1	MUSK2	Type of Testing
MILES	$86.3 : [84.9, 87.7]$	$87.7 : [86.3, 89.1]$	tenfold cross-validation
	87.0	**93.1**	leave-one-out test
APR [19]	92.4	89.2	tenfold cross-validation
Bagging-APR [63]	**92.8**	**93.1**	tenfold cross-validation
Bayesian-kNN [53]	90.2	82.4	leave-one-out test
Citation-kNN [53]	92.4	86.3	leave-one-out test
DD [35]	88.9	82.5	tenfold cross-validation
DD-SVM [16]	85.8	91.3	tenfold cross-validation
EM-DD [61]	84.8	84.9	tenfold cross-validation
mi-SVM [2]	87.4	83.6	tenfold cross-validation
MI-SVM [2]	77.9	84.3	tenfold cross-validation
MI-NN [43]	88.0	82.0	tenfold cross-validation
Multinst [4]	$76.7 : [73.6, 79.8]$	$84.0 : [81.4, 86.6]$	tenfold cross-validation
RELIC [46]	83.7	87.3	tenfold cross-validation

The 95 percent confidence interval of the average accuracy of tenfold cross-validation for different random runs is Included only for MILES and MULTINST.

[16], EM-DD [61],[2] MI-SVM and mi-SVM [2], MI-NN [43], Multinst [4], Citation-kNN and Bayesian-kNN [53], RELIC [46], and Bagging-APR [63].[3] The prediction accuracies for Bayesian-kNN and Citation-kNN were based on the leave-one-out test, while the performance of other methods was evaluated using tenfold cross-validation.

Table 1 shows that Bagging-APR achieves the best performance on both MUSK1 and MUSK2 data sets. Bagging-APR applied an ensemble algorithm, bagging [11], to a base learner that used the APR method. Therefore, its excellent performance should be credited to the ensemble scheme as well as the base learner. Since the bagging technique can potentially be applied to all algorithms listed in Table 1 to improve the prediction accuracy, it might be more meaningful to compare only the base learners. Among all 12 base multiple-instance learners, the APR method gave the best overall prediction accuracy on MUSK1 and MUSK2 data sets (the average accuracy on two data sets is 90.8 percent). It is observed that the proposed framework, MILES, is the second best in terms of the average prediction accuracy (90.1 percent, leave-one-out test) over the two data sets. The tenfold cross-validation results of MILES are also comparable with those of other techniques. Even though MILES is not designed specially for the drug activity prediction problem, its performance is comparable with that of the APR method. In fact, MILES generated the best performance on MUSK2 (93.1 percent, leave-one-out test).

We tested the sensitivity of MILES with respect to the parameters λ and σ^2. Our algorithm is rather stable with choices of parameter values as long as they are chosen from a reasonable range. Fig. 2 shows the gray-level encoded leave-one-out test accuracy under different values of λ and σ^2. For MUSK1, λ varies from 0.2 to 0.5 with step size 0.01; σ^2 varies from 3×10^5 to 7×10^5 with step size 5×10^4. The maximal and minimal leave-one-out test accuracies are, respectively, 89.1 percent and 81.5 percent (recall that the accuracy based on the parameters chosen from twofold cross-validation is 87.0 percent). For MUSK2, λ varies from 0.1 to 0.6 with step size 0.02; σ^2 varies from 5×10^5 to 9×10^5 with step size 5×10^4. The maximal and minimal leave-one-out test accuracies are, respectively, 94.1 percent and 83.3 percent (recall that the accuracy based on the parameters chosen from twofold cross-validation is 93.1 percent).

5.2 Region-Based Image Categorization

Image categorization refers to the labeling of images into predefined categories. Depending on the imagery features used in the classification, image categorization algorithms can be divided into two groups: *global approaches* and *component-based approaches*. The global image classification approaches use features that characterize the global information of an image. Although the global features can usually be computed with little cost and are effective for certain classification tasks, some visual contents of images could only be locally defined. A number of component-based approaches have been proposed to exploit local and spatial properties of an image. Component-based image categorization has been formulated and tackled as a multiple-instance learning problem in the past works [37], [56], [60], [2], [16].

5.2.1 Experimental Setup

The image data set consists of 2,000 images taken from 20 CD-ROMs published by the COREL Corporation. Each COREL CD-ROM contains 100 images representing a distinct concept.[4] Therefore, the data set has 20 thematically diverse

2. The EM-DD results reported in [61] were obtained by selecting the optimal solution using the test data. The EM-DD result cited in this paper was provide by [2] using the correct algorithm.

3. The best performance reported in [63] was achieved by combining EM-DD learners (prediction accuracy: 96.9 percent for MUSK1 and 97.0 percent for MUSK2). However, it seems that the EM-DD learner implemented in [63] used the test data to select the optimal solution. Therefore, the results were not included in Table 1.

4. The image data sets are available at http://www.cs.uno.edu/~yixin/ddsvm.html/.

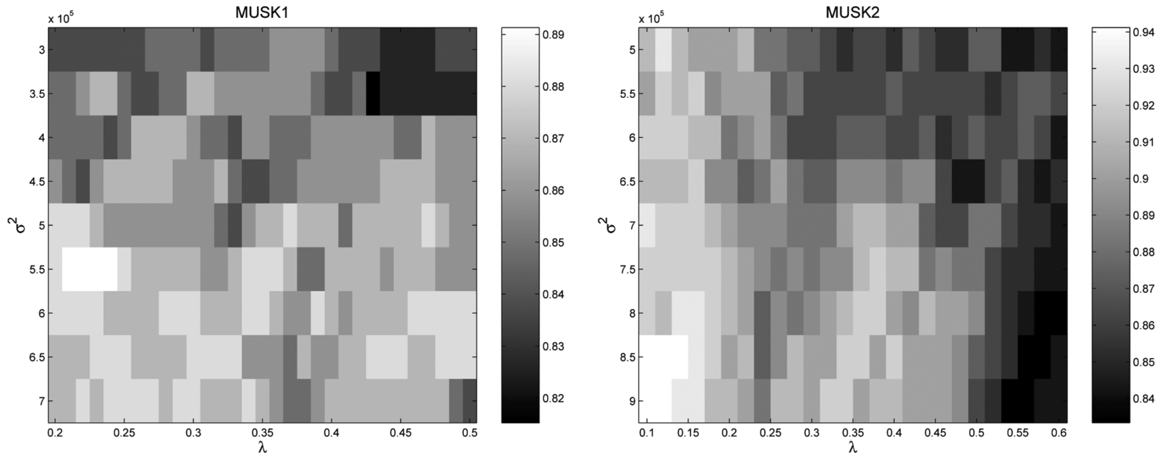


Fig. 2. Sensitivity analysis: classification accuracy of leave-one-out test under different values of parameters. The correspondence between the accuracies and gray-scale levels are indicated by the scale bar images.

image categories, each containing 100 images. Images are in JPEG format of size 384×256 or 256×384. The category names are listed in Table 2 along with the identifiers (IDs) for 20 categories. Since the classification problem is multiclass, we apply the simple one-against-the-rest heuristics.

This data set has been used in [16] to demonstrate much improved performance of a MIL algorithm, DD-SVM, in comparison with several other techniques. Since we will compare MILES with the DD-SVM approach, we adopt the same image segmentation algorithm as described in [16], [54]. A brief summary about the imagery features is given as follows: To segment an image, the system first partitions the image into nonoverlapping blocks of size 4×4 pixels. A feature vector is then extracted for each block. Each feature vector consists of six features. Three of them are the average color components in a block. The LUV color space is used, where L encodes luminance, and U and V encode color information (chrominance). The other three represent the square root of energy in the high-frequency bands of the wavelet transforms, i.e., the square root of the second order moment of wavelet coefficients in high frequency bands. The coefficients in different frequency bands show variations in different directions, hence capture the texture properties. To calculate these moments, a Daubechies-4 wavelet transform is applied to the L component of the image. After one-level wavelet transform, a 4×4 block is decomposed into four frequency bands: the LL (low low), LH (low high), HL, and HH bands, each containing 2×2 coefficients. If the coefficients in the HL band are given as $c_{i,j}$, where $i, j = 1, 2$, then a feature is defined as $(\frac{1}{4}\sum_{i=1}^{2}\sum_{j=1}^{2} c_{i,j}^2)^{\frac{1}{2}}$. The other two features are computed similarly from the LH and HH bands.

A modified k-means algorithm is applied to group the feature vectors into clusters each corresponding to a region in the segmented image. The algorithm does not require the number of clusters be specified. Instead, the number of clusters gradually increases until a stop criterion is met. The number of regions in an image can vary depending on the complexity of the image content. The average number of regions per image changes in accordance with the adjustment of the stop criteria. Table 2 lists the average number of regions per image in each category. Fig. 3 shows some images randomly sampled from the 20 categories and the corresponding segmentation results. After segmentation, three extra features are computed for each region to describe shape properties. They are normalized inertia of order 1, 2, and 3. As a result, each region in any image is characterized by a nine-dimensional feature vector characterizing the color, texture, and shape properties of the region.

In our experiments, images within each category were randomly partitioned in half to form a training set and a test set. We repeated each experiment for five random splits, and reported the average of the results obtained over five different test sets. The parameter μ was set to be 0.5. Parameters σ^2 and λ were selected according to a twofold

TABLE 2
Twenty Image Categories and the Average Number of Regions per Image for Each Category

Category ID	Category Name	Regions per Image
0	African people and villages	4.84
1	Beach	3.54
2	Historical building	3.1
3	Buses	7.59
4	Dinosaurs	2.00
5	Elephants	3.02
6	Flowers	4.46
7	Horses	3.89
8	Mountains and glaciers	3.38
9	Food	7.24
10	Dogs	3.80
11	Lizards	2.80
12	Fashion models	5.19
13	Sunset scenes	3.52
14	Cars	4.93
15	Waterfalls	2.56
16	Antique furniture	2.30
17	Battle ships	4.32
18	Skiing	3.34
19	Desserts	3.65

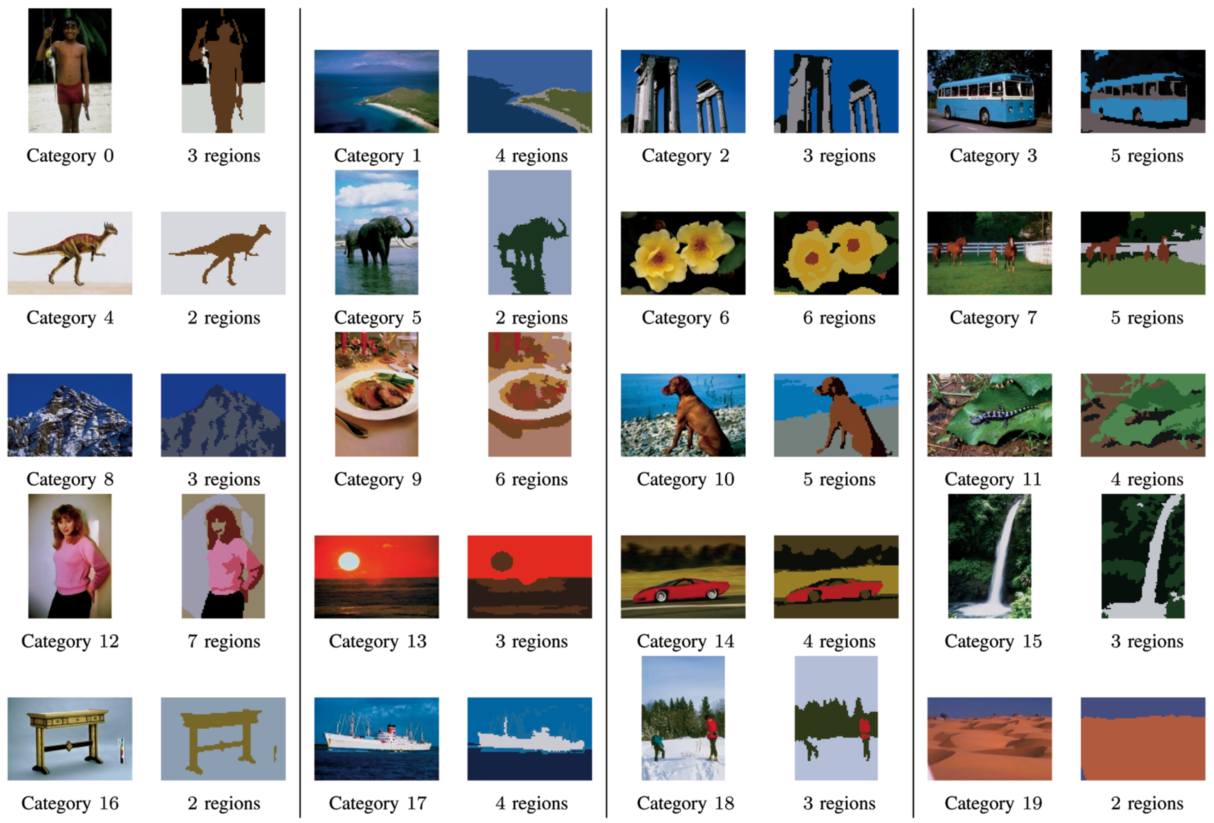


Fig. 3. Images randomly sampled from 20 categories and the corresponding segmentation results. Segmented regions are shown in their representative colors.

TABLE 3
The Confusion Matrix of Image Categorization Experiments (Over Five Randomly Generated Test Sets) Using MILES

	Cat. 0	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8	Cat. 9
Cat. 0	**68.8**	1.8	9.3	0.3	1.3	10.3	1.3	4.3	1.3	1.3
Cat. 1	4.0	**66.0**	4.0	3.6	0.7	3.0	1.3	0.0	15.7	1.7
Cat. 2	4.6	2.0	**75.7**	4.0	0.7	6.3	0.7	0.3	3.7	2.0
Cat. 3	0.0	2.7	2.3	**90.3**	0.0	1.0	0.0	0.0	2.0	1.7
Cat. 4	0.0	0.0	0.0	0.0	**99.7**	0.3	0.0	0.0	0.0	0.0
Cat. 5	2.0	1.3	5.0	0.0	0.7	**77.7**	0.0	6.6	6.0	0.7
Cat. 6	3.0	0.3	0.0	0.0	0.0	0.0	**96.4**	0.0	0.0	0.3
Cat. 7	1.0	1.3	0.7	0.0	0.0	1.3	0.7	**95.0**	0.0	0.0
Cat. 8	3.6	13.7	5.3	1.0	0.3	2.7	1.7	0.7	**71.0**	0.0
Cat. 9	5.0	1.3	0.3	1.0	0.7	2.3	1.0	2.0	1.0	**85.4**

Each row lists the average percentage of images (test images) in one category classified to each of the 10 categories. Numbers on the diagonal show the classification accuracies (in percent).

cross-validation on the training set. We chose λ from 0.1 to 0.6 with step size 0.05 and σ^2 from 5 to 15 with step size 1. We found that $\lambda = 0.5$ and $\sigma^2 = 11$ gave the minimum twofold cross-validation error. We then fixed $\lambda = 0.5$ and $\sigma^2 = 11$ in all subsequent experiments.

5.2.2 Categorization Results

We first report the confusion matrix of the proposed method in Table 3 based on images in Category 0 to Category 9, i.e., 1,000 images. Each row lists the average percentages of images in a specific category classified to each of the 10 categories. The numbers on the diagonal show the classification accuracy for each category and off-diagonal entries indicate classification errors. A detailed examination of the confusion matrix shows that two of the largest errors (the underlined numbers in Table 3) are errors between Category 1 (Beach) and Category 8 (Mountains and glaciers): 13.7 percent of *Mountains and glaciers* are misclassified as *Beach* and 15.7 percent of *Beach* images are misclassified as *Mountains and glaciers*. This observation is in line with that presented in [16]. As stated in [16], the high classification errors are due to the fact that many images from these two categories have regions that are semantically

TABLE 4
Comparing the Image Categorization Accuracy Obtained Using MILES with Those of Other Methods

Algorithms	1000-Image Data Set	2000-Image Data Set
MILES	**82.6** : $[81.4, 83.7]$	**68.7** : $[67.3, 70.1]$
DD-SVM [16]	81.5 : $[78.5, 84.5]$	67.5 : $[66.1, 68.9]$
MI-SVM [2], [16]	74.7 : $[74.1, 75.3]$	54.6 : $[53.1, 56.1]$
k-means-SVM [17]	69.8 : $[67.9, 71.7]$	52.3 : $[51.6, 52.9]$

The numbers listed are the average classification accuracies (in percent) over five random test sets and the corresponding 95 percent confidence intervals. The 1,000-image data set contains images from Category 0 to Category 9. The 2,000-image data set contains images from all 20 categories. Training and test sets are of equal size.

related and visually similar, such as regions corresponding to mountain, river, lake, and ocean.

We compared the overall prediction accuracy of MILES with that of DD-SVM [16], MI-SVM [2], and a method proposed in [17] (we call it k-means-SVM to simplify the discussions). k-means-SVM constructed a region vocabulary by clustering regions using the k-means algorithm. Each image was then transformed to an integer-valued feature vector indicating the number of regions assigned to each cluster. SVMs were constructed using these integer-valued features. The size of the vocabulary and the parameters of k-means-SVM were chosen according to a twofold cross-validation using all 2,000 images. The average classification accuracies over five random test sets and the corresponding 95 percent confidence intervals are provided in Table 4. On both data sets, the performance of MILES is significantly better than that of MI-SVM and k-means-SVM. MILES outperforms DD-SVM, though the difference is not statistically significant as the 95 percent confidence intervals for the two methods overlap.

5.2.3 Sensitivity to Labeling Noise

The results in Section 5.2.2 demonstrate that the performance of MILES is highly comparable with that of DD-SVM. Next, we compare MILES with DD-SVM in terms of the sensitivity to the noise in labels. Sensitivity to labeling noise is an important performance measure for classifiers because in many practical applications, it is usually impossible to get a "clean" data set and the labeling processes are often subjective. In terms of binary classification, we define the labeling noise as the probability that an image is mislabeled. In this experiment, training sets with different levels of labeling noise were generated as follows. We first randomly picked $d\%$ of positive images and $d\%$ of negative images from a training set. Then, we modified the labels of the selected images by negating their labels, i.e., positive (negative) images were labeled as negative (positive) images. Finally, we put these images with new labels back to the training set. The new training set has $d\%$ of images with negated labels (or "noisy" labels).

We compared the classification accuracy of MILES with that of DD-SVM for $d = 0$ to 30 (with step size 2) based on 200 images from Category 2 (Historical buildings) and Category 7 (Horses). The reason of selecting these two categories is that both DD-SVM and MILES produce almost perfect classification accuracies at $d = 0$, which makes them a good data set for comparing sensitivities at different levels

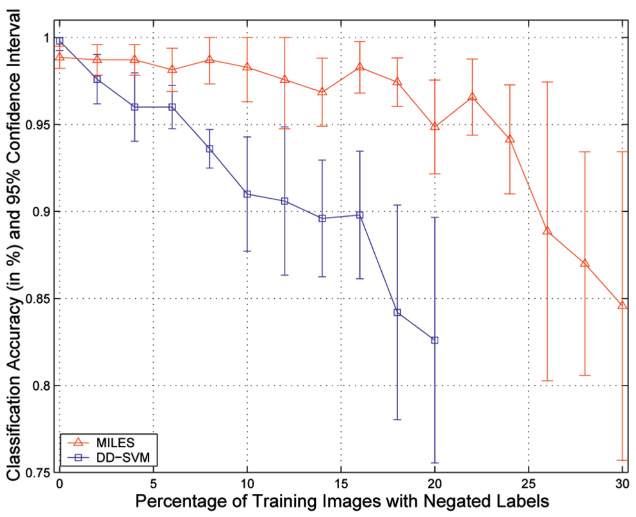


Fig. 4. Comparing MILES with DD-SVM on the robustness to labeling noise. The experiment is performed on 200 image in Category 2 and Category 7 (training and test sets are of equal size) with different levels of labeling noise on training images. The average classification accuracy and the corresponding 95 percent confidence intervals are computed over five randomly generated test sets.

of labeling noise. The training and test sets have equal size. The average classification accuracy (over five randomly generated test sets) and the corresponding 95 percent confidence interval are presented in Fig. 4. The average accuracy of DD-SVM drops quickly below 75 percent when $d > 20$; therefore, only the data associated with $d \leq 20$ are included in Fig. 4.

Fig. 4 shows that MILES and DD-SVM have similar classification accuracy when there is no labeling noise. As the noise level increases, the average classification accuracy of DD-SVM decreases rapidly, whereas the performance of MILES remains almost the same when d varies from 0 to 18. The lower bound of the 95 percent confidence interval for MILES is well above 90 percent even when 24 percent of training images have their labels negated. But for DD-SVM, the lower bound of the 95 percent confidence interval stays above 90 percent when the mislabeled images is less than 10 percent. This demonstrates the robustness of MILES against labeling noise. In contrast, the diverse density function in DD-SVM is very sensitive to instances in negative bags [16]. The diverse density value at a point is exponentially reduced if there is a single instance from a negative bag close to the point. Consequently, the local maximizers of the DD function are sensitive to labeling noise. MILES abandons the process of maximizing the DD function. The labeling noise can, in essence, be viewed as some "defective features" defined by the instances in the mislabeled bags. Because of the appropriate regularization in 1-norm SVM, the feature selection process tends to exclude those defective features. As a result, MILES is less sensitive to labeling noise.

5.3 Object Class Recognition

We carried out experiments on the Caltech data set used by Fergus et al. [21] and others [42], [5]. The goal is to learn and recognize object class models from unsegmented images. Each image is represented as a collection of "salient regions." Objects are modeled as flexible constellations of the regions. The labels of the training images are given, i.e., it is known

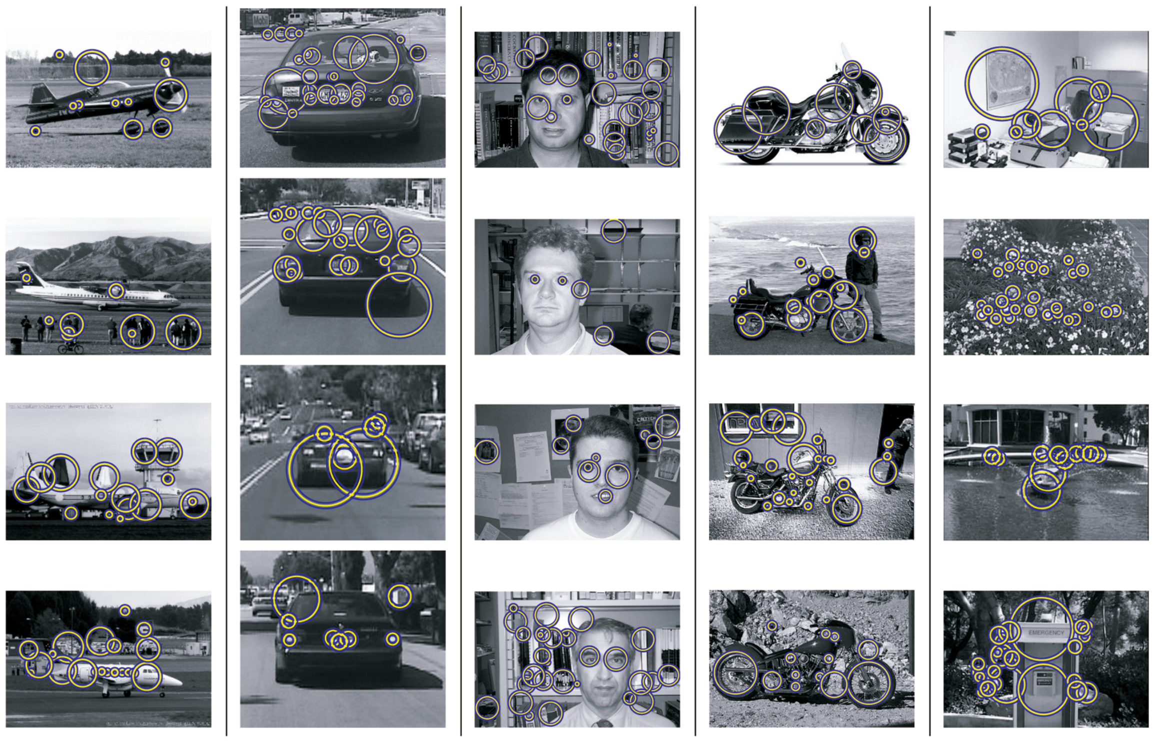


Fig. 5. Images randomly selected from five classes: Airplanes (first column), Cars (second column), Faces (third column), Motorbikes (fourth column), and Background images (fifth column). Each circle in an image represents a salient region.

whether or not a training image contains a certain object from a particular class. However, the labels of the salient regions in each training image are unknown, so it is not clear which salient regions correspond to the object, and which are not. This learning situation can be viewed as a multiple-instance problem. We compared MILES with the Fergus method [21] as well as the approaches described in [5], [42].

5.3.1 Experimental Setup

The image data set was downloaded from the Website of Robotics Research Group at the University of Oxford.[5] It contains the following four object classes and background images: Airplanes (800 images), Cars (800 images), Faces (435 images), Motorbikes (800 images), and Background (900 general background and 1,370 road background).

We followed the descriptions in [21] for feature extraction. A brief summary of the process is given below. All images are converted to gray-scale. Salient regions are found using the method introduced by Kadir and Brady [26] , which detects regions that are salient over both location and scale.[6] In our experiments, the scale varies between 5 and 50. Fig. 5 shows some images randomly selected from five classes along with the detected salient regions. Each salient region is cropped from the image and rescaled to an image patch of size 11×11 pixels so it corresponds to a 121-dimensional feature vector. Principal component analysis (PCA) is applied to all image patches for dimensionality reduction. Each patch is then represented by a vector of the coordinates within the first 15 principal components. The scale (radius of the salient region) and location of the salient region are also included. This leads to a total of 18 features for each image patch.

Images within each object class were partitioned in half to form a training set and a test set of equal size. Our partition is identical to that used in [21]. The background images were also divided equally into training and test sets. The parameter μ was chosen to be 0.5. The other two parameters σ^2 and λ were selected according to a twofold cross-validation on the training set. We found that $\lambda = 0.5$ and $\sigma^2 = 10^5$ gave the minimum twofold cross-validation error for airplanes, faces, and motorbikes, while $\lambda = 0.5$ and $\sigma^2 = 2 \times 10^5$ worked best for cars.

5.3.2 Recognition Results

The recognition decision is binary, i.e., *object present* versus *object absent*. To be consistent with the experimental setting in [21], [42], [5], images in the three object categories (Airplanes, Faces, and Motorbikes) were tested against the general background images, while the Cars images were tested against the road background. The performance is measured by the equal-error-rates point on the receiver-operating characteristic (ROC) curve, i.e., the point where the true positive rate is equal to the true negative rate. For example, a true positive rate of 90 percent at an equal-error-rates point means that 90 percent of the positive images (object present) are correctly classified, and 90 percent of the negative images (background images) are also correctly classified. The error rate is therefore 10 percent. The ROC curve of MILES is obtained by varying the b^* parameter in (10).

The comparison results are summarized in Table 5. The boosting technique proposed by Bar-Hillel et al. [5] can take two different methods of selecting weak learners, hence, we included in Table 5 the best results for each category. MILES gives the best performance on three object classes:

5. These data sets are available at http://www.robots.ox.ac.uk/~vgg/data.html/.

6. We used the salient region detector implemented by Timor Kadir. We used Version 1.5 in the experiments. The software is available at http://www.robots.ox.ac.uk/~timork/salscale.html/.

TABLE 5
Comparing the Object Class Recognition Performance Obtained Using MILES with that of Other Methods

Algorithms	Airplanes	Cars (Rear)	Faces	Motorbikes
MILES	**98.0**	94.5	**99.5**	**96.7**
Fergus et al. [21]	90.2	90.3	96.4	92.5
Opelt et al. [42]	88.9	90.1	93.5	92.2
Bar-Hillel et al. [5]	89.7	**97.7**	91.7	93.1

The numbers listed are the true positive rates (in percent) at the equal-error-rates point on the ROC curve (i.e., $\text{true positive rate} = \text{true negative rate}$*).*

The error rates of MILES over airplanes, faces, and motorbikes are around 20.4, 13.9, and 48 percent, respectively, of those of the second best (the Fergus method on Airplanes and Faces, the Bar-Hillel method on Motorbikes). The method proposed by Bar-Hillel et al. [5] gave the best results on Cars category. The overall recognition performance of MILES appears to be very competitive.

5.3.3 Selected Features and Instance Classification

Since the MILES classifier is determined entirely by the select features, these features describe the properties that are common to objects in the same class and separate objects from background clutters. According to (7), each selected feature is defined by a unique instance in the training bags. Next, we show some instances (or image patches) associated with the selected features. We divided the selected features into positive features and negative features according to the sign of their weights (w_k^*s). In the experiments for all four object classes, it is observed that every positive feature is defined by an image patch from a positive image (i.e., containing a certain object) and every negative feature is defined by an image patch from a background image. It is interesting that MILES succeeded in doing this without explicit constraint. On the other hand, this also suggests that these data sets may contain much more information than is typically assumed in the context of MIL [48]. Unless this issue can be fully investigated, the above object class recognition results should be interpreted cautiously.

Fig. 6 shows 32 image patches related to positive features for each object class. Within each class, the image patches are arranged (from left to right, top to bottom) in descending order of the magnitude of the weight coefficients, which indicates the degree of importance for the selected features. Even though all image patches related to positive features were selected from positive images, some patches were not taken from a region corresponding to the object. We call these features false positive features. An image patch for a false positive feature is displayed with a bounding box in Fig. 6. We

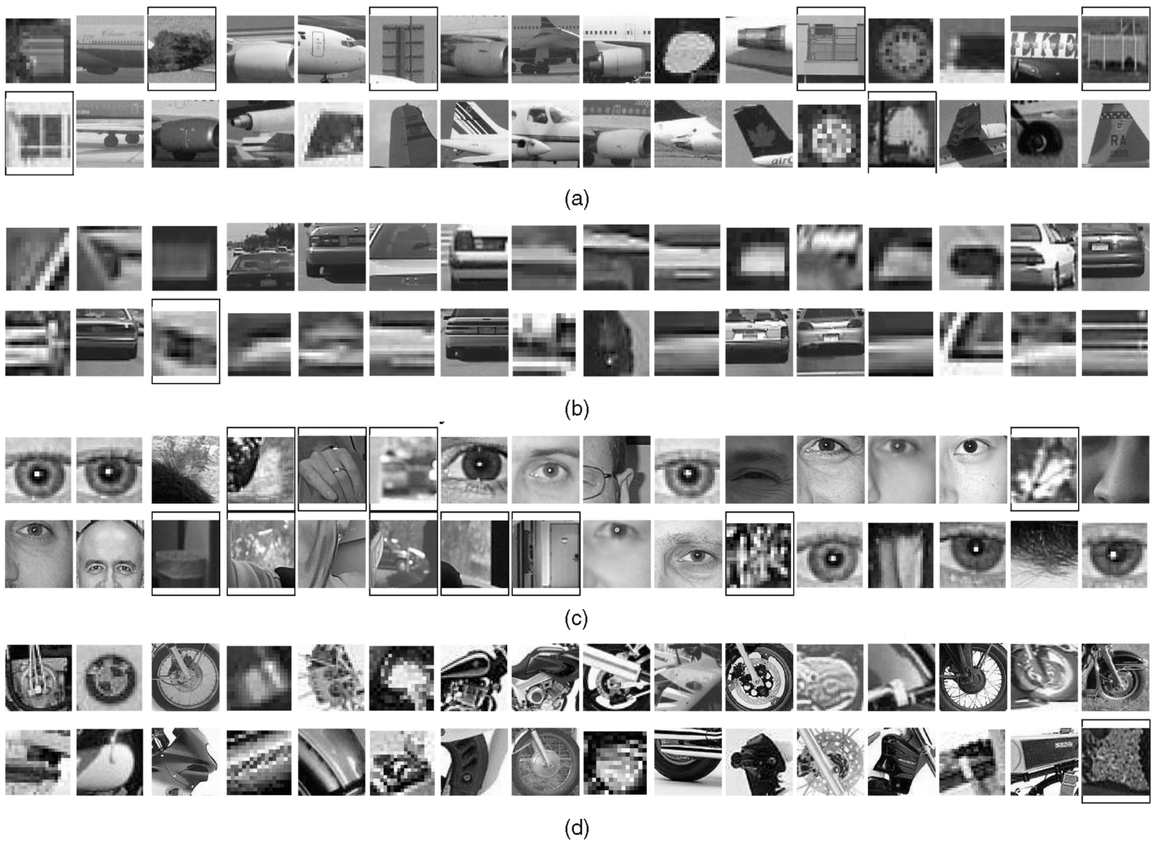
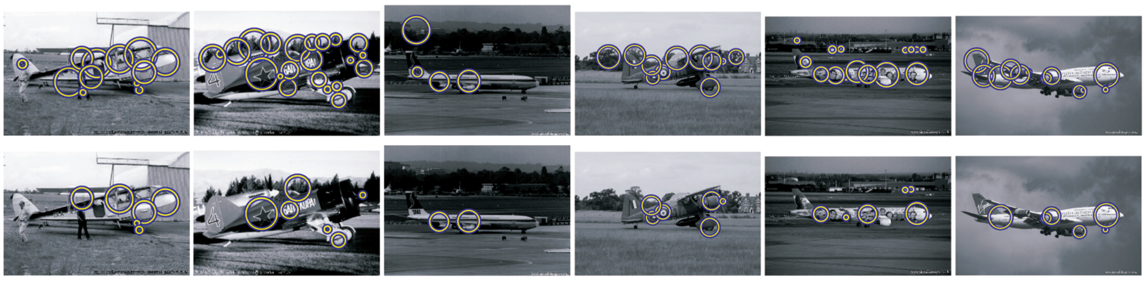
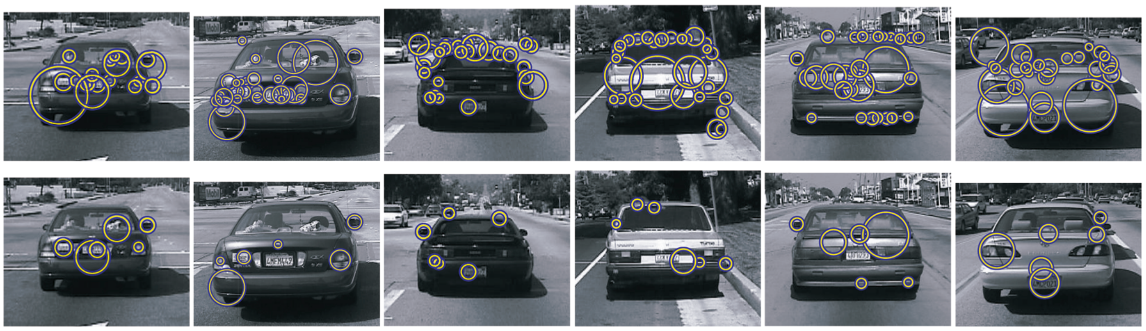


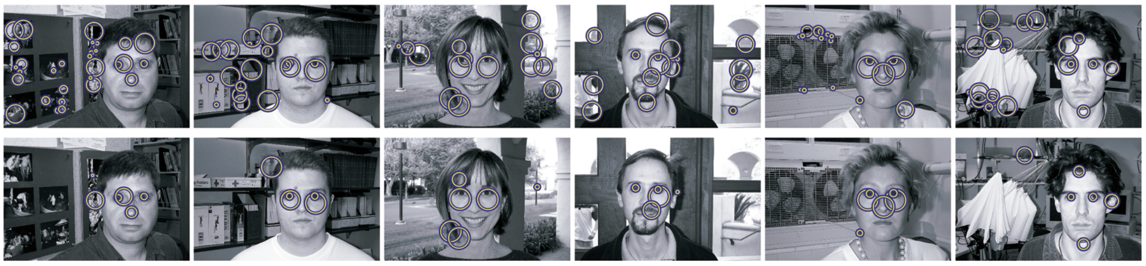
Fig. 6. Image patches corresponding to the selected positive features (i.e., features with positive weights). Each image patch defines a positive feature. The features defined by the patches enclosed by a bounding box are false positive features because these patches are not taken from the region associated with the targeted object. Only 32 patches are shown for each class. The four numbers below each block of image patches are the number of features before feature selection, the number of negative features, the number of positive features, and the number of false positive features. (a) Patches automatically selected from the images of airplanes (6,821, 97, 96, 16). (b) Patches automatically selected from the images of cars (10,041, 98, 97, 90. (c) Patches automatically selected from the images of faces (6,997, 27, 42, 14). (d) Patches automatically selected from the images of motorbikes (9995, 90, 101, 3).

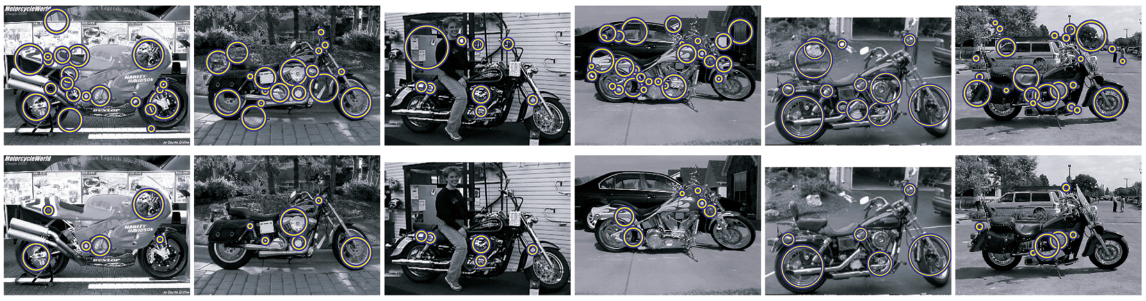


(a)



(b)





(c)

Fig. 7. Sample test images that are classified as containing a targeted object. For each block of images, images in the first row show all the detected salient regions enclosed by circles, and the corresponding images in the second row show only the regions that are identified by MILES as parts of the object. (a) Airplanes, (b) cars, (c) faces, and (d) motorbikes.

also included in the figure the number of features before feature selection (i.e., the number of instances in all training bags), the number of negative features, the number of positive features, and the number of false positive features. As indicated by these numbers, the solution of MILES is very sparse. On average, more than 98 percent of the weights are zero. The results also seem to suggest that the feature selection process of MILES can, to some extent, capture the characteristics of an object class.

Next, we tested the instance classification algorithm proposed in Section 3.2. Specifically, if an image is classified as containing a certain object, the contribution of each salient region within the image is evaluated using (11). A salient region is viewed as part of the object if its contribution is greater than $\tau = -\frac{b^*}{|\mathcal{U}|}$. Such salient regions are named positive regions. Fig. 7 shows some randomly selected test images from the four object classes. For each block of images, images in the first row show all the detected salient regions, and the corresponding images in the second row show only the regions that are identified by MILES as parts of the object. The results suggest that MILES performs well on instance classification. The selected positive regions may provide generative clues in identifying the location of the object.

TABLE 6
Comparing the Training Time (in Minutes) of MILES with that of the Fergus Method and DD-SVM

Algorithms	MUSK1	MUSK2	COREL	Caltech
MILES	**0.1 + 29**	**1.2 + 129**	**0.25 + 20**	**16 + 140**
DD-SVM [16]	500 + 112	1500 + 240	40 + 80	Not Available
Fergus method [21]	Not Available	Not Available	Not Available	2160 + 0

The numbers shown are the training time when the parameters are selected + the time needed in selecting the parameters.

5.4 Computation Time

Compared with the methods based on the diverse density framework (DD-SVM [16]) and the Fergus method, MILES is significantly more efficient in terms of computational complexity. We summarized in Table 6 the training time needed by MILES, DD-SVM, and the Fergus method on different data sets. The training of MILES, DD-SVM, and the Fergus method are performed on a 2.2 GHz PC, a 700 MHz PC, and a 2 GHz PC [21], respectively. The time on MUSK1 and MUSK2 data sets is based on the total training time of tenfold cross-validation (the parameters were selected from twofold cross-validation with 51 values for λ and 17 values for σ^2). The training time on the COREL data set is measured on a training set of 500 images (the parameters were selected from twofold cross-validation with 11 values for λ and 11 values for σ^2). The longest training time among the four object classes of the Caltech data set is shown in Table 6 (the parameters were chosen from twofold cross-validation with six values for λ and five values for σ^2). Overall, the learning process of MILES is more than one order of magnitude faster than that of the other two methods.

6 Conclusions and Future Work

We have proposed a learning algorithm, MILES, which transforms a MIL problem to a supervised learning problem. A bag is embedded in a feature space defined by the instances in all the training bags where the coordinates of a bag represent its similarities to various instances. A feature selection technique using 1-norm SVM is applied to identify discriminative features (or instances) and construct bag classifiers at the same time. We tested MILES over the benchmark data sets taken from applications of drug activity prediction, image categorization, and object class recognition. In comparison with other methods, MILES demonstrates competitive classification accuracy on all three data sets. In addition, it is on average more than one order of magnitude faster in terms of learning speed than the Fergus method and a method based on the diverse density framework [16]. Moreover, it demonstrates good performance in instance classification and high robustness to the labeling noise.

MILES has several limitations:

- The performance of MILES depends on whether there are "useful" features among those defined by the instances in the training bags. Constraining the features to only those derived from the training bags reduces the search space, but may also hamper the performance on small data sets where it is possible that none of the instances is close to a "target point." This may explain the observation that MILES performs better on theleave-one-out test than on tenfold cross-validation for the MUSK1 and MUSK2 data sets.
- In some applications, for example 3D object recognition [45], geometric constraints on the image patches are extremely useful in reducing the search space and improving the recognition accuracy. However, MILES is not designed to take advantage of this type of prior knowledge. The instance-based feature mapping cannot easily model the spatial relationships among the instances.
- The feature vectors generated by the mapping (7) are not sparse. Therefore, the LP requires the storage of a data matrix of size $(\ell^+ + \ell^-) \times n$, where $\ell^+ + \ell^-$ is the number of training bags, and n is the number of instances in all the training bags. For some applications, n can be several orders of magnitude greater than $\ell^+ + \ell^-$. Therefore, the storage requirement can be prohibitive. The needed storage space could be significantly reduced by an instance similarity measure that generates sparse features.

Continuations of this work could take several directions.

- The instance classification method can be improved. The current decision rule, which is based on a threshold $-\frac{b^*}{|\mathcal{U}|}$, may identify many false positive instances, especially when the number of instances in a bag is large. The false positive rate may be reduced by optimizing the threshold.
- The mapping used in MILES can be viewed as feature aggregation for bags, i.e., take the maximum of each feature across instance in a bag. In this sense, MILES is closely related to the work by Gärtner et al. [22], in which several different aggregation operators were proposed to build a bag-level representation and, subsequently, a bag kernel. Therefore, it will be interesting to test different aggregation operators within the MILES framework.
- In some applications, the training data are given in a one-class setting. For example, the domain-based protein interaction inference problem typically only has positive training samples, namely, pairs of proteins that interact [62]. The interaction of two proteins is determined by the interaction of their domains. Therefore, if we view a pair of proteins as a bag and the domain-domain pairs as the instances in the bag, we have a multiple-instance representation. Multiple-instance learning with one-class training data is an interesting direction that we would like to pursue.

MILES can be integrated into a target tracking system for object detection. Evaluation of different feature selection techniques and investigation of the connection between the current approach and generative probabilistic methods are also interesting.

ACKNOWLEDGMENTS

Y. Chen is supported by a Louisiana Board of Regents RCS program under Grant No. LBOR0077NR00C, the US National Science Foundation EPSCoR program under Grant No. NSF/LEQSF(2005)-PFUND-39, the Research Institute for Children, and the University of New Orleans. This research work was conducted when Y. Chen was with the Department of Computer Science at University of New Orleans, New Orleans, Lousiana and when J. Bi was with the Department of Mathematical Sciences at Rensselaer Polytechnic Institute, 110 8th Street, Troy, New York. J.Z. Wang is supported by the US National Science Foundation under Grant Nos. IIS-0219272, IIS-0347148, and ANI-0202007, The Pennsylvania State University, and the PNC Foundation. The authors would like to thank the anonymous reviewers and the associate editor for their comments which have led to improvements of this paper. They would also like to thank Timor Kadir for providing the salient region detector and Rob Fergus for sharing the details on the object class recognition experiments.

REFERENCES

[1] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. Seventh European Conf. Computer Vision,* vol. 4, pp. 113-130, 2002.

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support Vector Machines for Multiple-Instance Learning," *Advances in Neural Information Processing Systems 15,* pp. 561-568, 2003.

[3] S. Andrews and T. Hofmann, "Multiple-Instance Learning via Disjunctive Programming Boosting," *Advances in Neural Information Processing Systems 16,* 2004.

[4] P. Auer, "On Learning from Mult-Instance Examples: Empirical Evaluation of a Theoretical Approach," *Proc. 14th Int'l Conf. Machine Learning,* pp. 21-29, 1997.

[5] A. Bar-Hillel, T. Hertz, and D. Weinshall, "Object Class Recognition by Boosting a Part-Based Model," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 702-709, 2005.

[6] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D.M. Blei, and M.I. Jordan, "Matching Words and Pictures," *J. Machine Learning Research,* vol. 3, pp. 1107-1135, 2003.

[7] K.P. Bennett, "Combining Support Vector and Mathematical Programming Methods for Classification," *Advances in Kernel Methods-Support Vector Machines,* B. Schölkopf, C. Burges and A. Smola, eds., pp. 307-326, 1999.

[8] J. Bi, K.P. Bennett, M. Embrechts, C. Breneman, and M. Song, "Dimensionality Reduction via Sparse Support Vector Machines," *J. Machine Learning Research,* vol. 3, pp. 1229-1243, 2003.

[9] C.L. Blake and C.J. Merz, UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/, 1998.

[10] A. Blum and A. Kalai, "A Note on Learning from Multiple-Instance Examples," *Machine Learning,* vol. 30, no. 1, pp. 23-29, 1998.

[11] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, pp. 123-140, 1996.

[12] M. Bressan and J. Vitria, "On the Selection and Classification of Independent Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 10, pp. 1312-1317, Oct. 2003.

[13] B.G. Buchanan and T.M. Mitchell, "Model-Directed Learning of Production Rules," *Pattern-Directed Inference Systems,* pp. 297-312, Academic Press, 1978.

[14] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM J. Scientific Computing,* vol. 20, no. 1, pp. 33-61, 1998.

[15] Y. Chen and J.Z. Wang, "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 9, pp. 1252-1267, Sept. 2002.

[16] Y. Chen and J.Z. Wang, "Image Categorization by Learning and Reasoning with Regions," *J. Machine Learning Research,* vol. 5, pp. 913-939, 2004.

[17] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual Categorization with Bags of Keypoints," *Proc. ECCV '04 Workshop Statistical Learning in Computer Vision,* pp. 59-74, 2004.

[18] L. De Raedt, "Attribute-Value Learning versus Inductive Logic Programming: The Missing Links," *Lecture Notes in Artificial Intelligence,* vol. 1446, pp. 1-8, 1998.

[19] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez, "Solving the Multiple Instance Problem with Axis-Parallel Rectangles," *Artificial Intelligence,* vol. 89, nos. 1-2, pp. 31-71, 1997.

[20] G. Dorkó and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 634-639, 2003.

[21] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 264-271, 2003.

[22] T. Gärtner, A. Flach, A. Kowalczyk, and A.J. Smola, "Multi-Instance Kernels," *Proc. 19th Int'l Conf. Machine Learning,* pp. 179-186, 2002.

[23] F.J. Iannarilli, Jr. and P.A. Rubin, "Feature Selection for Multiclass Discrimination via Mixed-Integer Linear Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 6, pp. 779-783, June 2003.

[24] ILOG, *ILOG CPLEX 6.5 Reference Manual,* ILOG CPLEX Division, Incline Village, NV, 1999.

[25] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 153-158, Feb. 1997.

[26] T. Kadir and M. Brady, "Scale, Saliency and Image Description," *Int'l J. Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.

[27] T. Kadir, A. Zisserman, and M. Brady, "An Affine Invariant Salient Region Detector," *Proc. Eighth European Conf. Computer Vision,* pp. 404-416, 2004.

[28] B. Krishnapuram, A.J. Hartemink, L. Carin, and M.A.T. Figueiredo, "A Bayesian Approach to Joint Feature Selection and Classifier Design," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1105-1111, Sept. 2004.

[29] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 273-324, 1997.

[30] N. Kwak and C.-H. Choi, "Input Feature Selection by Mutual Information Based on Parzen Window," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1667-1671, Dec. 2002.

[31] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain, "Simultaneous Feature Selection and Clustering Using Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1154-1166, Sept. 2004.

[32] J. Li and J.Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 9, pp. 1075-1088, Sept. 2003.

[33] P.M. Long and L. Tan, "PAC Learning Axis-Aligned Rectangles with Respect to Product Distribution from Multiple-Instance Examples," *Machine Learning,* vol. 30, no. 1, pp. 7-21, 1998.

[34] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 837-842, Aug. 1996.

[35] O. Maron and T. Lozano-Pérez, "A Framework for Multiple-Instance Learning," *Advances in Neural Information Processing Systems 10,* pp. 570-576, 1998.

[36] O. Maron, "Learning from " Dept. of Electrical and Computer Science, Massachusetts Inst. of Technology, Cambridge, 1998.

[37] O. Maron and A.L. Ratan, "Multiple-Instance Learning for Natural Scene Classification," *Proc. 15th Int'l Conf. Machine Learning,* pp. 341-349, 1998.

[38] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *Int'l J. Computer Vision,* vol. 60, no. 1, pp. 63-86, 2004.

[39] P. Mitra, C.A. Murthy, and S.K. Pal, "Unsupervised Feature Selection Using Feature Similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 301-312, Mar. 2002.

[40] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 4, pp. 349-361, Apr. 2001.

[41] J. Novovicova, P. Pudil, and J. Kittler, "Divergence Based Feature Selection for Multimodal Class Densities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 2, pp. 218-223, Feb. 1996.

[42] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak Hypotheses and Boosting for Generic Object Detection and Recognition," *Proc. Eighth European Conf. Computer Vision,* vol. 2, pp. 71-84, 2004.

[43] J. Ramon and L. De Raedt, "Multi Instance Neural Networks," *Proc. ICML-2000 Workshop Attribute-Value and Relational Learning,* 2000.

[44] S. Ray and M. Craven, "Supervised versus Multiple Instance Learning: An Empirical Comparison," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 697-704, 2005.

[45] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 272-277, 2003.

[46] G. Ruffo, "Learning Single and Multiple Decision Trees for Security Applications," PhD Dissertation, Dept. of Computer Science, Univ. of Turin, Italy, 2000.

[47] S.D. Scott, J. Zhang, and J. Brown, "On Generalized Multiple-Instance Learning," *Int'l J. Computational Intelligence and Applications,* vol. 5, no. 1, pp. 21-35, 2005.

[48] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering Object Categories in Image Collections," *Proc. Int'l Conf. Computer Vision,* vol. I, pp. 370-377, 2005.

[49] A.J. Smola, B. Schölkopf, and G. Gätsch, "Linear Programs for Automatic Accuracy Control in Regression," *Proc. Int'l Conf. Artificial Neural Networks,* pp. 575-580, 1999.

[50] P. Somol, P. Pudil, and J. Kittler, "Fast Branch & Bound Algorithms for Optimal Feature Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 7, pp. 900-912, July 2004.

[51] R. Tibshirani, "Regression Shrinkage and Selection via the LASSO," *J. Royal Statistical Soc., Series B,* vol. 58, pp. 267-288, 1996.

[52] T. Tuytelaars and L. Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *Int'l J. Computer Vision,* vol. 59, no. 1, pp. 61-85, 2004.

[53] J. Wang and J.-D. Zucker, "Solving the Multiple-Instance Problem: A Lazy Learning Approach," *Proc. 17th Int'l Conf. Machine Learning,* pp. 1119-1125, 2000.

[54] J.Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 9, pp. 947-963, Sept. 2001.

[55] N. Weidmann, E. Frank, and B. Pfahringer, "A Two-Level Learning Method for Generalized Multi-Instance Problems," *Proc. European Conf. Machine Learning,* pp. 468-479, 2003.

[56] C. Yang and T. Lozano-Pérez, "Image Database Retrieval with Multiple-Instance Learning Techniques," *Proc. IEEE Int'l Conf. Data Eng.,* pp. 233-243, 2000.

[57] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," *J. Machine Learning Research,* vol. 5, pp. 1205-1224, 2004.

[58] X. Xu and E. Frank, "Logistic Regression and Boosting for Labeled Bags of Instances," *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining,* pp. 272-281, 2004.

[59] M.-L. Zhang and Z.-H. Zhou, "Improve Multi-Instance Neural Networks through Feature Selection," *Neural Processing Letters,* vol. 19, no. 1, pp. 1-10, 2004.

[60] Q. Zhang, S.A. Goldman, W. Yu, and J. Fritts, "Content-Based Image Retrieval Using Multiple-Instance Learning," *Proc. 19th Int'l Conf. Machine Learning,* pp. 682-689, 2002.

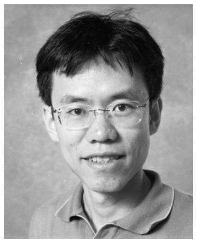
[61] Q. Zhang and S.A. Goldman, "EM-DD: An Improved Multiple-Instance Learning Technique," *Advances in Neural Information Processing Systems 14,* pp. 1073-1080, 2002.

[62] Y. Zhang, H. Zha, C. Chu, and X. Ji, "Towards Inferring Protein Interactions: Challenges and Solutions," *EURASIP J. Applied Signal Processing,* special issue on advanced signal/image processing techniques for bioinformatics, 2006.

[63] Z.-H. Zhou and M.-L. Zhang, "Ensembles of Multi-Instance Learners," *Lecture Notes in Artificial Intelligence,* vol. 2837, pp. 492-502, 2003.

[64] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-Norm Support Vector Machines," *Advances in Neural Information Processing Systems,* vol. 16, 2004.

[65] J.-D. Zucker and Y. Chevaleyre, "Solving Multiple-Instance and Multiple-Part Learning Problems with Decision Trees and Rule Sets, Application to the Mutagenesis Problem," *Lecture Notes in Artificial Intelligence,* vol. 2056, pp. 204-214, 2001.



Yixin Chen (S'99-M'03) received the BS degree from the Department of Automation, Beijing Polytechnic University, China, in 1995, the MS degree in control theory and application from Tsinghua University, China, in 1998, and the MS and PhD degrees in electrical engineering from the University of Wyoming, Laramie, in 1999 and 2001, respectively. In 2003, he received the PhD degree in computer science from The Pennsylvania State University, University Park. From August 2003 to July 2006, he was an assistant professor in the Department of Computer Science, University of New Orleans. Since August 2006, he has been an assistant professor at the Department of Computer and Information Science, The University of Mississippi, University. His research interests include machine learning, data mining, computer vision, bioinformatics, and robotics and control. Dr. Chen is a member of the ACM, the IEEE, the IEEE Computer Society, the IEEE Neural Networks Society, and the IEEE Robotics and Automation Society.

Jinbo Bi received the PhD degree in mathematics from Rensselaer Polytechnic Institute in 2003. She also received the BS and MS degrees, respectively, in applied mathematics and in automatic control, from the Beijing Institute of Technology. Since 2003, she has been a staff scientist at Siemens Medical Solutions at Malvern, Pennsylvania, where she works on machine learning theory and algorithms with application to computer aided cancer and abnormality detection and diagnosis. Her research interests include mathematical programming, statistical learning, machine learning, data mining, and dimensionality reduction. She is a member of the IEEE.

James Z. Wang received the BS degree (summa cum laude) in mathematics and computer science from the University of Minnesota, Twin Cities, in 1994, the MSc degree in mathematics and the MSc degree in computer science from Stanford University, Stanford, California, in 1997, and the PhD degree in medical information sciences from the Stanford University Biomedical Informatics Program and Computer Science Database Group in 2000. Since 2000, he has been the holder of the endowed PNC Technologies Career Development Professorship and a faculty member at the College of Information Sciences and Technology, the Department of Computer Science and Engineering, and the Integrative Biosciences Program at the Pennsylvania State University, University Park. He is a recipient of a US National Science Foundation Career award in support of his research program. He is a senior member of the IEEE and the IEEE Computer Society.