MICCLLR: A Generalized Multiple-Instance Learning Algorithm Using Class Conditional Log Likelihood Ratio

Yasser EL-Manzalawy and Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University
Ames, IA 50011-1040, USA
Email: $\{yasser, honavar\}$ $@cs.iastate.edu$

Abstract

We propose a new generalized multiple-instance learning (MIL) algorithm, MICCLLR (multiple-instance class conditional likelihood ratio), that converts the MI data into a single meta-instance data allowing any propositional classifier to be applied. Experimental results on a wide range of MI data sets show that MICCLLR is competitive with some of the best performing MIL algorithms reported in literature.

1. Introduction

Dietterich et al. [5] introduced the multiple-instance learning (MIL) problem motivated by his work on classifying aromatic molecules according to whether or not they are ”musky”. In this classification task, each molecule can adopt multiple shapes as a consequence of rotation of some internal bonds. Dietterich et al. [5] suggested representing each molecule by multiple conformations (instances) representing possible shapes or conformations that the molecule can assume. The multiple conformations yield a multiset (bag) of instances (where each instance corresponds to a conformation) and the task of the classifier is to assign a class label to such a bag. Dietterich's proposed solution to the MIL problem is based on the standard multiple-instance assumption, that all the instances in a bag, in order for it be labeled negative, must contain no positively labeled instance, and a positive bag must have at least one positive instance. The resulting classification task finds application in drug discovery, identifying Thioredoxin-fold proteins [19], content-based image retrieval (CBIR) [11, 24, 2], and computer aided diagnosis (CAD) [7].

Several approaches to MIL have been investigated in the literature including a MIL variant of the backpropagation algorithm [14], variants of the k-nearest neighbor (k-NN) algorithm [20], the Diverse Density (DD) method [10] and EM-DD [23] which improves on DD by using Expectation Maximization (EM), DD-SVM [4] which trains an SVM in a feature space constructed from a mapping defined by the local maximizers and minimizers of the DD function, and MI logistic regression (MI/LR) [15]. Most of these methods rely on the assumption that a bag is positive if and only if it has at least one positive instance. Alternatively, a number of MIL methods [21, 17, 3] have a generalized view of the MIL problem where all the instances in a bag are assumed to participate in determining the bag label.

Against this background, we introduce MICCLLR, a new generalized MIL algorithm which relies on class conditional likelihood ratio (CCLLR) statistics derived from the MI training data to map each bag into a single meta-instance and trains a support vector machine (SVM) classifier from the meta-instances data. Our experimental results on a broad range of real world and artificial data sets show that MICCLLR has a consistent and comparable performance to the state-of-the-art MI methods.

The rest of this paper is organized as follows: Section 2 summarizes the formulations of the MIL problem and overviews two related MIL methods, TLC [21] and statistical kernel [8], that uses the same idea of mapping each bag into a single instance. Section 3 introduces our method. Experimental results on data sets from two MI classification tasks and on artificially generated data sets is given in Section 4 . Section 5 concludes with a brief summary and discussion.

2 Preliminaries

2.1 Multiple-Instance Learning

In the standard (single-instance) supervised classifier learning scenario, each instance (input to the classifier) is represented by an ordered tuple of attribute values. The instance space $I = D_1 \times D_2 \times ... \times D_n$ where D_i is the domain of the i^{th} attribute. The output of the classifier is a class label drawn from a set C of mutually exclusive classes. A training example is a labeled instance in the form $< X_i, c(X_i) >$ where $X_i \in I$ and $c : I \rightarrow C$ is unknown function that assigns to an instance X_i its corresponding class label $c(X_i)$. For simplicity we consider only the binary classification problem in which $C = \{-1, 1\}$. Given a collection of training examples, $E = \{< X_1, c(X_1) >, < X_2, c(X_2) >, ..., < X_n, c(X_n) >\}$, the goal of the (single-instance) learner is to learn a function c^* that approximates c as well as possible.

In the multiple-instance supervised classifier learning scenario, instead of labeling single instances, the task of the classifier is to label a *bag* of instances. Under standard MIL assumption, a bag is labeled negative if and only if all of its instances are negatively labeled and a bag is labeled positive if at least one of its instances is labeled positive. More precisely, Let B_i denotes the i^{th} bag in a set of bags B. Let $X_{ij} \in I$ denotes the j^{th} instance in the bag B_i and X_{ijk} be the value of the k^{th} feature in the instance X_{ij}. The set of MI training examples, E_{MI}, is a collection of ordered pairs $< B_i, f(B_i) >$ where f is unknown function that assigns to each bag B_i a class label $f(B_i) \in \{-1, 1\}$. Under the standard MIL assumption [5], $f(B_i) = -1$ iff $\forall_{X_{ij} \in B_i} c(X_{ij}) = -1$; and $f(B_i) = 1$ iff $\exists_{X_{ij} \in B_i}$, such that $c(X_{ij}) = 1$. Given E_{MI}, a collection of MI training examples, the goal of the multiple instance learner is to learn a good approximation function of f. It should be noted that the function f is defined in terms of a function $c : I \rightarrow C$. However, learning c from the MI training data is challenging since we have labels only associated with bags and we do not have labels for each instance. In other words, for instances within a negative bag, we can assign a negative label to each instance but for instances in a positive bag, we do not know which of them has a positive label.

2.2 Generalized Multiple-Instance Learning

A generalization of the MIL problem has been considered by Weidmann et al. [21] and Tao et al. [17]. In this setting, all the bag instances contribute the label assigned to the bag and negative bags may contain some positive instances. Instead of a single concept, the generalized MIL problem considered a set of underlying concepts and requires a positive bag to have a certain number of instances in each of them. Weidmann et al. [21] explored three different models for the generalized MIL. In *presence-based MIL*, a positive bag must contain at list one instance in each of the underlying concepts. The *threshold-based MIL* model requires each positive bag to have a minimum number of instances in each concept. Finally, in the *count-based MIL* model, the number of instances in each concept is bounded by a lower and lower threshold. It should be noted that the standard MIL problem is a special case of the presence-based MIL problem when the number of concepts is one. Therefore, a generalized MIL classifier is expected to be able to deal with data sets that confirm to the standard MIL assumption. The two-level MIL classifier (TLC) [21] and methods based on MI kernels [17, 8] offer examples of multiple instance learning under the generalized MIL assumption. Several recent papers [21, 17, 8, 15] have reported good performance of these generalized algorithms on the Musk data sets [5] in which the bags are labeled according to the standard MIL assumption.

2.3 Two level classifier

As the name suggests, the “two level classifier” [21] has two classifiers trained from the data at two different levels of abstraction. The first classifier is trained from the MI data at the instance level by assigning the label of each bag to its instances and assigning a weight to each instance such that bags of different size will end up with the same weight. Specifically, each instance in a bag B_i will be assigned a weight equals $\frac{1}{|B_i|} \times \frac{N}{b}$, where N is the total number of instances in the training data and b is the number of training bags. Weidmann et al. [21] used a pruned decision tree trained form the MI data at the instance level to represent the structure of the instance space. They used the tree to map each training bag into a single meta-instance with a number of attributes equals to the size of the tree. Each attribute in the meta-instance corresponds to a node in the tree and its value is an integer value that counts how many instances in a bag under consideration visited this node during the task of classifying each instance in this bag by the decision tree. Once the training bags have been transformed into a set of meta-instances by the first classifier, the second classifier is trained from the meta-instances data set. In their experiments, Weidman et al. used Logit-boosted decision stumps [6] with 10 boosting iterations as the second level classifier.

2.4 Statistical kernel

Gartner et al. [8] mapped each bag into a single meta-instance using an aggregation function (e.g. mean, median, minimum, maximum, etc.) applied to each instance attribute. The resulting labeled meta-instances data set is then used to train a SVM classifier. Gartner used a kernel k_{stat} defined as $k_{stat}(B_i, B_j) = k(s(B_i), s(B_j))$, where $s(B_i) = \{min_l\, X_{il1}, ..., min_l\, X_{iln}, max_l\, X_{il1}, ..., max_l\, X_{iln}\}$. This simple approach of mapping each bag into a meta-instance has two limitations. First, each bag is mapped independently of any other training bags. Therefore, the transformation process of a bag into a single meta-instance does not make use of the other available training data to improve the mapping. Second, the statistical kernel is not applicable to data sets in binary representation because there is a high chance that two bags with different labels will be mapped into a meta-instance with a value of zero for the first n attributes and a value of one for the remaining attributes.

3 Algorithm

We now proceed to describe MICCLLR, a MIL algorithm that uses CCLLR statistics extracted from the MI training data to map each bag into a single meta-instance. Figure 1 presents the pseudo code for MICCLLR. In step 1, we assign the label of each bag to its instances and associate a weight with each instance such that bags of different sizes will be treated equally. Step 2 estimates the probability of each attribute value given the instance label. Under Naive Bayes assumption, the posterior probability of each attribute is independent from other attributes given the instance label. Therefore, the posterior probability of each attribute can be easily estimated from the training data using standard probability methods based on relative frequencies of each attribute value and class label occurrences observed in the training labeled instances [13]. Step 3 uses the collected statistics to map each bag into a single meta-instance. Let $B_i = \{X_{i1}, \ldots, X_{ik}\}$ be a bag of k instances. Each instance is represented by an ordered tuple of n attribute values. We define a function s that maps B_i into a single meta-instance of n real value attributes as; $s(B_i) = \{s_1, s_2, \ldots, s_n\}$ where each meta-instance attribute is computed using Eq. 1. The MI kernel K is then defined as: $K(B_i, B_j) = k(s(B_i), s(B_j))$ where k is the RBF kernel.

$$s_q = \frac{1}{k} ln \sum_{l=1}^{k} \frac{Pr(X_{ilq} = a_q|c = 1)}{Pr(X_{ilq} = a_q)|c = -1)} \tag{1}$$

Once the MI data has been transformed into a standard supervised learning data, any propositional classifier can be trained from such data. In this work, we used support vector machine (SVM) classifier [18] with an RBF kernel as the propositional classifier. To classify an unlabeled bag B, we first transform it into a single meta-instance using Eq. 1 and then we use the SVM classifier h to classify the meta-instance.

```
Algorithm: MICCLLR
Input: A collection of labeled bags $E_{MI}$
Output: MIL classifier $h$

  1. Use $E_{MI}$ to construct the collection of all instances $E_{AV}$ by labeling each instance with its bag's class label and assign to instances in a bag $B_i$ a weight equal to $\frac{1}{|B_i|} \cdot \frac{N}{M}$ , where $N = \sum_i |B_i|$ and $M$ denotes the number of bags in the training data set
  2. Estimate the posterior probabilities of each attribute, $Pr(a_q|c_j)$, from $E_{AV}$.
  3. convert each bag in $E_{MI}$ to a single meta-instance $\{s_1, s_2, ..., s_n\}$ using Eq. 8.
  4. Train an SVM using a kernel $K$ defined as $K(B_i, B_j) = k(s(B_i), s(B_j))$ to build a classifier $h$ from the set of single meta-instances.
```

Figure 1. An algorithm for building a MICCLR classifier from a collection of training bags E_{MI}. In the experiments reported in this paper, k is a radial basis function (RBF) kernel.

4 Experiments and Results

We implemented MICCLLR and the statistical kernel [8] using the Weka machine learning workbench [22]. For both methods, we trained an SMO classifier with the RBF kernel. We tuned the C and γ and used the default values for the remaining parameters to get the optimal performance of the SMO classifier. The C parameter determines the tradeoff between margin maximization and training error minimization. The γ parameter determines the RBF kernel width. To get the best performance out of MICCLLR and statistical kernel classifiers, we applied a grid search over the range $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$ to optimize C and γ parameters by selecting (C, γ) pair that yields the highest correlation coefficient (as estimated from a 10-fold stratified cross-validation experiment). We then evaluated the classifiers trained using the optimal values for (C, γ) using 10 independent 10-fold cross validation experiments. Each of the 10 independent experiments was set up as a 10-fold stratified cross validation experiment, with a different random seed. The results of each such experiment represent averages over the 10 runs (with each run using 9 subsets of the bags for training the classifier and the remaining set of bags for evaluating the classifier). The reported results correspond to averages and standard deviations over 10 such 10-fold cross-validation experiments. It is important to note that each of the 10 10-fold cross validation experiments used to evaluate the classifier uses a different random partition of the dataset from each other and from the partition used in the 10-fold cross validation experiment used to optimize the C and γ parameters of the classifiers.

4.1 Drug Activity prediction

Prediction of drug activity was the first application of the MIL problem [5]. In this classification task, each molecule can adopt multiple shapes (conformations) as a result of the rotation of some internal bonds. Each molecule is represented as a bag of instances where each instance represents a possible conformation of the molecule. A bag is assigned a positive label if and only if at least one of its instances represents an active conformation i.e., one that interacts with a target molecule. Dietterich et al. [5] introduced the Musk data sets, Musk1 and Musk2, which have been used to evaluate almost every MIL method reported in literature. Musk1 has smaller number of bags and smaller number of instances per bag compared with Musk2.

Table 1. Comparison of the performance (% correct ± std. deviation) of MICCLLR and our implementation of statistical kernel, k_{stat}, with those of other methods on the Musk data sets. All methods had been evaluated using 10-fold cross validation test except IAPR [5] which had been evaluated using leave one out test.

Method	Musk1	Musk2
MICCLLR	89.3 ± 1	88 ± 1.7
k_{stat}	91.3 ± 2	85.5 ± 1.7
TLC [21]	88.7 ± 1.6	83.1 ± 3.23
DD-SVM [4]	85.8	91.3
MILES [3]	86.3	87.7
IAPR [5]	92.4	89.2
DD [10]	88.9	82.5
EM-DD [16]	84.8	84.9
MI-SVM [16]	77.9	84.3
mi-SVM [16]	87.4	83.6
MI-NN [14]	88	82
Multinst [1]	76.7	84
MICA [9]	88.4	90.5
CH-FD [7]	88.8	85.7

Table 1 compares the performance of MICCLLR and our implementation of the statistical kernel with several MIL methods. It should be noted that many of the reported methods has a performance close to the optimal on one data set but does not perform well on the other data set. Although MICCLLR is not the best classifier on Musk1 or Musk2, its performance in the two data sets is relatively well and close the best reported performance. Compared with the TLC, MICCLLR outperformed the TLC classifier on the two data sets. The statistical kernel has slightly better accuracy on Musk1 than MICCLLR but MICCLLR has a lower standard deviation. MICCLLR is significantly better than the statistical kernel on Musk2 data set.

4.2 Content-Based Image Retrieval

In content-based image retrieval (CBIR), the user submits a query image which contains an object of interest and the task of the classifier is to retrieve images that contain the query object from a database of images. For example, the user may submit an image with an elephant and the task is to search a database of images for images containing an elephant. In [11, 24], each image is viewed as a bag of segments. Although there is no one-to-one mapping from objects to image segments, the underlying assumption is that the object of interest (e.g. elephant) is contained in at least one image segment of the target image. In our experiments, we used three CBIR data sets [16]. Each data set corresponds to one of three different categories, namely Elephant, Fox, and Tiger. For each category, the data set has 100 positive and 100 negative example images. In Table 2, we compare our results with EM-DD [23], mi-SVM and MI-SVM [16], MICA [9], and CH-FD [7]. For Elephant and Fox data sets, MICCLLR and statistical kernel have the best reported performance while MI-SVM has the best performance on Tiger data set. The statistical kernel is not performing well on the Tiger data set while MICCLLR has a consistent performance on the three data sets.

4.3 Artificial Data Sets

In this experiment, we evaluate MICCLLR on the three models of the generalized MIL problem introduced by Weidmann et al [21]. Unlike Musk and CBIR data sets, these artificial data sets have only binary attributes. Therefore, we could not test the statistical kernel on these data sets because the min and max aggregation operators mapped all the bags into the same instance, an instance with zero value in the first n attributes and the value of one in the remaining n attributes.

We followed the procedure described in [21] to generate the artificial data sets. In these data sets, instances are drawn from $\{0, 1\}^{(r+i)}$, where r is the number of relevant attributes and i denotes the number of irrelevant ones. A

Table 2. Comparison of the performance (% correct ± std. deviation) of MICCLLR and our implementation of statistical kernel, k_{stat}, with those of other methods on the CBIR data sets [16].

Method	Elephant	Fox	Tiger
MICCLLR	84.4 ± 3.4	63.7 ±1.8	81.5 ±2.1
k_{stat}	83.5 ±0.9	63.3 ±3.1	78.8 ± 1.1
EM-DD [16]	78.3	56.1	72.1
mi-SVM [16]	82.2	58.2	78.9
MI-SVM [16]	81.4	59.4	84
MICA [9]	80.5	58.7	82.6
CH-FD [7]	82.4	60.4	82.2

concept, c_i is represented by a binary string in $\{0,1\}^r$. An instance is a member of a concept c_i if and only if its first r attributes match the binary string string c_i. For each classification task, MICCLLR is trained on five different training sets of 50 positive and 50 negative bags each. Then the average performance, accuracy ± standard deviation of five runs on a test set of 5000 positive and 5000 negative bags is reported. In the following, we report MICCLLR performance on the three generalized MIL problems.

Presence-based MIL. The names of a presence-based data set take the form $c-r-i$ where c is the number of concepts, r and i are the numbers of relevant and irrelevant features respectively. As stated before, the standard MIL problem is a special case of the presence-based MIL where the number of concepts equals one. Table 3 shows that the DD method [10], developed based on the standard MIL assumption, has the best performance on the standard MIL artificial data sets while MICCLLR has the lowest reported performance on these data sets. When the number of concepts is greater than one, Weidmann [21] reported a poor performance of that method and avoided reporting its performance in the remaining data sets. Therefore, the remaining of the results will consider only MI-SVM, TLC, and MICCLLR methods. In Table 4, the performance of each of three methods is comparable to the others except on data sets with zero irrelevant attributes we observe that TLC has a better performance.

Threshold-based MIL. For threshold-based data sets, the names take the form $t_1 t_2 \dots t_n - r - i$, where t_i is the threshold for the i^{th} concept. For example, the threshold-based data set 42-10-5 means that a positive bag must have at least 4 instances in the first concept and 2 instances in the second concept and each instance is composed of 15 features, 10 of them are relevant features. Table 5 shows that both MICCLLR and TLC are competitive to each other and both has a better performance than MI-SVM. However, TLC outperforms MICCLLR on data sets with zero irrelevant attributes.

Table 4. Results for the presence-based MI data sets using two or three underlying concepts.

	MI-SVM	TLC	MICCLLR
2-5-0	80.96 ± 1.9	100 ± 0	81.7 ± 1.5
2-5-5	81.17 ± 1.79	88.38 ± 11.91	79.5 ± 1.8
2-5-10	79.21 ± 1.66	78.64 ± 13.15	83.7 ± 2.3
2-10-0	84.18 ± 0.52	99.01 ± 1.32	84.6 ± 1.4
2-10-5	82 ± 1.53	85.18 ± 10.07	84.9 ± 0.3
2-10-10	80.74 ± 0.79	86.63 ± 8.69	83.7 ± 1.2
3-5-0	82 ± 2.13	100 ± 0	79.9 ± 2.3
3-5-5	82.12 ± 0.98	81.93 ± 2.9	76.9 ± 2.3
3-5-10	81.43 ± 0.96	86.32 ± 6.48	73.7 ± 3.4
3-10-0	84.39 ± 1.25	95.68 ± 3.78	74.9 ± 1.3
3-10-5	84.27 ± 1.44	78.07 ± 0.91	75.2 ± 1.6

Table 5. Results for the threshold-based MI data sets using two or three underlying concepts.

Data	MI-SVM	TLC	MICCLLR
42-5-0	84.35 ± 3.07	100 ± 0	94.6 ± 1
42-5-5	81.54 ± 2.24	95.93 ± 9.1	90.9 ± 3.2
42-5-10	81.59 ± 0.4	84.67 ± 14.31	91.9 ± 3.1
42-10-0	86.28 ± 1.33	99.35 ± 0.45	94.4 ± 1.1
42-10-5	85.36 ± 0.92	88.65 ± 10.12	95.4 ± 0.3
42-10-10	83.93 ± 0.36	84.59 ± 8.08	94.2 ± 1.6
275-5-0	84.75 ± 1.03	97.2 ± 2.78	86.2 ± 7.8
275-5-5	83.9 ± 1.29	90.62 ± 6.57	85.1 ± 2.6
275-5-10	82.73 ± 0.85	86.42 ± 5.39	86.9 ± 2
275-10-0	88.66 ± 1.12	95.44 ± 1.21	89.2 ± 0.5
275-10-5	87.05 ± 0.75	86.92 ± 6.56	87 ± 2.2

Count-based MIL. The count-based data set 42-10-5 means that a positive bag is required to have exactly 4 instances in the first concept and 2 instances in the second concept and each instance has 10 and 5 relevant and irrelevant features respectively. In this setting, a bag with 5 and 2 instances that are members in the first and second concepts respectively is considered negative. Therefore, the count-based MIL seems more challenging than the other two problems. In Table 6, the TLC has a good performance only on data sets with zero irrelevant attributes. On the remaining data sets the performance of the three methods is just few percents above a classifier that randomly assigns labels to the bags.

Table 3. Results for the presence-based MI data sets using only one underlying concept.

Data	DD	MI-SVM	TLC	MICCLLR
1-5-0	100 ± 0	94.35 ± 0.74	100 ± 0	88.7 ± 0.6
1-5-5	100 ± 0	92.26 ± 0.95	100 ± 0	88.4 ± 0.2
1-5-10	100 ± 0	90.74 ± 0.76	100 ± 0	87.9 ± 0.6
1-10-0	99.57 ± 0.59	96.2 ± 0.85	97.46 ± 0.92	88.8 ± 0.5
1-10-5	99.41 ± 0.54	94.67 ± 1.35	97.57 ± 0.87	89.2 ± 0.6
1-10-10	99.8 ± 0.44	91.66 ± 2.3	97.85 ± 0.89	88.6 ± 0.2

Table 6. Results for the count-based MI data sets using two or three underlying concept.

Data	MI-SVM	TLC	MICCLLR
42-5-0	52.78 ± 2	99.55 ± 0.64	52.1 ± 1.1
42-5-5	52.7 ± 1.04	57.89 ± 11.09	51.5 ± 0.7
42-5-10	53.83 ± 1.46	57.63 ± 7.64	51.5 ± 1.9
42-10-0	55.21 ± 1.76	90.89 ± 6.25	52.5 ± 0.6
42-10-5	54.62 ± 0.5	57.8 ± 8.55	51.7 ± 1.2
42-10-10	55.59 ± 2.81	51.05 ± 1.6	51.9 ± 0.5
275-5-0	54.31 ± 2.07	95.15 ± 2.4	54.6 ± 1.9
275-5-5	51.6 ± 0.45	55.2 ± 6.13	53 ± 1.1
275-5-10	52.34 ± 0.5	50.33 ± 0.72	52.1 ± 1.2
275-10-0	54.52 ± 1.54	87.85 ± 4.26	52.9 ± 1.3
275-10-5	54.5 ± 1.81	54.11 ± 4.79	53.1 ± 0.5

5 Summary and Discussion

We have proposed MICCLLR, a generalized MIL algorithm that uses the class conditional log likelihood ratio (CCLLR) to convert each bag into a single meta-instance. We have conducted extensive experiments on a large number of MIL data sets, including both real world and artificial data, in order to demonstrate the applicability of our approach. Our results show that MICCLLR has a competitive performance with a large number of MIL methods. We also compared MICCLLR with two MIL methods, TLC [21] and statistical kernel [8], that use the same approach of mapping each bag into a single instance and showed that MICCLLR has a better performance than TLC on Musk data sets, and on artificial data sets with irrelevant attributes. Compared with the statistical kernel, MICCLLR has a consistent performance on the Musk and CBIR data sets while the statistical kernel is not competitive with MICCLLR on Musk2 and Tiger data sets. This can be justified by the fact that MICCLLR is making use of the available training data to map bags into meta-instances while statistical kernel is applying aggregation operators to each bag without considering other available training data. Both TLC and MICCLLR can be applied to data sets with real-value or nominal attributes while the statistical kernel may not be applicable to data sets with binary attributes.

In the current implementation of MICCLLR, in estimating the relevant probabilities, we assume that instances within a bag are independently identically distributed (iid) given the label assigned to the bag. This assumption is unrealistic and unlikely to hold in practice. The consequences of violating the iid assumption as a result of autocorrelation among instances in probability estimation have been explored and addressed in the context of multi-relational learning in the work of [12]. Because the instances within a bag are likely to be autocorrelated, it would be interesting to explore variants of MIL algorithms (including MICCLLR) that use statistical estimators that correct for such autocorrelation.

References

[1] P. Auer. On learning from multi-instance examples: Empirical evaluation of a theoretical approach, 1997.

[2] J. Bi, Y. Chen, and J. Wang. A sparse support vector machine approach to region-based image categorization. *In Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 1121–1128, 2005.

[3] Y. Chen, J. Bi, and J. Wang. MILES: Multiple-instance learning via embedded instance selection. *IEEE Trans Pattern Anal Mach Intell*, 28(12):1931–1947, 2006.

[4] Y. Chen and J. Wang. Image categorization by learning and reasoning with regions. *The Journal of Machine Learning Research*, 5:913–939, 2004.

[5] R. H. Dietterich, T. G.; Lathrop and T. Lozano-Perez. Solving the multiple-instance problem with axis parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *Annals of Statist*, 28(2):337–407, 2000.

[7] G. Fung, M. Dundar, et al. Multiple instance learning for computer aided diagnosis, Advances in Neural Information Processing Systems (NIPS 2006).

[8] T. Gartner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *Proceedings 19th International Conference on Machine Learning*, pages 179–186, 2002.

[9] O. Mangasarian and E. Wild. Multiple instance classification via successive linear programming.

[10] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, 10, 1998.

[11] O. Maron and A. Ratan. Multiple-instance learning for natural scene classification. In *Proc. of the 5th International Conference on Machine Learning*, pages 341–349.

[12] A. McGovern and D. Jensen. Identifying predictive structures in relational data using multiple instance learning. In *Proc. of the 20th International Conference on Machine Learning*, pages 528–535, 2003.

[13] T. M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

[14] J. Ramon and L. De Raedt. Multi instance neural networks. *Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learning*, 2000.

[15] S. Ray and M. Craven. Supervised versus multiple instance learning: an empirical comparison. In *In Proc. of the 22nd international conference on Machine learning*, 2005.

[16] I. T. Stuart Andrews and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, 15, 2002.

[17] Q. Tao, S. D. Scott, and N. V. Vinodchandran. Svm-based generalized multiple-instance learning via approximate box counting. In *Proc. of the 21st International Conference on Machine Learning (ICML 2004)*, pages 779–806, July 2004.

[18] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[19] C. Wang, S. Scott, J. Zhang, Q. Tao, D. E. Fomenko, and V. N. Gladyshev. A study in modeling low-conservation protein superfamilies. Technical Report TR-UNL-CSE-2004-3, Dept. of Computer Science, University of Nebraska, 2004.

[20] J. Wang and J. D. Zucker. Solving the multiple-instance problem: a lazy learning approach. In *Proceedings 17th International Conference on Machine Learning*, pages 1119–1125, 2000.

[21] N. Weidmann, E. Frank, and B. Pfahringer. A two-level learning method for generalized multi-instance problems. In *Proc. of the European Conference on Machine Learning*, pages 468–479, 2003.

[22] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition edition, 2005.

[23] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. *Neural Information Processing Systems*, 14, 2001.

[24] Q. Zhang, S. A. Goldman, W. Yu, and J. Fritts. Content-based image retrieval using multiple-instance learning. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 682–689, 2002.