DICTIONARY-BASED MULTIPLE INSTANCE LEARNING

Ashish Shrivastava[1], *Jaishanker K. Pillai*[2], *Vishal M. Patel*[1], *Rama Chellappa*[1]

[1]Center for Automation Research, University of Maryland, College Park, MD 20742
[2]Google, Mountain View, CA 94043
{ashish, pvishalm, rama}@umiacs.umd.edu, jaypillai@google.com

ABSTRACT

We present a multi-class, multiple instance learning (MIL) algorithm using the dictionary learning framework where the data is given in the form of bags. Each bag contains multiple samples, called instances, out of which at least one belongs to the class of the bag. We propose a noisy-OR model-based optimization framework for learning the dictionaries. Our method can be viewed as a generalized dictionary learning algorithm since it reduces to a novel discriminative dictionary learning framework when there is only one instance in each bag. Various experiments using the popular MIL datasets show that the proposed method performs better than existing methods.

Index Terms— Multiple instance learning, dictionary learning, object recognition.

1. INTRODUCTION

Machine learning has played a significant role in developing robust computer vision algorithms for object detection and classification. Most of these algorithms are supervised learning methods, which assume the availability of labeled training data. Label information often includes the type and location of the object in the image, which are typically provided by a human annotator. The human annotation is expensive and time consuming for large datasets. Furthermore, multiple human annotators often provide inconsistent labels which could affect the performance of subsequent learning algorithm [1]. However, it is relatively easy to obtain weak labeling information either from search queries on Internet or from amateur annotators providing the category but not the location of the object in the image. This necessitates the development of learning algorithms from weakly labeled data.

A popular approach to incorporate partial label information during training is through Multiple Instance Learning (MIL) [2]. Unlike supervised learning algorithms, the MIL framework does not require label information for each training instance, but just for a collection of instances called bags. For two class problems, e.g. object detection, a positive bag contains at least one positive instance while a negative bag contains all the negative instances. However, in multi-class cases, at least one instance in each bag is guaranteed to belong to the class of its bag. One of the first algorithms for MIL, namely Axis-Parallel Rectangle (APR), was proposed in [2]. This method attempts to find an APR by manipulating a hyper rectangle in the instance feature space to maximize the number of instances from different positive bags enclosed by the rectangle. At the same time, it tries to minimize the number of instances from the negative bags within the rectangle. Following this, a general framework, called Diverse Density (DD), was proposed in [3] which measures the co-occurrence of similar instances from different positive bags. An approach based on Expectation-Maximization and DD, called EM-DD, for MIL was proposed in [4]. More recently, an MIL algorithm for randomized trees, named MIForest, was proposed in [5]. An interesting approach, called Multiple Instance Learning via Embedded instance Selection (MILES), was proposed in [6]. This method converts the MIL problem to a standard supervised learning problem that does not impose the assumption relating instance labels to bag labels.

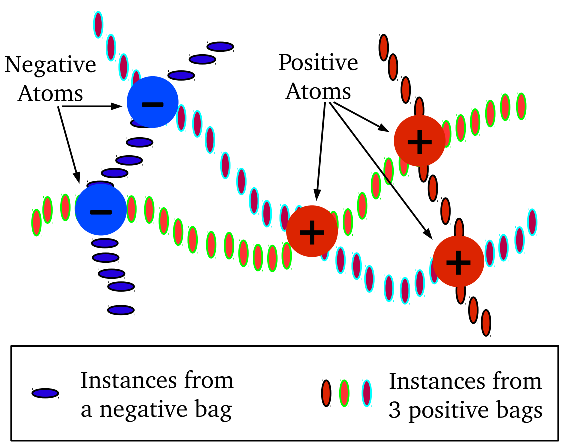


Fig. 1: Motivation for the proposed DD-based MIL framework.

In recent years, sparse representation and dictionary learning-based methods have gained a lot of traction in computer vision and image understanding fields [7], [8], [9]. While the MIL algorithms exist for popular classification methods like Support Vector Machines (SVM) [10] and decision trees [5], such algorithms have not been studied much in the literature using the dictionary learning framework. In this paper, we develop a DD-based dictionary learning framework for MIL where labels are available only for the bags, and not for the individual samples. Figure 1 provides the intuition behind our method. Instances in a bag can be thought of as curves on a data manifold. In this figure, we show instances from one negative bag and 3 positive bags. These curves laid by each bag intersect at different locations. From the problem definition, the negative bag contains only negative class samples, hence the region around the negative curve is very likely to be a negative concept, even if it intersects with positive bags. However, the intersection of positive bags, is likely to belong to the positive concept. Traditional diverse density-based approaches [3] can find only one positive concept that is close to the intersection of positive bags and away from the negative bags. Since one point in feature space can not describe the

This work was partially supported by an ONR grant N00014-12-1-0124.

positive class distribution, these approaches tend to compute different positive concepts with multiple initializations. In the proposed method, the multiple concepts are naturally captured by dictionary atoms leading to a better performance.

Recently, a dictionary-based MIL algorithm was proposed for event detection in [11] that iteratively prunes negative samples from positive bags based on the dictionary learned from the negative bags. This approach may not generalize well for multiclass classification where computing a negative dictionary might be difficult. In contrast, our method seeks to learn a dictionary that captures the intersection of positive bags. Another max-margin dictionary learning algorithm was recently proposed for computing spatial pyramid features from gray images for object and scene recognition in [12] . This dictionary consists of rows containing SVM weight vectors computed using an approach similar to the MI-SVM. This dictionary is pre-multiplied to the dense features and the resulting coefficients are max-pooled. This algorithm takes dense features as its input and is difficult to extend to more general image features.

The rest of the paper is organized as follows. Section 2 presents an overview and formulation of proposed problem. Section 3 describes an optimization approach to solve the proposed formulation. Next, in Section 4 we present approach to classify the bags. Experiments are presented in Section 5 and, finally, the paper is concluded in Section 6.

2. MIL DICTIONARY LEARNING

In this section, we first give an overview of the proposed MIL dictionary learning framework. We then formulate the multi-class MIL dictionary learning objective.

Overview: Assume that we are given N labeled bags $\mathbf{Y}_i$ and their corresponding labels l_i for all $i = 1, \dots N$. Each label can be from one of the C classes, i.e. $l_i \in \{1, \dots, C\}$. A bag $\mathbf{Y}_i$ can have one or more samples, called instances, denoted by $\mathbf{y}_{ij}, j = 1, \dots M_i$, where M_i is the number of instances in the i^{th} bag, i.e., $\mathbf{Y}_i \triangleq [\mathbf{y}_{i1}, \dots, \mathbf{y}_{iM_i}]$. In multi-class MIL setting, if the label of a bag is l_i, then at least one of its instances should belong to class l_i while the others can be from any class. In many computer vision applications a bag corresponds to an image and its instances can be created by varying the scale, position or region of interest. For example, in tracking by detection application [13] multiple overlapping patches can be used as instances and in object recognition application [14, 15, 5] multiple regions of an image can be treated as its instances.

The main focus of this work is to obtain a good representation by learning a dictionary for each class with the given labeled training bags. We represent each instance as a sparse linear combination of the dictionary elements (called atoms) that are representative of the true class. However, for learning the underlying structure in each class, it is important to consider only those instances which belong to the bag's class and disregard other class instances. Existing algorithms for dictionary learning need samples as input and do not work with bags. Hence, in this work we propose a novel noisy-OR model-based dictionary learning algorithm that can learn the representation of each class from bags under the MIL setting.

Our approach relies on learning a dictionary $\mathbf{D}_c$ for each class c. We define the probability of an instance $\mathbf{y}_{ij}$ belonging to the c^{th} class as,

$$p_{ij} = \exp\big(-\|\mathbf{y}_{ij} - \mathbf{D}_c\mathbf{x}_{ij}\|_2^2\big),$$

where $\mathbf{x}_{ij}$ is the sparse coefficient corresponding to $\mathbf{y}_{ij}$ and $\|\mathbf{y}_{ij} - \mathbf{D}_c\mathbf{x}_{ij}\|_2$ is the reconstruction error. Our goal is to learn $\mathbf{D}_c$ for which at least one instance in each bag of class c is well represented (i.e., the probability is high) and the bags of all the other classes (i.e., not c) are poorly represented. This objective can be captured using the noisy-OR model (or the diverse density function [3]) as follows,

$$\tilde{\mathcal{J}} = \prod_{i:l_i=c}\Big(1 - \prod_{j=1}^{M_i}(1-p_{ij})\Big)\prod_{i:l_i\neq c}\Big(\prod_{j=1}^{M_i}(1-p_{ij})\Big). \quad (1)$$

Note that, for $\tilde{\mathcal{J}}$ to be high, at least one instance from each bag of class c should have high p_{ij}, while all the instances in the bags of other classes should have low probability. If we maximize the cost function in (1) with respect to the matrix $\mathbf{D}_c$, we can learn the structure common to all the c^{th} class bags and absent from the bags of the other classes. We refer to this method as dictionary-based multiple instance learning (DMIL).

Formualtion: We denote the data matrix by $\mathbf{Y} = [\mathbf{Y}_1, \dots, \mathbf{Y}_N] \in \mathbb{R}^{d\times M}$. Here, $M = M_1 + \cdots + M_N$ is the total number of instances in all the bags, M_i is the number of instances in the i^{th} bag and d is the dimension of the features for each instance. Let $\mathbf{Y}^c$ be the concatenation of all the c^{th} class bags, i.e, $\mathbf{Y}^c = [\mathbf{Y}_i : l_i = c] \in \mathbb{R}^{d\times M^c}$. Note that the subscript i in $\mathbf{Y}_i$ denotes the bag index and superscript c in $\mathbf{Y}^c$ denotes the matrix of all the bags that belong to class c. Similarly, M^c is the total number of instances in all the c^{th} class bags, i.e. $M^c = \sum_{i:l_i=c} M_i$. For notational clarity, we re-index instances of all the c^{th} class bags and write $\mathbf{Y}^c = [\mathbf{y}_1^c, \dots, \mathbf{y}_{M^c}^c]$, where $\mathbf{y}_i^c$ is the i^{th} instance of the c^{th} class after re-indexing.

Given a dictionary $\mathbf{D}_c$, we can represent an instance $\mathbf{y}$ as a sparse linear combination of the atoms in $\mathbf{D}$. Finding such a sparse representation entails solving the following optimization problem

$$\mathbf{x} = \arg\min_{\mathbf{z}} \|\mathbf{y} - \mathbf{D}_c\mathbf{z}\|_2^2 \ \text{ subject to } \ \|\mathbf{z}\|_0 \le T_0, \quad (2)$$

where, $\|.\|_2$ and $\|.\|_0$ denote the ℓ_2 and ℓ_0-norm, respectively. The problem in (2) can be solved using the orthogonal matching pursuit (OMP) algorithm. Next, we represent the j^{th} instance of the i^{th} bag using the dictionary $\mathbf{D}_c$ and write its probability p_{ij} in terms of the representation error as follows,

$$p_{ij}(\mathbf{D}_c, \mathbf{x}_{ij}) = \exp\big(-\|\mathbf{y}_{ij} - \mathbf{D}_c\mathbf{x}_{ij}\|_2^2\big) \quad (3)$$

where $\mathbf{x}_{ij}$ is the sparse coefficient of $\mathbf{y}_{ij}$ and $(.)^T$ denotes the matrix transposition operation.

In order to learn the dictionary $\mathbf{D}_c = [\mathbf{d}_1, \cdots, \mathbf{d}_{K_c}]$ for class c, we need to optimize the cost in (1) with respect to $\mathbf{D}_c$ and all the sparse coefficients $\mathbf{x}_{ij}$. We denote all the sparse coefficients for the c^{th} class dictionary by the matrix $\mathbf{X}^c = [\mathbf{X}_1, \dots, \mathbf{X}_N] \in \mathbb{R}^{K_c\times N}$ where $\mathbf{X}_i = [\mathbf{x}_{i1}, \dots, \mathbf{x}_{iM_i}] \in \mathbb{R}^{K_c\times M_i}$. In other words, $\mathbf{X}_i$ contains the sparse coefficients for all the instances of the i^{th} bag. Note that, for notational simplicity, we have not used any subscript/superscript c with $\mathbf{X}_i$ and $\mathbf{x}_{ij}$ to indicate that these sparse coefficients are computed using the c^{th} class dictionary. Next, we take the negative log of the cost $\tilde{\mathcal{J}}$ in (1), and introduce a parameter α that controls the influence of the non-c^{th} class bags,

$$\mathcal{J}(\mathbf{D}_c, \mathbf{X}^c) = -\sum_{i:l_i=c}\log\Big(1 - \prod_{j=1}^{M_i}(1-p_{ij})\Big) - \alpha\sum_{i:l_i\neq c}\sum_{j=1}^{M_i}\log(1-p_{ij}). \quad (4)$$

With the above notations, the overall problem of learning the dictionaries can be captured in following optimization problem,

$$\hat{\mathbf{D}}_c, \hat{\mathbf{X}}^c = \arg\min_{\mathbf{D}_c, \mathbf{X}^c} \mathcal{J}(\mathbf{D}_c, \mathbf{X}^c),$$
$$\text{subject to } \|\mathbf{x}_{ij}\|_0 \leq T_0, \|\mathbf{d}_m\|_2 =, 1 \ \ m = 1, \cdots, K_c. \quad (5)$$

3. OPTIMIZATION APPROACH

In this section, we develop an approach to solve (5) by updating one atom, $\mathbf{d}_k$, at a time while keeping the others fixed. We first write instance probabilities p_{ij} as a function of $\mathbf{d}_k$, and then utilize it to update $\mathbf{d}_k$.

Instance Probabilities p_{ij} in terms of $\mathbf{d}_k$: Using (3), we can rewrite p_{ij} as a function of the k^{th} atom $\mathbf{d}_k$, i.e.,

$$\begin{aligned} p_{ij}(\mathbf{d}_k, x_{ijk}) &= \exp\left(-\|\mathbf{y}_{ij} - \mathbf{D}_c\mathbf{x}_{ij}\|_2^2\right) \\ &= \exp\left(-\|\mathbf{r}_{ij} - \mathbf{d}_k x_{ijk}\|_2^2\right), \end{aligned} \quad (6)$$

where x_{ijk} is the k^{th} element of the sparse vector $\mathbf{x}_{ij}$ and

$$\mathbf{r}_{ij} = \mathbf{y}_{ij} - \sum_{\substack{m=1 \\ m\neq k}}^{K_c} \mathbf{d}_m x_{ijm}.$$

Atom Update: We propose an iterative procedure to optimize the k^{th} atom $\mathbf{d}_k$, $k = 1, \ldots, K_c$. Let the cost for optimizing $\mathbf{d}_k$ be denoted by $\mathcal{J}_{\mathbf{d}_k}$. Note that $\mathcal{J}_{\mathbf{d}_k}$, from (4), is a function of p_{ij} and, together with the definition of p_{ij} in (3), can be written as,

$$\begin{aligned} \mathcal{J}_{\mathbf{d}_k}(\mathbf{d}_k) = &- \sum_{i:l_i=c} \log\left(1 - \prod_{j=1}^{M_i}\left(1 - p_{ij}\left(\mathbf{d}_k, x_{ijk}\right)\right)\right) \\ &- \alpha \sum_{i:l_i\neq c}\sum_{j=1}^{M_i} \log\left(1 - p_{ij}(\mathbf{d}_k, x_{ijk})\right). \end{aligned} \quad (7)$$

Hence, the optimization problem in (5) can be reformulated for the k^{th} atom as,

$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \mathcal{J}_{\mathbf{d}_k}(\mathbf{d}_k), \ \text{ subject to } \ \|\mathbf{d}_k\|_2 = 1. \quad (8)$$

Optimization for $\mathbf{d}_k$ in (8) can be viewed as minimizing the negative log likelihood and, similar to [16], is solved using the gradient descent algorithm. In order to satisfy the unit norm constraint, we re-normalize the solution at each step of the gradient descent and line search. Different steps of the optimization for $\mathbf{D}_c$ are summarized in Algorithm 1.

Connection to the Traditional Dictionary Learning: It is interesting to note that first part of our cost $\mathcal{J}$ in (4) is identical to the traditional dictionary learning cost [17] when there is only one instance in each bag. Let this first part of the cost be denoted by $\mathcal{J}_1$. By setting $M_i = 1, \forall i$ it becomes,

$$\begin{aligned} \mathcal{J}_1 &= -\sum_{i:l_i=c} \log\left(1 - \prod_{j=1}^{M_i}(1 - p_{ij})\right) = -\sum_{i:l_i=c} \log p_{i1} \\ &= \sum_{i:l_i=c} \|\mathbf{y}_{i1} - \mathbf{D}_c\mathbf{x}_{i1}\|_2^2. \end{aligned} \quad (9)$$

Hence, in the case of one instance per bag, our problem formulation can also be viewed as a discriminative dictionary learning approach

Algorithm 1: Algorithm for Learning c^{th} Class Dictionary $\mathbf{D}_c$

Input: Bags $\mathbf{Y}_i$, Labels $l_i, \forall i = 1, \ldots N$, Parameters $\alpha, T_0, maxItr$.
Output: $\mathbf{D}_c$.
for $itr = 1, \ldots, maxItr$ **do**
  for $k = 1, \ldots, K_c$ **do**
    Update $\mathbf{d}_k$ by solving (8) with the gradient descent method.
  end
  Update the coefficient matrix $\mathbf{X}^c$ by solving (2) for each instance one at a time.
end
return $\mathbf{D}_c$.

where the first part $\mathcal{J}_1$ ensures that the instances are well represented by the dictionary of the corresponding class, and the second part of the cost $\mathcal{J}$ in (4) ensures that the samples of the non-c^{th} classes are not represented well by the dictionary $\mathbf{D}_c$.

4. CLASSIFICATION

Let $\mathbf{D}$ denote the combined dictionary defined as diag$(\mathbf{D}_1, \cdots, \mathbf{D}_C)$. We compute the sparse coefficients of all the training instances on the combined dictionary by solving the following sparse coding problem

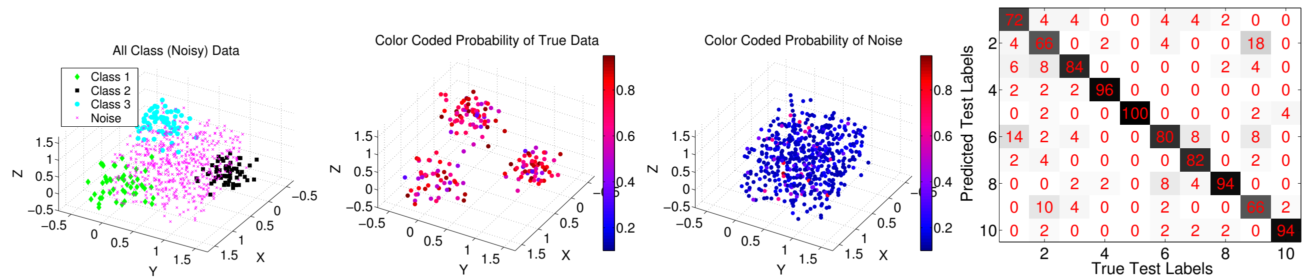
$$\mathbf{x}_{ij} = \arg\min_{\mathbf{z}} \|\mathbf{y}_{ij} - \mathbf{D}\mathbf{z}\|_2^2 \ \text{ subject to, } \ \|\mathbf{z}\|_0 \leq T.$$

We then compute the probability p_{ij} of this instance by (6) after replacing $\mathbf{D}_c$ by $\mathbf{D}$. The sparse representation of the training bags $\mathbf{Y}_i$ is obtained as the weighted combination of the sparse coefficients of its instances. For example, the sparse representation of the i^{th} training bag, denote by $\mathbf{x}_i$ is computed as, $\mathbf{x}_i = \sum_{j=1}^{M_i} p_{ij}\mathbf{x}_{ij}$. Once we obtain the sparse codes for the training bags, any classification algorithm can be used to classify the samples. In our experiments, we utilize an SVM for classification.

5. EXPERIMENTAL RESULTS

In this section we first analyze our algorithm using a synthetic dataset to gain additional insights. We then evaluate our method on popular datasets for MIL like the Tiger, Fox, Elephant [10], Musk [2] and the Corel dataset [6].

Synthetic Experiment: We analyze our algorithm using a three class synthetic dataset. We create 50 bags for each class by first drawing one sample per bag from three different normal distributions (one for each class) with means $[1, 0, 0]^T$, $[0, 1, 0]^T$ and $[0, 0, 1]^T$ and covariance matrix $\begin{bmatrix} 0.051 & 0.01 & 0.01 \\ 0.01 & 0.051 & 0.01 \\ 0.01 & 0.01 & 0.051 \end{bmatrix}$. Then, in each bag, we add 3 more samples from uniformly distributed noise as shown in Figure 2(a). After learning the dictionary for each class with the noisy bags, we project all the instances on the dictionaries of their respective classes and compute the probability for each instance using (3). The color coded probabilities are displayed in Figure 2(b)-(c). As can be observed from this figure, the instances from the normal distributions have higher probabilities (depicted by red color) and instances from the noise distributions have lower probabilities (blue color). This clearly demonstrates that



(a) (b) (c) (d)

Fig. 2: Demonstrating the probabilities of instances after projecting them onto their respective dictionaries: (a) Original noisy bags. Each bag contains one instance from a normal distribution of its class and 3 instances from the uniformly distributed noise. (b) Color coded probabilities of instances from three normal distributions, (c) Color coded probabilities of the uniformly distributed noise. Note that the probabilities of true instances are much higher than that of the noise. (d) Confusion matrix for one of the splits of Corel-1000 image dataset.

the proposed method learns the true representations of each class and reconstructs them well, despite the presence of multiple noise samples in each bag,. Furthermore, the method does not learn the structure in the noise samples and hence gives high reconstruction errors for them.

Intuitive Reasoning: Atoms are learned by the proposed method to capture the common structure across all the bags of a class, at the same time suppressing the structure of other classes. Thus, the linear combination of these atoms will reconstruct the true samples of the class while suppressing the background or noise. Since the classifier is learned over the reconstructed signal, it can discriminate the classes well despite the presence of noisy samples in the training bags.

MIL Benchmark Datasets: Next, we evaluate the proposed approach on the benchmark MIL datasets namely Tiger, Elephant and Fox introduced in [10], and the Musk1 and Musk2 proposed by [2]. Each of the Tiger, Elephant, and Fox datasets have 100 positive and 100 negative bags. A positive bag corresponds to the true image of an animal and negative bags are randomly drawn from the pool of other animals. The instances in each bag are created by segmenting the images, and color, texture, and shape features are used as described in [10]. The Musk1 and Musk2 datasets are publicly available datasets that were introduced in drug activity problem proposed by Dietterich *et al.* [2]. We use the same features and experimental set up as used [10] and compare our results in Table 1. In these experiments, we use 50 atoms per class. The sparsity $T_0 = 10$ and $\alpha = 0.1$ were used for dictionary learning for all the datasets. These two parameters were found using 5-fold cross-validation. We believe that the main reason why our method performs better is that we learn dictionary in such a way that the learned atoms can represent well the commonalities among the bags of the same class while they result in high reconstruction error for the non-common structure. By translating these reconstruction error into probabilities, we are able to reduce the effect of the background of each image while computing the bag features (sparse coefficients).

Algorithms	Elephant	Fox	Tiger	Musk1	Musk2
mi-SVM [10]	82	58	79	87	84
MI-SVM [10]	81	59	84	78	84
MILES [6]	81	62	80	88	83
SIL-SVM	85	53	77	88	87
AW-SVM [18]	82	64	83	86	84
AL-SVM [18]	79	63	78	86	83
EM-DD [4]	78	56	72	85	85
MILBoost-NOR [19]	73	58	56	71	61
MIForests [5]	84	64	82	85	82
DMIL	**87**	**68**	**89**	**92**	**91**

Table 1: Average accuracy of five random splits on the benchmark datasets.

Corel Dataset: The Corel dataset consists of 20 object categories with 100 images per category. These images are taken from CD-ROMs published by the COREL Corporation. Each image is segmented into regions and each region is then called an instance [6]. The regions of an image can greatly vary depending on its complexity. We use the same instance features as [6] and report our result in Table 2. Here, we perform two categorization tasks: first on 10 object categories (corel-1000) and then on all the 20 object categories (corel-2000). For corel-1000 task, we analyze the class accuracy for each category using the confusion matrix in Figure 2(d). Each column in the confusion matrix corresponds to the predicted accuracy of the test samples. As we can see from the figure, class 2 ('Beach') is confused mostly with class 9 ('Mountains and glaciers') which is possibly due to their similar appearances. In both tasks, we used 50 atoms for dictionary learning, sparsity $T_0 = 5$, and $\alpha = 0.1$. As before, T_0 and α were selected by 5-fold cross-validation.

Algorithms	1000-Image Dataset	2000-Image Dataset 2
MILES [6]	82.6 : [81.4, 83.7]	68.7 : [67.3, 70.1]
MI-SVM [10]	74.7 : [74.1, 75.3]	54.6 : [53.1, 56.1]
DD-SVM [14]	81.5 : [78.5, 84.5]	67.5 : [66.1, 68.9]
k-means-SVM [20]	69.8 : [67.9, 71.7]	52.3 : [51.6, 52.9]
DMIL	**84.1** : [82.8, 85.4]	**70.2** : [68.3, 72.1]

Table 2: Average accuracy along with the 95 percent confidence interval over five random test sets of the Corel Dataset.

6. CONCLUSION

We proposed a general dictionary learning method using multiple instance learning framework. It was shown that a special case of our method reduces to a novel discriminative dictionary learning formulation. An efficient algorithm was proposed for updating atoms of the dictionary.

7. REFERENCES

[1] T. Leung, Y. Song, and J. Zhang, "Handling label noise in video classification via multiple instance learning," in *International Conference on Computer Vision*, 2011.

[2] T. G. Dietterich and R. H. Lathrop, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, 1997.

[3] O. Maron and T. Pérez, "A Framework for Multiple-Instance Learning," in *Neural Information Processing Systems*, 1998.

[4] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Neural Information Processing Systems*, 2001.

[5] C. Leistner, A. Safari, and H. Bischof, "MIForests: Multiple-instance learning with randomized trees," in *European Conference on Computer Vision*, 2010.

[6] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

[7] J. Wright, Yi Ma, J. Mairal, G. Sapiro, T.S. Huang, and Shuicheng Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[8] V. M. Patel and R. Chellappa, "Sparse representations, compressive sensing and dictionaries for pattern recognition," in *Asian Conference on Pattern Recognition*, 2011.

[9] V. M. Patel and R. Chellappa, *Sparse representations and compressive sensing for imaging and vision*, SpringerBriefs, 2013.

[10] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Neural Information Processing Systems*, 2003.

[11] J. Huo, Y. Gao, W. Yang, and H. Yin, "Abnormal event detection via multi-instance dictionary learning," in *Intelligent Data Engineering and Automated Learning - IDEAL*, 2012.

[12] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *ICML*, 2013.

[13] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009.

[14] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *Journal of Machine Learning Research*, vol. 5, pp. 913–939, 2004.

[15] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349–361, 2001.

[16] O. Maron, *Learning from Ambiguity*, Ph.D. thesis, Massachusetts Institute of Technology, 1998.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K -SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[18] P. V. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning," in *JMLR Workshop and Conference Proceedings Volume 2: AISTATS 2007*, 2007, pp. 123–130.

[19] P. A. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Neural Information Processing Systems*, 2005.

[20] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.