Multi-Instance Dimensionality Reduction via Sparsity and Orthogonality

Article *in* Neural Computation · October 2018

DOI: 10.1162/neco_a_01140

CITATIONS
0

READS
78

3 authors, including:

Zhu Hong
Hong Kong Baptist University
9 PUBLICATIONS **92** CITATIONS

SEE PROFILE

Li-Zhi Liao
Hong Kong Baptist University
86 PUBLICATIONS **2,263** CITATIONS

SEE PROFILE

LETTER **Communicated by Soumya Ray**

Multi-Instance Dimensionality Reduction via Sparsity and Orthogonality

Hong Zhu
zhuhongmath@126.com
Faculty of Science, Jiangsu University, Zhenjiang, Jiangsu 212013, China

Li-Zhi Liao
liliao@hkbu.edu.hk
Michael K. Ng
mng@math.hkbu.edu.hk
Department of Mathematics, Hong Kong Baptist University, Hong Kong, China

We study a multi-instance (MI) learning dimensionality-reduction algorithm through sparsity and orthogonality, which is especially useful for high-dimensional MI data sets. We develop a novel algorithm to handle both sparsity and orthogonality constraints that existing methods do not handle well simultaneously. Our main idea is to formulate an optimization problem where the sparse term appears in the objective function and the orthogonality term is formed as a constraint. The resulting optimization problem can be solved by using approximate augmented Lagrangian iterations as the outer loop and inertial proximal alternating linearized minimization (iPALM) iterations as the inner loop. The main advantage of this method is that both sparsity and orthogonality can be satisfied in the proposed algorithm. We show the global convergence of the proposed iterative algorithm. We also demonstrate that the proposed algorithm can achieve high sparsity and orthogonality requirements, which are very important for dimensionality reduction. Experimental results on both synthetic and real data sets show that the proposed algorithm can obtain learning performance comparable to that of other tested MI learning algorithms.

1 Introduction

In multi-instance (MI) learning (Dietterich, Lathrop, & Lozano-Pérez, 1997), data observations (bags) can have different alternative descriptions (instances). Labels are assigned to bags, but they are not assigned to instances. Under the standard MI assumption, a bag is positively labeled if it contains at least one positive instance and negatively labeled if all instances contained in a bag are negative. The goal of MI learning is to learn from a training data set and build a classifier for correctly labeling unseen bags. MI

Neural Computation **30, 3281–3308** (2018)
doi:10.1162/neco_a_01140

learning naturally fits various real-world applications—for example, drug activity prediction (Dietterich et al., 1997), text categorization (Andrews, Tsochantaridis, & Hofmann, 2003), image retrieval (Andrews et al., 2003), medical diagnosis (Fung & Ng, 2007), and face detection (Viola & Jones, 2004).

In the literature, MI learning on drug activity prediction was first represented in Dietterich et al. (1997). The task of drug activity prediction is to predict whether a given molecule is qualified to make a drug. We note that molecules are qualified to make a drug when one of its low-energy shapes can bind tightly to the target object. In contrast, molecules are not qualified to make a drug when none of its low-energy shapes can bind tightly to the target object. The main challenge of drug activity prediction is that each molecule can have many possible low-energy shapes, and only one or a few of them can bind tightly to the target object. In practice, few training molecules are known to be qualified to make a drug, but we may not know low-energy shapes for such qualification. In MI learning, each molecule can be described with a bag, and each instance corresponds to one low-energy shape.

1.1 Related Work on MI Learning. The axis-parallel rectangle (APR) algorithm based on greedy selection was proposed to predict drug activity. The learnability and computational complexities of the APR algorithm were analyzed in Auer, Long, and Srinivasan (1997), Long and Tan (1998), Blum and Kalai (1998). Then several MI learning algorithms were developed and studied, such as diverse density (Maron & Lozano-Pérez, 1998), citation-KNN (Wang & Zucker, 2000), ID3-MI (Chevaleyre & Zucker, 2001), RIPPER-MI (Chevaleyre & Zucker, 2001), BP-MIP (Zhang & Zhou, 2004), and MI-SVM (Andrews et al., 2003). The diverse density algorithm searches for a point with the maximum diverse density in the feature space, which is defined as measuring how many different positive bags are nearby and how far the negative bags are from this point. Citation-KNN is a nearest-neighbor algorithm that measures the distance between bags by the minimal Hausdorff distance. ID3-MI is a decision tree algorithm that follows the divide-and-conquer strategy and uses MI entropy to distinguish bags instead of splitting instances. RIPPER-MI is a rule induction algorithm that uses MI coverage to follow the divide-and-conquer method for rule inducers. BP-MIP is a feedforward neural network algorithm that uses diverse density for feature scaling and PCA (principal component analysis; Jolliffe, 2002) for feature reduction. Basically, these MI learning algorithms are studied at bag level (i.e., discrimination on bags instead of discrimination on instances). More detailed information on the differences among these methods can be found in Zhou (2004). There are some survey papers and books on MI learning (Foulds & Frank, 2010; Amores, 2013; Herrera et al., 2016). Foulds and Frank classified MI learning methods according to the assumptions of each method. Amores (2013) looks into what level of information is

used in MI learning (i.e., instance level or bag level). However, dimensionality reduction in MI has not attracted much attention.

Many MI learning problems involve high-dimensional data: each instance has a large number of features. In addition, MI data may contain noisy and redundant features. Feature selection and dimension reduction are two ways to manage high-dimensional data. Feature selection attempts to select a subset of the original features according to some measurements. Dimensionality reduction attempts to identify a small set of features in a new feature space from the set of original features. HyDR-MI (Zafra, Pechenizkiy, & Ventura, 2013) is a feature selection method that uses the filter method to determine the important attributes in the feature space and then adopts the wrapper method to select the best feature subset. MIDR (Sun, Ng, & Zhou, 2010) aims to make the posterior probability of a truly positive bag to be one and zero otherwise. The projection matrix for dimensionality reduction was required to satisfy both sparsity and orthogonality. MidLABS (Ping, Xu, Ren, Chi, & Shen, 2010) uses the trace-ratio expression to simultaneously maximize between-class scattering and minimize within-class scattering. MidLABS constructs scattering matrices by directly evaluating the scattering among bags and takes the structure information of each bag by building an ϵ-graph. MIDA (Chai, Ding, Chen, & Li, 2014) uses the selected positive instance in each positive bag and the mean of all negative instances in all negative bags to construct scattering matrices and simultaneously maximizes the between-class scattering and minimizes the within-class scattering by solving the trace-difference formulation. CLFDA (Kim & Choi, 2010) prelabels all instances with their bag labels and then adopts neighborhood information to detect the false-positive ones. MidLABS, MIDA, and CLFDA can be treated as MI extensions of LDA (linear discriminant analysis; Fukunaga, 1990) using different scattering matrices.

1.2 Motivation. The main aim of this letter is to study MI dimensionality reduction by sparsity and orthogonality. The optimization problem of the MIDR method (Sun et al., 2010) is solved by the gradient descent method along the tangent space of the orthogonal matrices. In order to improve efficiency, sparsity and orthogonality constraints were further approximated in the method. Therefore, the calculated solution is not necessarily sparse and orthogonal, and learning performance may be affected. There is no proof of convergence of the MIDR algorithm; the computed solution cannot ensure that the optimality conditions are satisfied. Unlike Sun et al. (2010), we do not use any approximation for sparsity and orthogonality. Our idea is to formulate the optimization problem using the corresponding scaled augmented Lagrangian function with sparsity and orthogonality. The resulting Lagrangian function can be solved iteratively by updating the involved variables. In particular, variables associated with sparsity and orthogonal constraints can be treated by using the inertial proximal alternating linearized minimization (iPALM) method (Pock & Sabach, 2016; Zhu, 2016).

The advantage of this method is that these variables can be managed separately and updated effectively. We show the global convergence of the proposed algorithm combining the outer approximate augmented Lagrangian iteration step and the inner iPALM iteration step. The computed solution can be sparse and orthogonal, and it also satisfies the optimality conditions of the optimization problem. Experimental results on both synthetic and real data sets demonstrate that a good MI learning performance and both sparsity and orthogonality can be achieved, by the proposed algorithm.

The outline of this letter is as follows. In section 2, we review the MI dimensionality-reduction formula with sparsity and orthogonality and study the convergence of the proposed algorithm. In addition, we present the iPALM solver for the subproblems arising from the proposed algorithm. In section 3, we present experimental results to show the effectiveness of the proposed algorithm. Finally, we give some concluding remarks in section 4.

2 Dimensionality Reduction

In this section, we review the MIDR method proposed by Sun et al. (2010) and apply the approximate augmented Lagrangian method to solve the corresponding optimization problem.

2.1 The Optimization Problem. Let $\{(X_1, y_1), \ldots, (X_N, y_N)\}$ be the training data set, where $X_i = \{x_{i,1}, \ldots, x_{i,n_i}\} \subset \mathbb{R}^D$ is the ith bag, which contains n_i instances (n_i can vary across the bags), and $y_i \in \{0, 1\}$ is the label of X_i. Here $x_{i,j} \in \mathbb{R}^D$ denotes the jth instance in the ith bag, and its hidden label is $y_{i,j} \in \{0, 1\}$. Each instance contains D attributes. Under the standard assumption of MI learning, X_i has a label $y_i = 1$ and is said to be a positive bag if there exists at least one instance $x_{i,j} \in X_i$ with a label $y_{i,j} = 1$ (the concrete value of the index j is usually unknown). Otherwise, X_i is said to be a negative bag with a label $y_i = 0$.

In MIDR, Sun et al. (2010) studied the projection matrix $A \in \mathbb{R}^{D\times d}$ ($d \ll D$) to discriminate positive and negative bags: project $X_i \subset \mathbb{R}^D$ to $\{A^T x_{i,1}, \ldots, A^T x_{i,n_i}\} \subset \mathbb{R}^d$. A is required to be orthogonal to guarantee that the resulting features are uncorrelated and nonredundant in the new feature representation. Obviously each new feature $A^T x_{i,j} \in \mathbb{R}^d$ is a linear combination of all features in original data $x_{i,j} \in \mathbb{R}^D$, and the coefficients of such a linear combination are generally nonzero. Therefore, the importance of the original features and the interpretation of the features obtained in the lower-dimensional space may be difficult to be considered, especially when the data dimension D is large. To improve the ability of the model to interpret and visualize the results, $\|A\|_1 (= \sum_{i,j} |A_{ij}|)$ is incorporated into the new feature representation. Sparse representation of features for some real data sets has been studied and reported (see Dundar, Fung, Bi, Sathyakama, & Rao,

2005; Fung & Ng, 2007; Qiao, Zhou, & Huang, 2009; Ng, Liao, & Zhang, 2011) and references therein.

MIDR aims to solve the following optimization problem

$$\min_{A\in\mathbb{R}^{D\times d},\, w\in\mathbb{R}^d} \phi(A,w) := f(A,w) + \alpha\|A\|_1$$
$$\text{subject to} \quad A^TA = I_d, \tag{2.1}$$

where α is a positive number to control the balance between the fitting term $f(A,w)(=\sum_{i=1}^{N}(P_i(A,w)-y_i)^2)$ and the sparsity term $\|A\|_1$. Here,

$$P_i(A,w) = \text{softmax}_\beta(P_{i,1}(A,w),\ldots,P_{i,n_i}(A,w)) = \frac{\sum_{j=1}^{n_i} P_{i,j}(A,w)\exp(\beta P_{i,j}(A,w))}{\sum_{j=1}^{n_i}\exp(\beta P_{i,j}(A,w))}$$

is the softmax approximation of $\max_{1\le j\le n_i}\{P_{i,1}(A,w),\ldots,P_{i,n_i}(A,w)\}$ (when β is sufficient large, the formula would be a good approximation). The logistic regression parameter w was used to estimate the posterior probability $P_{i,j}$,

$$P_{i,j}(A,w) = \text{Prob}(y_{i,j}=1 \mid A^Tx_{i,j}) = \frac{1}{1+\exp(-w^TA^Tx_{i,j})}, \tag{2.2}$$

where $\text{Prob}(y_{i,j}=1 \mid A^Tx_{i,j})$ refers to the probability of $y_{i,j}=1$ under the condition $A^Tx_{i,j}$ is considered. Softmax approximation and posterior probability $P_{i,j}$ are usually employed in the multiple instance logistic regression (MILR) method (Ray & Craven, 2005).

2.2 Our Algorithm. Next, we study the approximate augmented Lagrangian method as used in (Zhu, Zhang, Chu, & Liao, 2017) to deal with problem 2.1. By introducing an auxiliary variable B, problem 2.1 can be rewritten as

$$\min_{A,B\in\mathbb{R}^{D\times d},\, w\in\mathbb{R}^d} \quad f(B,w) + \alpha\|B\|_1 + \delta(A) \tag{2.3}$$
$$\text{subject to} \quad A - B = 0,$$

where

$$\delta(A) = \begin{cases} 0, & \text{if } A^TA = I_d, \\ +\infty, & \text{otherwise,} \end{cases}$$

is the indicator function defined on the Stiefel manifold $\mathtt{St}(d,D) = \{A\in\mathbb{R}^{D\times d} \mid A^TA = I_d\}$.

Algorithm 1: MI-ALM.
Input: $\{\epsilon^{(k)}\}_{k\in\mathbb{N}} \downarrow 0$, $-\infty < \Lambda_{min} \le \Lambda_{max} < +\infty$ (pointwise), $\tau \in (0,1)$, $\mu > 1$, $k = 1$, $\rho^{(0)}$, $A^{(0)}$, $B^{(0)}$, $\Lambda_{min} \le \bar{\Lambda}^{(0)} \le \Lambda_{max}$ (pointwise), $\epsilon > 0$.
Output: $\{(A^{(k)}, B^{(k)}, w^{(k)}, \bar{\Lambda}^{(k)}, \rho^{(k)})\}_{k\in\mathbb{N}}$.
Step 0 Compute the logistic regression parameter $w^{(0)}$ using $B^{(0)}$.
Step 1 For given $\rho^{(k-1)}$ and $\bar{\Lambda}^{(k-1)}$, compute $(A^{(k)}, B^{(k)}, w^{(k)})$ such that $(A^{(k)})^T A^{(k)} = I_d$, and there exists $\xi^{(k)} \in \partial L_k(A^{(k)}, B^{(k)}, w^{(k)})$ satisfying

$$||\xi^{(k)}||_\infty \le \frac{\epsilon^{(k-1)}}{\rho^{(k-1)}}. \tag{2.5}$$

Step 2 Update the Lagrangian multiplier

$$\Lambda^{(k)} = \bar{\Lambda}^{(k-1)} + \rho^{(k-1)}(A^{(k)} - B^{(k)}),$$
$$\bar{\Lambda}^{(k)} = \mathcal{P}(\Lambda^{(k)}),$$

where $\mathcal{P}(\cdot)$ is the projection of a matrix on $\{\Lambda \mid \Lambda_{min} \le \Lambda \le \Lambda_{max}\}$.
Step 3 Update the penalty parameter

$$\rho^{(k)} = \begin{cases} \rho^{(k-1)}, & \text{if } ||A^{(k)} - B^{(k)}||_\infty \le \tau ||A^{(k-1)} - B^{(k-1)}||_\infty, \\ \mu\rho^{(k-1)}, & \text{otherwise.} \end{cases}$$

Step 4 If $\|B^{(k)} - B^{(k-1)}\|_\infty / \|B^{(k-1)}\|_\infty \le \varepsilon$, then Stop; otherwise, set $k \leftarrow k+1$ and go to **Step 1**.

The augmented Lagrangian function associated with equation 2.3 is given by

$$\tilde{L}(A, B, w, \Lambda) = f(B, w) + \alpha\|B\|_1 + \delta(A) + \langle \Lambda, A - B\rangle + \frac{\rho}{2}\|A - B\|_F^2,$$

where $\rho > 0$ is the penalty parameter and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product for matrices. In the following discussion, we consider the scaled form,

$$L(A, B, w, \Lambda) = \frac{1}{\rho}\tilde{L}(A, B, w, \Lambda). \tag{2.4}$$

There are four variables (A, B, w, and Λ) in equation 2.4. An iterative algorithm can be developed by updating each variable one by one based on the scaled augmented Lagrangian function. More precisely, we compute A, B, and w in the first step by fixing Λ, and then we update the Lagrange multiplier Λ by fixing A, B, and w. For given $\Lambda(= \bar{\Lambda}^{(k-1)})$, we denote $L_k(A, B, w) \triangleq L(A, B, w, \bar{\Lambda}^{(k-1)})$. Here, k refers to the iteration index. The outline of this iterative scheme is given in algorithm 1.

In step 1 of algorithm 1, we would like to find a vector $\xi^{(k)} (\in \partial L_k)$ such that its norm is small enough (see equation 2.5 in the algorithm). We note

that

$$L_k(A, B, w) = \frac{1}{\rho^{(k-1)}} f(B, w) + \frac{\alpha}{\rho^{(k-1)}} \|B\|_1 + \frac{1}{\rho^{(k-1)}} \delta(A) + \frac{1}{\rho^{(k-1)}} \langle \bar{\Lambda}^{(k-1)}, A - B \rangle + \frac{1}{2} \|A - B\|_F^2, \tag{2.6}$$

and ∂L_k can be computed by

$$\partial L_k(A, B, w) = \begin{bmatrix} \partial_A L_k(A, B, w) \\ \partial_B L_k(A, B, w) \\ \partial_w L_k(A, B, w) \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho^{(k-1)}} \partial \delta(A) + \frac{\bar{\Lambda}^{(k-1)}}{\rho^{(k-1)}} + (A - B) \\ \frac{1}{\rho^{(k-1)}} \nabla_B f(B, w) + \frac{\alpha}{\rho^{(k-1)}} \partial \|B\|_1 - \frac{\bar{\Lambda}^{(k-1)}}{\rho^{(k-1)}} - (A - B) \\ \frac{1}{\rho^{(k-1)}} \nabla_w f(B, w)^T \end{bmatrix}. \tag{2.7}$$

Inequality 2.5 can be guaranteed by using the iPALM method (a detailed discussion is in section 2.3). Moreover, step 1 implies that the logistic regression parameter w is updated by minimizing the distance between the posterior probability $P_i(B, w)$ and the label y_i of the ith bag X_i. Step 2 updates the Lagrangian multipliers in the algorithm. The following theorem provides the convergence result of the MI-ALM algorithm.

Theorem 1. *Let $\{(A^{(k)}, B^{(k)}, w^{(k)})\}_{k\in\mathbb{N}}$ be a sequence generated by algorithm 1. If the sequence $\{(A^{(k)}, B^{(k)}, w^{(k)})\}_{k\in\mathbb{N}}$ is bounded, then any cluster point (A^*, B^*, w^*) of the sequence $\{(A^{(k)}, B^{(k)}, w^{(k)})\}_{k\in\mathbb{N}}$ satisfies the first-order optimality condition of problem 2.3. Moreover, (A^*, w^*) satisfies the first-order optimality condition of problem 2.1.*

The proof of theorem 1 is given in appendix A. Theorem 1 implies that the MI-ALM algorithm can provide a theoretically guaranteed optimal solution with the required sparsity and orthogonality, which is important to MI dimensionality reduction.

2.3 iPALM Algorithm. We note that step 1 is the crucial part of the MI-ALM algorithm, which requires that criterion 2.5 must be satisfied in each iteration. Next, we will show how the iPALM method can be employed to provide $(A^{(k)}, B^{(k)}, w^{(k)})$ such that equation 2.5 holds.

Algorithm 2: iPALM.
Input: $(A^{(k,0)}, B^{(k,0)}, w^{(k,0)}), (A^{(k,1)}, B^{(k,1)}, w^{(k,1)}) \in \mathbb{R}^{D\times d} \times \mathbb{R}^{D\times d} \times \mathbb{R}^d$.
Output: $\{(A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\}_{j\in\mathbb{N}}$.
For each $j = 1, 2, \cdots$,

$$A^{(k,j+1)} = \arg\min_A \left\{ \frac{\alpha^{(k)}}{\rho^{(k-1)}}\delta(A) + \alpha^{(k)}\langle A - A^{(k,j)}, \nabla_A H_k(A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\rangle + \frac{a^{(j)}}{2}\|A - A^{(k,j)}\|_F^2 + \beta^{(k)}\langle A, A^{(k,j-1)} - A^{(k,j)}\rangle \right\}, \tag{2.9}$$

$$B^{(k,j+1)} = \arg\min_B \left\{ \frac{\alpha\alpha^{(k)}}{\rho^{(k-1)}}\|B\|_1 + \alpha^{(k)}\langle B - B^{(k,j)}, \nabla_B H_k(A^{(k,j+1)}, B^{(k,j)}, w^{(k,j)})\rangle + \frac{b^{(j)}}{2}\|B - B^{(k,j)}\|_F^2 + \beta^{(k)}\langle B, B^{(k,j-1)} - B^{(k,j)}\rangle \right\}, \tag{2.10}$$

$$w^{(k,j+1)} = \arg\min_w \left\{ \alpha^{(k)}\langle w - w^{(k,j)}, \nabla_w H_k(A^{(k,j+1)}, B^{(k,j+1)}, w^{(k,j)})\rangle + \frac{c^{(j)}}{2}\|w - w^{(k,j)}\|_F^2 + \beta^{(k)}\langle w, w^{(k,j-1)} - w^{(k,j)}\rangle \right\}, \tag{2.11}$$

where for each $j \in \mathbb{N}$, $a^{(j)}$, $b^{(j)}$ and $c^{(j)}$ satisfy the following requirements:

$$a^{(j)} > \beta^{(k)}+\alpha^{(k)}, \quad b^{(j)} > \beta^{(k)}+l_k^1(A^{(k,j+1)}, w^{(k,j)})\alpha^{(k)} \quad c^{(j)} > \beta^{(k)}+l_k^2(B^{(k,j+1)})\alpha^{(k)},$$

here $l_k^1(A^{(k,j+1)}, w^{(k,j)})$ and $l_k^2(B^{(k,j+1)})$ (given in Appendix B) are the Lipschitz constants of $\nabla_B H_k(A^{(k,j+1)}, B, w^{(k,j)})$ and $\nabla_w H_k(A^{(k,j+1)}, B^{(k,j+1)}, w)$, respectively.

We rewrite formula 2.6 as follows:

$$L_k(A, B, w) = \frac{\alpha}{\rho^{(k-1)}}\|B\|_1 + \frac{1}{\rho^{(k-1)}}\delta(A) + H_k(A, B, w), \tag{2.8}$$

where

$$H_k(A, B, w) = \frac{f(B, w)}{\rho^{(k-1)}} + \langle\frac{\bar{\Lambda}^{(k-1)}}{\rho^{(k-1)}}, A - B\rangle + \frac{1}{2}\|A - B\|_F^2.$$

For each fixed k, the implementation of the iPALM method employed in step 1 of MI-ALM is listed in algorithm 2.

In iPALM algorithm, we set $(A^{(1,0)}, B^{(1,0)}, w^{(1,0)}) = (A^{(1,1)}, B^{(1,1)}, w^{(1,1)})$ randomly and $(A^{(k,0)}, B^{(k,0)}, w^{(k,0)}) = (A^{(k,1)}, B^{(k,1)}, w^{(k,1)}) = (A^{(k-1)}, B^{(k-1)}, w^{(k-1)})$ if $k > 1$ (i.e., we use the iteration at $(k-1)$th outer iteration as the initial guess for step 1 in algorithm 1). We note that the A-subproblem 2.9 is equivalent to

$$A^{(k,j+1)} = \arg\min_{A^TA=I_d} \frac{a^{(j)}}{2}\left\| A - \left[A^{(k,j)} - \frac{\alpha^{(k)}}{a^{(j)}}\nabla_A H_k(A^{(k,j)}, B^{(k,j)}, w^{(k,j)}) - \frac{\beta^{(k)}}{a^{(j)}}(A^{(k,j-1)} - A^{(k,j)}) \right] \right\|_F^2.$$

Therefore, $A^{(k,j+1)}$ can be computed via the singular value decomposition (Lai & Osher, 2014), $A^{(k,j+1)} = U^{(k,j)}(V^{(k,j)})^T$, where

$$U^{(k,j)}\Sigma^{(k,j)}(V^{(k,j)})^T = A^{(k,j)} - \frac{\alpha^{(k)}}{a^{(j)}}\nabla_A H_k\big(A^{(k,j)}, B^{(k,j)}, w^{(k,j)}\big) - \frac{\beta^{(k)}}{\alpha^{(j)}}\big(A^{(k,j-1)} - A^{(k,j)}\big).$$

B-subproblem (2.10) requires solving an l_1-norm proximal operator. Therefore, $B^{(k,j+1)}$ can be obtained by shrinkage operation (Boyd et al., 2011),

$$B^{(k,j+1)} = \text{shrink}\left(B^{(k,j)} - \frac{\alpha^{(k)}}{b^{(j)}}\nabla_B H_k(A^{(k,j+1)}, B^{(k,j)}, w^{(k,j)}) - \frac{\beta^{(k)}}{b^{(j)}}(B^{(k,j-1)} - B^{(k,j)}), \frac{\alpha\alpha^{(k)}}{\rho^{(k-1)}b^{(j)}}\right),$$

where $\text{shrink}(X, \eta) = \text{sign}(X) \odot \max\{|X| - \eta, 0\}$ is the soft-shrinkage operator and $\odot$ denotes the component-wise product.

We note that w-subproblem (2.11) is an unconstrained quadratic optimization problem, which means that $w^{(k,j+1)}$ can be solved exactly:

$$w^{(k,j+1)} = w^{(k,j)} - \frac{\alpha^{(k)}}{c^{(j)}}\nabla_w H_k(A^{(k,j+1)}, B^{(k,j+1)}, w^{(k,j)}) - \frac{\beta^{(k)}}{c^{(j)}}(w^{(k,j-1)} - w^{(k,j)}).$$

Moreover, we define

$$\xi_A^{(k,j+1)} = \left(1 - \frac{a^{(j)}}{\alpha^{(k)}}\right)(A^{(k,j+1)} - A^{(k,j)}) + \frac{\beta^{(k)}}{\alpha^{(k)}}(A^{(k,j)} - A^{(k,j-1)}) - (B^{(k,j+1)} - B^{(k,j)}), \tag{2.12}$$

$$\xi_B^{(k,j+1)} = \frac{1}{\rho^{(k-1)}}[\nabla_B f(B^{(k,j+1)}, w^{(k,j+1)}) - \nabla_B f(B^{(k,j)}, w^{(k,j)})] + \frac{\beta^{(k)}}{\alpha^{(k)}}(B^{(k,j)} - B^{(k,j-1)}) + (1 - \frac{b^{(j)}}{\alpha^{(k)}})(B^{(k,j+1)} - B^{(k,j)}), \tag{2.13}$$

$$\xi_w^{(k,j+1)} = \frac{1}{\rho^{(k-1)}}[\nabla_w f(B^{(k,j+1)}, w^{(k,j+1)}) - \nabla_w f(B^{(k,j+1)}, w^{(k,j)})], \tag{2.14}$$

and $\xi^{(k,j)} = (\xi_A^{k,j}, \xi_B^{k,j}, \xi_w^{k,j})$. The following theorem indicates that $\xi^{(k,j)}$ indeed satisfies criterion 2.5, which means that step 1 of MI-ALM is well defined with the iPALM method as the inner solver.

Theorem 2. *For each $k \geq 1$, let $\{(A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\}_{j\in\mathbb{N}}$ be a sequence generated by equations 2.9 to 2.11.*

i. $\xi^{(k,j+1)}$ defined in equations 2.12 to 2.14 satisfies

$$\xi^{(k,j+1)} \in \partial L_k(A^{(k,j+1)}, B^{(k,j+1)}, w^{(k,j+1)}), \ \forall j \in \mathbb{N}.$$

Moreover, $\|\xi^{(k,j)}\|_\infty \to 0$, as $j \to \infty$.

ii. The sequence $\{(A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\}_{j\in\mathbb{N}}$ has finite length:

$$\sum_{j=1}^{\infty} \|(A^{(k,j+1)}, B^{(k,j+1)}, w^{(k,j+1)}) - (A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\| < \infty.$$

Moreover, $\{(A^{(k,j)}, B^{(k,j)}, w^{(k,j)})\}_{j\in\mathbb{N}}$ converges to a critical point $(A^{(k,*)}, B^{(k,*)}, w^{(k,*)})$ of function $L_k(A, B, w)$.

The proof of theorem 2 is in appendix B.

In the next section, we test the performance of the proposed algorithm.

3 Experimental Results

We evaluate the effectiveness and efficiency of the MI-ALM algorithm on some synthetic data sets and five MI benchmark data sets.[1] In the following experiments, we set $\beta = 3$ as the approximate degree of softmax function, $\epsilon^{(k)} = 0.999^k$, $\tau = 0.99$, $\mu = 1.02$, $\bar{\Lambda}_{\min} = -10^2$, $\bar{\Lambda}_{\max} = 10^2$ in algorithm 1. For each $k \geq 1$, we set $a^{(j)} = \gamma_1(\alpha^{(k)} + \beta^{(k)})$, $b^{(j)} = \gamma_2(\alpha^{(k)}(2N\|w^{(k,j)}\| \max_{1\leq i\leq N} \sum_{u=1}^{n_i} \|x_{i,u}\|) + \beta^{(k)})$, and $c^{(j)} = \gamma_3\big(\alpha^{(k)} \frac{(|1-\beta|+\beta)\|B^{(k,j+1)}\|}{2} \sum_{i=1}^{N} \frac{\sum_{u=1}^{n_i} \|x_{iu}\|}{n_i} + \beta^{(k)}\big)$ for all $j \in \mathbb{N}$ in the iPALM method, where $\alpha^{(k)} \equiv 0.95$, $\beta^{(k)} \equiv 0.05$, and $\gamma_i = 1.01$, $i = 1, 2, 3$. All experiments were performed in Matlab R2013a on a MacBook Pro laptop with an Intel core i7 CPU at 2.2 GHz $\times$ 4 and 16 GB of RAM.

3.1 Synthetic Data Sets. Two experiments were carried out on synthetic data sets to verify the effectiveness of algorithm 1. In this section, we set $\alpha = 0.2$ in problem 2.1 for algorithm 1.

Figure 1 shows a simple two-dimensional synthetic example. Our goal is to reduce the dimensionality of the synthetic data from two to one. The structure in this test is shown in Table 1. There are four bags; the first three

[1] The code of the proposed MI-ALM algorithm can be found in www.math.hkbu.edu.hk/~mng/mi-alm.html.

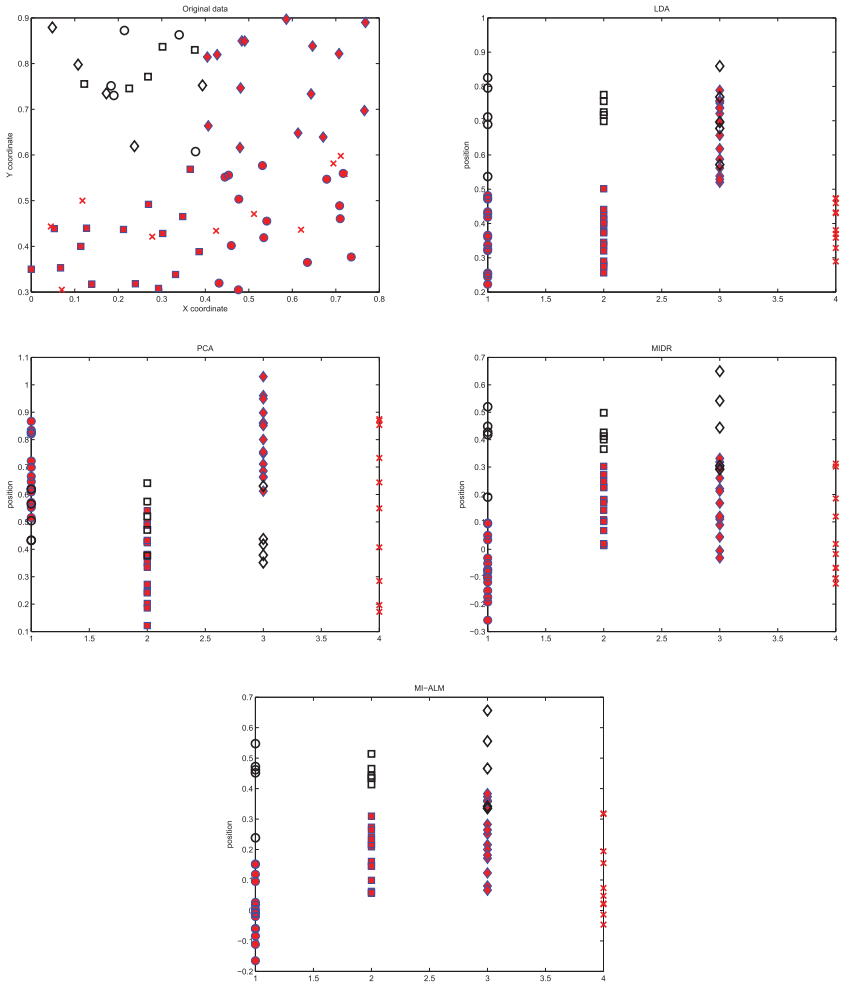

Figure 1: Original data, with the location of instances from different bags on the projected space: LDA, PCA, MIDR, and MI-ALM.

Table 1: Number of Negative Instances (Neg.) and Positive Instances (Pos.) in Each Bag.

	Bag 1		Bag 2		Bag 3		Bag 4	
Bags	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive
Figure 1	15	5	15	5	15	5	10	0
legend	●	○	■	□	◆	◇	×	

are positive, and the last one is negative. Figure 1 shows one-dimensional results after dimension reduction generated by methods LDA, PCA, MIDR, and MI-ALM. For MIDR, $\min_{A,H} \sum_i (P_i(A) - y_i)^2 + \frac{c_2}{2}\|A - H\|_F^2 + c_1\|H\|_1$ is used as an approximation of problem 2.1. Here, we set $c_1 = 0.5$, $c_2 = 20$ to keep A and H close. Figure 1 shows that LDA is misled by negative instances in the positive bags. After dimensionality reduction, the positive and negative instances in the positive bags 2 and 3 are very close. This is consistent with the fact that LDA, as a supervised learning algorithm, assigns each instance to the label of the bag. It can be seen from bag 4 in the PCA panel in Figure 1 that after dimensionality reduction, the positive and negative bags are not well separated. This can be explained: PCA does not consider label information as an unsupervised learning method. MIDR and MI-ALM both try to enlarge the distances between positive and negative instances to separate positive and negative bags. It can be seen in the figures that the performance of MIDR and MI-ALM are better than those of PCA and LDA.

Next, we generate a synthetic data set (data set II) in a high-dimensional feature space containing more bags and instances. Each data set has 80 bags (60 positive bags and 20 negative bags). Each positive bag contains 5 positive instances and 15 negative instances. Each negative bag contains 10 negative instances. There are D features in each instance, d relevant dimensions, and $D - d$ noisy (or irrelevant) dimensions. For a positive instance, the means of relevant dimensions are $[-4, 4, -4, 4, \ldots] \in \mathbb{R}^d$; for a negative instance, the means of relevant dimensions are $[4, -4, 4, -4, \ldots] \in \mathbb{R}^d$. The covariance matrices for relevant dimensions in both positive instances and negative instances are $2I_d$. The noisy dimensions follow a normal distribution with a mean of 0 and a standard deviation of 8. The d relevant dimensions were randomly selected among D dimensions. In this setting, we would like to reduce the dimension from D to d.

Table 2 reports the average results obtained by MI-ALM, MIDR, PCA, and LDA algorithms on 20 randomly generated data sets; their standard deviations are given in parentheses. We randomly selected 50% positive bags and 50% negative bags as training data and the remaining bags as testing data. The competitive algorithms are used only for dimensionality reduction, so an additional classifier is needed to evaluate classification performance. Here we use MILR solved by the BFGS method (Ray & Craven, 2005) as the classifier of these four methods. For comparison, we further tested MILR using the original features. The value of AUROC gives the classification accuracy based on the area under ROC (AUROC; Bradley, 1997; Fawcett, 2006) by constructing the receiver operating characteristic (ROC) curve. The value of sparsity refers to the ratio between the number of zero entries in the calculated solution A_c and its size. We note that when the sparsity value of each dimension approaches one (zero), the solution becomes sparser (denser). The values of orthogonality for MI-ALM and MIDR are calculated from the computed solution A_c: $\|A_c^T A_c - I\|_F/\sqrt{d}$. Table 3 shows

Table 2: Results Obtained by Different Algorithms for Data Set II.

		AUROC									
		MILR		MI-ALM		MIDR		PCA		LDA	
$D = 40$	0.2D	0.68	(1.71E-01)	0.84	(2.58E-01)	**0.98**	(6.49E-02)	0.79	(2.10E-01)	0.81	(2.14E-01)
	0.3D	0.82	(2.10E-01)	**0.96**	(1.15E-01)	**0.96**	(1.38E-01)	**0.96**	(7.72E-02)	0.82	(2.0E-01)
	0.4D	0.94	(9.25E-02)	0.95	(1.01E-01)	**1.00**	(0.00E+00)	0.92	(1.51E-01)	0.84	(1.6E-01)
	0.5D	0.96	(1.13E-01)	**1.00**	(0.00E+00)	**1.00**	(0.00E+00)	**1.00**	(0.00E+00)	0.89	(1.5E-01)
	0.6D	0.99	(2.52E-02)	**1.00**	(0.00E+00)	**1.00**	(1.58E-02)	0.99	(1.54E-02)	0.90	(9.4E-02)
D=100	0.2D	0.71	(2.37E-01)	**1.00**	(1.58E-02)	0.99	(2.64E-02)	0.93	(1.33E-01)	0.77	(9.35E-2)
	0.3D	0.96	(8.17E-02)	**0.99**	(3.27E-02)	0.96	(9.28E-02)	0.98	(4.96E-02)	0.77	(9.08E-02)
	0.4D	0.91	(1.67E-01)	**1.00**	(0.00E+00)	0.94	(1.86E-01)	0.99	(3.21E-02)	0.76	(1.10E-01)
	0.5D	0.95	(1.00E-01)	**1.00**	(0.00E+00)	0.99	(4.43E-02)	**1.00**	(1.26E-02)	0.78	(1.25E-01)
	0.6D	0.94	(1.33E-01)	**1.00**	(0.00E+00)	0.96	(1.41E-01)	**1.00**	(0.00E+00)	0.77	(1.03E-01)
$D = 500$	0.2D	0.76	(2.50E-01)	**0.97**	(5.02E-02)	0.96	(1.32E-01)	0.82	(3.34E-01)	0.56	(1.13E-01)
	0.3D	0.99	(2.71E-02)	**1.00**	(0.00E+00)	**1.00**	(0.00E+00)	0.74	(4.09E-01)	0.55	(1.37E-01)
	0.4D	0.95	(1.12E-01)	**1.00**	(0.00E+00)	0.99	(2.11E-02)	0.96	(6.27E-02)	0.54	(1.22E-01)
	0.5D	0.93	(1.87E-01)	**1.00**	(0.00E+00)	**1.00**	(0.00E+00)	0.94	(1.76E-01)	0.54	(1.04E-01)
	0.6D	0.92	(3.16E-03)	0.94	(1.99E-01)	0.81	(4.00E-01)	**1.00**	(0.00E+00)	0.54	(1.40E-01)

Notes: Values in parentheses are standard deviations. The best results are highlighted in bold.

Table 3: Sparsity and Orthogonality Results Obtained by MI-ALM and MIDR for Data Set II.

		Sparsity				Orthogonality				Error	
		MI-ALM		MIDR		MI-ALM		MIDR		MIDR	
D = 20	0.2D	0.87	(1.22E-01)	0.19	(2.99E-01)	5.12E-04	(5.82E-04)	5.82E-01	(6.63E-01)	3.63E-03	(2.04E-04)
	0.3D	0.94	(2.91E-02)	0.46	(3.20E-01)	2.53E-04	(2.59E-04)	8.64E-02	(1.89E-01)	3.54E-03	(1.07E-04)
	0.4D	0.92	(7.12E-02)	0.66	(3.06E-01)	3.95E-04	(3.97E-04)	9.41E-02	(1.89E-01)	3.67E-03	(5.45E-04)
	0.5D	0.94	(2.81E-02)	0.56	(2.31E-01)	2.15E-04	(2.58E-04)	4.22E-02	(2.50E-02)	3.50E-03	(4.30E-05)
	0.6D	0.94	(2.77E-02)	0.61	(1.00E-01)	3.78E-04	(3.51E-04)	5.75E-02	(1.16E-02)	4.23E-03	(1.48E-03)
D = 100	0.2D	0.90	(1.07E-01)	0.90	(8.35E-02)	1.15E-03	(1.13E-03)	2.53E-02	(1.00E-02)	9.79E-03	(8.89E-04)
	0.3D	0.80	(2.12E-01)	0.91	(3.67E-02)	1.64E-03	(4.83E-04)	2.65E-02	(6.75E-03)	9.69E-03	(6.47E-04)
	0.4D	0.91	(9.91E-02)	0.84	(4.71E-02)	1.33E-03	(1.30E-03)	3.51E-02	(7.26E-03)	9.63E-03	(2.28E-04)
	0.5D	0.83	(1.82E-01)	0.77	(6.80E-02)	1.85E-03	(1.09E-03)	4.30E-02	(5.26E-03)	9.38E-03	(1.89E-04)
	0.6D	0.91	(5.48E-02)	0.63	(2.93E-02)	1.60E-03	(7.07E-03)	5.68E-02	(6.95E-03)	1.04E-02	(2.09E-03)
D = 500	0.2D	0.95	(5.59E-02)	0.96	(3.47E-03)	1.29E-03	(1.52E-03)	9.78E-03	(1.28E-03)	1.23E-01	(8.61E-03)
	0.3D	0.96	(1.62E-02)	0.94	(1.10E-02)	7.50E-04	(1.97E-04)	1.22E-02	(2.37E-03)	1.19E-01	(1.04E-03)
	0.4D	0.96	(2.06E-02)	0.94	(3.86E-02)	1.44E-03	(9.51E-04)	1.61E-02	(1.66E-03)	1.23E-01	(1.56E-02)
	0.5D	0.93	(3.14E-02)	0.92	(6.42E-02)	1.76E-03	(7.36E-04)	2.48E-02	(7.14E-03)	1.18E-01	(5.33E-04)
	0.6D	0.96	(1.93E-02)	0.64	(3.19E-01)	1.85E-03	(1.40E-03)	1.02E-01	(1.41E-01)	1.18E-01	(3.05E-03)

Note: Values in the parentheses are standard deviations.

that the MIDR method provides sparse and orthogonal approximation, and it is not effective in dealing with sparsity and orthogonal constraints simultaneously. Notice that MIDR separates sparse and orthogonality constraints by introducing constraint $H = A$, requiring H to meet sparsity constraints and A to meet orthogonal constraints. In the objective function of MIDR, the penalty parameter is used to penalize the constraint $A = H$. Table 3 also reports the difference between A_c and H by calculating $\|A_c - H\|_F/\sqrt{d}$ ("Error") for reference. It can be seen that when $D = 500$, MIDR cannot obtain the high accuracy of constraints $A = H$. This is because the penalty method requires that the penalty parameter should be sufficiently large. However, the number of iterations required for solving such minimization problem is very large. Therefore, it is difficult to balance the computational trade-off by tuning the penalty parameter. In contrast, the proposed algorithm MI-ALM can provide sparse and orthogonal solutions well. Table 3 does not list the sparsity and orthogonality results of PCA and LDA because PCA and LDA do not provide sparse solutions, and their solutions satisfy orthogonality via the eigendecomposition procedure. We also remark that the upper bound of its subspace dimension is equal to the number of classes minus one in LDA setting. This is valid for binary classes (positive and negative labels) in multiple-instance learning.

According to the results of the mean values and standard deviations listed in Table 2, the classification performance of MI-ALM is quite competitive compared with MIDR, PCA, LDA, and MILR. We can also see from Table 2 that for each fixed D, MILR obtains improved accuracy as d (the number of related dimensions) increases. For most cases, PCA can achieve better accuracy than LDA and MILR. We find that LDA misclassified negative bags into positive bags, especially for large D. Table 2 demonstrates that the classification performance can be improved by dimension reduction.

3.2 Real Data Sets. In this section, we test MI-ALM algorithm on five MI benchmark data sets: Musk1, Musk2, Tiger, Elephant (Elep), and Fox.[2] Musk1 and Musk2 are drug activity prediction tasks and aim to predict whether a drug molecule can bind well to a target protein related to certain disease states, which is primarily determined by the low-energy shape of the molecule. A molecule can bind well if at least one of its shapes can bind well. Hence, MI learning can be used to learn the right shape that determines the binding and predict whether a new molecule can bind to the target protein by modeling a molecule as a bag and taking low-energy shapes as the instances. Tiger, Elephant, and Fox are image classification tasks. The image is considered positive when at least one segment of the image contains the desired animal. Therefore, by modeling an image as a bag and

[2] These five data sets can be downloaded from http://www.uco.es/grupos/kdis/momil/.

Table 4: Summary of the Structure of Five Benchmark Data Sets.

	Data Set	(D) Features	Bags			Instances			
			Positive	Negative	Total	Positive	Negative	Total	Average
Bioinformatics	Musk1	166	47	45	92	207	269	476	5.57
	Musk2	166	39	63	102	1017	5581	6598	64.69
Image classification and retrieval	Tiger	230	100	100	200	544	676	1220	6.69
	Elep	230	100	100	200	762	629	1391	6.10
	Fox	230	100	100	200	647	673	1320	6.60

modeling the image segment as an instance, MI learning can be applied to find out whether the image contains the required animals. A description of these five data sets is shown in Table 4. Before we perform multi-instance dimensionality reduction and learning for each data set, we employ Z-score standardization such that each feature of all instances has zero mean and standard deviation one.

In the following experiments, we repeat 10 times 10-fold cross validation with random partitions. More precisely, each data set is randomly partitioned into 10 samples. A single sample among 10 samples is retained as the testing data, and the remaining 9 samples are used as training data. Each of these 10 samples is used exactly once as the testing data. The whole process is then repeated 10 times. Then we compute the average performance of 10 random repetitions of different MI algorithms. Here, we compare the performance of MI-ALM with other MI learning methods: PCA, LDA, HyDR-MI, CLFDA, and MIDR. For the dimensionality-reduction methods PCA, CLFDA, MIDR, and MI-ALM, we reduce the dimensionality to (20%, 30%, 40%, 50%, 60%) of the size of original features. We use MILR as the classifier to evaluate the classification performance. We also test MILR without dimensionality reduction for reference. For HyDR-MI, we use the adapted Hausdorff distance to select the initial (15%, 30%, 45%, 60%, 75%, 90%) features of the original data in the filter component. We use AUROC as the evaluation criterion. For MI-ALM, we set $\alpha = 0.3$; for MIDR, we set $c_1 = 0.1$ and $c_2 = 10$ to make A and H close.

Table 5 reports AUROC values generated by each algorithm. It can be seen that in most of cases, the results obtained by dimension-reduction methods can be higher than those based on the original feature space (e.g., MILR). For convenience, we highlight the best result under dimensional reduction for each data set. We see that the learning performance of the MI-ALM algorithm is better than that of the other dimension-reduction algorithms.

The performance of LDA is not good in most cases since LDA misclassified negative bags as positive bags. It is worth noting that for image classification data sets, the performance of CLFDA is worse than that of LDA.

Table 5: Accuracy (AUROC) for Benchmark Data Sets.

	Dimension Reduction 10%					Dimension Reduction 20%					Dimension Reduction 30%				
Algorithm	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
PCA	**0.87**	0.84	**0.86**	**0.89**	0.62	0.82	0.88	0.86	0.88	**0.64**	0.82	**0.88**	0.85	**0.89**	**0.61**
CLFDA	0.77	0.83	0.84	0.86	0.57	0.79	0.85	**0.90**	0.88	0.56	**0.83**	0.86	0.87	**0.89**	0.54
MIDR	0.83	**0.87**	0.76	0.85	0.61	0.82	**0.89**	0.78	0.87	0.62	0.82	0.87	0.79	0.80	0.58
MI-ALM	0.85	0.83	**0.86**	**0.89**	**0.64**	**0.83**	0.88	0.87	**0.90**	0.62	**0.83**	**0.88**	**0.89**	0.87	0.59
	Dimension Reduction 40%					Dimension Reduction 50%					Dimension Reduction 60%				
	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
PCA	0.81	0.86	**0.87**	**0.88**	**0.59**	0.81	0.88	0.86	**0.87**	0.59	0.82	0.87	0.85	**0.88**	**0.58**
CLFDA	0.83	0.87	0.85	0.86	0.54	0.82	0.88	0.87	0.86	0.52	0.82	**0.89**	0.85	0.87	0.52
MIDR	0.82	**0.88**	0.81	0.83	0.55	0.80	**0.90**	0.80	0.83	0.53	**0.86**	0.88	0.81	0.87	0.54
MI-ALM	**0.85**	**0.88**	**0.87**	**0.88**	**0.59**	**0.86**	0.88	**0.92**	0.84	**0.60**	0.82	0.86	**0.89**	**0.88**	**0.58**
	Initial Selection 15%					Initial Selection 30%					Initial Selection 45%				
	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
HyDR-MI	0.75	0.78	0.87	0.86	0.54	0.87	0.89	0.87	0.85	0.58	0.80	0.88	0.88	0.87	0.60
	8.9%	9.5%	13.0%	11.3%	13.2%	15.7%	15.8%	23.2%	23.1%	22.7%	17.5%	24.9%	30.7%	33.4%	31.2%
	Initial Selection 60%					Initial Selection 75%					Initial Selection 90%				
	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
HyDR-MI	0.79	0.86	0.89	0.85	0.53	0.80	0.84	0.88	0.88	0.57	0.81	0.88	0.90	0.85	0.58
	24.2%	31.3%	43.8%	39.9%	44.7%	33.2%	39.3%	50.5%	50.8%	54.4%	41.1%	43.6%	66.9%	46.6%	58.2%
	Musk1	Musk2	Tiger	Elep	Fox										
MILR	0.83	0.88	0.90	0.90	0.55										
LDA	0.80	0.54	0.85	0.90	0.60										

Note: The best result is highlighted in bold, and the underlined result refers to the best result obtained by HyDR-MI for each data set.

CLFDA can be seen as supervised dimensionality reduction for multiple instance learning, which incorporates both citation and reference information to detect false-positive instance. In most cases, HyDR-MI can obtain better accuracy results than CLFDA and LDA. We list the percentages of features finally selected by HyDR-MI in Table 5 and underline the best results obtained by HyDR-MI for each data set. We see that the performance of the MI-ALM algorithm is better than that of the LDA, CLFDA, and HyDR-MI methods.

Sparsity results generated by MIDR and MI-ALM methods are reported in Table 6. Under the current parameter setting, the sparsity results by MI-ALM are better than MIDR except the Fox data set. The orthogonality results generated by MI-ALM are better than those by MIDR except the Musk2 data set. Table 6 also reports the error between A_c and H (Error) by MIDR algorithm. MI-ALM does not have this issue, and it provides solution A directly. These results demonstrate that sparse and orthogonal solution can provide better projection matrix to discriminate positive and negative bags. The important result is that the MI-ALM algorithm can provide better learning performance via sparsity and orthogonality than the MIDR algorithm does.

4 Conclusion

In this letter, we have proposed an augmented Lagrangian method to deal with MI learning via sparsity and orthogonality. The subproblems arising from the augmented Lagrangian method are solved by the iPALM method. The convergence of the proposed algorithm is also given. The important result is that the algorithm can guarantee both sparsity and orthogonality constraints for solutions. The effectiveness of the proposed algorithm is verified by both synthetic and real data sets. The learning performance of the proposed algorithm is better than that of existing projected algorithms. As future research, we would like to extend the approach to MI multilearning problems by incorporating both sparsity and orthogonality constraints in finding better solutions in learning. The identification of features by sparsity projection matrices would be useful to deal with high-dimensional problems.

Appendix A: The Convergence of the MI-ALM algorithm

Before we study the convergence result of the MI-ALM algorithm (theorem 1), we discuss the optimality condition of problem (2.3).

The Lagrangian function of problem (2.3) is given by

$$\mathcal{L}(A, B, w) = f(B, w) + \alpha\|B\|_1 + \langle \Lambda, A - B\rangle + \langle \Gamma, A^T A - I_d\rangle,$$

Table 6: Results of Sparsity, Orthogonality, and Error by MIDR and MI-ALM for Benchmark Data Sets.

		Dimension Reduction 10%					Dimension Reduction 20%				
	Algorithm	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
Sparsity	MIDR	0.581	0.584	0.814	0.808	**0.848**	0.693	0.669	0.861	0.835	**0.868**
	MI-ALM	**0.895**	**0.593**	**0.817**	**0.848**	0.741	**0.945**	**0.789**	**0.910**	**0.892**	0.788
Orthogonality	MIDR	2.1E-03	1.1E-01	3.4E-03	1.3E-02	5.9E-02	3.6E-02	4.8E-03	6.2E-03	5.5E-03	4.7E-02
	MI-ALM	**1.1E-03**	**1.5E-03**	**2.1E-03**	**1.7E-03**	**1.4E-03**	**8.4E-04**	**1.9E-03**	**1.7E-03**	**2.3E-03**	**2.8E-03**
Error	MIDR	9.9E-02	9.8E-02	7.9E-02	7.9E-02	7.1E-02	8.9E-02	9.3E-02	7.0E-02	7.4E-02	7.0E-02
		Dimension Reduction 30%					Dimension Reduction 40%				
	Algorithm	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
Sparsity	MIDR	0.720	0.674	0.904	0.871	**0.904**	0.772	0.696	0.898	0.864	**0.897**
	MI-ALM	**0.944**	**0.848**	**0.917**	**0.910**	0.827	**0.959**	**0.851**	**0.922**	**0.916**	0.868
Orthogonality	MIDR	4.8E-02	**1.7E-03**	1.4E-02	8.0E-03	1.9E-02	6.1E-02	**2.0E-03**	7.4E-03	1.7E-01	1.3E-02
	MI-ALM	**1.1E-03**	1.8E-03	**1.9E-03**	**2.8E-03**	**3.2E-03**	**8.5E-04**	2.2E-03	**2.1E-03**	**2.4E-03**	**2.7E-03**
Error	MIDR	8.7E-02	9.4E-02	5.8E-02	6.5E-02	6.0E-02	7.9E-02	9.2E-02	6.1E-02	6.7E-02	6.3E-02
		Dimension Reduction 50%					Dimension Reduction 60%				
	Algorithm	Musk1	Musk2	Tiger	Elep	Fox	Musk1	Musk2	Tiger	Elep	Fox
Sparsity	MIDR	0.803	0.713	0.906	0.880	**0.906**	0.850	0.710	0.914	0.896	**0.922**
	MI-ALM	**0.963**	**0.797**	**0.930**	**0.922**	0.876	**0.965**	**0.884**	**0.933**	**0.916**	0.901
Orthogonality	MIDR	6.0E-02	**2.6E-03**	1.1E-01	4.9E-03	4.4E-02	5.1E-02	2.6E-03	5.0E-03	7.6E-03	7.2E-02
	MI-ALM	**1.0E-03**	2.7E-03	**2.2E-03**	**2.4E-03**	**3.2E-03**	**1.0E-03**	**2.1E-03**	**2.1E-03**	**2.6E-03**	**2.9E-03**
Error	MIDR	7.5E-02	9.0E-02	6.0E-02	6.3E-02	6.1E-02	6.5E-02	9.2E-02	5.7E-02	5.8E-02	5.5E-02

Note: The best result is highlighted in bold.

where $\Gamma \in \mathbb{R}^{d\times d}$ is the Lagrange multiplier refering to the orthogonal constraints. The first-order optimality condition of problem 2.3 can be described by the following lemma. The proof is similar to lemma 2 in Zhu et al. (2017). We prove it here for completeness.

Lemma 1. *Suppose that* (A^*, B^*, w^*) *is a stationary point of problem 2.3. Then there exist* $\Lambda^* \in \mathbb{R}^{D\times d}$, $\Gamma^* \in \mathbb{R}^{d\times d}$, $(\Gamma^*)^T = \Gamma^*$ *such that* $(A^*, B^*w^*; \Lambda^*, \Gamma^*)$ *satisfies the first-order optimality conditions of problem 2.3,*

$$\begin{bmatrix} \nabla_B f(B^*, w^*) + \alpha v^* \\ 0 \\ \nabla_w f(B^*, w^*) \end{bmatrix} + \begin{bmatrix} -I_d & 0 \\ I_d & 2A^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Lambda^* \\ \Gamma^* \end{bmatrix} = 0, \tag{A.1}$$

where $v^* \in \partial\|B^*\|_1$, $A^* - B^* = 0$, *and* $(A^*)^T A^* = I_d$. *Furthermore,* $(A^*, w^*; \Gamma^*)$ *satisfies the first-order optimality conditions of problem 2.1:* $(A^*)^T A^* = I_d$ *and*

$$\nabla_A f(A^*, w^*) + \alpha v^* + 2A^*\Gamma^* = 0, \quad \nabla_w f(A^*, w^*) = 0. \tag{A.2}$$

Proof of Lemma 1. The equalities $A^* - B^* = 0$ and $(A^*)^T A^* = I_d$ hold since (A^*, B^*, w^*) is feasible as a stationary point.

For analytical convenience, we denote $W := \begin{bmatrix} B \\ A \end{bmatrix} \in \mathbb{R}^{2D\times d}$, and define $g_0 : \mathbb{R}^{2D\times d} \to \mathbb{R}^{D\times d}$ as

$$g_0(W) = \begin{bmatrix} -I_D & I_D \end{bmatrix} W.$$

Let $\Omega = \{W \in \mathbb{R}^{2D\times d} \mid g_0(W) = 0\}$. Then problem 2.3 is equivalent to

$$\min_{W,w} \; f(B, w) + \alpha\|B\|_1 + \delta(A) + \delta_\Omega(W).$$

Notice that $\mathtt{St}(d, D)$ is a smooth manifold, Ω is a closed convex set, and (A^*, B^*, w^*) is a stationary point. Then by using the generalized Fermat rule (Clarke, Ledyaev, Stern, & Wolenski, 2008; Rockafellar & Wets, 2009), we have

$$0 \in \begin{bmatrix} \nabla_B f(B^*, w^*) + \alpha\partial\|B^*\|_1 \\ \partial\delta(A^*) \\ \nabla_w f(B^*, w^*) \end{bmatrix} + \begin{bmatrix} \partial\delta_\Omega(W^*) \\ 0 \end{bmatrix}.$$

Since

$$\nabla g_0(W^*) = \begin{bmatrix} -I_D \\ I_D \end{bmatrix}$$

has full column rank, $\partial\delta_\Omega(W^*)$ is the normal cone to Ω at W^*:

$$\partial\delta_\Omega(W^*) = N_\Omega(W^*) \equiv \left\{ \begin{bmatrix} -I_D \\ I_D \end{bmatrix} \Lambda \mid \Lambda \in \mathbb{R}^{D\times d} \right\}.$$

$\partial\delta(A^*)$ is the normal cone to $\mathtt{St}(d, D)$ at A^*:

$$\partial\delta(A^*) = N_{\mathtt{St}}(A^*) \equiv \{A^*S \mid S \in \mathbb{R}^{d\times d}, S^T = S\}.$$

Therefore, there exist $v^* \in \partial\|B^*\|_1$, $\Lambda^* \in \mathbb{R}^{D\times d}$, $\Gamma^* \in \mathbb{R}^{d\times d}$, $(\Gamma^*)^T = \Gamma^*$ such that

$$\begin{aligned} 0 &= \begin{bmatrix} \nabla_B^* f(B^*, w^*) + \alpha v^* \\ 2A^*\Gamma^* \\ \nabla_w f(B^*, w^*) \end{bmatrix} + \begin{bmatrix} -\Lambda^* \\ \Lambda^* \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \nabla_B f(B^*, w^*) + \alpha v^* \\ 0 \\ \nabla_w f(B^*, w^*) \end{bmatrix} + \begin{bmatrix} -I_d & 0 \\ I_d & 2A^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Lambda^* \\ \Gamma^* \end{bmatrix}, \end{aligned}$$

which proves equation (A.1). Moreover, it yields $\Lambda^* = -2A^*\Gamma^*$. By substituting this result into equation (A.1) and eliminating B^* by $A^* - B^* = 0$, we obtain equation (A.2). □

Next, we prove theorem 1, which implies that any cluster point generated by the ML-ALM algorithm satisfies the first-order optimality conditions in equation (A.1). In the analysis, we assume step 1 of MI-ALM is established.

Proof of Theorem 1. For any cluster point (A^*, B^*, w^*) of the bounded sequence $\{(A^{(k)}, B^{(k)}, w^{(k)})\}_{k\in\mathbb{N}}$, there exists an index set $\mathcal{K} \subset \mathbb{N}$ such that $\{(A^{(k)}, B^{(k)}, w^{(k)})\}_{k\in\mathcal{K}}$ converges to (A^*, B^*, w^*). To prove that (A^*, B^*, w^*) satisfies the first-order optimality condition, equation (A.1), we first show that it is a feasible point.

The equality $(A^*)^T A^* = I_d$ is easy to check since $(A^{(k)})^T A^{(k)} = I_d$ holds for any $k \in \mathbb{N}$.

If $\{\rho^{(k)}\}$ is bounded, then there exist a $k_0 \in \mathbb{N}$ such that $\|A^{(k)} - B^{(k)}\|_\infty \le \tau\|A^{(k-1)} - B^{(k-1)}\|_\infty$, $\forall k \ge k_0$ (by the updating rule of $\rho^{(k)}$ in MI-ALM). Hence, we have $A^* - B^* = 0$.

Now we assume $\{\rho^{(k)}\}$ is unbounded. By using the generalized Fermat rule and equation (2.7), finding a solution satisfying constraint (2.5) is

equivalent to calculating a point $(A^{(k)}, B^{(k)}, w^{(k)})$ such that

$$\left\| \frac{1}{\rho^{(k-1)}} \begin{bmatrix} \nabla_B f(B^{(k)}, w^{(k)}) + \alpha v^{(k)} \\ 0 \\ \nabla_w f(B^{(k)}, w^{(k)}) \end{bmatrix} + \begin{bmatrix} -I_d & 0 \\ I_d & 2A^{(k)} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\bar{\Lambda}^{(k-1)}}{\rho^{(k-1)}} + (A^{(k)} - B^{(k)}) \\ \frac{\Gamma^{(k)}}{\rho^{(k-1)}} \end{bmatrix} \right\|_\infty \leq \frac{\epsilon^{(k-1)}}{\rho^{(k-1)}}, \tag{A.3}$$

for some $v^{(k)} \in \partial \|B^{(k)}\|_1$ and $2A^{(k)}\Gamma^{(k)} \in \partial \delta(A^{(k)})$. We note that $\{\bar{\Lambda}^{(k)}\}$ is bounded in algorithm 1. $\{v^{(k)}\}_{k\in\mathcal{K}}$ is bounded by the convexity of $\|B\|_1$ (Bertsekas, 1999). Moreover,

$$\|\nabla_B f(B, w)\| \leq \frac{(1+\beta)\|w\|}{2} \sum_{i=1}^{N} \max_{1\leq j\leq n_i} \{\|x_{ij}\|\},$$

$$\|\nabla_w f(B, w)\| \leq \frac{(|1-\beta|+\beta)\|B\|}{2} \sum_{i=1}^{N} \frac{\sum_{j=1}^{n_i} \|x_{ij}\|}{n_i}.$$

Thus, $\{(\nabla_B f(B^{(k)}, w^{(k)}), \nabla_w f(B^{(k)}, w^{(k)}))\}_{k\in\mathcal{K}}$ is bounded. Let $k \in \mathcal{K}$ go to infinity, equation (A.3) implies that $A^* - B^* = 0$. Therefore, in both cases, we show that (A^*, B^*) is a feasible point.

Next we show that there exist $\Lambda^* \in \mathbb{R}^{D\times d}$, $\Gamma^* \in \mathbb{R}^{d\times d}$, $(\Gamma^*)^T = \Gamma^*$ such that $(A^*, B^*; \Lambda^*, \Gamma^*)$ satisfies equation (A.1). Since $\{v^{(k)}\}_{k\in\mathcal{K}}$ is bounded, there exists a subsequence $\mathcal{K}_2 \subseteq \mathcal{K}$ such that $\lim_{k\in\mathcal{K}_2} v^{(k)} = v^*$. Moreover, by the closedness property of the limiting Fréchet subdifferentiable (Mordukhovich, 2006), it follows that

$$v^* \in \partial \|B^*\|_1.$$

Combined with the updating formula of $\Lambda^{(k)}$ in step 2 of the MI-ALM algorithm (algorithm 1), equation (A.3) implies that there exists an $\xi^{(k)}$ with $\|\xi^{(k)}\|_\infty \leq \frac{\epsilon^{(k-1)}}{\rho^{(k-1)}}$ such that

$$\rho^{(k-1)}\xi^{(k)} = \frac{1}{\rho^{(k-1)}} \begin{pmatrix} \nabla f(B^{(k)}) + \alpha v^{(k)} \\ 0 \end{pmatrix} + \begin{bmatrix} -I_d & 0 \\ I_d & 2A^{(k)} \end{bmatrix} \begin{bmatrix} \Lambda^{(k)} \\ \Gamma^{(k)} \end{bmatrix}. \tag{A.4}$$

Let $\Xi^{(k)} = \begin{bmatrix} -I_d & 0 \\ I_d & 2A^{(k)} \end{bmatrix}$, $\Upsilon^{(k)} = \begin{bmatrix} \Lambda^{(k)} \\ \Gamma^{(k)} \end{bmatrix}$. Then $(\Xi^{(k)})^T(\Xi^{(k)})$ is nonsingular since the columns of $\Xi^{(k)}$ are linearly independent. Hence, we have

$$\Upsilon^{(k)} = ((\Xi^{(k)})^T \Xi^{(k)})^{-1}(\Xi^{(k)})^T \left(\rho^{(k-1)}\xi^{(k)} - \begin{bmatrix} \nabla_B f(B^{(k)}, w^{(k)}) + \alpha v^{(k)} \\ 0 \end{bmatrix} \right). \tag{A.5}$$

By taking a limit on equation (A.5) as $k \in \mathcal{K}_2$ goes to infinity and noticing that $\|\xi^{(k)}\|_\infty \leq \frac{\epsilon^{(k-1)}}{\rho^{(k-1)}}$ with $\epsilon^{(k)} \downarrow 0$ as $k \to \infty$, we have

$$\Upsilon^{(k)} \to \Upsilon^* := -((\Xi^*)^T \Xi^*)^{-1}(\Xi^*)^T \begin{bmatrix} \nabla_B f(B^*, w^*) + \alpha v^* \\ 0 \end{bmatrix},$$

where $\Xi^* = \begin{bmatrix} -I & 0 \\ I & 2A^* \end{bmatrix}$ has full column rank. From the definition of $\Upsilon^{(k)}$, taking limit $k \in \mathcal{K}_2$ goes to infinity on both sides of equation A.4 yields

$$0 = \begin{bmatrix} \nabla_B f(B^*, w^*) + \alpha v^* \\ 0 \end{bmatrix} + \begin{bmatrix} -I_d & 0 \\ I_d & 2A^* \end{bmatrix} \begin{bmatrix} \Lambda^* \\ \Gamma^* \end{bmatrix}.$$

Moreover, $(\Gamma^*)^T = \Gamma^*$ since $(\Gamma^{(k)})^T = \Gamma^{(k)}$ for any $k \in \mathbb{N}$. According to lemma 1, (A^*, B^*, w^*) satisfies the first-order optimality condition of problem 2.3. Moreover, (A^*, w^*) satisfies the first-order optimality condition of problem 2.1. The result follows.

We remark that $v^{(k)} \in \partial\|B^{(k)}\|_1$ and $2A^{(k)}\Gamma^{(k)} \in \partial\delta(A^{(k)})$ are just for theoretical analysis; $v^{(k)}$ and $\Gamma^{(k)}$ do not need to be calculated in implementation. □

Appendix B: Details and Analysis of iPALM Algorithm

The iPALM method was first proposed in Pock et al. (2016) and Zhu (2016) independently as an extension of the PALM method to solve optimization problems in the form

$$\min_{x,y} \; f(x) + g(y) + h(x, y).$$

An inertial term (e.g., $\langle A, A^{(k,j-1)} - A^{(k,j)} \rangle$ in equation 2.9) was introduced to improve the performance of the PALM method. Algorithm 2, which we used mainly, followed the update form proposed in Zhu (2016).

Before we study the convergence result of algorithm 2 (theorem 2), we first demonstrate that the scaled augmented Lagrangian function, equation 2.8, satisfies the following blanket assumptions for the iPALM method proposed in Zhu (2016).

Blanket Assumptions

1. $\|\cdot\|_1 : \mathbb{R}^{D\times d} \to (-\infty, +\infty]$ and $\delta : \mathbb{R}^{D\times d} \to (-\infty, +\infty]$ are proper and lower semicontinuous functions.
2. $H_k(\cdot,\cdot,\cdot) : \mathbb{R}^{D\times d} \times \mathbb{R}^{D\times d} \times \mathbb{R}^d \to (-\infty, +\infty]$ is a C^1 function.
3. For any fixed B and w, the function $A \to H_k(A, B, w)$ is $C_1^{1,1}$ (i.e., $\nabla_A H_k(A, B, w)$) is globally Lipschitz with moduli 1. Similarly, for any fixed A and w, the function $B \to H_k(A, B, w)$ is $C^{1,1}_{l_k^1(A,w)}$. For any fixed A and B, the function $w \to H_k(A, B, w)$ is $C^{1,1}_{l_k^2(B)}$.
4. There exist $\lambda_1^- > -\infty$ and $\lambda_1^+ < +\infty$ such that
$$\inf\{l_k^1(A, w) : k \in \mathbb{N}\} \geq \lambda_1^- \quad \text{and} \quad \sup\{l_k^1(A, w) : k \in \mathbb{N}\} \leq \lambda_1^+.$$
There also exist $\lambda_2^- > -\infty$ and $\lambda_2^+ < +\infty$ such that
$$\inf\{l_k^2(B) : k \in \mathbb{N}\} \geq \lambda_2^- \quad \text{and} \quad \sup\{l_k^2(B) : k \in \mathbb{N}\} \leq \lambda_2^+.$$
5. ∇H_k is Lipschitz continuous on bounded subsets of $\mathbb{R}^{D\times d} \times \mathbb{R}^{D\times d} \times \mathbb{R}^d$. For each bounded subset $\Omega_1 \times \Omega_2 \times \Omega_3$ of $\mathbb{R}^{D\times d} \times \mathbb{R}^{D\times d} \times \mathbb{R}^d$, there exists $\theta > 0$ such that for all $(A, B, w), (A', B', w') \in \Omega_1 \times \Omega_2 \times \Omega_3$:
$$\left\| \begin{bmatrix} \nabla_A H_k(A, B, w) - \nabla_A H_k(A', B', w') \\ \nabla_B H_k(A, B, w) - \nabla_B H_k(A', B', w') \\ \nabla_w H_k(A, B, w)^T - \nabla_w H_k(A', B', w')^T \end{bmatrix} \right\| \leq \theta \left\| \begin{bmatrix} A - A' \\ B - B' \\ w^T - (w')^T \end{bmatrix} \right\|.$$

Now we check that assumptions 1 to 4 hold. We note that assumptions 1 and 2 are trivial. For assumption 3,

$$\|\nabla_A H_k(A, B, w) - \nabla_A H_k(A', B, w)\| = \|A - A'\|, \quad \forall B, w.$$

Thus, for any fixed B and w, $\nabla_A H_k(A, B, w)$ is Lipschitz continuous with moduli 1. Moreover,

$$\nabla_B H_k(A, B, w) = \nabla_B f(B, w)/\rho^{(k-1)} - \bar{\Lambda}^{k-1}/\rho^{(k-1)} + B - A,$$
$$\nabla_w H_k(A, B, w) = \nabla_w f(B, w)/\rho^{(k-1)}.$$

By simple calculations, for any fixed A and w, $\nabla_B^2 f(B, w)$ is bounded. Thus, $\nabla_B H_k(A, B, w)$ is Lipschitz continuous with bounded moduli. Similarly, for any fixed A and B, $\nabla_w H_k(A, B, w)$ is Lipschitz continuous with bounded moduli. Then assumption 4 holds as a by-product. By the definition of

$H_k(A, B, w)$ and the boundedness of $\nabla_B f(B, w)$ and $\nabla_w f(B, w)$ on bounded subset $\Omega_1 \times \Omega_2 \times \Omega_3$, assumption 5 is valid.

Next, we prove theorem 2, which implies that step 1 in algorithm 1 is well defined with the iPALM method as a solver.

Proof of Theorem 2. We consider a fixed value of k. For simplicity, we set $V := (A, B, w)$ and $L_k(V) := L_k(A, B, w)$.

i. By the first-order optimality conditions of subproblems 2.9 to 2.11, it is easy to check that

$$\xi_A^{(k,j)} \in \partial_A L_k(V^{(k,j)}), \quad \xi_B^{(k,j)} \in \partial_B L_k(V^{(k,j)}) \quad \text{and} \quad \xi_w^{(k,j)} \in \partial_w L_k(V^{(k,j)}).$$

Since $H_k(V)$ is continuously differentiable, by subdifferentiability property,

$$\partial L_k(V) = \partial_A L_k(V) \times \partial_B L_k(V) \times \partial_w L_k(V),$$

which implies

$$\xi^{(k,j)} \in \partial L_k(V^{(k,j)}) = \partial L_k(A^{(k,j)}, B^{(k,j)}, w^{(k,j)}), \quad j \in \mathbb{N}.$$

To show $\|\xi^{(k,j)}\|_\infty \to 0$, we only need to verify that $\{V^{(k,j)}\}_{j\in\mathbb{N}}$ generated by equations 2.9 to 2.11 is bounded and use lemma 4.6 and theorem 4.3 in Zhu (2016) (or proposition 4.4 in Pock & Sabach (2016)). The boundedness of $\{V^{(k,j)}\}_{j\in\mathbb{N}}$ is proved by contradiction. Notice that for any $\rho^{(k-1)} > 0$, $\tilde{L}_k(V) = \rho^{(k-1)} L_k(V) = f(B, w) + \alpha\|B\|_1 + \delta(A) + \frac{\rho^{(k-1)}}{2}\|A - B + \bar{\Lambda}^{(k-1)}/\rho^{(k-1)}\|_F^2 - \frac{1}{2\rho^{(k-1)}}\|\bar{\Lambda}^{(k-1)}\|_F^2$ is a coercive function ($f(B, w)$ is bounded, $\alpha\|B\|_1$ and $\delta(A)$ are coercive, $\|A - B + \bar{\Lambda}^{(k-1)}/\rho^{(k-1)}\|_F^2 > 0$, $-\frac{1}{2\rho^{(k-1)}}\|\bar{\Lambda}^{(k-1)}\|_F^2$ is bounded from below since $\{\rho^{(k)}\}_{k\in\mathbb{N}}$ is nondecreasing). Suppose $\lim_{j\to\infty} \|V^{(k,j)}\|_\infty = +\infty$. Then there must hold

$$\lim_{j\to\infty} \tilde{L}_k(V^{(k,j)}) = +\infty.$$

We know from proposition 4.3 in Zhu (2016) that there exists $M > 0$ such that $\{L_k(V^{(k,j)}) + M\|V^{(k,j)} - V^{(k,j-1)}\|_F^2\}_{j\in\mathbb{N}}$ is a decreasing sequence, which implies that

$$\lim_{j\to\infty} \tilde{L}_k(V^{(k,j)}) < +\infty.$$

Hence, by contradiction argument, $\{V^{(k,j)}\}_{j\in\mathbb{N}}$ is bounded.

ii. From the proof of the previous point (i), we know that $\{V^{(k,j)}\}_{j\in\mathbb{N}}$ is bounded. Then by theorem 4.3 in Zhu (2016) (or theorem 4.1 in Pock

& Sabach, 2016), it remains to verify that $L_k(V)$ is a K-L function. Notice that

$$L_k(V) = \frac{1}{\rho^{(k-1)}} f(B, w) + \frac{\alpha}{\rho^{(k-1)}} \|B\|_1 + \frac{1}{\rho^{(k-1)}} \delta(A) + \frac{1}{2} \|A - B + \frac{\bar{\Lambda}^{(k-1)}}{\rho^{(k-1)}}\|_F^2, \tag{B.1}$$

where $f(B, w)$ satisfies KL properties since the exponential function is definable (Wilkie, 1996) and the composition of definable function is a definable. Therefore, $L_k(V)$ is definable in an o-minimal structure. The result holds directly.

□

Acknowledgments

The work of H.Z. was supported by NSF of China grant NSFC11701227, NSF of Jiangsu Province under project BK20170522, and Jiangsu University 17JDG013. The work of L.-Z. L. was supported in part by HKRGC GRF 12319816 and HKBU FRG2/15-16/058. The work of M.N. was supported in part by HKRGC GRF 12302715, 12306616, 12200317 and 12300218, HKBU RC-ICRS/16-17/03.

References

Amores, J. (2013). Multiple instance classification: Review, taxonomy and comparative Study. *Artificial Intelligence*, *201*, 81–105.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-Instance learning. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems* (pp. 577–584). Cambridge, MA: MIT Press.

Auer, P., Long, P. M., & Srinivasan, A. (1997). Approximating hyper-rectangles: Learning and pseudo-random sets. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (pp. 314–323). New York: ACM.

Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont, MA: Athena Scientific.

Blum, A., & Kalai, A. (1998). A note on learning from multiple-instance examples. *Machine Learning*, *30*, 23–29.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*, 1–122.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithm. *Pattern Recognition*, *30*, 1145–1159.

Chai, J., Ding, X., Chen, H., & Li, T. (2014). Multiple-instance discriminant analysis. *Pattern Recognition*, *47*, 2517–2531.

Chevaleyre, Y., & Zucker, J. D. (2001). Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. Application to the

mutagenesis problem. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 204–214). New York: Springer.

Clarke, F. H., Ledyaev, Y. S., Stern, R. J., & Wolenski, P. R. (2008). *Nonsmooth analysis and control theory*. New York: Springer.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Dundar, M. M., Fung, G., Bi, J., Sathyakama, S., & Rao, B. (2005). Sparse Fisher discriminant analysis for computer aided detection. In *Proceedings of the 2005 SIAM International Conference on Data Mining* (pp. 476–480). Philadelphia: SIAM.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*, 861–874.

Foulds, J. R., & Frank, E. (2010). A review of multi-instance learning assumptions. *Knowledge Engineering Review*, *25*, 1–25.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Orlando, FL: Academic Press.

Fung, E. S., & Ng, M. K. (2007). On sparse Fisher discriminant method for microarray data analysis. *Bioinformation*, *2*, 230–234.

Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D., & Vluymans, S. L. (2016). *Multiple instance learning: Foundations and algorithms*. Cham, Switzerland: Springer.

Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). New York: Wiley.

Kim, S., & Choi, S. (2010). Local dimensionality reduction for multiple instance learning. In *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing* (pp. 13–18). Piscataway, NJ: IEEE.

Lai, R. J., & Osher, S. (2014). A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, *58*, 431–449.

Long, P. M., & Tan, L. (1998). PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, *30*, 7–21.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. M. I. Jordan, M. J. Kearns, & S. A, Solla (Eds.), In *Advances in neural information processing systems*, *10* (pp. 570–576). Cambridge, MA: MIT Press.

Mordukhovich, B. S. (2006). *Variational analysis and Generalized Differentiation I: Basic Theory*. New York: Springer.

Ng, M. K., Liao, L. Z., & Zhang, L. (2011). On sparse linear discriminant analysis algorithm for high-dimensional data classification. *Numerical Linear Algebra with Applications*, *18*, 223–235.

Ping, W., Xu, Y., Ren, K., Chi, C., & Shen, F. (2010). Non-I.I.D. multi-instance dimensionality reduction by learning a maximum bag margin subspace. In *Proceedings of the 24th AAAI National Conference on Artificial Intelligence* (pp. 551–556). Menlo Park, CA: AAAI.

Pock, T., & Sabach, S. (2016). Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, *9*, 1756–1787.

Qiao, Z., Zhou, L., & Huang, J. Z. (2009). Sparse linear discriminant analysis with applications to high dimensional low sample size data. *International Journal of Applied Mathematics*, *39*, 48–60.

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the International Conference on Machine Learning* (pp. 697–704). New York: ACM.

Rockafellar, R. T., & Wets, R. J. B. (2009). *Variational analysis*. New York: Springer.

Sun, Y. Y., Ng, M. K., & Zhou, Z. H. (2010). Multi-instance dimensionality reduction. *In Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI.

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*, 137–154.

Wang, J., & Zucker, J. D. (2000). Solving multiple-instance problem: A lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning* (pp. 1119–1125). San Mateo, CA: Morgan Kaufmann.

Wilkie, A. J. (1996). Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, *9*, 1051–1094.

Zafra, A., Pechenizkiy, M., & Ventura, S. (2013). HyDR-MI: A hybrid algorithm to reduce dimensionality in multiple instance learning. *Information Sciences*, *222*, 282–301.

Zhang, M.-L., & Zhou, Z. H. (2004). Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, *19*, 1–10.

Zhou, Z. H. (2004). *Multi-instance learning: A Survey* (Technical Report). Department of Computer Science and Technology, Nanjing University.

Zhu, H. (2016). Solving optimization problems with generalized orthogonality constraints. Ph.D. diss., Hong Kong Baptist University.

Zhu, H., Zhang, X., Chu, D., & Liao, L. Z. (2017). Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method. *Journal of Scientific Computing*, *72*, 1–42.

Received August 25, 2017; accepted August 8, 2018.