Multiple Kernel-based Dictionary Learning for Weakly Supervised Classification

Ashish Shrivastava

Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

Jaishanker K. Pillai

Google, Mountain View, CA 94043

Vishal M. Patel

Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742

Abstract

In this paper, we develop a multiple instance learning (MIL) algorithm using the dictionary learning framework where the labels are given in the form of positive and negative bags, with each bag containing multiple samples. A positive bag is guaranteed to have only one positive class sample while all the samples in a negative bag belong to the negative class. Given positive and negative bags of data, our method learns appropriate feature space to select positive samples from the positive bags as well as optimal dictionaries to represent data in these bags. We apply this method for digit recognition, action recognition, and gender recognition tasks and demonstrate that the proposed method is robust and can perform significantly better than many

Email addresses: ashish@umiacs.umd.edu (Ashish Shrivastava), jaypillai@google.com (Jaishanker K. Pillai), pvishalm@umiacs.umd.edu (Vishal M. Patel)

competitive two class MIL classification algorithms.
Keywords: Dictionary learning, multiple Instance learning, multiple kernel learning.

---

1. Introduction

Acquiring good quality labeled training data is one of the critical steps in building an object recognition system. While human annotation is the popular choice for obtaining labeled data for training, it is expensive and time consuming. However, it is relatively easy to obtain weakly labeled data in the Internet. Such data can be obtained through web search queries, captions, subtitles of movies and from amateur raters without full knowledge about the object categories. This has lead to the development of algorithms for weakly supervised object classification.

A popular machine learning paradigm to handle weakly supervised data is the Multiple Instance Learning (MIL) [7]. In MIL paradigm, examples are not individually labeled but grouped into sets or bags which either contain at least one positive examples or only negative examples. Various MIL algorithms have been proposed in the literature for classification [17], [25], [2]. The MIL algorithms have been used to handle label errors, by collecting multiple samples with possible label errors into positive bags. Effect of alignment errors in training data can be reduced by forming bags with multiple shifted templates. The MIL-based algorithms have also been developed for robust tracking of objects [3].

In recent years, the field of sparse representation and dictionary learning has undergone rapid development, both in theory and in algorithms. It

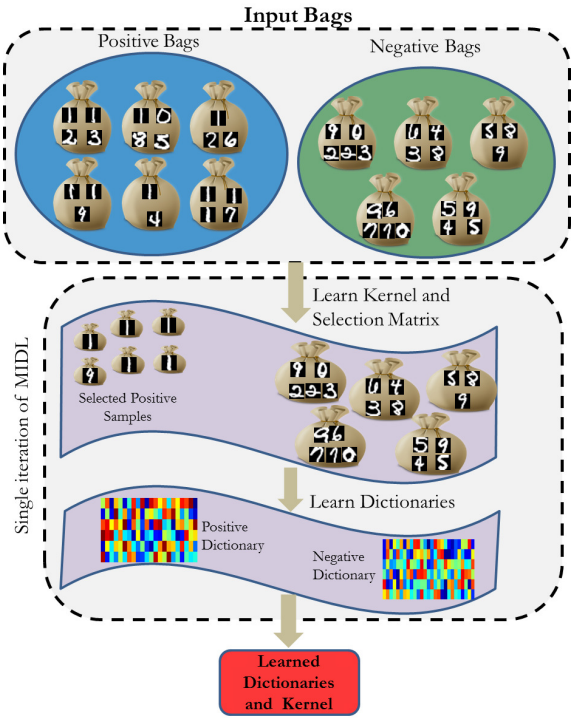


Figure 1: Overview of our method. Given positive and negative bags of data, our method learns appropriate feature space to select positive samples from the positive bags as well as optimal dictionaries to represent data in these bags.

has also been successfully applied to numerous image understanding applications. This is partly due to the fact that signals or images of interest, though high dimensional, can often be coded using few representative atoms in some dictionary. These dictionaries can be either analytic or they can be learned directly from the data. Often, learning a dictionary directly from data usually leads to improved results in many practical applications such as classification and restoration [27]. This has motivated researchers to develop

robust dictionary learning algorithms for various learning scenarios ranging from fully supervised [15], [10], [21], [14], [16], to weakly supervised [23], to unsupervised [24], [20], [5], [8]. Note that this work is fundamentally different than the work of Shrivastava *et al.* [23]. Input to the method of Shrivastava *et al.* is labeled or unlabeled samples and their method does not handle bags. In other words, their method works under semi-supervised setting and not under the MIL setting. Furthermore, Shrivastava *et al.* use a predefined kernel, while the proposed method learns an optimal one based on the Multiple Kernel Learning (MKL) method [9].

While the MIL algorithms exist for popular classification methods like Support Vector Machines (SVM), logistic regression and boosting, such algorithms have not been studied thoroughly in the literature using the dictionary learning framework. Recently, Huo et al. [12] explore dictionary based MIL method for detecting abnormal events in videos by predicting the labels of the instances in the positive bag. Similarly, Wang et al. [26] learn a multi-class classification matrix for object representation. In this paper, we develop an MIL algorithm using the non-linear dictionary learning framework by projecting the data into a feature space. We formulate the multiple instance learning problem as a kernel learning problem and iteratively learn the dictionary in the embedded space of the learned kernel. Multiple kernel learning essentially combines multiple kernels instead of using a single predefined kernel [9]. Different kernels correspond to different notions of similarity between two data samples. In particular, in a high dimensional feature space, it is not optimal to choose one kernel for all the datasets. In the case of MIL, the kernel is learned in a discriminative manner, ensuring that the

negative samples have high reconstruction error on the positive dictionary in the embedded space. This in turn reduces the effect of negative samples in the positive bag on the learned positive dictionary. A block diagram of the proposed algorithm is given in Fig. 1.

The key contributions of our work are:

1. We develop a multiple instance dictionary learning framework to handle weakly supervised data.
2. We also demonstrate how kernel learning can be incorporated into the dictionary learning framework so that data from negative and positive bags are well represented at the same time positive and negative classes are separated in the feature space.
3. We propose a novel classification procedure based on the proposed multiple instance dictionary framework.
4. We demonstrate the effectiveness our approach on three publicly available image classification datasets.

1.1. Organization of the paper

This paper is organized as follows. Section 2 defines and formulates the multiple instance dictionary learning problem. Details of the optimization problem are presented in Section 3. A classification procedure using our proposed dictionary learning method is presented in Section 4. Experimental results are presented in Section 5 and Section 6 concludes the paper with a brief summary and discussion.

2. Problem Formulation

Given a set of training samples, $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{d \times N}$, one can learn a dictionary $\mathbf{D} \in \mathbb{R}^{d \times K}$ with K atoms, that leads to the best representation for each member in this set, under strict sparsity constraints by solving the following optimization problem

$$\arg\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ subject to } \forall i \ \|\mathbf{x}_i\|_0 \leq T_0, \tag{1}$$

where $\mathbf{x}_i$ represents the i^{th} column of coefficient matrix $\mathbf{X} \in \mathbb{R}^{K \times N}$ and T_0 is the sparsity parameter. Here, the Frobenius norm is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{ij}^2}$ and the norm $\|\mathbf{x}\|_0$ counts the number of non-zero elements in $\mathbf{x}$. Various algorithms have been proposed in the literature that can solve the above optimization problem [1], [27].

Using the kernel trick, one can also make the dictionary learning model (1) non-linear [19]. Let $\mathbf{\Phi} : \mathbb{R}^d \rightarrow G$ be a non-linear mapping from a d dimensional space into a dot product space G. A non-linear dictionary can be trained in the feature space G by solving the following optimization problem

$$\arg\min_{\mathbf{A},\mathbf{X}} \|\mathbf{\Phi}(\mathbf{Y}) - \mathbf{\Phi}(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2 \text{ s. t. } \forall i \ \|\mathbf{x}_i\|_0 \leq T_0, \tag{2}$$

where $\mathbf{\Phi}(\mathbf{Y}) = [\mathbf{\Phi}(\mathbf{y}_1), \cdots, \mathbf{\Phi}(\mathbf{y}_N)]$. Since the dictionary lies in the linear span of the samples $\mathbf{\Phi}(\mathbf{Y})$, in (2) we have used the following model for the dictionary in the feature space, $\tilde{\mathbf{D}} = \mathbf{\Phi}(\mathbf{Y})\mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{N \times K}$ is a matrix with K atoms [19]. This model provides adaptivity via modification of the matrix $\mathbf{A}$. After some algebraic manipulations, the cost function in (2) can be rewritten as,

$$\|\mathbf{\Phi}(\mathbf{Y}) - \mathbf{\Phi}(\mathbf{Y})\mathbf{A}\mathbf{X}\|_F^2 = \mathbf{tr}((\mathbf{I} - \mathbf{A}\mathbf{X})^T \boldsymbol{\mathcal{K}}(\mathbf{Y}, \mathbf{Y})(\mathbf{I} - \mathbf{A}\mathbf{X})),$$

where $\boldsymbol{\mathcal{K}}(\mathbf{Y},\mathbf{Y})$ is a kernel matrix whose elements are computed from $\kappa(i,j) = \boldsymbol{\Phi}(\mathbf{y}_i)^T\boldsymbol{\Phi}(\mathbf{y}_j)$. It is apparent that the objective function is feasible since it only involves a matrix of finite dimension $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times N}$, instead of dealing with a possibly infinite dimensional dictionary.

An important property of this formulation is that the computation of $\boldsymbol{\mathcal{K}}$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\boldsymbol{\Phi}$. Some commonly used kernels include polynomial kernels $\kappa(\mathbf{x},\mathbf{y}) = \langle(\mathbf{x},\mathbf{y}\rangle + c)^d$ and Gaussian kernels $\kappa(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c}\right)$, where c and d are the parameters.

In multiple instance learning setting, we have labeled bags instead of samples for training. Each bag is labeled either $+1$ or -1, called a positive or a negative bag, respectively. A negative bag will have only negative samples. A positive bag is guaranteed to have at least one positive sample, while the remaining ones can be either positive or negative. We denote the b^{th} positive bag by a matrix $\mathbf{Y}_b^p \triangleq [\mathbf{y}_{b,1}^p,\ldots,\mathbf{y}_{b,m_b}^p] \in \mathbb{R}^{d\times m_b}$, whose columns are the m_b positive samples. Here, d is the dimension of data sample. Similarly, let $\mathbf{Y}_b^n \triangleq [\mathbf{y}_{b,1}^n,\ldots,\mathbf{y}_{b,n_b}^n] \in \mathbb{R}^{d\times n_b}$ be the b^{th} negative bag containing the n_b negative samples. We denote concatenation of all positive bags with $\mathbf{Y}_p$ and that of negative bags with $\mathbf{Y}_n$, i.e. $\mathbf{Y}_p \triangleq [\mathbf{Y}_1^p,\ldots,\mathbf{Y}_{N_p}^p] = [\mathbf{y}_1^p,\ldots,\mathbf{y}_M^p] \in \mathbb{R}^{d\times M}$ and $\mathbf{Y}_n \triangleq [\mathbf{Y}_1^n,\ldots,\mathbf{Y}_{N_n}^n] = [\mathbf{y}_1^n,\ldots,\mathbf{y}_N^n] \in \mathbb{R}^{d\times N}$, where $M \triangleq \sum_{b=1}^{N_p} m_b$ is the total number of positive samples in all the positive bags and $N \triangleq \sum_{b=1}^{N_n} n_b$ is the total number of negative samples in all the negative bags. There are N_p positive bags and N_n negative bags in total. The b^{th} positive bag contains m_b samples, while the b^{th} negative bag has n_b samples. Given $\mathbf{Y}_n$ and $\mathbf{Y}_p$,

the objective is to learn a non-linear dictionary-based model that can classify a novel test sample to a positive or a negative class.

We denote the negative dictionary, in feature space, as $\tilde{\mathbf{D}}_n = \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_n$ and positive dictionary as $\tilde{\mathbf{D}}_p = \mathbf{\Phi}(\mathbf{Y}_p)\mathbf{A}_p$, where $\mathbf{A}_n \in \mathbb{R}^{N \times K_n}$ and $\mathbf{A}_p \in \mathbb{R}^{M \times K_p}$ are matrices with K_n and K_p number of atoms, respectively. Since, the dictionaries are learned by adapting $\mathbf{A}_p$ and $\mathbf{A}_n$ we will henceforth refer to $\mathbf{A}_p$ as the positive dictionary and $\mathbf{A}_n$ as the negative dictionary. Let $\mathbf{X}_n \triangleq [\mathbf{x}_1^n, \ldots, \mathbf{x}_N^n] \in \mathbb{R}^{K_n \times N}$ be the coefficient matrix for negative samples using negative dictionary where $\mathbf{x}_i^n$ is the coefficient vector for i^{th} negative sample. Likewise, let $\mathbf{X}_p \triangleq [\mathbf{x}_1^p, \ldots, \mathbf{x}_M^p] \in \mathbb{R}^{K_p \times d}$ denote the coefficient matrices for positive samples using the positive dictionary.

Equipped with the above notations, in what follows we formulate the costs to be optimized for learning the dictionaries in the features space. Since we have the labels for all the negative samples from the negative bags, we seek a dictionary such that the following reconstruction error, subject to a sparsity constraint on $\mathbf{x}_i^n$, is minimized,

$$\mathcal{R}_n(\mathbf{A}_n, \mathbf{\Phi}(.), \mathbf{X}_n) = \|\mathbf{\Phi}(\mathbf{Y}_n) - \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\|_F^2. \tag{3}$$

In the positive bags, only one sample is guaranteed to be positive, hence, we use exactly one sample per positive bag. Since we do not know which sample is true positive, we also need to learn a selection matrix $\mathbf{\Omega} \in \mathbb{R}^{M \times N_p}$ that selects a true positive sample from each positive bag. This can be done by

defining the following matrix,

$$\mathbf{\Omega}(i,j) = \begin{cases} 1, & \text{if } \mathbf{y}_i \text{ is the true positive sample} \\ & \text{of } j^{\text{th}} \text{ positive bag } \mathbf{Y}_j^p \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$i = 1, \ldots M, \qquad j = 1, \ldots N_p. \tag{5}$$

With the above definition, we see that $\mathbf{Y}_p\mathbf{\Omega}$ has as many columns as the number of positive bags, i.e. N_p, and each column is picked from different positive bags. The reconstruction error from the selected true positive samples can be written as,

$$\mathcal{R}_p(\mathbf{A}_p, \mathbf{X}_p, \mathbf{\Phi}(.), \mathbf{\Omega}) = \|\mathbf{\Phi}(\mathbf{Y}_p)\mathbf{\Omega} - \mathbf{\Phi}(\mathbf{Y}_p)\mathbf{A}_p\mathbf{X}_p\mathbf{\Omega}\|_F^2 \tag{6}$$

Along with learning a dictionary, our goal is also to learn a feature space where the data of a given class is well represented using the corresponding dictionary and maximally orthogonal to samples from other classes. This will make the dictionaries more incoherent and data easily classifiable in separate classes. Therefore, we want to learn the feature space $\mathbf{\Phi}$ where selected positive samples and negative dictionaries (in which all the negative samples can be represented) are maximally orthogonal. This can be done by minimizing the following objective function

$$\mathcal{F}(\mathbf{\Phi}, \mathbf{A}_n, \mathbf{A}_p, \mathbf{\Omega}) = \|\mathbf{A}_n^T\mathbf{\Phi}(\mathbf{Y}_n)^T\mathbf{\Phi}(\mathbf{Y}_p)\mathbf{\Omega}\|_F^2 = \|\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_p)\mathbf{\Omega}\|_F^2 \tag{7}$$

While one can directly learn this kernel matrix, such non parametric kernel learning approaches are computationally expensive, involving optimization over hundreds of variables. Therefore, we use multiple base kernels and

seek to learn their optimum linear combination. Assume that there are a total of B base kernels and each of them is denoted by $\boldsymbol{\mathcal{K}}_b$, where $b = 1, \ldots, B$. A kernel $\boldsymbol{\mathcal{K}}(\mathbf{P}, \mathbf{Q})$, where $\mathbf{P}, \mathbf{Q}$ can be either $\mathbf{Y}_p$ or $\mathbf{Y}_n$, can be represented as a linear combination of the base kernels as follows

$$\boldsymbol{\mathcal{K}}(\mathbf{P}, \mathbf{Q}) = \sum_{b=1}^{B} \beta_b \boldsymbol{\mathcal{K}}_b(\mathbf{P}, \mathbf{Q}) \tag{8}$$

$$\text{s. t. } 0 \leq \beta_b \leq 1; \forall b = 1, \ldots, B, \text{ and } \sum_{b=1}^{B} \beta_b = 1 \tag{9}$$

where $\boldsymbol{\beta} \triangleq [\beta_1, \ldots, \beta_B]^T$ is a vector of linear weights to combine the predetermined base kernels. By combining the above discussed objectives, the overall cost to be minimized becomes

$$\begin{aligned} \mathcal{J}(\mathbf{A}_p, \mathbf{X}_p, \mathbf{A}_n, \mathbf{X}_n, \boldsymbol{\Omega}, \boldsymbol{\beta}) &= \mathcal{R}_n(\mathbf{A}_n, \mathbf{X}_n, \boldsymbol{\beta}) + \eta \mathcal{R}_p(\mathbf{A}_p, \mathbf{X}_p, \boldsymbol{\Omega}, \boldsymbol{\beta}) \\ &\quad + \lambda \mathcal{F}(\mathbf{A}_p, \mathbf{A}_n, \boldsymbol{\Omega}, \boldsymbol{\beta}) \end{aligned} \tag{10}$$

$$\begin{aligned} \text{subject to,} \quad & \|\mathbf{x}_i^p\|_0 \leq T_0, \quad \|\mathbf{x}_j^n\|_0 \leq T_0, \\ & i = 1, \ldots, M, \; j = 1, \ldots, N \end{aligned} \tag{11}$$

where η and λ are the hyper-parameters that control the contribution from the individual cost functions. We refer to the framework of learning dictionaries by optimizing the above objective as the Multiple Instance Dictionary Learning (MIDL).

3. Multiple Instance Dictionary Learning

Since our objective in Eq. (10) is jointly non-convex in all the variables, we update one variable at a time, with remaining ones fixed. In what follows, we discuss our optimization approach to update each of the variables $\mathbf{X}_n, \mathbf{X}_p, \mathbf{A}_n, \mathbf{A}_p, \boldsymbol{\Omega}$ and $\boldsymbol{\beta}$.

3.1. Updating Coefficient Matrices $\mathbf{X}_n$ and $\mathbf{X}_p$

With fixed dictionary and feature kernel, the sparse coefficients for the i^{th} samples can be computed by minimizing the following cost:

$$\mathcal{J}_{\mathbf{x}_i^n} = \|\mathbf{\Phi}(\mathbf{y}_i^n) - \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_n\mathbf{x}_i^n\| \tag{12}$$

$$\hat{\mathbf{x}}_i^n = \arg\min_{\mathbf{x}_i^n} \mathcal{J}_{\mathbf{x}_i^n} \quad \text{subject to} \quad \|\mathbf{x}_i^n\|_0 \leq T_0 \quad \forall i. \tag{13}$$

This optimization problem can be solved using the kernel orthogonal matching pursuit (KOMP) algorithm proposed in [19], and $\mathbf{X}_n$ can be updated as,

$$\mathbf{X}_n = [\hat{\mathbf{x}}_1^n, \ldots, \hat{\mathbf{x}}_N^n]. \tag{14}$$

Likewise, we compute the positive sample coefficient $\hat{\mathbf{x}}_i^p$ as,

$$\hat{\mathbf{x}}_i^p = \arg\min_{\mathbf{x}_i^p} \mathcal{J}_{\mathbf{x}_i^p} \quad \text{subject to} \quad \|\mathbf{x}_i^p\|_0 \leq T_0 \quad \forall i, \tag{15}$$

$$\text{where} \quad \mathcal{J}_{\mathbf{x}_i^p} = \|\mathbf{\Phi}(\mathbf{y}_i^p) - \mathbf{\Phi}(\mathbf{Y}_p)\mathbf{A}_p\mathbf{x}_i^p\|. \tag{16}$$

Finally, $\mathbf{X}_p$ can be updated as,

$$\mathbf{X}_p = [\hat{\mathbf{x}}_1^p, \ldots, \hat{\mathbf{x}}_M^p]. \tag{17}$$

3.2. Updating Negative Dictionary $\mathbf{A}_n$

To update the negative dictionary and the corresponding coefficients, we need to optimize $\mathcal{R}_n$ over $\mathbf{A}_n$ and $\mathbf{X}_n$. Similar to [19], we update one atom at a time in an efficient way. To update the k^{th} atom $\mathbf{a}_k$, we minimize the following error:

$$\mathcal{E}(\mathbf{a}_k) = \|\mathbf{\Phi}(\mathbf{Y}_n) - \Phi(\mathbf{Y}_n)\Big(\sum_{j\neq k} \mathbf{a}_j\mathbf{x}_T^j + \mathbf{a}_k\mathbf{x}_T^k\Big)\|_F^2. \tag{18}$$

In order to keep the sparsity same, we consider only those samples that use the current atom $\mathbf{a}_k$. Let these indices be $\mathbf{z}_k \triangleq \{i \mid 1 \leq i \leq K_n,$ such that $\mathbf{x}_T^k(i) \neq 0\}$ and $\mathbf{Z}_k \in \mathbb{R}^{K_n \times |\mathbf{z}_k|}$ be a binary valued matrix with $\mathbf{Z}_k(\mathbf{z}_k(j), j) = 1, j = 1, \ldots, |\mathbf{z}_k|$ and 0 otherwise. Now, the error $\mathcal{E}$ can be re-written as,

$$\mathcal{E}(\mathbf{a}_k) = \|\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R - \Phi(\mathbf{Y}_n)\mathbf{a}_k\mathbf{x}_R^k\|_F^2, \tag{19}$$

where $\mathbf{x}_R^k = \mathbf{x}_T^k\mathbf{Z}$ and $\mathbf{E}_k^R = \left(\mathbf{I} - \sum_{j \neq k} \mathbf{a}_j\mathbf{x}_T^j\right)\mathbf{Z}$. Note that $\Phi(\mathbf{Y}_n)\mathbf{a}_k\mathbf{x}_R^k$ is the rank-1 approximation of $\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R$. We write the singular value decomposition (SVD) of $\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ and then,

$$\Phi(\mathbf{Y}_n)\mathbf{a}_k\mathbf{x}_R^k = \sigma_1\mathbf{u}_1\mathbf{v}_1, \tag{20}$$

where, $\sigma_1 = \mathbf{\Sigma}(1,1)$ is the largest eigenvalue and $\mathbf{u}_1$ and $\mathbf{v}_1$ are the corresponding eigen vectors of $\mathbf{U}$ and $\mathbf{V}$, respectively. To keep the atom norm to unity, we set $\mathbf{a}_k = \mathbf{u}_1$ and $\mathbf{x}_R^k = \sigma_1\mathbf{v}_1^T$. However, it is difficult to compute the direct SVD of $\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R$. Hence, we approximate it with the SVD of the gram matrix,

$$(\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R)^T(\mathbf{\Phi}(\mathbf{Y}_n)\mathbf{E}_k^R) = (\mathbf{E}_k^R)^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{E}_k^R = \mathbf{V}\mathbf{\Delta}\mathbf{V}^T. \tag{21}$$

We then set

$$\mathbf{a}_k = \sigma_1^{-1}\mathbf{E}_k^R\mathbf{v}_1, \qquad \mathbf{x}_R^k = \sigma_1\mathbf{v}_1^T. \tag{22}$$

3.3. Updating Positive Dictionary $\mathbf{A}_p$

The positive dictionary update is very similar to that of the negative dictionary except the fact that positive dictionary is learned using the selected signals according to the $\mathbf{\Omega}$ matrix. In order to account for the additional

matrix $\mathbf{\Omega}$ in the dictionary update, we represent the kernel matrix and the reduced data as follows,

$$\boldsymbol{\mathcal{K}}(\mathbf{Y}_p^R, \mathbf{Y}_p^R) = \mathbf{\Omega}^T \boldsymbol{\mathcal{K}}(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{\Omega} \tag{23}$$

$$\mathbf{Y}_p^R = \mathbf{Y}_p\mathbf{\Omega}, \qquad \mathbf{X}_p^R = \mathbf{X}_p\mathbf{\Omega}. \tag{24}$$

We learn the reduced dictionary $\mathbf{A}_p^R$ and $\mathbf{X}_p^R$ exactly as negative dictionary with $\mathbf{Y}_n$ replaced with $\mathbf{Y}_p^R$ and $\mathbf{X}_n$ replaced with $\mathbf{X}_n^R$. Finally, to obtain $\mathbf{A}_p$ from $\mathbf{A}_p^R$, we copy the rows of $\mathbf{A}_p^R$ into those rows of $\mathbf{A}_p$ that correspond to the selected positive samples according to the matrix $\mathbf{\Omega}$ and set the rest of the rows to 0. In other words,

$$\mathbf{A}_p = \mathbf{\Omega}\mathbf{A}_p^R. \tag{25}$$

3.4. Updating Kernel Mixing Coefficients $\boldsymbol{\beta}$

While updating the kernel, we want to consider the following two criteria:

1. We want the dictionary atoms to have unit norms in the new feature space, i.e., $\mathbf{a}_p^T \boldsymbol{\mathcal{K}}(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p = 1,\ \forall p = 1, \ldots, K_p$ and $\mathbf{a}_n^T \boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n = 1,\ \forall n = 1, \ldots, K_n$. Here, $\mathbf{a}_p$ is the p^{th} atom of the positive dictionary and $\mathbf{a}_n$ is the n^{th} atom of the negative dictionary.
2. The cost $\mathcal{J}(.)$ in Eq. (10) should be minimized with respect to $\boldsymbol{\beta}$.

Considering these two criteria, we seek to minimize the following quadratic cost with respect to β

$$\mathcal{J}_{\boldsymbol{\beta}} = \boldsymbol{\beta}^T\mathbf{e}_n + \eta\boldsymbol{\beta}^T\mathbf{e}_p + \lambda\boldsymbol{\beta}^T\mathbf{G}\boldsymbol{\beta} + \lambda_L(\boldsymbol{\beta}^T\mathbf{Q}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\mathbf{q}) \tag{26}$$

$$\text{s. t. } 0 \leq \boldsymbol{\beta}_b \leq 1;\ \forall b = 1, \ldots, B, \text{ and } \sum_{b=1}^{B} \boldsymbol{\beta}_b = 1. \tag{27}$$

First and second part of the above cost minimizes the reconstruction error of the negative and the selected positive samples. The third enforces the orthogonality between the two classes and the last part of the cost enforces the atom norms in feature space to be close to unity. Here, λ_L is set to a large value to penalize heavily on any norm other than unity. In our implementation, we use $\lambda_L = 100$. Various terms in the cost are computed as follows (see appendix for the derivation),

$$\mathbf{e}_n(b) = \sum_{i=1}^{N} \big(\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^n, \mathbf{y}_i^n) + (\mathbf{x}_i^n)^T \mathbf{A}_n^T \boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n) \mathbf{A}_n \mathbf{x}_i^n - 2\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^n, \mathbf{Y}_n) \mathbf{A}_n \mathbf{x}_i^n \big). \tag{28}$$

$$\mathbf{e}_p(b) = \sum_{i=1}^{M} \big(\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^p, \mathbf{y}_i^p) + (\mathbf{x}_i^p)^T \mathbf{A}_p^T \boldsymbol{\mathcal{K}}_b(\mathbf{Y}_p, \mathbf{Y}_p) \mathbf{A}_p \mathbf{x}_i^p - 2\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^p, \mathbf{Y}_p) \mathbf{A}_p \mathbf{x}_i^p \big) \mathbf{1}_{[\mathbf{y_i} \in \boldsymbol{\Omega}_s]}, \tag{29}$$

where $\mathbf{1}_{[\mathbf{y_i} \in \boldsymbol{\Omega}_s]}$ is the indicator variable which is 1 when $\mathbf{y}_i$ is one of the selected true positive samples and is 0 otherwise. Here, $\boldsymbol{\Omega}_s$ is defined as,

$$\boldsymbol{\Omega}_s = \{ i \mid 1 \leq i \leq M, \sum_{i=1}^{N_p} \boldsymbol{\Omega}[i, :] \neq 0 \}. \tag{30}$$

The third term in Eq. (26), minimizes the similarity between positive and negative dictionary atoms by learning a feature space in which both the classes are maximally orthogonal. We minimize the projection of the negative dictionary atoms onto the selected positive samples $\mathcal{F}$ which can be written

as,

$$\begin{aligned}\mathcal{F} &= \text{trace}\{(\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega})(\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega})^T\} \\ &= \text{trace}\{(\mathbf{A}_n^T\sum_{b=1}^{B}\beta_b\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega})(\mathbf{A}_n^T\sum_{b=1}^{B}\beta_b\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega})^T\} \quad (31) \\ &= \sum_{bi=1}^{B}\sum_{bj=1}^{B}\boldsymbol{\beta}_{bi}\mathbf{G}(bi,bj)\boldsymbol{\beta}_{bj}. \quad (32)\end{aligned}$$

Here, $\mathbf{G}(bi,bj)$ is defined as

$$\mathbf{G}(bi,bj) = \text{trace}[\big(\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_{bi}(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega}\big)\big(\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_{bj}(\mathbf{Y}_n,\mathbf{Y}_p)\boldsymbol{\Omega}\big)^T] \quad (33)$$

where, $bi = 1,\ldots,B$, and, $bj = 1,\ldots,B$.

Finally, the matrix $\mathbf{Q} \in \mathbb{R}^{B\times B}$ and the vector $\mathbf{q} \in \mathbb{R}^B$ in the fourth term of Eq.(26) are computed as follows,

$$\begin{aligned}\mathbf{Q} = &\sum_{p=1}^{K_p}\begin{bmatrix}\mathbf{a}_p^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\\ \cdots\\ \mathbf{a}_p^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\end{bmatrix}\begin{bmatrix}\mathbf{a}_p^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\\ \cdots\\ \mathbf{a}_p^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\end{bmatrix}^T \\ &+\sum_{n=1}^{K_n}\begin{bmatrix}\mathbf{a}_n^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\\ \cdots\\ \mathbf{a}_n^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\end{bmatrix}\begin{bmatrix}\mathbf{a}_n^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\\ \cdots\\ \mathbf{a}_n^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\end{bmatrix}^T, \quad (34)\end{aligned}$$

where $\mathbf{a}_p$ is the p^{th} atom of the positive dictionary and $\mathbf{a}_n$ is the n^{th} atom of the negative dictionary and

$$\mathbf{q} = \sum_{p=1}^{K_p}\begin{bmatrix}\mathbf{a}_p^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\\ \cdots\\ \mathbf{a}_p^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p\end{bmatrix} + \sum_{n=1}^{K_n}\begin{bmatrix}\mathbf{a}_n^T\boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\\ \cdots\\ \mathbf{a}_n^T\boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n\end{bmatrix}. \quad (35)$$

Since $\mathcal{J}_{\boldsymbol{\beta}}$ in Eq. (26) is a quadratic cost with linear constraints in $\boldsymbol{\beta}$, it can be solved with any quadratic program (QP) solver [6].

3.5. Updating Positive Sample Selection Matrix $\mathbf{\Omega}$

According to the cost $\mathcal{J}$ in Eq. (10), we select those samples which give minimum error with the positive dictionary and are maximum orthogonal to the negative dictionary. However, a positive dictionary is required in order to select the samples based on the positive dictionary. To avoid this 'chicken and egg' problem, we select positive samples by maximizing their reconstruction error onto the negative dictionary and minimizing their projection onto the negative dictionary. For a given positive bag $\mathbf{Y}_b^p$, we select the sample that maximizes the following cost,

$$\mathcal{E}_b(i) = \|\mathbf{\Phi}(\mathbf{y}_{b,i}) - \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_p\mathbf{x}_{b,i}\|_2^2 - \lambda\|\mathbf{A}_n^T\mathbf{\Phi}(\mathbf{Y}_n)^T\mathbf{\Phi}(\mathbf{y}_{b,i})\|_2^2 \tag{36}$$

$$= \boldsymbol{\mathcal{K}}(\mathbf{y}_{b,i}, \mathbf{y}_{b,i}) + \mathbf{x}_{b,i}^T\mathbf{A}_p^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_p\mathbf{x}_{b,i} - 2\boldsymbol{\mathcal{K}}(\mathbf{y}_{b,i}, \mathbf{Y}_n)\mathbf{A}_p\mathbf{x}_{b,i} - \lambda\|\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{y}_{b,i})\|_2^2. \tag{37}$$

Sample selected as true positive from the b^{th} positive bag is denoted by $\mathbf{y}_{b,m^*}$, where,

$$m^* = \arg\max_i \mathcal{E}_b(i). \tag{38}$$

Finally, we update $\mathbf{\Omega}$ according to Eq. (4).

3.6. Algorithm Initialization

We initialize $\mathbf{A}_n$ with all zeros except for a randomly selected location in each column, which is set equal to 1. This corresponds to initializing the dictionary with randomly selected samples in the feature space. Since we have the labels of all the negative sample, we run a few alternate iterations of updating $\mathbf{A}_n$ and $\mathbf{X}_n$. Positive sample selection matrix $\mathbf{\Omega}$ is updated by choosing those samples that give the maximum reconstruction error with

the negative dictionary. Kernel mixing coefficient vector $\boldsymbol{\beta}$ is initialized to a uniform distribution, giving equal weight to all the base kernels. Finally, the columns of positive dictionary $\mathbf{A}_p$ are initialized with all zeros but a single 1. The position of this 1 in a column is randomly selected, such that this corresponds to the sample in the feature space which has been chosen as the true positive sample according to $\boldsymbol{\Omega}$. The complete MIDL algorithm is summarized in Algorithm 1.

Algorithm 1: MIDL Algorithm

Input: $\mathbf{Y}_n, \mathbf{Y}_p, \boldsymbol{\mathcal{K}}_b, T_0, J \triangleq$ iteration count

Output: $\mathbf{A}_n, \mathbf{A}_p, \mathbf{X}_n, \mathbf{X}_p, \boldsymbol{\beta}, \boldsymbol{\Omega}$

Initialize $\mathbf{A}_n, \mathbf{A}_p, \mathbf{X}_n, \mathbf{X}_p, \boldsymbol{\beta}, \boldsymbol{\Omega}$ as described in sub-section 3.6.

for $t = 1, \ldots, J$ **do**

1. Update $\boldsymbol{\beta}$ by optimizing cost in Eq. (26).
2. Update $\mathbf{X}_n$ and $\mathbf{X}_p$ using Eq. (14) and (17).
3. Update $\mathbf{A}_n$ and $\mathbf{A}_p$ using Eq (19) and (25).
4. Update $\boldsymbol{\Omega}$ using Eq.(38) and (4).

end

return $\mathbf{A}_n, \mathbf{A}_p, \mathbf{X}_n, \mathbf{X}_p, \boldsymbol{\beta}, \boldsymbol{\Omega}$

4. Classification

Given a test bag, we may need to predict the label of the whole bag or each instance in it. Once we decide the label of an instance, the bag label can trivially be determined as +1 if and only if at least one instance is positive. To determine the label of a given test instance $\mathbf{y}_t$, we compute its sparse code

$\mathbf{x}_t \in \mathbb{R}^{K_p+K_n}$ by solving the following optimization problem using the kernel OMP algorithm

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(\mathbf{Y})\mathbf{A}\|_2^2 \qquad \text{subject to} \quad \|\mathbf{x}_t\|_0 \leq T_1,$$

where,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0}_{K_p \times K_n} \\ \mathbf{0}_{K_n \times K_p} & \mathbf{A}_n \end{bmatrix} \text{ and } \mathbf{Y} = [\mathbf{Y}_p \mid \mathbf{Y}_n]. \tag{39}$$

Let the $\mathbf{x}_t^p \in \mathbb{R}^{K_p}$ be a vector consisting of the first K_p elements of $\mathbf{x}_t$ and $\mathbf{x}_t^n \in \mathbb{R}^{K_n}$ be the last K_n elements of $\mathbf{x}_t$. We compute the reconstruction error using the positive dictionary $\boldsymbol{\epsilon_p}$ and the reconstruction error using the negative dictionary $\boldsymbol{\epsilon_n}$ as follows,

$$\boldsymbol{\epsilon_p} = \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(Y_p)\mathbf{A}_p\mathbf{x}_t^p\|_2^2 \tag{40}$$

$$= \boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{y}_t) + (\mathbf{x}_t^p)^T\mathbf{A}_p^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{A}_p\mathbf{x}_t^p - 2\boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{y}_t)\mathbf{A}_p\mathbf{x}_t^p. \tag{41}$$

Similarly,

$$\boldsymbol{\epsilon_n} = \boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{y}_t) + (\mathbf{x}_t^n)^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{x}_t^n - 2\boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{y}_t)\mathbf{A}_n\mathbf{x}_t^n. \tag{42}$$

Finally, the label of the instance is decided as,

$$\text{class label of } \mathbf{y}_t = \begin{cases} +1 & \text{if } \boldsymbol{\epsilon}_p \leq \boldsymbol{\epsilon}_n \\ -1 & \text{otherwise.} \end{cases} \tag{43}$$

5. Experiments

In this section, we present several experimental results demonstrating the effectiveness of the proposed dictionary learning method for classification tasks. We present classification results on the USPS digit dataset [11], MSR2

Action dataset [4], and gender recognition using the AR face dataset [18]. The positive dictionary is not very reliable in the beginning of the learning process because it has been learned using the mixture of negative and positive samples. Therefore, we set λ and η values to 0 at the beginning of the first iteration and slowly increase their values by 10% at each iteration until they reach their predefined values. We use a total of 50 base kernels for learning the optimal kernel. We use a linear kernel, the histogram intersection kernel, 10 polynomial kernels of degree $d = 2$ (with coefficients c varying from 0.5 to 5 in steps of 0.5), 10 polynomial kernels of degree $d = 3$ (with coefficients c varying from 0.5 to 5 in steps of 0.5), and remaining 28 Gaussian kernel functions with parameter c increasing in the steps of 0.01. We compare the performance of our method with that of several recent state-of-the-art MIL methods including diverse density (DD) [17], EM diverse density (EM-DD) [28], nearest neighbor (C-kNN) [25], multiple instance support vector machine (MI-SVM) [2] and recently proposed max-margin dictionary learning (MMDL) [26]. In order to compare it with MMDL, we computed the max-margin dictionary as proposed in [26] and the instance probabilities were computed based on its projection on the learned dictionary. The parameters of all the algorithms were optimized using two-fold cross validation. The training data was divided into two sets, the algorithm was learned using the data from one set and optimized by testing on the other set. Finally, the learned parameters were used to learn the algorithm on all the training data and tested on the test data.

5.1. Digit Recognition

The USPS digit dataset [11] consists of 10 classes (digits $0-9$) with total of 7291 training images and 2001 test images. We perform 10 experiments in which one digit is chosen as the positive class and the remaining digits are considered as the negative class samples. For training, we form 25 positive and 25 negative bags. Each of the positive bags contains 2 positive samples and 2 randomly chosen negative samples, while each of the negative bags contains 4 randomly picked negative samples. We test the learned model on 75 positive bags and 75 negative bags. For all the experiments we have set $\lambda = 1$, $\eta = 1$, the sparsity level $T_0 = 4$ and the number of iterations to 40.

As we see from Table 1, the proposed method performs better than the other compared methods for most of the digits. The parameters of the competing algorithms were set using two-fold cross validation. The training instances were divided into two sets and parameters were adjusted to maximize the accuracy on half of the instances while using the data for training from the other half. The closest performing algorithm is the MI-SVM which is similar to our method in the sense that MI-SVM also selects one sample from each bag and alternate between learning SVM plane and selecting one positive sample from each positive bag. Selecting positive samples helps to improve the accuracy of our method.

5.1.1. Pre-images of learned atoms

For the USPS digit dataset we use the pixel values as features, hence, we can analyze our results by visualizing the learned atoms of the positive dictionary in the feature space using their pre-images. Recall that the p^{th} atom of positive dictionary in feature space is represented by $\mathbf{\Phi}(\mathbf{Y}_p)\mathbf{a}_p$, where

	Bag/ Inst	DD [17]	EMDD [28]	C-kNN [25]	mi-SVM [2]	MI-SVM [2]	MMDL [26]	**MIDL**
Digit 0	Bag	43.3	81.3	54.0	84.0	88.0	82.7	**94.7**
	Inst	72.2	83.6	68.5	92.3	86.7	86.5	**95.0**
Digit 1	Bag	**95.3**	50.0	76.8	92.0	88.5	88.7	94.7
	Inst	**95.3**	75.0	84.7	76.7	81.5	92.8	93.0
Digit 2	Bag	52.0	37.3	56.0	84.0	**90.7**	85.3	89.3
	Inst	73.8	52.7	71.5	74.6	82.3	88.2	**90.7**
Digit 3	Bag	46.7	52.7	66.7	90.0	61.3	84.0	**96.0**
	Inst	73.8	52.7	73.3	95	78.0	87.0	**94.7**
Digit 4	Bag	80.0	80.0	56.0	82.0	82.0	86.7	**94.0**
	Inst	84.2	84.2	71.0	90.2	85.7	86.2	**92.3**
Digit 5	Bag	76.0	50.0	54.7	89.3	55.3	80.0	**93.3**
	Inst	83.7	75.0	71.3	90.7	76.3	84.3	**91.3**
Digit 6	Bag	48.0	86.7	71.3	89.3	73.3	84.0	**99.3**
	Inst	71.8	87.8	74.2	94.2	81.5	86.0	**96.2**
Digit 7	Bag	98.7	50.0	73.3	83.3	84.0	84.0	**99.3**
	Inst	**96.0**	75.0	74.8	92.5	85.8	91.7	95.7
Digit 8	Bag	47.3	80.0	46.0	86.0	86.0	84.7	**94.0**
	Inst	72.2	83.8	65.8	90.8	87.3	86.0	**90.8**
Digit 9	Bag	88.6	79.3	64.7	88.0	90.7	84.7	**90.7**
	Inst	86.2	82.5	68.8	88.3	88.0	85.3	**92.7**
Avg.	Bag	67.6	64.7	61.9	86.8	80.0	84.5	**94.3**
	Inst	80.9	75.7	72.4	88.5	83.3	87.4	**93.2**

Table 1: Bag and Instance accuracy for the proposed method, compared to the competing ones for the USPS digit [11] recognition experiment. Each row compares the accuracy for a digit chosen as the positive class and the rest of the digits as the negative class.

$\mathbf{a}_p \in \mathbb{R}^M$ is the representation of the kernel dictionary atom with respect to the base $\mathbf{\Phi}(\mathbf{Y}_p)$ in the feature space G. The pre-image of $\mathbf{\Phi}(\mathbf{Y}_p)\mathbf{a}_k$ is obtained by seeking a vector in the input space $\mathbf{d}_p \in \mathbb{R}^d$ that minimizes the cost function $\|\mathbf{\Phi}(\mathbf{d}_p) - \mathbf{\Phi}(\mathbf{Y}_p)\mathbf{a}_p\|^2$. Due to various noise effects and the generally non-invertible mapping $\mathbf{\Phi}$, the exact pre-image does not always exist. However, the approximated pre-image can be reconstructed without venturing into the feature space using the techniques described in [22]. We show the pre-images of 10 atoms of each of the digits in Figure 2. Here, c^{th} row corresponds to the positive dictionary atom when the digit c is chosen as the positive class and the rest of the digits as the negative class.

Figure 2: Pre-Images of the USPS digit's positive dictionary atoms for different digits as the positive class.

5.1.2. Noisy data

Furthermore, we study the behavior of our method when the data is contaminated by Gaussian noise and contain missing pixels. For the missing pixel scenario, we randomly set certain percentage of pixel count to zero. The Gaussian noise is added with varying standard deviation. As shown in Figure 3, our method performs better than the competing algorithms. This is the case because dictionary-based methods are known to be robust in the presence of noise [1], [27].

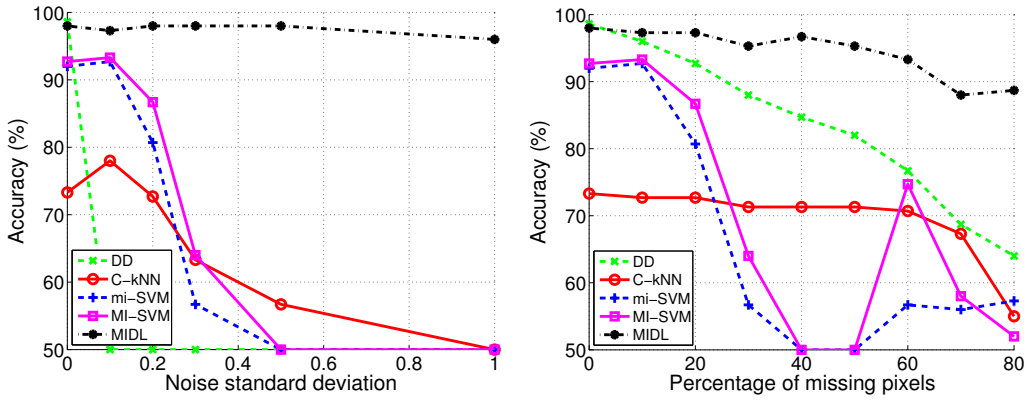
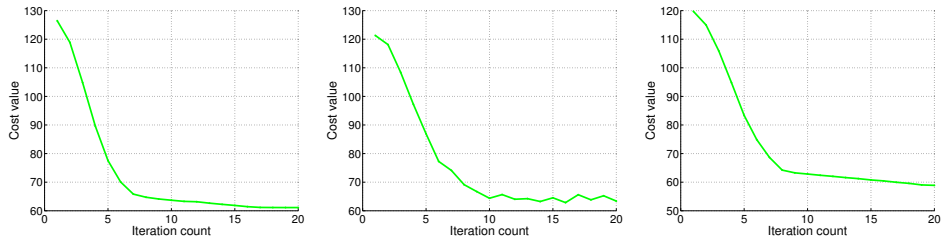


Figure 3: Comparison of USPS digit recognition accuracies for different methods in the presence of (a) Gaussian noise and (b) missing-pixel effects.

5.1.3. Convergence of the Proposed Method

The proposed method iterates until the cost converges or the maximum number of iterations are reached. At each iteration, the dictionary, the selection matrix, and the kernel weights are updated. Although, there is no theoretical guarantee that the cost should converge to a global minima, we

empirically observe that the proposed optimization approach improves the cost at each iteration. This is essentially what is shown in Fig. 4 for the experiment with the USPS digit dataset. One can clearly see the decrease of the cost as the iterations increase.



(a) (b) (c)

Figure 4: Convergence of the cost for three experiments (a) USPS digit 0 (b) USPS digit 1, and, (c) USPS digit 2

5.2. Action Recognition

The MSR2 action dataset has 54 videos and each video sequence has the following three actions: *clapping*, *hand-waving* and *boxing*. In our experimental setup, we use 27 videos for training and the remaining 27 videos for testing. We create one positive bag and one negative bag from each video sequence, resulting in 27 positive and 27 negative bags for training and the same number of positive and negative bags for testing. Since most of the video sequences have just 1 or 2 action samples (cuboids) per class, we add more action samples for each class as follows. For each action cuboid (sample), we include two more action cuboids, each with the same spatial co-ordinates but overlapping by 50% in temporal dimension. One action cuboid can degenerate into maximum of 3 action cuboids. Each positive bag contains all

the positive action cuboids and two negative cuboids from the same video sequence. The negative bag has all the remaining negative action cuboids. We compute the bag of words of dense spatial temporal interest point (STIP) features [13]. We set $\lambda = 0.01$ and $\eta = 1$, and the sparsity $T_0 = 5$ for this experiment. The results shown in Table 2 demonstrate that the proposed method performs significantly better than other compared methods. One of the reasons why our method performs better is that we learn the appropriate feature space in which the data is well represented as well as separated, while the other methods work in the pre-determined feature space.

	Bag/ Inst	DD [17]	EMDD [28]	C-kNN [25]	mi-SVM [2]	MI-SVM [2]	MMDL [26]	**MIDL**
Clapping	Bag	56.3	56.3	56.3	64.5	58.3	58.3	**72.9**
	Inst	77.0	77.0	77.0	74.5	76.0	71.7	**80.6**
Waving	Bag	50.0	50.0	50.0	75.9	70.3	61.1	**77.8**
	Inst	66.7	66.7	66.7	80.8	76.6	71.2	**85.0**
Boxing	Bag	50.0	50.0	50.0	59.2	64.8	64.8	**81.5**
	Inst	62.1	62.1	62.1	76.3	73.5	64.7	**77.3**

Table 2: Bag and Instance accuracy for the proposed method, compared to competing ones for MSR2 Action dataset [4]. Each row compares the accuracy for an action chosen as positive class and remaining ones as negative class.

We further study our results by looking at the incorrectly selected features (by examining the selection matrix $\mathbf{\Omega}$) for each of the classes (*clapping*, *hand-*

waving, and *boxing*). Figure 5 shows the incorrectly selected positive samples from the positive bags according to the $\mathbf{\Omega}$ matrix. We find that the *boxing* class selects all the samples correctly, which probably is the reason why it has better bag accuracy than the other classes.

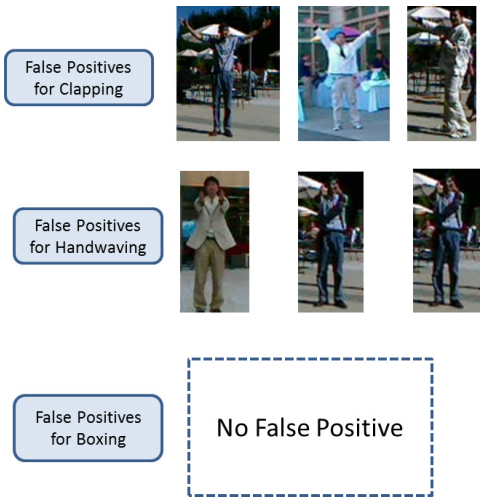


Figure 5: False positives selected by the $\mathbf{\Omega}$ matrix. The false positives for *clapping* are, two *hand-waving* and one *boxing* actions, while false positives for *hand-waving* are 3 *clapping* action units. *Boxing* class does not have any false positive. One frame for each false positive is shown in the figure.

5.3. Gender Recognition

Gender recognition task is a two class problem with male gender as the positive class and female gender as the negative class. For this purpose we use 50 male subjects and 50 female subjects of the AR face database. Each subject has 14 faces (7 from each of the two sessions). We use 25 male and 25 female subjects for training and the remaining ones for testing. With $350(= 25 * 14)$ positive samples and 350 negative samples, we form

175 positive bags and 175 negative bags. Each positive bag consists of 2 positive samples and 1 negative sample, while a negative bag has 1 negative sample. We set $\lambda = 0.5$ and $\eta = 1$ for this experiment. Table 3 compares the performance of our method with that of the other competitive methods.

Bag/ Inst	DD [17]	EMDD [28]	C-kNN [25]	mi-SVM [2]	MI-SVM [2]	MMDL [26]	**MIDL**
Bag	85.7	52.7	55.4	89.7	90.0	64.8	**96.6**
Inst	83.0	59.7	64.2	88.6	88.4	64.7	**92.14**

Table 3: Bag and Instance accuracy for various methods on the gender recognition task. Male faces are chosen as the positive class while female faces are chosen as the negative class.

6. Conclusion

In this paper, we proposed a non-linear dictionary learning method for the MIL framework. We formulated the MIL problem as a kernel learning problem and iteratively learned the dictionary in the embedded space of the learned kernel. We also described how we can learn the appropriate kernel matrix instead of using a pre-defined one. This idea of multiple kernel learning can also be extended to the discriminative dictionary learning setting, where labels of all the samples are known exactly. Although, this work addresses the two class classification problem, in the future work, we plan to extended it to the case of multi-class classification problem, where data for all the classes are available in the form of bags.

Appendix

Derivation of (28) and (29):

Reconstruction error of the negative samples can be written as,

$$\mathcal{R}_n = \text{trace}\{\big(\mathbf{\Phi}(\mathbf{Y}_n) - \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\big)\big(\mathbf{\Phi}(\mathbf{Y}_n) - \mathbf{\Phi}(\mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\big)^T\} \tag{44}$$
$$= \text{trace}\{\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n) + \mathbf{X}_n^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n - 2\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\}.$$

Writing $\boldsymbol{\mathcal{K}}(\mathbf{Y}_n, \mathbf{Y}_n)$ as the linear combination of the base kernels $\sum_b \beta_b\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)$ gives

$$\mathcal{R}_n = \text{trace}\{\sum_b \beta_b\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n) + \sum_b \beta_b\mathbf{X}_n^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n$$
$$- 2\sum_b \beta_b\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\} \tag{45}$$
$$= \sum_b \beta_b\text{trace}\{\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n) + \mathbf{X}_n^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n$$
$$- 2\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\}. \tag{46}$$

Now, we define $\mathbf{e}_n(b)$ in (28) as follows,

$$\mathbf{e}_n(b) \triangleq \text{trace}\{\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n) + \mathbf{X}_n^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n - 2\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{X}_n\}$$
$$= \sum_{i=1}^{N}\big(\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^n, \mathbf{y}_i^n) + (\mathbf{x}_i^n)^T\mathbf{A}_n^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{x}_i^n - 2\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^n, \mathbf{Y}_n)\mathbf{A}_n\mathbf{x}_i^n\big).$$

$\mathbf{e}_p(b)$ for the positive samples in (29) is similarly defined with the difference

that only selected positive samples (using selection matrix $\mathbf{\Omega}$) are considered.

$$\begin{aligned}\mathbf{e}_p(b) \triangleq \text{trace}\{&\mathbf{\Omega}^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{\Omega}+\mathbf{\Omega}^T\mathbf{X}_p^T\mathbf{A}_p^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{A}_p\mathbf{X}_p\mathbf{\Omega}\\&-2\mathbf{\Omega}^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{A}_p\mathbf{X}_p\mathbf{\Omega}\}\end{aligned}\tag{47}$$

$$\begin{aligned}=\sum_{i=1}^{M}\big(&\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^p,\mathbf{y}_i^p)+(\mathbf{x}_i^p)^T\mathbf{A}_p^T\boldsymbol{\mathcal{K}}_b(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{A}_p\mathbf{x}_i^p\\&-2\boldsymbol{\mathcal{K}}_b(\mathbf{y}_i^p,\mathbf{Y}_p)\mathbf{A}_p\mathbf{x}_i^p\big)\mathbf{1}_{[\mathbf{y_i}\in\mathbf{\Omega}_s]}.\end{aligned}\tag{48}$$

Derivation of (34) and (35):

These two equations enforce dictionary atom norms to unity by minimizing the following cost,

$$\mathcal{C}=\sum_{p=1}^{K_p}(\mathbf{a}_p^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_p,\mathbf{Y}_p)\mathbf{a}_p-1)^2+\sum_{n=1}^{K_n}(\mathbf{a}_n^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{a}_n-1)^2.\tag{49}$$

For notational simplicity, we denote a general atom (positive or negative) by $\mathbf{a}$, a kernel matrix ($\boldsymbol{\mathcal{K}}(\mathbf{Y}_n,\mathbf{Y}_n)$ or $\boldsymbol{\mathcal{K}}(\mathbf{Y}_p,\mathbf{Y}_p)$) by $\boldsymbol{\mathcal{K}}$, and the b^{th} base kernel by $\boldsymbol{\mathcal{K}}_b$ and then solve for the term under the sum:

$$\begin{aligned}(\mathbf{a}^T\boldsymbol{\mathcal{K}}\mathbf{a}-1)^2&=(\mathbf{a}^T\boldsymbol{\mathcal{K}}\mathbf{a})(\mathbf{a}^T\boldsymbol{\mathcal{K}}\mathbf{a})+1-2\mathbf{a}^T\boldsymbol{\mathcal{K}}\mathbf{a}\\&=\Big(\mathbf{a}^T(\sum_{bi=1}^{B}\beta_{bi}\boldsymbol{\mathcal{K}}_{bi})\mathbf{a}\Big)\Big(\mathbf{a}^T(\sum_{bj=1}^{B}\beta_{bj}\boldsymbol{\mathcal{K}}_{bj})\mathbf{a}\Big)+1-2\mathbf{a}^T(\sum_{b=1}^{B}\beta_b\boldsymbol{\mathcal{K}}_b)\mathbf{a}\\&=\sum_{bi=1}^{B}\sum_{bj=1}^{B}\beta_{bi}\beta_{bj}\Big(\mathbf{a}^T\boldsymbol{\mathcal{K}}_{bi}\mathbf{a}\Big)\Big(\mathbf{a}^T\boldsymbol{\mathcal{K}}_{bj}\mathbf{a}\Big)+1-2\sum_{b=1}^{M}\beta_b(\mathbf{a}^T\boldsymbol{\mathcal{K}}_b\mathbf{a})\end{aligned}\tag{50}$$

$$(\mathbf{a}^T\boldsymbol{\mathcal{K}}\mathbf{a}-1)^2=\boldsymbol{\beta}^T\begin{bmatrix}\mathbf{a}^T\boldsymbol{\mathcal{K}}_1\mathbf{a}\\\dots\\\mathbf{a}^T\boldsymbol{\mathcal{K}}_B\mathbf{a}\end{bmatrix}\begin{bmatrix}\mathbf{a}^T\boldsymbol{\mathcal{K}}_1\mathbf{a}\\\dots\\\mathbf{a}^T\boldsymbol{\mathcal{K}}_B\mathbf{a}\end{bmatrix}^T\boldsymbol{\beta}+1-2\boldsymbol{\beta}^T\begin{bmatrix}\mathbf{a}^T\boldsymbol{\mathcal{K}}_1\mathbf{a}\\\dots\\\mathbf{a}^T\boldsymbol{\mathcal{K}}_B\mathbf{a}\end{bmatrix}.\tag{51}$$

Now, we define $\mathbf{Q}$, and $\mathbf{q}$ as

$$\mathbf{Q} = \sum_{p=1}^{K_p} \begin{bmatrix} \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \\ \cdots \\ \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \end{bmatrix} \begin{bmatrix} \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \\ \cdots \\ \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \end{bmatrix}^T + \sum_{n=1}^{K_n} \begin{bmatrix} \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \\ \cdots \\ \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \end{bmatrix} \begin{bmatrix} \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \\ \cdots \\ \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \end{bmatrix}^T, \tag{52}$$

$$\mathbf{q} = \sum_{p=1}^{K_p} \begin{bmatrix} \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \\ \cdots \\ \mathbf{a}_p^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_p, \mathbf{Y}_p)\mathbf{a}_p \end{bmatrix} + \sum_{n=1}^{K_n} \begin{bmatrix} \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_1(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \\ \cdots \\ \mathbf{a}_n^T \boldsymbol{\mathcal{K}}_B(\mathbf{Y}_n, \mathbf{Y}_n)\mathbf{a}_n \end{bmatrix}. \tag{53}$$

Thus, the cost $\mathcal{C}$ in Eq (49) can be written as,

$$\mathcal{C}(\boldsymbol{\beta}) = \boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta} - 2\boldsymbol{\beta}^T \mathbf{q} + K_p + K_n. \tag{54}$$

K_p and K_n are constants and hence can be dropped from the optimization cost. This explains the fourth term in $\mathcal{J}_{\boldsymbol{\beta}}$ Eq. (26).

Acknowledgment

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

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 561–568. 2003.

[3] B. Babenko, Ming-Hsuan Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 983–990, 2009.

[4] L. Cao, Z. Liu, and T.S. Huang. Cross-dataset action detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1998–2005, 2010.

[5] Y. Chen, C. S. Sastry, V. M. Patel, J. P. Phillips, and R. Chellappa. Rotation invariant simultaneous clustering and dictionary learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1053–1056, 2012.

[6] T.F. Coleman and Y. Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, April 1996.

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

[8] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2797, 2009.

[9] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, July 2011.

[10] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 609–616. 2007.

[11] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, May 1994.

[12] 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*, pages 76–83, 2012.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[14] J. Mairal, F. Bach, J. Pnce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[15] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, April 2012.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1033–1040. 2008.

[17] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 570–576. 1998.

[18] A.M. Martinez and R. Benavente. The AR face database. *CVC Technical Report No. 24,*, June 1998.

[19] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Kernel dictionary learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pages 2021–2024, 2012.

[20] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3508, 2010.

[21] M. Ranzato, F. Haung, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[22] B. Scholkopf and A. J. Smola. *Learning With Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[23] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa. Learning discriminative dictionaries with partially labeled data. In *IEEE International Conference on Image Processing (ICIP)*, pages 3113–3116, 2012.

[24] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pages 2042–2045, 2010.

[25] J. Wang. Solving the multiple-instance problem: A lazy learning approach. In *International Conference on Machine Learning (ICML)*, pages 1119–1125, 2000.

[26] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu. Max-margin multiple-instance dictionary learning. In *International Conference on Machine Learning (ICML)*, pages 846–854, 2013.

[27] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, June 2010.

[28] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1073–1080. 2001.