An Instance Selection Approach to Multiple Instance Learning

Zhouyu Fu*[1] and Antonio Robles-Kelly[2,3]
[1] Gippsland School of IT, Monash University, Churchill, VIC 3842, Australia
[2] National ICT Australia †, Locked Bag 8001, Canberra, ACT 2601, Australia
[3] RSISE, Australian National University, Canberra, ACT 0200, Australia

Abstract

Multiple-instance Learning (MIL) is a new paradigm of supervised learning that deals with the classification of bags. Each bag is presented as a collection of instances from which features are extracted. In MIL, we have usually confronted with a large instance space for even moderately sized data sets since each bag may contain many instances. Hence it is important to design efficient instance pruning and selection techniques to speed up the learning process without compromising on the performance. In this paper, we address the issue of instance selection in multiple instance learning and propose the IS-MIL, an Instance Selection framework for MIL, to tackle large-scale MIL problems. IS-MIL is based on an alternative optimisation framework by iteratively repeating the steps of instance selection/updating and classifier learning, which is guaranteed to converge. Experimental results demonstrate the utility and efficiency of the proposed approach compared to the alternatives.

1. Introduction

Multiple-instance learning (MIL) [9] is a new paradigm in machine learning that addresses the classification of bags. In MIL, each bag is a collection of instances with features associated to the instances. The aim of MIL is to infer bag level labels based on the assumption that a positive bag contains at least one positive instance, whereas a negative bag contains negative instances only. MIL has a lot of potential in many applications in computer vision and pattern recognition. For instance, in content based image retrieval (CBIR), each image contains many regions, but only a subset of them are of interest. Here an image is a bag and the image regions are instances so that the CBIR problem can be cast in a MIL setting.

Due to false positives in positive bags, traditional supervised classification approaches may not apply to MIL and alternative algorithms are needed to cope with the new scenario. Previous methods for solving the MIL problem can be divided into two major categories. The first category is conformed by algorithms based on generative models such as axis parallel hyper-rectangles [9], Diverse Density (DD) [15], DD with Expectation Maximisation (EM-DD) [18]. The second category is comprised of discriminative learning based algorithms such as mil/MIL Support Vector Machines (SVM) [2], DD-SVM [6] and MILES [5].

One problem that hinders the wide application of MIL is the trade-off between efficiency and performance. Generative methods like DD and EM-DD are quite efficient in learning, but they are based on a strong assumption that all true positive instances form a compact cluster in the feature space. This is, however, not necessarily the case in real applications, as the distributions of positive instances can be arbitrary and most likely multi-modal. Hence, learning a single target distribution to represent positive instances is inadequate in capturing their distributions. Moreover, the optimisation of DD and EM-DD comes with no optimality guarantee. On the other hand, large margin discriminative methods are much more robust and achieve an improved performance, especially DD-SVM and MILES. DD-SVM overcomes the limitation of DD-based generative approaches by learning multiple target distributions for positive instances given by the local extrema of the EM-DD cost function. However, the local extrema are obtained by applying EM-DD optimisation to *all* training instances, which is very time-consuming for large data sets. MILES, in contrast, does not apply instance selection and use all instances in the training set to construct the feature map. This gives rise to a very high-dimensional bag-level feature vector, whose dimensionality is given by the total number of instances. Feature selection is implicitly done by training a one-norm SVM that produces a sparse feature map.

Despite effective, MILES does not scale-up to large data sets due to its high-dimensional feature space, which results in high complexity for both, the feature computation and the

*This work was done while Z. Fu was with the ANU

†National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

SVM optimization. To achieve comparable performance to MILES with much less complexity, explicit instance pruning and selection is necessary. That is, we require a principled way to reduce complexity devoid of clustering or quantisation procedures. This is an important observation since clustering and quantisation do not exploit discriminative information in the instances and may introduce additional noise in the process or information loss. Moreover, it is also important that instance selection, as a preprocessing step, should be done efficiently. However, the current DD-based instance extraction methodologies involve solving an unconstrained optimisation problem for each possible evaluation of the target concept. This is impractical for large-scale data sets.

2. Preliminaries and Algorithm Overview

In this paper, we propose a principled MIL approach for adaptive instance selection and feature learning called IS-MIL (Instance Selection for MIL). The proposed algorithm has three advantages. Firstly and most importantly, it combines instance selection and classifier learning into a unified alternative optimisation framework with convergence guarantee. Secondly, we propose a method for initial instance selection based on modelling the distributions of negative instances efficiently. Note that although the idea of modelling negative examples has found application in image detection [13], it is based on the supervised learning setting and novel in the context of MIL. Thirdly, due to the reduced complexity of the algorithm for the data sets under study, the resulting classifier yields comparable results to those yielded by competing SVM-based MIL alternatives while being much more efficient in training.

To describe the IS-MIL algorithm, we need to introduce some notation. Denote $\mathcal{B}^{\text{tr}} = \{B_1^+, \ldots, B_{m^+}^+, B_1^-, \ldots, B_{m^-}^-\}$ as the set of bags for training, where $B_i^{+(-)}$ denotes the ith bag from the positive(negative) class and $m_{+(-)}$ denotes the number of positive(negative) bags. From now on, for the sake of simplicity, we will omit the $+/-$ sign when there is no need for disambiguation.

A bag B_i contains n_i instances denoted by $\mathbf{x}_{i,j}$ for $j = 1, \ldots, n_i$. Without disambiguation, $\mathbf{x}_{i,j}$ also denotes the feature vector for the instance depending on the context. Different bags can have different number of instances, hence n_i may vary for different i's. To each instance $\mathbf{x}_{i,j}$ also corresponds a label which is not directly observable. Two assumptions are made about instance-level labels. First, all instances in each negative bag are negative. On the other hand, at least one instance in each positive bag is positive. The purpose is, therefore, to predict the label value for the novel testing data $B = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$.

With the above ingredients, we can now describe our framework for MIL. We focus on the binary case here, as any multiclass problem can be converted to several binary problems using the one-against-others strategy.The basic framework of IS-MIL is illustrated in Figure 1, where each block represents an object to operate on. Each arrow indicates an operation defined on the objects. In the training phase, we first perform instance pruning to the training instances. This is achieved by modeling the distributions of negative instances and picking the least negative instance from each positive bag. After doing this, we obtain a set of instance prototypes (IP) representing the true positive instances in positive bags, which we denote $x_i^{*(t)}$. Each of these is chosen from the corresponding positive bag. We then construct a bag-level feature map using bag-to-instance similarities between the training bags and the selected IPs. In the figure, $s(B_i, x_j^{*(t)})$ represents the similarity between bag i and the jth IP. Then, a standard linear SVM classifier is trained on the bag features and, based on the classification results on the training data, we update and reselect the instance prototypes. This step-sequence is interleaved until convergence. The above process is reminiscent of EM-like methods and guaranteed to converge.

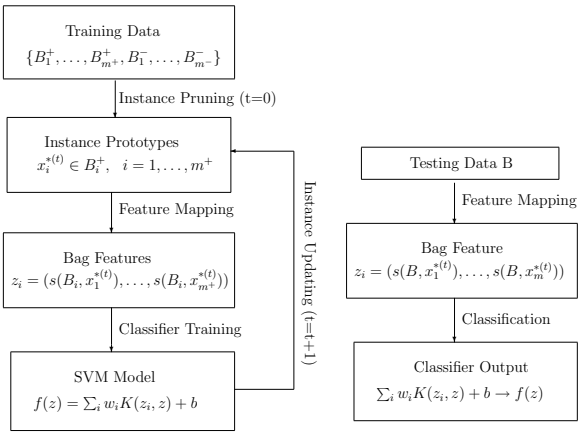


Figure 1. Framework of the proposed IS-MIL algorithm. Left Column: Training Phase; Right Column: Testing Phase

In the testing phase, we commence by extracting the feature vector for the testing bag using the feature mapping defined over the IPs obtained in the training phase. The trained SVM classifier is then applied so as to obtain the classification result.

3. Instance Selection and Feature Learning

3.1. An Alternative Optimisation Framework

In this section, we discuss the details of IS-MIL for joint instance selection and learning. We cast it in an optimisa-

tion setting with the following cost function

$$\min_{\mathbf{w},\phi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} L(\mathbf{z}_i(\phi), y_i; \mathbf{w}) \quad (1)$$

$$L(\mathbf{z}_i, y_i; \mathbf{w}) = \max(0, 1 - y_i \mathbf{w}^T \mathbf{z}_i) \quad (2)$$

$$\mathbf{z}_i(\phi) = [s(B_i, \mathbf{x}_{1,\phi_1}), \ldots, s(B_i, \mathbf{x}_{m^+,\phi_{m^+}})] \quad (3)$$

where $\phi = \{\phi_i \in \{1, \ldots, n_i\} | i = 1, \ldots, m^+\}$ is the set of indices of the IPs selected. Here, $\mathbf{z}_i(\phi)$ is the bag-level feature vector for the ith bag to which the classifier is applied. We choose a single IP from each positive bag. The reason for doing so will become clear shortly. We adopt a similarity based feature representation scheme here which represents each $\mathbf{z}_i(\phi)$ based on the similarity between the bag and all IPs. The cost function in the above equation is that of the linear SVM in the primal formulation, where $L(\mathbf{z}_i(\phi), y_i; \mathbf{w})$ is the hinge loss function used by the SVM classifier. Since it is usually more convenient to deal with the dual SVM formulation, we convert the above primal form into the following dual form,

$$\min_{\alpha,\phi} f(\alpha, \phi) = \frac{1}{2}\sum_{i,j} \alpha_i y_i \mathbf{z}_i(\phi)^T \mathbf{z}_j(\phi) \alpha_j y_j - \sum_i \alpha_i$$

$$s.t. \quad \sum_i \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0 \quad (4)$$

where the α_i's are the Lagrangian multipliers corresponding to the constraints in Equation 1 that arise from relaxing the hinge loss term 2, $\mathbf{z}_j(\phi)$ is the feature mapping given by Equation 3. The classifier weight $\mathbf{w}$ is then given by the KKT condition $\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{z}_i(\phi)$.

There are two sub-problems mixed in the optimisation problem in Equation 4, one over continuous variables α equivalent to the learning of classifier weights $\mathbf{w}$, and the other over discrete variables ϕ equivalent to the selection of instances. This is a difficult optimisation problem overall. We tackle it via alternative optimisation over the two sets of variables. This leads to three main components of IS-MIL - initial instance pruning that corresponds to the initialisation of ϕ_i's, classifier training that corresponds to minimising over α by fixing ϕ, and instance update that corresponds to minimising over ϕ while fixing α.

3.2. Initial Instance Pruning

The key challenge for MIL is how to distinguish between true positive and false positive instances in the positive bags. Simple aggregation rules, as used in MILSVM and MI kernel by averaging all instances in the training bag bias towards negative instances and may potentially lose discriminatory power, especially for sparse positive bags where the majority of the instances in the bag are negative. Also, the true positives may not form a Gaussian distribution. Therefore MIL based on a single IP selected such as DD and EM-DD may also fail to model the distributions of true positives.

The trade-off would be to use a single IP or target from each positive bag. This includes all potential true positive instances in the set of IPs, because by the assumption of MIL, there exists at least one positive instance in positive bags and negative bags do not contain positive instances. We do the pruning by directly modelling the distribution of instances in negative bags, which are supposed to be all negative, and keeping the single "least negative" instance in each positive bag. The notion of least negative is reciprocal to most positive and measured by the likelihood of the instance being negative based on the distributions of negative instances. The set of IPs is formed by those selected instances in positive bags.

Since negative instances can have very general distributions, we use the following Kernel Density Estimator (KDE) [10] to model the distributions of negative instances in the negative bags.

$$p(\mathbf{x}|B^-) = \frac{1}{\sum_i n_i^-} \sum_{i,j} K(||\mathbf{x} - \mathbf{x}_{i,j}^-||) \quad (5)$$

where $x_{i,j}^-$ denotes the jth instance from the ith negative bag. Here, we employ an isotropic Gaussian kernel K and set the band-width parameter σ^2 to the empirical variance of the training data. Notice the above equation defines a normalized probability density function (*PDF*) for the negative population. The initial value of ϕ_i, the index of the IP selected for the ith bag, is then given by

$$\phi_i = \arg \min_{i=1,\ldots,n_i} p(\mathbf{x}_{i,j}|B^-) \quad (6)$$

The major advantage in modeling the negative population in this manner resides in the fact that, due to its large quantity, negative instances usually dominate the joint PDF in MIL settings. Their distributions can then be modeled more reliably than true positives making use of KDE. For computational concerns, we have used a fast implementation of KDE based on the improved fast Gauss transform [17], which has been shown to have linear time complexity with respect to the size of data sets being modeled. The total number of IPs obtained by the initial pruning step is equal to the number of positive bags. This compares quite favorably with the number of prototypes used in MILES, which is given by the number of all training instances. This results in a much lower-dimensional feature space without much loss of discriminatory power, as each positive bag is represented by a corresponding IP. It is also noteworthy that our pruning method selects IPs directly from the training instances. DD-based methods, in contrast, extract novel instances that are not in the training set and are better characterised as instance extraction approaches. Compared to instance extraction approaches, instance selection is much more efficient

as it does not involve solving any optimisation problem and a larger set of IPs are usually obtained with potentially improved discriminative power. Moreover, since our instance pruning method is governed by PDFs, it is more robust to noise corruption and the existence of outliers in the data set.

3.3. Classification

After obtaining the initial set of IPs whose indices are given by ϕ_i's, we can form the bag-level feature for each training bag according to Equation 3. The jth feature element for the ith training example is given by the similarity between bag i and the jth IP as defined below

$$s(B_i, \mathbf{x}_{j,\phi_j}) = \max_{k=1,\ldots,n_i} \exp\left(-\gamma d(\mathbf{x}_{i,k}, \mathbf{x}_{j,\phi_j})\right) \quad (7)$$

where $d(\mathbf{x}, \mathbf{y})$ is the distance between two instances, γ is the band-width parameter. The choice of distance measure is application dependent and the Euclidean distance is used by default. The above defined similarity measure compares the jth IP with the most similar instance in bag i or the closest instance in the bag in the sense of distance. This is a special case of the Hausdorff distance defined over two collection of sets, where the second set is a singleton here. The idea is that the closest instance in each bag to an IP carries the maximum amount of category information, which is consistent with the assumption of MIL. This similarity based formulation allows for more flexibility and robustness in feature mapping. Even if the closeness assumption breaks down for certain prototypes, as long as it is true for the majority of the prototypes, this feature mapping is still informative and quite tolerant to possible inaccuracies introduced in the initial pruning stage. This is somewhat expected since the formulation above is related to majority voting formulations.

For classification, we apply the standard linear SVM in Equation 4 to the current feature vectors $\mathbf{z}_i$'s and obtain the classifier weights α_i's which is used later for instance update. Note that the use of linear SVM is sufficient here for discrimination purpose, since the feature computation defined in Equation 7 is inherently nonlinear. In the single instance setting, our framework is equivalent to that of a nonlinear SVM with Gaussian kernel. One disadvantage with L_2 norm is that it delivers less sparse feature weights as compared to one-norm SVM used in [5] and hence not suitable for feature selection. This might lead to inferior results for very high dimensional features. However, since we already reduce the feature dimension by pruning the instances before training the classifier, it is justified to use L_2 SVMs, which can be trained more efficiently.

3.4. Instance Update

With the SVM classifier for the current feature map at hand, we can substitute it back to the cost function in Equation 4 to validate and update the currently selected IPs. This is equivalent to minimising the function with respect to discrete variable set ϕ using the following coordinate descent scheme,

$$\begin{aligned}
\phi_1^{(t+1)} &= \arg\min_{\phi_1=1}^{n_1} f(\alpha, \{\phi_1, \phi_2^{(t)}, \ldots\}) \quad (8)\\
&\ldots\\
\phi_i^{(t+1)} &= \arg\min_{\phi_i=1}^{n_i} f(\alpha, \{\ldots, \phi_{i-1}^{(t)}, \phi_i, \phi_{i+1}^{(t)}, \ldots\})\\
&\ldots\\
\phi_{m^+}^{(t+1)} &= \arg\min_{\phi_{m^+}=1}^{n_{m^+}} f(\alpha, \{\ldots, \phi_{m^+-1}^{(t)}, \phi_{m^+}\})
\end{aligned}$$

where $\phi_i^{(t)}$ and $\phi_i^{(t+1)}$ correspond to the index values of the old and updated IPs for the ith bag. Since α is unchanged during the process of updating ϕ, only the first term on the RHS of Equation 4 is affected, which is related to the margin of the SVM. We can easily see that each IP update leads to a lower cost function value. The order for which the IPs are updated is not fixed and shuffled randomly across different iterations.

To optimise over each ϕ_k for the sub-problem defined in the above equation, a naive implementation requires an exhaustive search procedure over all possible values and is quite costly. By noticing that change of a single ϕ_k only affects the kth column of feature matrix $\mathbf{Z}$, we can develop an incremental update scheme by recycling the intermediate results from previous iteration. Specifically, we take the difference between the cost with new index value ϕ_k and the old cost in the following

$$\delta(\phi_k, \phi_k^{(t)}) = \sum_{\alpha_i,\alpha_j>0} t_{i,j}(\delta z_{i,k}\delta z_{j,k} + \delta z_{i,k} z_{j,k}^{(t)} + z_{i,k}^{(t)}\delta z_{j,k}) \quad (9)$$

where $t_{i,j} = \alpha_i\alpha_j y_i y_j$, and $\delta z_{i,k} = z_{i,k} - z_{i,k}^{(t)}$ is the difference between the old and new feature map for entry (i,k). $z_{i,k}$ is the new feature map value by using $\mathbf{x}_{k,\phi_k}$ as the new IP, and $z_{i,k}$ is the old value. It is also worth mention that the sum is taken over pairs of α_i and α_j with positive values. This further reduces the number of operations needed for each such test. Instance $\mathbf{x}_{k,\phi_k}$ is selected as the new IP iff $\phi_k = \arg\min_{\phi_k} \delta(\phi_k, \phi_k^{(t)})$ and $\delta(\phi_k, \phi_k^{(t)}) < 0$.

3.5. Iterative Framework and Algorithm

The above two steps of classifier training and instance selection can be repeated iteratively in an EM-like alternative optimisation fashion. Once a classifier is trained, we can update the instance prototypes and recalculate the feature mapping. Then a new classifier can be learned from the updated feature mapping. It is straightforward to see that each iteration decreases the value of $f(\alpha, \phi)$. We can do this making use of the relations below

$$f(\alpha^{(t)}, \phi^{(t)}) \geq f(\alpha^{(t+1)}, \phi^{(t)}) \geq f(\alpha^{(t+1)}, \phi^{(t+1)}) \quad (10)$$

where the first inequality corresponds to the classifier updating step and the second arises from instance updating.

Moreover, since ϕ can only take a finite set of values due to the discrete nature of the ϕ_l variables, this updating process is guaranteed to converge towards $\phi_k^{(t+1)} = \phi_k^{(t)}$ for every k. That is, at convergence, the instance prototypes selected from two adjacent iterations do not change. The whole process is similar in spirit to the EM algorithm [8]. Instance selection/updating is akin to the E-step, where the posterior probabilities (i.e. the values of Equation 8 for different k's) are updated by the current model and the hidden variable with the largest posterior probability is chosen. Classifier training can be regarded as the M-step where the parameters are updated by maximizing the likelihood. The complete framework for IS-MIL with integrated instance selection and classifier learning is outlined in Figure 2.

Input: training bags $(B_1^+, \ldots, B_{m^+}^+, B_1^-, \ldots, B_{m^-}^-)$

- Apply KDE to the negative instances via Equation 5 and select $\phi^{(0)}$ via Equation 6
- Form bag-level features via Equations 3 and 7 and train an SVM classifier (Equation 4) on bag-level features.
- Update ϕ based on the learned SVM classifier via Equations 8 and 9
- Repeat the above two steps until convergence

Output: the set of chosen instance prototypes indexed by ϕ_i's and the SVM classifier $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b$ with $\mathbf{w} = \sum_i y_i \alpha_i \mathbf{z}_i$.

Figure 2. Summary of the proposed iterative framework for instance selection and classifier learning

3.6. Multiclass MIL

For multiclass MIL problems, we adopt the one-against-others strategy by applying binary IS-MIL to c binary sub-problems, where c is the number of classes. Each sub-problem has the same training data but different labels. The purpose of the jth sub-problem is to distinguish class j against all other classes. Thus, c SVM classifiers are obtained with c distinctive feature maps during the training process. A testing bag is fed to each of the trained classifiers using the class-specific feature map and assigned to the class with the largest decision value output by the SVM classifier. The steps of training and testing for each class are the same as the binary case, except for initial instance selection, where KDE is applied in a slightly different manner. For multiclass instance selection, we rather use the class conditional distribution of each class to model the negative distributions for other classes. Denote $p(x|\Omega_k)$ the class conditional distribution of training instances from the kth class obtained from KDE,

$$r(x_{i,j}) = \frac{p(x_{i,j}|\Omega_{y_i})}{\max_{k \neq y_i} p(x_{i,j}|\Omega_k)} \tag{11}$$

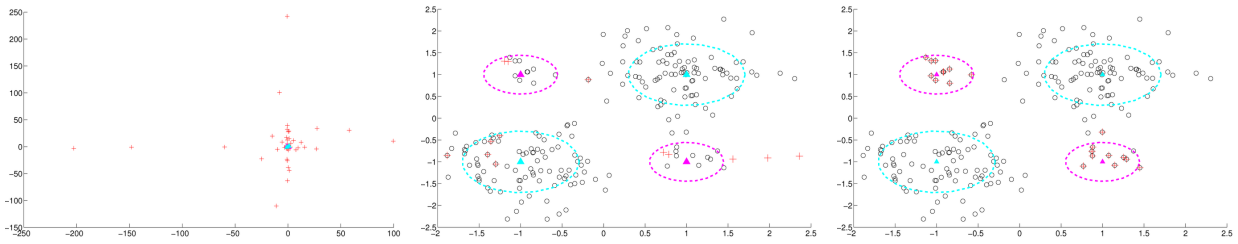
where $y_i \in \{1, \ldots, c\}$ is the label for bag i. The instance with the largest likelihood ratio value in the above equation is selected as the initial prototype for bag i. That is, $\phi_i^{(0)} = \arg\max_j r(x_{i,j})$. The idea here is similar to the binary case, except in the way the distributions for negative instances are modeled. We take the maximum over all other classes for two reasons. Firstly, this is more efficient than running the density estimator over all instances from all other classes. As a result, we apply KDE for each class only once. Secondly, the maximum of class conditional probability is a robust indicator for the discriminability. The instance carries more discriminative information if it has low likelihood for all other classes and a high likelihood for the class it represents.

4. Experimental Results

In this section, we perform several experiments to demonstrate the utility of IS-MIL, the proposed MIL algorithm based on instance selection. First, we test IS-MIL on synthetic data to show its power in instance selection and compare it against EM-DD based instance selection methods [18, 6]. Next, we apply it to two real-world applications, namely region-based image classification and object class categorisation. For region-based image classification, we compare IS-MIL with MILES [5], the state-of-the-art MIL technique. For object class categorisation, which is prohibitively large for alternative methods, we have compared IS-MIL with the state-of-the-art single instance algorithm for categorization [14, 3]. IS-MIL gains a margin of advantage in accuracy compared to MILES for region-based image classification and considerably faster.

We fixed the number of iterations of our algorithm to 5. The parameter γ in Equation 7 was set to the inverse mean between pairs of instances and fixed over the iterations, and the SVM parameter C was chosen via cross validation. For image classification, we have adopted the optimal parameter setting for MILES as reported in [5].The code has been implemented in Matlab with optimization routines written in C++. We used the liblinear package [4] for training linear SVMs and the commercial MOSEK optimization software [1] for solving linear programs resulting in one-norm SVMs.

Synthetic Data In the first example, we illustrate the ability of instance selection for IS-MIL with a synthetic data set shown in Figure 3. We have generated a data-set of 20 positive bags and 20 negative bags. Each bag has 5 instances, and there is only a single positive instance in each



(a) EM-DD (b) EM-DD close-up view (c) IS-MIL

Figure 3. Example of instance selection on synthetic data.

positive bag. The positive instances are randomly drawn from a mixture of Gaussian (MoG) distribution with two Gaussian components, as shown in broken magenta curves in Figure 3(b). The negative instances are drawn from a different MoG distribution with two Gaussian components shown in broken cyan curves. Figure 3(a) shows the IPs extracted by EM-DD. We can see that a large amount of IPs lie further away from the data distribution. This is due to the non-convexity of EM-DD cost function, which are likely to generate outliers with high likelihood value.Figure 3(b) provides a close-up view of the IPs extracted by EM-DD falling in the range of the data set superimposed on the training instances. Figure 3(c) shows the IPs selected by the nonparametric KDE approach of IS-MIL. We can see that IS-MIL has successfully selected the single positive instance from each positive bag as the initial IP whereas the initial IPs selected by EM-DD are not as discriminative as IS-MIL with many miss-hits. Moreover, IS-MIL takes less than 0.1 second for initial instance selection on this data set and is much more efficient than EM-DD, which spends more than 15 seconds and is thus impractical for large-scale applications.

Region-based Image Classification In the next experiment, we tested IS-MIL with MILES for region based image classification on the COREL image data set. The data set contains 2000 images taken from 20 different categories, with 100 images in each category. Each image is segmented into several regions and features are extracted from each region. Hence this is a typical MIL problem with images as bags and region features as instances. Details of segmentation and feature extraction are beyond the scope of this paper and interested readers are referred to [6, 5] for more details along with the introduction of the database. Again, we compared our algorithm with MILES in terms of both accuracy and efficiency. Two tests have been performed. The first test uses only the first 10 categories in the data set for training and testing, while the second test uses the complete data set with all 20 categories. For both tests, we randomly split all images into 50% of training data and the rest for testing. Training and testing were repeated for 5 different random partitions. The results of classification accuracy rates are outlined in Table 1 in terms of mean accuracy and 90% confidence intervals.

Algorithms	1000 Images	2000 Images
IS-MIL	$\mathbf{83.8} : [\mathbf{82.8}, \mathbf{84.8}]$	$\mathbf{69.3} : [\mathbf{68.3}, \mathbf{70.3}]$
MILES	$82.3 : [81.4, 83.2]$	$68.7 : [67.3, 70.1]$

Table 1. Comparison of classification accuracy for the proposed algorithm with MILES for image categorization.

Algorithms	1000 images	2000 images
IS-MIL	9.7 ± 1.9	59.0 ± 5.8
MILES Algorithm	180.3 ± 10.6	960 ± 30.5

Table 2. Comparison of training speed for the proposed algorithm with MILES for image categorization.

Figure 5. IPs selected by IS-MIL for COREL image data.

From the table, we can see that our algorithm is very competitive for image categorization tasks, with a margin of advantage in the accuracy rate compared to MILES. This is especially encouraging, considering the fact that MILES is the state-of-the-art method on the COREL image data set. Also, from the running time comparison results in Table 2, we can see that our algorithm is much more efficient in training than MILES due to effective instance pruning. Notice that the timing results reported in Table 2 are the number of seconds for the training of *all* the classes involved.

In Figure 5, we show some results of IPs selection by IS-MIL. For region-based classification, each IP is a region with discriminant features pertaining to the class it belongs to. Figure 5 shows 3 pairs of images, where the first row shows the original color images, and the bottom row shows the corresponding segmented images where the region corresponding to the selected IP is highlighted in white. It can

(a) COREL (b) Caltech-101

Figure 4. Example images from the two image databases used in the experiment.

be seen that the highlighted regions indeed represent the intrinsic features of the class.

Object class categorisation For our last experiment, we applied our MIL algorithm to object class categorisation on the Caltech 101 image categorisation database [11]. We used a subset of the database that contains 102 object categories including the background category, with 30 images for each category. Some example images from the database are shown in Figure 4(b). Each row shows objects from the same category with huge intra-class variations. The discriminative bags-of-words (BOW) approaches [12, 14] are used as baseline. They start by extracting and quantising local feature descriptors and using a histogram of occurrences of the codewords from the quantised vocabulary as the image level feature. One problem with these approaches is that local features are extracted over the full image and thus inevitably include noisy features in the background. To tackle this problem, Bosch *et.al.* proposed a method for the automatic extraction of regions of interest (ROI) of the target objects in the images and use features within the ROI alone to build frequency histograms [3]. Here we propose an alternative approach based on the IS-MIL algorithm developed in this paper. We used multiple histograms of feature descriptors over different local regions of the image to represent the training images. Each local region represented by the histogram is a candidate ROI. If the image contains the target object, then at least one of the candidates will have high likelihood of being the target. Otherwise, all candidate histograms will have low similarity with the specified target. This is a typical multiclass MIL scenario that can be addressed by our proposed algorithm.

The candidate ROI locations are learned from the annotated ROIs for the training images. This is achieved by clustering on the normalised x-y coordinates of the ground truth. A vocabulary of 50 ROI vectors is formed after clustering, each specifying a candidate location of the ROI. This leads to over $100,000$ instances and a dense feature matrix plus overheads occupying half Gigabytes of the memory space for the MILES approach. On the other hand, IS-MIL can still be applied in an effective manner. A level-1 spatial histogram representation [14] was adopted as the instance level feature vector. For similarity computation, we used the χ-squared distance, which has been empirically shown to be more appropriate for comparing histograms. We tested our approach with the baseline BOW approach and BOW+ROI where features are taken over ROIs [3]. Table 3 lists the classification accuracy averaged over 10 repetitive trials on different random partitioning of training and testing sets using 15 and 20 training images respectively.

Algorithms	15 images	20 images
IS-MIL	$\mathbf{60.5} : [\mathbf{59.3}, \mathbf{61.7}]$	$\mathbf{63.8} : [\mathbf{63.2}, \mathbf{64.4}]$
BOW	$53.5 : [52.6, 54.4]$	$56.3 : [55.5, 57.1]$
BOW+ROI	$60.8 : [59.8, 61.8]$	$64.2 : [62.7, 65.7]$

Table 3. Performance comparison of the algorithms for object class categorisation.

From the results in Table 3, we can see that MIL outperforms the baseline single instance based BOW approach and achieves a comparable accuracy rate as BOW+ROI, which is the state-of-the-art technique for categorisation on the Caltech-101 database using single type of feature. Nevertheless, note that our MIL approach does not perform explicit ROI detection for training, nor does it require any ROI information in the testing stage. The annotated ground truth is only used at the training stage for learning candidate ROI regions and not needed for testing purposes. However, the instance prototypes (IP) selected for the training images do implicitly convey the ROI information. The IPs correspond to instances carry the discriminant information for the class, hence are likely to be located at the ROIs. Examples of the

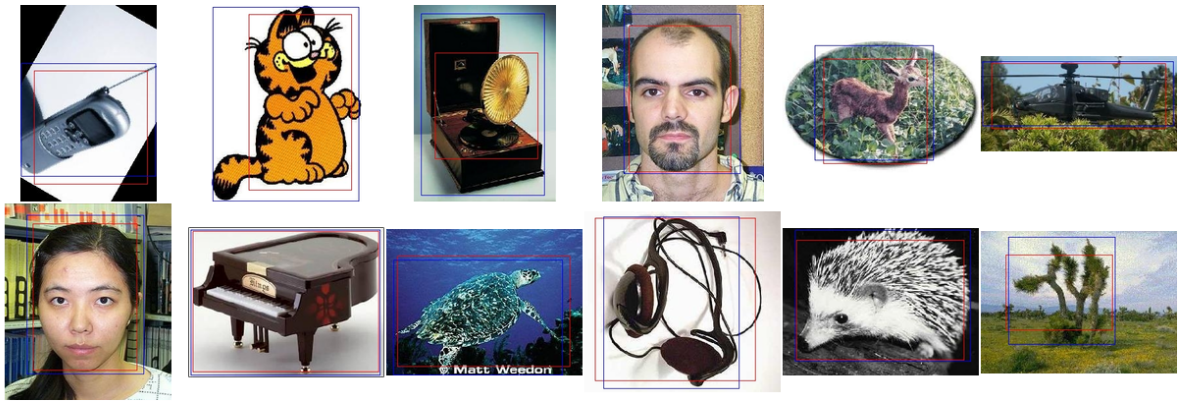


Figure 6. Examples of implicit ROI selection. Blue rectangles indicate the annotated ground truth, red rectangles indicate ROIs automatically selected by the MIL algorithm.

implicitly selected ROIs corresponding to the IPs are shown in Figure 6, where the ROIs chosen by instance pruning is depicted in red rectangles, and the ground truth ROIs are shown by blue rectangles.

Note that the method presented here is quite general. We could use more sophisticated features to get further improvement of the results, such as multi-resolution histograms [12]. We could also use a different SVM algorithm like the one proposed by Crammer and Singer [7] which is known to be better suited to multiclass classification tasks. When combined with MILES, it has been shown to outperform the original algorithm [16]. Our approach can be extended to incorporate the multiclass SVM formulation [7] in a straightforward manner.

5. Conclusions

We proposed IS-MIL, a principled approach for instance selection in MIL. By combining instance selection with classifier learning, we have developed an EM-like alternative optimization framework for iteratively updating the classifier, which is guaranteed to convergence. For the data sets used in our experiments, the proposed approach achieved classification rates comparable to the state-of-the-art MIL classifiers while being much more efficient.

References

[1] Mosek. http://www.mosek.com, 2001. 5

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Neural Info. Proc. Systems*, 2003. 1

[3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Intl. Conf. on Computer Vision*, 2007. 5, 7

[4] K.-W. C. C-J. Hsieh, C.-J. Lin, S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Intl. Conf. on Machine Learning*, 2008. 5

[5] Y. Chen, J. Bi, and J. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006. 1, 4, 5, 6

[6] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5(913-939), 2004. 1, 5, 6

[7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 8

[8] A. Dempster, N. Laird, and D. Rubin. Maximum-likehood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977. 5

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

[10] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973. 3

[11] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004. 7

[12] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007. 7, 8

[13] D. Keren, M. Osadchy, and C. Gotsman. Anti-faces: A novel, fast method for image detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(7):747–761, 2001. 2

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*, 2006. 5, 7

[15] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *NIPS*, 1998. 1

[16] X. Xu and B. Li. Evaluating multi-class multiple-instance learning for image categorization. In *Asian Conf. on Computer Vision*, pages 155–165, 2007. 8

[17] C. Yang, R. Duraiswami, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *Intl. Conf. Computer Vision*, pages 464–471, 2003. 3

[18] Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2002. 1, 5