Fragmentary Multi-Instance Classification

Jie Wu, Wenzhang Zhuge, Xinwang Liu, Li Liu, *Senior Member, IEEE*, and Chenping Hou, *Member, IEEE*

***Abstract*—Multi-instance learning (MIL) has been extensively applied to various real tasks involving objects with bags of instances, such as in drugs and images. Previous studies on MIL assume that data are entirely complete. However, in many real tasks, the instance is fragmentary. In this article, we present probably the first study on multi-instance classification with fragmentary data. In our proposed framework, called fragmentary multi-instance classification (FIC), the fragmentary data are completed and the multi-instance classifier is learned jointly. To facilitate the integration between the completion and classifier learning, FIC establishes the weighting mechanism to measure the importance levels of different instances. To validate the compatibility of our framework, four typical MIL methods, including multi-instance support vector machine (MI-SVM), expectation maximization diverse density (EM-DD), citation-*K* nearest neighbors (Citation-KNNs), and MIL with discriminative bag mapping (MILDM), are embedded into the framework to obtain the corresponding FIC versions. As an illustration, an efficient solving algorithm is developed to address the problem for MI-SVM, together with the proof of convergence behavior. The experimental results on various types of real-world datasets demonstrate the effectiveness.**

***Index Terms*—Fragmentary data, multi-instance learning (MIL), weighting mechanism.**

I. Introduction

THE MULTI-INSTANCE learning (MIL) model is a generalization of supervised classification in which each training example is a bag of instances, instead of a single instance [1]. Standard supervised learning, in which each bag contains a single instance, can be regarded as a special case of MIL [2]. There is only one single label for each bag in MIL and it depends on the maximum label among all instances in the bag, where the maximum label refers to the maximum value of the labels of all the instances in a bag. In the multi-instance classification task, a bag is positive if there is at least one positive instance in it. A bag is negative if all the member instances are negative [3]. The goal of MIL is to learn a classifier from the labeled bags of instances, which can predict the labels of unseen bags based on their member instances. MIL is receiving growing attentions in the machine-learning field [4]–[6].

MIL roots from many real applications. It is originally introduced to solve the drug activity prediction problem, where each molecule contains many possible conformations [3]. Each molecule is viewed as a bag and the conformations in it are viewed as its member instances. The efficacy of a molecule can be tested experimentally, but that of each individual conformation cannot be identified. A molecule is active if there are at least one of its conformations binding to the target protein and inactive otherwise. Dietterich *et al.* [3] solved this problem by learning axis-parallel rectangles (APRs). The second application is region-based image classification and retrieval [1], [6]–[8]. An image is divided into several regions and each region has different contents. Each image can be viewed as a bag of local image regions. When searching for the images with a predefined semantics meaning from the pool of images, an image is selected if at least one of its member regions possess that semantics meaning. For retrieval, an image receives a certain label if at least one of its instances possesses the label. The third application is the video recognition [6]. Each video scene is viewed as a bag of shots via shot detection. The key frame of each shot is extracted as an instance of the scene. Other application problems such as text categorization [1] and Web mining [9] can also be solved efficiently with the MIL model.

Different from the traditional supervised classification setting, only bag labels are available in MIL and each bag contains multiple instances. Besides, the numbers of instances may be different among various bags. These characters make it hard to deal with the MIL problem. Existing methods can be roughly divided into three categories: 1) modifying methods [1], [9], [10]; 2) redesigning methods [2], [11], [12]; and 3) deep MIL methods [13]–[15]. The first category includes modifying methods that modify the single instance algorithm to fit the multi-instance problem. Multi-instance support vector machine (MI-SVM) [1], a representative method, extends SVM [16] to MIL by finding the most positive instance in each bag to represent the bag. Here, the most positive instance refers to the one which is most likely to be positive. It is farthest from the interval in the positive area. The second category includes the redesigning methods that design new models to

Manuscript received May 14, 2019; revised August 6, 2019 and August 20, 2019; accepted August 22, 2019. This work was supported in part by the NSF of China under Grant 61922087 and Grant 61906201, and in part by the NSF for Distinguished Young Scholars of Hunan Province under Grant 2019JJ20020. This article was recommended by Associate Editor R. Tagliaferri. *(Corresponding author: Chenping Hou.)*

J. Wu, W. Zhuge, and C. Hou are with the Department of Systems Science, National University of Defense Technology, Changsha 410073, China (e-mail: wujienudt@yahoo.com; zgwznudt@yeah.net; hcpnudt@hotmail.com).

X. Liu is with the School of Computer Science, National University of Defense Technology, Changsha 410073, China (e-mail: xinwangliu@nudt.edu.cn).

L. Liu was with the Center for Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland. She is now with the College of Systems Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: liuli_nudt@nudt.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2938206

solve multi-instance problem directly. For example, MIL with discriminative bag mapping (MILDM) [12] uses bag mapping to transform a bag into a single instance in a new space via instance selection. It aims to identify the best instances to directly distinguish bags in the new mapping space. Recently, with the combination of MIL and deep learning, methods in the third category have significantly improved the prediction accuracy. The difference among these deep MIL methods is the variation in constructing the neural networks according to different combination mechanisms between deep learning and MIL. For example, the attention method [14] learns the Bernoulli distribution of the bag labels parameterized by neural networks. The multiple instance neural networks (MINNs) [13] construct MINN to learn bag representations. In the deep multiple instance learning (DMIL)-based spatial–spectral classification for PAN and MS imagery method [15], an end-to-end learning framework based on deep multiple instance learning is proposed using the joint spectral and spatial information based on the feature fusion.

Although the above-mentioned MIL methods have achieved prominent performance, they all assume that the data are entirely complete. Nevertheless, in some real applications, due to the environmental disturbances, detector faults, or transmission distortions [17], [18], the collected instances are often fragmentary. For example, in image and sound capture, the shelter and shadow can be regarded as environmental disturbances, which makes the data fragmentary. In video classification, captured video data may also be contaminated or incomplete. In drug activity prediction, drug molecules may be shielded from each other, which makes the detection instrument unable to locate the edge in some direction. Other applications such as text categorization and Web mining may also suffer from the data missing problem [19].

The fragmentary character of data prevents the direct utility of above-mentioned MIL methods. There are two surrogate ways to apply existing multi-instance classification methods. The first strategy is deleting, that is, removing the examples with missing attributes. It contradicts the target of classification, that is, learning the classifier from all examples. The second strategy is completion, that is, filling in the missing attribute with matrix completion (MC) approaches divided into two categories: 1) simple value completion and 2) complex estimation. The simple value completion methods include constant (like 0, etc.) completion and statistics (like feature average, etc.) completion. Complex estimation uses learning strategies, such as regression estimation [20], [21]; expectation maximization [22]; decision tree prediction; Bayesian prediction [23]; and low-rank completion [24], [25]. Different complex estimation methods have different assumptions. For example, low-rank completion assumes that the incomplete matrix is not full rank since its rows (or columns) are often correlated in many real applications, such as recommendation or rating systems [24], [25]. Although traditional MC methods make MIL methods applicable, they often neglect the requirement of following multi-instance classification. The completion and multi-instance classification are conducted separately and the complementarity between them is not fully considered. The ideal case is that the completion strategy takes the requirement of multi-instance classification into consideration and the multi-instance classifier is also aware of the completeness of fragmentary instances. Besides, there are also some researches about feature missing problem in multiview learning [26], where a single data is characterized by more than one kind of descriptions. Our setting is totally different from it since we studied a feature missing problem in MIL.

In this article, we have proposed the fragmentary multi-instance classification (FIC) framework to handle such fragmentary multi-instance data, which completes data and learns MIL classifier simultaneously. Considering that various instances have different levels of importance in MIL, in our method, more positive instances have been given larger weights in completing. To verify the compatibility and efficiency of our framework, we have embedded four typical MIL methods into this framework to obtain the corresponding FIC models, that is, fragmentary MI-SVM (F-MI-SVM), fragmentary expectation maximization diverse density (F-EM-DD), fragmentary citation-KNN (F-C-KNN), and fragmentary aMILGDM (F-aMILGDM). Since the meanings of more positive in different MIL methods are different, we have designed different weight functions. We have also extended the augmented Lagrange multiplier (ALM) method to solve the proposed problem within our framework in an efficient way. Together with the provable convergence for F-MI-SVM model, comparison of experimental results on various types of benchmark datasets have indicated that our FIC framework can improve the performances of all the four traditional MIL methods. The contributions of this article are as follows.

1) We have proposed a unified framework for MIL with fragmentary data, which can jointly complete the fragmentary data and learn the multi-instance classifier by considering the measurement of instance's importance.
2) The proposed framework is compatible with different MIL methods. We have derived different weight functions for different FIC models based on the definition of positiveness in different MIL methods.
3) As an illustration, we have developed an efficient algorithm with provable convergence to solve our formulated F-MI-SVM problem.

The remainder of this article is organized as follows. Section II provides the notations and related works. Section III presents the proposed framework and its optimization. Section IV provides our F-MI-SVM model and the algorithm. Section V analyzes the convergence and complexity. Section VI provides the comparison results on various kinds of datasets and Section VII provides the concluding remarks.

II. Related Work

A. Notations

Denote $\mathbf{D} = \{\langle \mathbf{B}_1, y_1\rangle, \ldots, \langle \mathbf{B}_N, y_N\rangle\}$ as the labeled dataset. It contains a set of N bags and the corresponding labels. Here, bag $\mathbf{B}_I = \{\mathbf{B}_{I1}, \ldots, \mathbf{B}_{Ij}, \ldots, \mathbf{B}_{IN_I}\}$, with $\mathbf{B}_{Ij}$ denotes the jth instance in bag $\mathbf{B}_I$. The labels of the instances in $\mathbf{B}_I$ are $y_{I1}, \ldots, y_{Ij}, \ldots, y_{IN_I}$, which cannot be obtained. $y_I = y_{I1} \vee \ldots \vee y_{IN_I}$ for Boolean labels and $y_I = \max\{y_{I1}, \ldots, y_{IN_I}\}$

TABLE I
NOTATIONS

Notations	Descriptions
$\mathbf{D}$	Labeled Dataset
$\mathbf{B}_I$	The Ith bag
$\boldsymbol{B}_I^{\phi}$	A single instance corresponding to $\mathbf{B}_I$ in a new feature space
$\mathbf{B}_{Ij}$	The jth instance in bag $\mathbf{B}_I$
N_I	The number of instances in $\mathbf{B}_I$
N	The number of bags
n	The number of all instances
d	Common dimensionality of all instances
$\mathbf{X} \in \mathbb{R}^{d\times n}$	The learned completed data matrix
Ω	The set of indexes of observed elements
$\mathbf{X}_\Omega$	The set of the elements in the observed positions of $\mathbf{X}$
$\mathbf{T} \in \mathbb{R}^{d\times n}$	Observed data matrix
$\hat{\mathbf{X}} \in \mathbb{R}^{d\times N}$	The matrix consisting of the key instances of all bags
$\boldsymbol{x}_i(\boldsymbol{x}_j) \in \mathbb{R}^d$	The ith (jth) column of $\mathbf{X}$
$\hat{\boldsymbol{x}}_I \in \mathbb{R}^d$	The key instance of the Ith bag
$\hat{y}_I$	The label of key instance $\hat{\boldsymbol{x}}_I$
y_I	The label of bag $\mathbf{B}_I$
δ	A parameter to adjust the weight
$\boldsymbol{h} \in \mathbb{R}^d$	Concept point
$\boldsymbol{P}_{center} \in \mathbb{R}^d$	The center point of all positive instances
$\boldsymbol{P}_{center}^{\phi} \in \mathbb{R}^m$	The center point of all positive instances in the new feature space
$\boldsymbol{s} \in \mathbb{R}^n$	The ancillary variable introduced to optimize

for real-value labels. N_I is the number of instances in $\mathbf{B}_I$. n is the total number of instances of all bags, that is, $n = \sum_{i=1}^{N} N_i$. $\mathbf{X} \in \mathbb{R}^{d\times n}$ denotes the completed data matrix which we want to learn. We simply stack all the instances of all bags to construct the data matrix $\mathbf{X}$, that is, $\mathbf{X} = [\mathbf{B}_{11}, \ldots, \mathbf{B}_{1N_1}, \mathbf{B}_{21}, \ldots, \mathbf{B}_{2N_2}, \ldots, \mathbf{B}_{N1}, \ldots, \mathbf{B}_{NN_N}]$. Thus, each column of $\mathbf{X}$ is an instance and each row is a feature. For simplicity, denote $\boldsymbol{x}_i \in \mathbb{R}^d (i = 1, 2, \ldots, n)$ as the ith column of $\mathbf{X}$ and $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$. Denote $\mathbf{T} \in \mathbb{R}^{d\times n}$ as the observed data matrix, in which all the missing indices have no elements. Denote Ω as the set of indices of the available elements and $\mathbf{X}_\Omega$ as the observed elements of $\mathbf{X}$. Assume that $\hat{\mathbf{X}} \in \mathbb{R}^{d\times N}$ consists of the key instances of all bags. There is only one key instance in each bag, which refers to the most positive instance and best represents the bag. Denote $\hat{\boldsymbol{x}}_I \in \mathbb{R}^d (I = 1, 2, \ldots, N)$ as the key instance of the Ith bag. Thus, $\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_N]$. Denoting $\hat{y}_I$ as the label of key instance $\hat{\boldsymbol{x}}_I$. It is also the label of the Ith bag. Denoting $\boldsymbol{h} \in \mathbb{R}^d$ as the concept point (the point with the highest density of all the instances in a positive bag) and $\boldsymbol{P}_{\text{center}} \in \mathbb{R}^d$ as the center point of all positive instances. In summary, the notations are listed in Table I and we will explain its concrete meaning when it is first used.

B. MI-SVM

MI-SVM [1] is an extension of SVM [16], which leads to a mixed-integer quadratic program (QP) that can be solved heuristically. It is based on the fact that the margin is defined by the distance of the most positive instance (witness) in each bag. In other words, once these witness instances have been identified, the other instances in each bag become irrelevant to the classification boundary.

MI-SVM can be cast as a mixed-integer program. In deriving the optimization heuristics, for the given selector variables, that is, the key instance of each bag, the problem reduces to a QP. It alternates the following two steps: 1) for the given key instances, solving the associated QP and finding the optimal discriminant and 2) for the given discriminant, updating the key instance in each bag that is the most positive. With this alternation, MI-SVM has achieved competitive results.

C. EM-DD

EM-DD [2] is a general-purpose MI learning method that combines EM [22] with the extended diversity density (DD) [11] algorithm. It starts with an initial concept point $\boldsymbol{h}$, which has the highest density of all the instances in a positive bag, then repeats the following two steps (E-step and M-step) to search for the maximum-likelihood hypothesis. In the E-step, the current hypothesis concept $\boldsymbol{h}$ is used to identify the most representative instance in each bag. In the M-step, the two-step gradient ascent search of the standard DD algorithm [7] is used to find a new concept $\boldsymbol{h}'$ that maximizes DD($\boldsymbol{h}$). Once the final optimal concept $\boldsymbol{h}$ is identified, a bag is deemed as positive if the weighted distance from $\boldsymbol{h}$ to any of its instances is below the threshold, an artificially given parameter, which is often set to 0.5 as in [2].

D. Citation-KNN

Based on the Hausdorff distance among bags, C-KNN [10] is an adapted KNN algorithm to the multiple-instance problem. It is inspired by the notion of citation from library and information science [27]. When identifying the label of bag $\boldsymbol{b}$, it takes into account not only the neighbors (R-nearest references) of $\boldsymbol{b}$ but also the bags that count $\boldsymbol{b}$ as a neighbor (C-nearest citers).

For the R-nearest references of a bag $\boldsymbol{b}$, R_p and R_n denote the numbers of positive and negative bags, respectively. For the C-nearest citers of $\boldsymbol{b}$, C_p and C_n denote the numbers of positive and negative bags, respectively. Let $p = R_p + C_p$ and $n = R_n + C_n$. If $p > n$, then the bag $\boldsymbol{b}$ is predicted as positive; otherwise, negative.

E. MILDM

MILDM [12] is one of the most recent and advanced MIL methods, which uses bag mapping to transform a bag $\mathbf{B}_I$ into a single instance $\boldsymbol{B}_I^{\phi}$ in a new space via instance selection. It aims to identify the best instances to directly distinguish bags in the new mapping space. It consists of two steps. The first is to map each bag to a new feature space using the selected discriminative instances pool (DIP), a hidden instance set constructed from bags of instances. The second is to utilize any instance-based learning algorithm to derive multi-instance classification models. This article contains four different bag mapping models, that is, aMILGDM, pMILGDM, aMILLDM, and pMILLDM. They differ in at least one of the following two aspects to select instance in constructing the DIP: 1) the number of selected instance from each bag and 2) the scope of

training bags in selecting instance. aMILGDM and pMILGDM measure the discriminative power of the instances across the bags. aMILGDM uses all the training bags to generate the global DIP. pMILGDM only uses the positive bags. aMILLDM and pMILLDM compare the discriminative scores inside each individual bag. aMILLDM uses all the training bags to generate the local DIP. pMILGDM only uses the positive bags. Experiments show that aMILGDM performs better than the other models because more information is used to construct the DIP [12].

III. Proposed Framework

A. Fragmentary Multi-Instance Classification Framework

Inspired by the low-rank methods [25], [28]–[30], we aim to seek a matrix that appropriately fits the observed values of the fragmentary data matrix and fills the missing values, by constraining the rank of the approximation matrix lower than or equal to a predefined value k. This task can be addressed by solving the following problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}_\Omega - \mathbf{T}_\Omega\|^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \le r \end{aligned} \tag{1}$$

where $\mathbf{T}$ is the given incomplete data matrix. $\mathbf{X}$ is the low-rank matrix to approximate $\mathbf{T}$. Ω is the set of indices of observed elements in $\mathbf{T}$.

Aiming to propose an MIL framework which can deal with fragmentary data efficiently, we design a complementary mechanism between completion and MIL. Here, the complementary mechanism means the combination of completion and MIL. They are expected to benefit from each other and make the entire performance better. In MIL, the importance of an instance can be reflected by its possibility to be a positive instance. With the consideration that more positive instance is more important, we assign a more positive instance with a larger weight so that the completing loss of more positive instance in the observed domain should be less. Here, the completing loss of more positive instance means the loss function in completing the more positive instance. Intuitively, the more positive the instance is, the more important it is and it should be completed more accurately by making more emphasis on its completing loss. Therefore, it enables the completion of more positive instances to be more accurate. Given the measurements on data matrix $\mathbf{T}$, we propose the following FIC framework to fit most of the MIL methods:

$$\begin{aligned} \min_{\alpha,\mathbf{X}} \quad & C(\alpha, \mathbf{X}, \mathbf{Y}) + \mu \|F(\alpha, \mathbf{X})(\mathbf{X}_\Omega - \mathbf{T}_\Omega)^T\|^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \le r \end{aligned} \tag{2}$$

where $C(\cdot)$ is the objective of the MIL classifier to minimize the classification error. α represents the set of all model parameters of the classifier and $\mathbf{X}$ is the completed data matrix. $\mathbf{Y}$ denotes the label vector of all bags. μ is the regularization coefficient that balances the first term and second term. $F(\cdot)$ is a weight function which counts the weights of instances. The weights of different instances are determined according to the classification results and the positive degree of instances. The second term is to minimize the weighted loss of observed entries. The constraint term requires $\mathbf{X}$ to be of low rank.

To validate the effectiveness of the FIC framework, we embed four typical MIL methods, that is, MI-SVM, EM-DD, citation-K nearest neighbor (Citation-KNN), and aMILGDM into our framework and generate four corresponding FIC models, that is, F-MI-SVM, F-EM-DD, F-C-KNN, and F-aMILGDM. Since the meanings of more positive in these four methods are different, for each method, we design the corresponding weight functions. In the next section, we will present the concrete forms of the MIL classifier objective $C(\alpha, \mathbf{X}, \mathbf{Y})$ and the weight function $F(\alpha, \mathbf{X})$.

B. Typical Examples

1) F-MI-SVM: Here, the MIL parameter set α contains $\boldsymbol{w}$ and b. $C(\alpha, \mathbf{X}, \mathbf{Y})$ is based on the formulation of MI-SVM

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\xi_I \\ \text{s.t.} \quad & \forall I : y_I \max_{i\in I}(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \ge 1 - \xi_I, \xi_I \ge 0 \end{aligned} \tag{3}$$

where $\boldsymbol{w}$ is the projection matrix, b is the bias vector, $\{\xi_I\}_{I=1}^N$ are the slack variables of SVM, y_I is the label of the bag $\mathbf{B}_I$, and L is the tradeoff parameter. It is noticed that for each bag $\mathbf{B}_I$, we identify a representative instance to characterize it. Other instances have no impact on the objective. If we utilize the square hinge loss, it is considered to minimize the following objective function:

$$C(\boldsymbol{w}, b, \mathbf{X}, \mathbf{Y}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\left(1 - \max_{i\in I}\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)y_I\right)_+^2. \tag{4}$$

Here, $(\cdot)_+$ is a function, which is equal to the variable if it is positive and 0 otherwise.

As in MI-SVM, the more positive instance means that it is farther from the hyperplane on positive direction of the normal vector of the hyperplane. Thus, we design the weight of instance $\boldsymbol{x}_j$ as

$$F\left(\boldsymbol{w}, b, \boldsymbol{x}_j\right) = \left(\delta\left(\boldsymbol{w}\boldsymbol{x}_j^T + b\right) + 1\right)^{0.5} \tag{5}$$

where $(\delta(\boldsymbol{w}^T\boldsymbol{x}_j + b) + 1)$ denotes the weight of the instance $\boldsymbol{x}_j$ and $\delta \ge 0$ is a parameter to adjust the weight. If $\delta = 0$, $\delta(\boldsymbol{w}^T\boldsymbol{x}_j + b) + 1 = 1(\forall j = 1, \ldots, n)$ and instances of all bags have equal importance.

2) F-EM-DD: In this model, the MIL parameter α represents the concept point $\boldsymbol{h}$ to be optimized. Considering the fact that the importance of each feature varies greatly in most applications, we associate each attribute with an unknown scale factor [7]. To estimate the label of bag $\mathbf{B}_I$ for hypothesis $\boldsymbol{h} = \{h_1, \ldots, h_d, v_1, \ldots, v_d\}$, the following generative model is introduced in advance:

$$\text{Label}(\mathbf{B}_I|\boldsymbol{h}) = \max_j \exp\left[-\sum_{d'=1}^{d}\left(v_{d'}\left(B_{Ijd'} - h_{d'}\right)\right)^2\right] \tag{6}$$

where $B_{Ijd'}$ is the feature value of instance $\mathbf{B}_{Ij}$ for dimension d', $h_{d'}$ is the feature value of $\boldsymbol{h}$ on dimension d', and $v_{d'}$

TABLE II
FOUR SPECIFIC METHODS AND THEIR CORRESPONDING TERMS AND PARAMETERS IN OUR FRAMEWORK

Method	α	$C(\alpha, \mathbf{X}, \mathbf{Y})$	$F(\alpha, \mathbf{X})$
F-MI-SVM	$\boldsymbol{w}, b$	$\frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + L\sum_{I=1}^{N}(1-(\boldsymbol{w}^T\hat{\boldsymbol{x}}_I+b)y_I)_+^2$	$[(\delta(\boldsymbol{w}\boldsymbol{x}_1^T+b)+1)^{0.5}, (\delta(\boldsymbol{w}\boldsymbol{x}_2^T+b)+1)^{0.5}, \cdots, (\delta(\boldsymbol{w}\boldsymbol{x}_n^T+b)+1)^{0.5}]$
F-EM-DD	$\boldsymbol{h}$	$\sum_{I=1}^{N}(-logPr(y_I\|\boldsymbol{h}, \mathbf{B}_I))$	$[\parallel \boldsymbol{x}_1-\boldsymbol{h} \parallel^{-0.5}, \parallel \boldsymbol{x}_2-\boldsymbol{h} \parallel^{-0.5}, \cdots, \parallel \boldsymbol{x}_n-\boldsymbol{h} \parallel^{-0.5}]$
F-C-KNN	$\boldsymbol{P}_{center}$	$none$	$[(\delta\exp(-\parallel \boldsymbol{x}_1-\boldsymbol{P}_{center} \parallel^2)+1)^{0.5}, \cdots, (\delta\exp(-\parallel \boldsymbol{x}_n-\boldsymbol{P}_{center} \parallel^2)+1)^{0.5}]$
F-aMILGDM	$\boldsymbol{P}_{center}^{\phi}$	$none$	$[(\delta\exp(-\parallel \boldsymbol{B}_1^{\phi}-\boldsymbol{P}_{center} \parallel^2)+1)^{0.5}, \cdots, (\delta\exp(-\parallel \boldsymbol{B}_N^{\phi}-\boldsymbol{P}_{center} \parallel^2)+1)^{0.5}]$

is the scale factor to indicate the importance of feature d'. Label$(\mathbf{B}_I|\boldsymbol{h})$ is the label which would be given to $\mathbf{B}_I$ if $\boldsymbol{h}$ is the correct hypothesis concept point.

Inherited from EM-DD, $C(\alpha, \mathbf{X}, \mathbf{Y})$ is

$$C(h, \mathbf{X}, \mathbf{Y}) = \sum_{I=1}^{N}(-\log \Pr(y_I|\boldsymbol{h}, \mathbf{B}_I)) \tag{7}$$

where $\mathbf{B}_I$ denotes the Ith bag. $\Pr(y_I|\boldsymbol{h}, \mathbf{B}_I)$ is estimated as $1 - |y_I - \text{Label}(\mathbf{B}_I|\boldsymbol{h})|$.

In F-EM-DD, the closer an instance $\boldsymbol{x}_j$ to the concept point $\boldsymbol{h}$ is, the more likely $\boldsymbol{x}_j$ is positive. Therefore, the weight of $\boldsymbol{x}_j$ is estimated by

$$F\big(\boldsymbol{h}, \boldsymbol{x}_j\big) = \|\boldsymbol{x}_j - \boldsymbol{h}\|^{-0.5}. \tag{8}$$

3) F-C-KNN: In this model, $C(\alpha, \mathbf{X}, \mathbf{Y})$ has no explicit formulation since Citation-KNN is a lazy approach. In other words, it has no training process.

In F-C-KNN, we assume that the closer an instance $\boldsymbol{x}_j$ to center $\boldsymbol{P}_{\text{center}}$ of all the positive instances is, the more likely $\boldsymbol{x}_j$ is positive. Thus, the weight function of the $\boldsymbol{x}_j$ is

$$F\big(\boldsymbol{P}_{\text{center}}, \boldsymbol{x}_j\big) = \Big(\delta\exp\Big(-\|\boldsymbol{x}_j - \boldsymbol{P}_{\text{center}}\|^2\Big) + 1\Big)^{0.5}. \tag{9}$$

4) F-aMILGDM: In this model, $C(\alpha, \mathbf{X}, \mathbf{Y})$ also has no explicit formulation since F-aMILGDM takes KNN as classifier when bags are transformed into single instances in a new space via selected instance pool. It has no training process.

In F-aMILGDM, the bag $\mathbf{B}_I$ is transformed into $\boldsymbol{B}_I^{\phi} \in \mathbb{R}^m$ as a single instance in a new feature space, whose dimensionality is m, using the DIP, denoted as P. A transitional supervised learning classifier, that is, KNN, is then trained on the instances in the new feature space. We assume that the closer an instance $\boldsymbol{B}_I^{\phi}$ to center $\boldsymbol{P}_{\text{center}}^{\phi} \in \mathbb{R}^m$ of all the positive instances is, the more likely $\boldsymbol{B}_I^{\phi}$ is positive. Thus, the weight function of $\mathbf{B}_{Ij}$ in Ith bag is

$$F\Big(\boldsymbol{P}_{\text{center}}^{\phi}, \mathbf{B}_{Ij}\Big) = \Big(\delta\exp\Big(-\|\boldsymbol{B}_I^{\phi} - \boldsymbol{P}_{\text{center}}^{\phi}\|^2\Big) + 1\Big)^{0.5}. \tag{10}$$

In summary, Table II presents our four specific models and their corresponding parameters, MIL classifiers and weight functions within this framework.

C. Optimization

To solve the problem of our framework, we will derive an algorithm by introducing auxiliary variables based on the ALM method. Different from other penalty-based approaches, the ALM method estimates the solution and Lagrange multipliers simultaneously in an iterative way. In order to estimate the classifier accurately, we introduce the auxiliary function $t(\alpha, \mathbf{X}, \boldsymbol{s})$ to approximate $C(\alpha, \mathbf{X}, \mathbf{Y})$ and rewrite (2) to minimize the following augmented Lagrangian function:

$$\begin{aligned} &AL\big(\alpha, \mathbf{X}, \boldsymbol{s}, \boldsymbol{\rho}, \rho'\big) \\ &= t(\alpha, \mathbf{X}, \boldsymbol{s}) + \mu\big\|F(\alpha, \mathbf{X})(\mathbf{X}_\Omega - \mathbf{T}_\Omega)^T\big\|^2 \\ &\quad + \frac{\rho'}{2}\left\|C(\alpha, \mathbf{X}, \mathbf{Y}) - t(\alpha, \mathbf{X}, \boldsymbol{s}) + \frac{\boldsymbol{\rho}}{\rho'}\right\|^2 \\ &\text{s.t.} \quad \text{rank}(\mathbf{X}) \le r \end{aligned} \tag{11}$$

where ρ' is the penalty coefficient. $\boldsymbol{\rho} \in \mathbb{R}^{N\times 1}$ is used to adjust the difference between $C(\alpha, \mathbf{X}, \mathbf{Y})$ and $t(\alpha, \mathbf{X}, \boldsymbol{s})$. They are parameters of ALM and their update rules will be provided. As ρ' augments to infinity, the last term of (11) will force $C(\alpha, \mathbf{X}, \mathbf{Y}) = t(\alpha, \mathbf{X}, \boldsymbol{s})$.

In order to decouple the term $F(\alpha, \mathbf{X})$ from the term $(\mathbf{X}_\Omega - \mathbf{T}_\Omega)^T$, we introduce the auxiliary function $f(\boldsymbol{s})$ that approximates $F(\alpha, \mathbf{X})$. Besides, we introduce the auxiliary $\mathbf{M}$ that approximates $\mathbf{X}$. Here, for maintaining the concision and convergence of the optimization, we use quadratic penalty method to approximate $F(\alpha, \mathbf{X})$ and $\mathbf{X}$ instead of ALM method. Thus, we rewrite the following approximation to (11) for large enough values of λ and η:

$$\begin{aligned} &AL\big(\alpha, \mathbf{X}, \boldsymbol{s}, \mathbf{M}, \boldsymbol{\rho}, \rho'\big) \\ &= t(\alpha, \mathbf{X}, \boldsymbol{s}) + \mu\big\|f(\boldsymbol{s})(\mathbf{X}_\Omega - \mathbf{T}_\Omega)^T\big\|^2 \\ &\quad + \frac{\rho'}{2}\left\|C(\alpha, \mathbf{X}, \mathbf{Y}) - t(\alpha, \mathbf{X}, \boldsymbol{s}) + \frac{\boldsymbol{\rho}}{\rho'}\right\|^2 \\ &\quad + \lambda\Big\|F(\alpha, \mathbf{X}) - f(\boldsymbol{s})\|^2 + \eta\|\mathbf{X} - \mathbf{M}\Big\|^2 \\ &\text{s.t.} \quad \text{rank}(\mathbf{M}) \le r. \end{aligned} \tag{12}$$

When $\mathbf{M}$ is fixed, (12) becomes

$$\begin{aligned} \min_{\alpha, \mathbf{X}, \boldsymbol{s}} \;& t(\alpha, \mathbf{X}, \boldsymbol{s}) + \mu\big\|f(\boldsymbol{s})(\mathbf{X}_\Omega - \mathbf{T}_\Omega)^T\big\|^2 \\ &+ \frac{\rho'}{2}\left\|C(\alpha, \mathbf{X}, \mathbf{Y}) - t(\alpha, \mathbf{X}, \boldsymbol{s}) + \frac{\boldsymbol{\rho}}{\rho'}\right\|^2 \\ &+ \lambda\|F(\alpha, \mathbf{X}) - f(\boldsymbol{s})\|^2 + \eta\|\mathbf{X} - \mathbf{M}\|^2 \end{aligned} \tag{13}$$

where the gradient of any term can be obtained directly. Thus, (13) can be solved by the cyclic coordinate decent method.

When α, $\mathbf{X}$, and $\boldsymbol{s}$ are fixed, (2) becomes

$$\begin{aligned} \min_{\mathbf{M}} \quad & \|\mathbf{X} - \mathbf{M}\|^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{M}) \le r. \end{aligned} \tag{14}$$

Assuming the SVD decomposition of $\mathbf{X}$ is $\mathbf{FSG}^T$, then the solution to $\mathbf{M}$ is

$$\mathbf{M} = \mathbf{F}_r\mathbf{S}_r\mathbf{G}_r^T \tag{15}$$

Algorithm 1 Algorithm to Solve FIC

Input: Observed data matrix $\mathbf{T}$; bag labels $\mathbf{Y}$; parameters: $\{\mu, \lambda, \eta, r\}$;
Output: Completed data matrix $\mathbf{X}$; MIL parameter set α; auxiliary variables $\{\boldsymbol{s}, \mathbf{M}\}$
1. Initialize $\mathbf{X}, \alpha, \boldsymbol{s}, \mathbf{M}$
Repeat
2. Fixing $\mathbf{M}$, update $\mathbf{X}, \alpha, \boldsymbol{s}$ by Eq. (13) with cyclic coordinate decent method
3. Fixing $\mathbf{X}, \alpha, \boldsymbol{s}$, update $\mathbf{M}$ by Eq. (15)
4. Update $\boldsymbol{\rho}$ by Eq. (16)
Until converges

where $\mathbf{S}_r$ contains the top r largest values and $\mathbf{F}_r$ and $\mathbf{G}_r$ are the singular vector matrices corresponding to $\mathbf{S}_r$.

In the kth iteration, the ALM parameter $\boldsymbol{\rho}$ is updated by

$$\boldsymbol{\rho}_{(k)} = \boldsymbol{\rho}_{(k-1)} + \rho'_{(k)}(C(\alpha, \mathbf{X}, \mathbf{Y}) - t(\alpha, \mathbf{X}, \boldsymbol{s})). \tag{16}$$

The detailed algorithm to solve our proposed FIC is shown in Algorithm 1.

IV. Fragmentary MI-SVM

As an illustration, we present the detailed model and optimization of the F-MI-SVM.

A. Model

In MI-SVM, once the representative instances have been identified, the relative position of other instances in each bag with respect to the classification boundary becomes irrelevant. The most positive instances have crucial effect on learning the classification boundary. Thus, these key instances should be completed correctly in observed data domain and given relatively large weight. The detailed illustration is presented in Fig. 1. According to the above analysis, we have the following minimization problem for F-MI-SVM model:

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi,\mathbf{X}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\xi_I \\ & + \mu \sum_{(i,j)\in\Omega} \big(\delta\big(\boldsymbol{w}^T\boldsymbol{x}_j + b\big) + 1\big)\big(X_{ij} - T_{ij}\big)^2 \\ \text{s.t.} \quad & \forall I : y_I \max_{i\in I}(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_I \\ & \xi_I \geq 0, \text{rank}(\mathbf{X}) \leq r \end{aligned} \tag{17}$$

where $(\delta(\boldsymbol{w}^T\boldsymbol{x}_j + b) + 1)$ denotes the weight of the instance $\boldsymbol{x}_j$ and $\delta \geq 0$ is a parameter to adjust the weight. This model is obtained by embedding the MI-SVM objective function $C(\cdot)$ and the corresponding weight function $F(\cdot)$ into our FIC framework (2) directly.

B. Solution

A scheme of optimization is the alternation of the following two steps: 1) for the given representative instances of each bag, finding the optimal discriminant and completing the data matrix and 2) for the given discriminant and completed data, updating the representative instance of each bag.

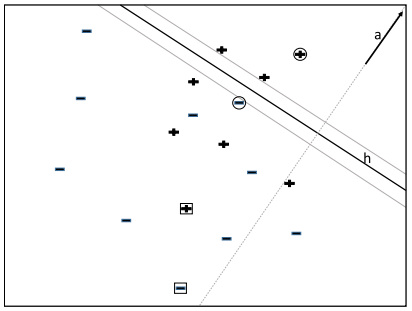


Fig. 1. Weight for different instances. There are two bags with one positive and one negative. The instances in the positive bag P are marked with "+" and instances in the negative bag N are marked with "-." Here, all instances are complete. We can calculate the weight of instances in every bag based on the optimal hyperplane "h" (the normal vector "a" of "h"). Intuitively, among these instances in the two bags, instance $\oplus$ has the largest weight and $\boxminus$ has the smallest weight. For bag P, $\oplus$ is the key instance because it is the "most positive" among the instances in P. For bag N, $\ominus$ is the key instance because it is the "least negative" among the instances in N.

1) Finding the Optimal Discriminant and Completing the Data Matrix: Given the representative instances, (17) becomes

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{w},b,\xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\xi_I \\ & + \mu \sum_{(i,j)\in\Omega} \big(\delta\big(\boldsymbol{w}^T\boldsymbol{x}_j + b\big) + 1\big)\big(X_{ij} - T_{ij}\big)^2 \\ \text{s.t.} \quad & \forall I : y_I\big(\langle \boldsymbol{w}, \hat{\boldsymbol{x}}_I\rangle + b\big) \geq 1 - \xi_I \\ & \xi_I \geq 0, \text{rank}(\mathbf{X}) \leq r \end{aligned} \tag{18}$$

where $\hat{\boldsymbol{x}}_I$ denotes the representative instance of the Ith bag. We select the squared hinge loss and (18) becomes

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\big(1 - \big(\boldsymbol{w}^T\hat{\boldsymbol{x}}_I + b\big)y_I\big)_+^2 \\ & + \mu \sum_{(i,j)\in\Omega} \big(\delta\big(\boldsymbol{w}^T\boldsymbol{x}_j + b\big) + 1\big)\big(X_{ij} - T_{ij}\big)^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \leq r \end{aligned} \tag{19}$$

which is derived by employing the loss function to remove the constraint $\forall I : y_I(\langle \boldsymbol{w}, \hat{\boldsymbol{x}}_I\rangle + b) \geq 1 - \xi_I, \xi_I \geq 0$.

We observe that in (19), there is a coupling between $X_{ij}^2((i,j) \in \Omega)$ and $\boldsymbol{x}_j$ in the last term. Therefore, it is difficult to optimize $\mathbf{X}$ directly. We introduce the auxiliary variable $s_j(j = 1, 2, \ldots, n)$ that approximates $\boldsymbol{w}^T\boldsymbol{x}_j + b$ and rewrite (19) into the following problem for a large enough parameter λ:

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{w},b,\boldsymbol{s}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\big(1 - \big(\boldsymbol{w}^T\hat{\boldsymbol{x}}_I + b\big)y_I\big)_+^2 \\ & + \mu \sum_{(i,j)\in\Omega} \big(\delta s_j + 1\big)\big(X_{ij} - T_{ij}\big)^2 \\ & + \frac{\lambda}{2}\|\mathbf{X}^T\boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) \leq r. \end{aligned} \tag{20}$$

We introduce an auxiliary variable $\mathbf{M} \in \mathbb{R}^{d\times n}$ that approximates $\mathbf{X}$ and rewrite (20) to minimize the following

approximation for large enough η:

$$\begin{aligned}&\text{obj}(\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \mathbf{M})\\&= \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\big(1-\big(\boldsymbol{w}^T\hat{\boldsymbol{x}}_I+b\big)y_I\big)_+^2 + \frac{\eta}{2}\|\mathbf{X}-\mathbf{M}\|^2\\&\quad+\ \mu\sum_{(i,j)\in\Omega}\big(\delta s_j+1\big)\big(X_{ij}-T_{ij}\big)^2 + \frac{\lambda}{2}\|\mathbf{X}^T\boldsymbol{w}+\mathbf{1}b-\boldsymbol{s}\|^2\\&\text{s.t.}\ \ \text{rank}(\mathbf{M})\le r.\end{aligned}\tag{21}$$

By solving (21), we can complete the data matrix $\mathbf{X}$ and obtain the optimal discriminant.

2) Updating the Representative Instance of Each Bag: With the given discriminant $(\boldsymbol{w}, b)$ and completed data matrix $\mathbf{X}$, the representative instance is the one that makes $\boldsymbol{w}^T\boldsymbol{x}+b$ maximum in each bag.

The process of step 2) is direct. Next, we will solve the problem in (21) in step 1). The optimization of (21) consists of two steps: 1) updating $\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$ with fixed $\mathbf{M}$ and 2) updating $\mathbf{M}$ with fixed $\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$.

3) Updating $\boldsymbol{X}, \boldsymbol{w}, b, \boldsymbol{s}$ With Fixed $\boldsymbol{M}$: When $\mathbf{M}$ is fixed, (21) becomes

$$\begin{aligned}\min_{\mathbf{X},\boldsymbol{w},b,\boldsymbol{s}} &= \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\big(1-\big(\boldsymbol{w}^T\hat{\boldsymbol{x}}_I+b\big)y_I\big)_+^2\\&\quad+\mu\sum_{(i,j)\in\Omega}\big(\delta s_j+1\big)\big(X_{ij}-T_{ij}\big)^2\\&\quad+\frac{\lambda}{2}\big\|\mathbf{X}^T\boldsymbol{w}+\mathbf{1}b-\boldsymbol{s}\big\|^2\\&\quad+\frac{\eta}{2}\|\mathbf{X}-\mathbf{M}\|^2.\end{aligned}\tag{22}$$

Inspired by [31], we introduce the auxiliary variables $\hat{e}_I = \hat{y}_I - \boldsymbol{w}^T\hat{\boldsymbol{x}}_I - b$ and $\hat{\boldsymbol{e}} = [\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_N]^T \in \mathbb{R}^N$. Based on ALM [32], the augmented Lagrangian function of (22) is

$$\begin{aligned}&AL\big(\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \boldsymbol{\rho}, \rho'\big)\\&= \frac{1}{2}\|\boldsymbol{w}\|^2 + L\sum_{I=1}^{N}\big(\hat{y}_I\hat{e}_I\big)_+^2\\&\quad+\ \frac{\rho'}{2}\left\|\hat{\mathbf{X}}^T\boldsymbol{w}+\mathbf{1}b-\hat{\boldsymbol{y}}+\hat{\boldsymbol{e}}+\frac{\boldsymbol{\rho}}{\rho'}\right\|^2\\&\quad+\ \mu\sum_{(i,j)\in\Omega}\big(\delta s_j+1\big)\big(X_{ij}-T_{ij}\big)^2\\&\quad+\ \frac{\lambda}{2}\big\|\mathbf{X}^T\boldsymbol{w}+\mathbf{1}b-\boldsymbol{s}\big\|^2\\&\quad+\ \frac{\eta}{2}\|\mathbf{X}-\mathbf{M}\|^2\end{aligned}\tag{23}$$

where ρ' is the penalty coefficient and $\boldsymbol{\rho} \in \mathbb{R}^{N\times 1}$ is used to adjust the difference between $\hat{\boldsymbol{e}}$ and $\hat{\boldsymbol{y}} - \hat{\mathbf{X}}^T\boldsymbol{w} - \mathbf{1}b$. They are all parameters of ALM and their updating rules will be provided later. We now derive the optimal solutions with respect to $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$, and $\boldsymbol{s}$ in (22) by the cyclic coordinate decent method with the following alternative steps: optimizing $\hat{\boldsymbol{e}}$ with fixed $\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$; optimizing $\mathbf{X}$ with fixed $\hat{\boldsymbol{e}}, \boldsymbol{w}, b, \boldsymbol{s}$; optimizing $\boldsymbol{w}$ with fixed $\hat{\boldsymbol{e}}, \mathbf{X}, b, \boldsymbol{s}$; optimizing b with fixed $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, \boldsymbol{s}$; and optimizing $\boldsymbol{s}$ with fixed $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$.

Optimizing $\hat{\boldsymbol{e}}$ With Fixed $\boldsymbol{X}, \boldsymbol{w}, b, \boldsymbol{s}$: When $\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$ are fixed, optimizing $\hat{e}_I$ in (23) becomes

$$\begin{aligned}&\min_{\hat{e}_I}\ \ L\big(\hat{y}_I\hat{e}_I\big)_+^2 + \frac{\rho'}{2}\left\|\hat{e}_I-\left(\hat{y}_I-\hat{\mathbf{X}}_I^T\boldsymbol{w}-\mathbf{1}b-\frac{\rho_I}{\rho'}\right)\right\|^2\\&=\min_{\hat{e}_I}\ \ L\big(\hat{y}_I\hat{e}_I\big)_+^2 + \frac{\rho'}{2}\big(\hat{e}_I-v_I\big)^2\\&=\min_{\hat{e}_I}\ \ \gamma\big(\hat{y}_I\hat{e}_I\big)_+^2 + \frac{1}{2}\big(\hat{e}_I-v_I\big)^2\end{aligned}\tag{24}$$

where $\gamma = L/\rho'$, $v_I = \hat{y}_I - \hat{\mathbf{X}}_I^T\boldsymbol{w} - b - (\rho_I/\rho')$ is a constant. Equation (24) is similar to [31, eq. (10)] and the following optimal e_I is similar to [31, eq. (13)]:

$$\hat{e}_I = \begin{cases} v_I/(1+2\gamma) & \hat{y}_I v_I > 0\\ v_I & \hat{y}_I v_I \le 0.\end{cases}\tag{25}$$

Optimizing $\boldsymbol{X}$ With Fixed $\hat{\boldsymbol{e}}, \boldsymbol{w}, b, \boldsymbol{s}$: When $\hat{\boldsymbol{e}}, \boldsymbol{w}, b, \boldsymbol{s}$ are fixed, we denote the objective of the subproblem with respect to $\mathbf{X}$ as $O_{\mathbf{X}}$. Then, the optimization of $\mathbf{X}$ in (23) becomes

$$\begin{aligned}\min_{\mathbf{X}}\ \ O_{\mathbf{X}} \equiv &\frac{\rho'}{2}\|\hat{\mathbf{X}}^T\boldsymbol{w}+\mathbf{1}b-\hat{\boldsymbol{y}}+\hat{\boldsymbol{e}}+\frac{\boldsymbol{\rho}}{\rho'}\|^2\\&+\frac{\lambda}{2}\|\mathbf{X}^T\boldsymbol{w}+\mathbf{1}b-\boldsymbol{s}\|^2\\&+\mu\sum_{(i,j)\in\Omega}\big(\delta s_j+1\big)\big(X_{ij}-T_{ij}\big)^2+\frac{\eta}{2}\|\mathbf{X}-\mathbf{M}\|^2.\end{aligned}\tag{26}$$

For each element X_{ij} of $\mathbf{X}$, we set the derivative with respect to X_{ij} to be zero and the first-order equation with respect to $X_{1j}, X_{2j}, \ldots, X_{dj}$ can be derived. Thus, for each column $\boldsymbol{x}_j$ of X_{ij}, d equations with respect to $X_{1j}, X_{2j}, \ldots, X_{dj}$ can be derived so that these equation sets can be solved to obtain $X_{1j}, X_{2j}, \ldots, X_{dj}$. To obtain $\boldsymbol{x}_j (j = 1, 2, \ldots, n)$, there are two cases with respect to instance $\boldsymbol{x}_j$ ($\boldsymbol{x}_j \in \hat{\mathbf{X}}$ and $\boldsymbol{x}_j \notin \hat{\mathbf{X}}$). To obtain $X_{i,j} (i = 1, 2, \ldots, d)$ in $\boldsymbol{x}_j$, there are two cases with respect to X_{ij} ($X_{ij} \in \Omega$ and $X_{ij} \notin \Omega$).

When the instance $\boldsymbol{x}_j \in \hat{\mathbf{X}}$ and $X_{ij} \notin \Omega$, the derivative of $O_{\mathbf{X}}$ with respect to X_{ij} is

$$\begin{aligned}\frac{\partial O_{\mathbf{X}}}{\partial X_{ij}} = &\sum_{l\ne i}\big(\lambda+\rho'\big)w_i w_l X_{lj} + (\lambda w_i w_i + \eta)X_{ij}\\&+\lambda w_i\big(b-s_j\big)+\eta' M_{ij}+\lambda w_i\big(b-\hat{y}_j+\hat{e}_j\big).\end{aligned}\tag{27}$$

If $\boldsymbol{x}_j \in \hat{\mathbf{X}}$ and $X_{ij} \in \Omega$, the derivative of $O_{\mathbf{X}}$ with respect to X_{ij} is

$$\begin{aligned}\frac{\partial O_{\mathbf{X}}}{\partial X_{ij}} = &\sum_{l\ne i}\big(\lambda+\rho'\big)w_i w_l X_{lj} + \big(\lambda w_i w_i+\eta+2\mu\big(\delta s_j+1\big)\big)X_{ij}\\&+\lambda w_i\big(b-s_j\big)+\eta M_{ij}+\lambda w_i\left(b-\hat{y}_j+\hat{e}_j+\frac{\rho_j}{\rho'}\right)\\&-2\mu T_{ij}\big(\delta s_j+1\big).\end{aligned}\tag{28}$$

When the instance $\boldsymbol{x}_j \notin \hat{\mathbf{X}}$ and $X_{ij} \notin \Omega$, the derivative of $O_{\mathbf{X}}$ with respect to X_{ij} is

$$\frac{\partial O_{\mathbf{X}}}{\partial X_{ij}} = \sum_{l\ne i}\lambda w_i w_l X_{lj} + (\lambda w_i w_i+\eta)X_{ij} + \lambda w_l\big(b-s_j\big)+\eta M_{ij}.\tag{29}$$

If $\boldsymbol{x}_j \notin \hat{\mathbf{X}}$ and $X_{ij} \in \Omega$, the derivative of $O_{\mathbf{X}}$ with respect to X_{ij} is

$$\frac{\partial O_{\mathbf{X}}}{\partial X_{ij}} = \sum_{l \neq i} \lambda w_i w_l X_{lj} + \big(\lambda w_i w_i + \eta + 2\mu\big(\delta s_j + 1\big)\big)X_{ij} + \lambda w_l\big(b - s_j\big) + \eta M_{ij} - 2\mu T_{ij}\big(\delta s_j + 1\big). \tag{30}$$

Thus, we can solve for $\mathbf{X}$ with the gradient descent method.

Optimizing $\boldsymbol{w}, b$ With Fixed $\hat{\boldsymbol{e}}, \boldsymbol{X}, \boldsymbol{s}$: When $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{s}$ are fixed, (23) becomes

$$\min_{\boldsymbol{w},b} \quad O_{\boldsymbol{w},b} \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\rho'}{2}\|\hat{\mathbf{X}}^T\boldsymbol{w} + \mathbf{1}b - \hat{\boldsymbol{y}} + \hat{\boldsymbol{e}} + \frac{\boldsymbol{\rho}}{\rho'}\|^2 + \frac{\lambda}{2}\|\mathbf{X}^T\boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2. \tag{31}$$

The gradients of $O_{\boldsymbol{w},b}$ with respect to $\boldsymbol{w}$ and b are

$$\frac{\partial O_{\boldsymbol{w},b}}{\partial \boldsymbol{w}} = \boldsymbol{w} + \rho'\hat{\mathbf{X}}\hat{\mathbf{X}}^T w + \rho'\hat{\mathbf{X}}\left(\mathbf{1}b - \hat{\boldsymbol{y}} + \hat{\boldsymbol{e}} + \frac{\hat{\boldsymbol{\rho}}}{\rho'}\right) + \lambda\mathbf{X}\mathbf{X}^T\mathbf{w} + \lambda\mathbf{X}(\mathbf{1}b - \boldsymbol{s}) \tag{32}$$

$$\frac{\partial O_{\boldsymbol{w},b}}{\partial b} = \rho' N b + \rho'\mathbf{1}^T\left(\hat{\mathbf{X}}^T\boldsymbol{w} - \hat{\boldsymbol{y}} + \hat{\boldsymbol{e}} + \frac{\hat{\boldsymbol{\rho}}}{\rho'}\right) + \lambda n b + \lambda\mathbf{1}^T\big(\mathbf{X}^T\boldsymbol{w} - \boldsymbol{s}\big). \tag{33}$$

Following [31], we use an optimal step-size gradient method to update $\boldsymbol{w}$ and b in each iteration, which only costs $O(n\bar{d})$. Here, $\bar{d}$ is the average number of nonzero elements per instance. It is as costly as calculating the matrix inverse.

Optimizing $\boldsymbol{s}$ With Fixed $\hat{\boldsymbol{e}}, \boldsymbol{X}, \boldsymbol{w}, b$: When $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$ are fixed, (23) becomes

$$\min_{\boldsymbol{s}} \quad \mu \sum_{(i,j)\in\Omega} \big(\delta s_j + 1\big)\big(X_{ij} - T_{ij}\big)^2 + \frac{\lambda}{2}\|\mathbf{X}^T\boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2. \tag{34}$$

Denote $\tau_j = \sum_{(i,j)\in\Omega}(X_{ij} - T_{ij})^2,\ \forall j = 1, 2, \ldots, n$, (34) can be rewritten as

$$\min_{\boldsymbol{s}} \quad \frac{\lambda}{2}\|\boldsymbol{s} - \mathbf{X}^T\boldsymbol{w} - \mathbf{1}b\|^2 + \mu\sum_{j=1}^{n}\tau_j\big(\delta s_j + 1\big). \tag{35}$$

We can obtain the optimal $s_j (\forall j = 1, 2, \ldots, n)$ by solving

$$\min_{s_j} \quad \frac{\lambda}{2}\big(s_j - \boldsymbol{w}^T\boldsymbol{x}_j - b\big)^2 + \mu\tau_j\big(\delta s_j + 1\big). \tag{36}$$

Taking the derivative of (36) with respect to s_j and setting it to zero, we have

$$s_i = \boldsymbol{w}^T\boldsymbol{x}_j + b - \frac{\mu\delta}{\lambda}\tau_j. \tag{37}$$

Following the iterative thresholding method [33], at the kth iteration, the amount of violation linear computational cost solver is used to update the Lagrangian multiplier vector $\boldsymbol{\rho}$

$$\boldsymbol{\rho}_{(k)} = \boldsymbol{\rho}_{(k-1)} + \rho'_{(k)}\Big(\hat{\boldsymbol{e}} - \hat{\boldsymbol{y}} + \hat{\mathbf{X}}^T\boldsymbol{w} + \mathbf{1}b\Big). \tag{38}$$

Algorithm 2 Algorithm to Solve the Problem in (21)

Input: The observed data matrix $\mathbf{T}$; bag label $y_I (I = 1, 2, \cdots, N)$; representative instance index of each bag; parameters: $\{C, \mu, \delta, \lambda, \eta, k, \rho'_{(1)}, \rho'_{(2)}, \cdots, \rho'_{(\infty)}\}$;
Output: The completed data matrix $\mathbf{X}$; discriminant $\{\boldsymbol{w}, b\}$; auxiliary variables $\{\hat{\boldsymbol{e}}, \boldsymbol{s}, \mathbf{M}\}$
1. Initializing $\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \mathbf{M}$;
Repeat
2. Fixing $\mathbf{M}$, deriving the optimal solutions w.r.t $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$ and s in Eq. (22) by Eq. (23) with the cyclic coordinate decent method;
3. Updating $\boldsymbol{\rho}$ by Eq. (38);
4. Fixing $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$, updating $\mathbf{M}$ by Eq.(39);
Until converges

4) Updating $\boldsymbol{M}$ With Fixed $\boldsymbol{X}, \boldsymbol{w}, b, \boldsymbol{s}$: When $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}$ are fixed, optimizing $\mathbf{M}$ in (21) becomes

$$\min_{\mathbf{M}} \quad \|\mathbf{M} - \mathbf{X}\|^2 \quad \text{s.t.} \quad \operatorname{rank}(\mathbf{M}) \leq r \tag{39}$$

where the solution of $\mathbf{M}$ is in (15).

The algorithm to solve the problem in (21) is shown in Algorithm 2. Nevertheless, we have been able to achieve competitive results in experiments even with the following simpler optimization heuristic. With the data matrix completed initially, alternate the following three steps: 1) finding the optimal discriminant function; 2) calculating the weight of each instance by the weight function; and 3) completing the fragmentary data matrix with weight mechanism.

V. Discussion

A. Convergence Analysis

Step 1) is pivotal in solving the problem of F-MI-SVM. We now provide its convergence behavior as in Algorithm 2.

Theorem 1: Algorithm 2 will monotonically decrease the objective of (21) in each iteration and it converges to a local optimum.

As in [25], when $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$, and $\boldsymbol{s}$ are fixed, the solution to $\mathbf{M}$ obtained by Algorithm 2 will decrease the objective. Thus, Theorem 1 is true if the solution to $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$, and $\boldsymbol{s}$ obtained by Algorithm 2 with fixed $\mathbf{M}$ decreases the objective. To prove it, our convergence and optimality theorems require the boundedness of some sequences, which is based on the following lemma.

Lemma 1: Let $\mathcal{H}$ be a real Hilbert space endowed with an inner product $\langle\cdot,\cdot\rangle$ and a corresponding norm $\|\cdot\|$, u and v are functions, $v \in \partial\|u\|$, where ∂u is the subgradient of u. Then, $\|v\|^* = 1$ if $u \neq 0$, and $\|v\|^* \leq 1$ if $u = 0$, where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$.

In Lemma 1, both v and u are the elements in a real Hilbert space $\mathcal{H}$. If $\mathcal{H}$ is a function space, then both v and u are functions. If $\mathcal{H}$ is a vector space, then both v and u are vectors. Readers interested in the proof of Lemma 1 may refer to [28].

Lemma 2: The sequences $\{\boldsymbol{\rho}_{(k)}\}$ in Algorithm 2 are bounded.

Proof: From

$$\boldsymbol{w}_{(k)} = \arg\min_{\boldsymbol{w}} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)})$$

$$
\begin{aligned}
b_{(k)} &= \arg\min_{b} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}) \\
\hat{\boldsymbol{e}}_{(k)} &= \arg\min_{\hat{\boldsymbol{e}}} AL(\hat{\boldsymbol{e}}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}) \\
\hat{\mathbf{X}}_{(k)} &= \arg\min_{\hat{\mathbf{X}}} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}).
\end{aligned} \tag{40}
$$

We have

$$
\begin{aligned}
0 &\in \partial_{\boldsymbol{w}} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}) \\
0 &\in \partial_{b} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}) \\
0 &\in \partial_{\hat{\boldsymbol{e}}} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}) \\
0 &\in \partial_{\hat{\mathbf{X}}} AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)} \\
&\quad \rho'_{(k)}, \lambda_{(k-1)}, \lambda'_{(k)}, \eta_{(k-1)}, \eta'_{(k)}).
\end{aligned} \tag{41}
$$

Therefore,

$$
\begin{aligned}
\boldsymbol{\rho}_{(k)} &\in \partial \|1/\sqrt{2} \cdot \boldsymbol{w}_{(k)}\|^2 + \partial \|\mathbf{X}^T \boldsymbol{w}_{(k)} + \mathbf{1}b - \boldsymbol{s}\|^2 \\
\boldsymbol{\rho}_{(k)} &\in \partial \|C(\hat{\mathbf{y}}^T \hat{\boldsymbol{e}}_{(k)})_+\|^2 \\
\boldsymbol{\rho}_{(k)} &\in \partial \|\sqrt{(\lambda/2)} \cdot (\mathbf{X}^T \boldsymbol{w} + \mathbf{1}b_{(k)} - \boldsymbol{s})\|^2 \\
\boldsymbol{\rho}_{(k)} &\in \partial \mu \sum_{(i,j)\in\Omega} (\delta s_j + 1)(\mathbf{X}_{(k)ij} - T_{ij})^2 \\
&\quad + \partial \frac{\lambda}{2} \|\mathbf{X}_{(k)}^T \boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2 + \partial \frac{\eta}{2} \|\mathbf{X}_{(k)} - \mathbf{M}\|^2.
\end{aligned} \tag{42}
$$

Then, by Lemma 1, the sequence $\{\boldsymbol{\rho}_{(k)}\}$ in Algorithm 2 is bounded since the dual norm of $\|\cdot\|^2$ is $\|\cdot\|^2$ [34]. ■

Denote $G_{(k)} = 0.5\|\boldsymbol{w}_{(k)}\|^2 + \|L(\hat{\mathbf{y}}^T \hat{\boldsymbol{e}}_{(k)})_+\|^2 + 0.5\rho'_{(k)}\|\hat{\mathbf{X}}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \hat{\mathbf{y}} + \hat{\boldsymbol{e}}_{(k)}\|^2 + \mu \sum_{(i,j)\in\Omega} (\delta s_{(k)j} + 1)(X_{(k)ij} - T_{ij})^2 + 0.5\lambda \|\mathbf{X}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \boldsymbol{s}_{(k)}\|^2 + 0.5\eta \|\mathbf{X}_{(k)} - \mathbf{M}\|^2$, we have the following lemma.

Lemma 3: The sequences $\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b_{(k)}$, and $\boldsymbol{s}_{(k)}$ in Algorithm 2 are all bounded if $G_{(k+1)} \le G_{(k)}$, $\forall k > 0$ and $\sum_{k=1}^{\infty} [(\rho'_{(k+1)})/\rho'^2_{(k)}] < \infty$.

Proof: As $G_{(k)}$ is nonincreasing as Algorithm 2 iterates, by substituting (38) to eliminate $\boldsymbol{\rho}_{(k-1)}$, we have

$$
\begin{aligned}
&AL\Big(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \mathbf{M}, \boldsymbol{w}_{(k)}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}\Big) \\
&\le AL\Big(\hat{\boldsymbol{e}}_{(k-1)}, \mathbf{X}_{(k-1)}, \mathbf{M}, \boldsymbol{w}_{(k-1)}, b_{(k-1)}, \boldsymbol{s}_{(k-1)}, \boldsymbol{\rho}_{(k-2)} \rho'_{(k-1)}\Big) \\
&\quad + 0.5\rho'^{-2}_{(k-1)} \Big(\rho'_{(k-1)} + \rho_{(k-2)}\Big) \|\boldsymbol{\rho}_{(k-1)} - \boldsymbol{\rho}_{(k-2)}\|^2.
\end{aligned} \tag{43}
$$

Thus, $AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \mathbf{M}, \boldsymbol{w}_{(k)}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)})$ is upper bounded due to the boundedness of $\boldsymbol{\rho}_{(k)}$ and $\sum_{k=1}^{\infty} [(\rho'_{(k)} + \rho'_{(k+1)})/\rho'^2_{(k)}] \le \sum_{k=1}^{\infty} [(2\rho'_{(k+1)})/\rho'^2_{(k)}] \le \infty$.

Furthermore, we have

$$
\begin{aligned}
&\|\boldsymbol{w}_{(k)}\|^2 + 0.5 \left\| L\Big(\hat{\mathbf{y}}^T \hat{\boldsymbol{e}}_{(k)}\Big)_+ \right\|^2 \\
&\quad + \mu \sum_{(i,j)\in\Omega} \Big(\delta s_{(k)j} + 1\Big)\Big(X_{(k)ij} - T_{ij}\Big)^2 \\
&\quad + \lambda \|\mathbf{X}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \boldsymbol{s}_{(k)}\|^2 + \eta \|\boldsymbol{X}_{(k)} - \mathbf{M}\|^2 \\
&= AL\Big(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}\Big) - \frac{\|\boldsymbol{\rho}_{(k)}\|^2}{2\rho'_{(k)}}
\end{aligned} \tag{44}
$$

be upper bounded. Thus, $\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b_{(k)}$, and $\boldsymbol{s}_{(k)}$ in Algorithm 2 are all bounded. ■

Lemma 3 implies the upper limit of $\rho'_{(k)}$ to generate the sequence $\boldsymbol{\rho}_{(k)}$

$$
\begin{aligned}
\rho'_{(k+1)} = &\Big(0.5\rho'_{(k)} \|\hat{\mathbf{X}}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \hat{\mathbf{y}} + \hat{\boldsymbol{e}}_{(k)}\|^2 + G_{(k)} \\
&- G_{(k+1)}\Big) \Big/ \Big(0.5\|\hat{\mathbf{X}}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \hat{\mathbf{y}} + \hat{\boldsymbol{e}}_{(k)}\|^2\Big).
\end{aligned} \tag{45}
$$

Theorem 2: $\hat{\boldsymbol{e}}_{(\infty)}, \mathbf{X}_{(\infty)}, \boldsymbol{w}_{(\infty)}, b_{(\infty)}$, and $\boldsymbol{s}_{(\infty)}$ obtained by step 2 in Algorithm 2 are the optimal solutions to (22).

Proof: According to the property of the ALM algorithm, that is, when using ALM to solve the minimization problem, adding the constraint will make the minimum value of the objective function greater than or equal to the minimum value of the unconstrained objective function, the following is true:

$$
\begin{aligned}
&AL(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}) \\
&= \min_{\boldsymbol{e}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}} AL(\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}) \\
&\le \min_{\boldsymbol{e}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \hat{\mathbf{X}}^T \boldsymbol{w} + \mathbf{1}b - \hat{\mathbf{y}} + \hat{\boldsymbol{e}} = 0} AL(\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}) \\
&= \min_{\boldsymbol{e}, \mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}, \hat{\mathbf{X}}^T \boldsymbol{w} + \mathbf{1}b - \hat{\mathbf{y}} + \hat{\boldsymbol{e}} = 0} 0.5\|\boldsymbol{w}\|^2 + \|L(\hat{\mathbf{y}}^T \hat{\boldsymbol{e}})_+\|^2 \\
&\quad + \mu \sum_{(i,j)\in\Omega} (\delta s_j + 1)(X_{ij} - T_{ij})^2 + 0.5\lambda \|\mathbf{X}^T \boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2 \\
&\quad + 0.5\eta \|\mathbf{X} - \mathbf{M}\|^2 + \frac{\|\boldsymbol{\rho}_{(k-1)}\|^2}{2\rho'_{(k)}} \\
&= \min_{\mathbf{X}, \boldsymbol{w}, b, \boldsymbol{s}} 0.5\|\boldsymbol{w}\|^2 + \|L(1 - (\boldsymbol{w}^T \hat{\boldsymbol{x}}_I + b) y_I)_+\|^2 \\
&\quad + \mu \sum_{(i,j)\in\Omega} (\delta s_j + 1)(X_{ij} - T_{ij})^2 + 0.5\lambda \|\mathbf{X}^T \boldsymbol{w} + \mathbf{1}b - \boldsymbol{s}\|^2 \\
&\quad + 0.5\eta \|\mathbf{X} - \mathbf{M}\|^2 + \frac{\|\boldsymbol{\rho}_{(k-1)}\|^2}{2\rho'_{(k)}} \\
&= \text{obj}^* + \frac{\|\boldsymbol{\rho}_{(k-1)}\|^2}{2\rho'_{(k)}}
\end{aligned} \tag{46}
$$

where obj* is the minimum value of objective function in (22). The third equality holds due to the fact that when the constraint with respect to e is satisfied, the third term in (23) degenerates to $\|\boldsymbol{\rho}_{(k)}\|^2/2\rho'_{(k)}$.

We denote the term $0.5\|\boldsymbol{w}_{(k)}\|^2 + \|L(\hat{\mathbf{y}}^T \hat{\boldsymbol{e}}_{(k)})_+\|^2 + \mu \sum_{(i,j)\in\Omega} (\delta sj + 1)(X_{(k)ij} - T_{ij})^2 + 0.5\lambda \|\mathbf{X}_{(k)}^T \boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} - \boldsymbol{s}_{(k)}\|^2 + 0.5\eta \|\mathbf{X}_{(k)} - \mathbf{M}\|^2$ by $H_{(k)}$, the following equation is true:

$$
H_{(k)} = AL\Big(\hat{\boldsymbol{e}}_{(k)}, \mathbf{X}_{(k)}, \boldsymbol{w}_{(k)}, b_{(k)}, \boldsymbol{s}_{(k)}, \boldsymbol{\rho}_{(k-1)}, \rho'_{(k)}\Big) - \frac{\|\boldsymbol{\rho}_{(k)}\|^2}{2\rho'_{(k)}}. \tag{47}
$$

According to (46), we have

$$
H_{(k)} \le \text{obj}^* + \frac{\|\boldsymbol{\rho}_{(k-1)}\|^2}{2\rho'_{(k)}} - \frac{\|\boldsymbol{\rho}_{(k)}\|^2}{2\rho'_{(k)}}. \tag{48}
$$

Due to the boundedness of $\{\boldsymbol{\rho}_{(k)}\}$, the term $[\|\boldsymbol{\rho}_{(k-1)}\|^2/2\rho'_{(k)}] - [\|\boldsymbol{\rho}_{(k)}\|^2/2\rho'_{(k)}]$ is negligible when $k \to \infty$. Thus

$$H_{(\infty)} \leq \text{obj}^*. \tag{49}$$

Besides, (38) leads to $\hat{\boldsymbol{e}}_{(k)} - \hat{\boldsymbol{y}} + \hat{\mathbf{X}}^T_{(k)}\boldsymbol{w}_{(k)} + \mathbf{1}b_{(k)} = (\boldsymbol{\rho}_{(k)} - \boldsymbol{\rho}_{(k-1)})/\rho'_{(k)}$. The constraint is satisfied when $k \to \infty$

$$\hat{\boldsymbol{e}}_{(\infty)} - \hat{\boldsymbol{y}} + \hat{\mathbf{X}}^T_{(\infty)}\boldsymbol{w}_{(\infty)} + \mathbf{1}b_{(\infty)} = 0. \tag{50}$$

Therefore, $(\hat{\boldsymbol{e}}_{(\infty)}, \mathbf{X}_{(\infty)}, \boldsymbol{w}_{(\infty)}, b_{(\infty)}, \boldsymbol{s}_{(\infty)})$ is an optimal solution to (22), and the solution to $\hat{\boldsymbol{e}}, \mathbf{X}, \boldsymbol{w}, b$, and $\boldsymbol{s}$ obtained by step 2 in Algorithm 2 with fixed $\mathbf{M}$ will decrease the objective. Theorem 1 holds and step 1) converges. ■

Since step 2) contains a discrete optimization problem, analyzing the convergence of F-MI-SVM solution, which contains both steps 1) and 2), becomes complicated. We will analyze that in future work.

B. Complexity Analysis

Since F-MI-SVM is solved in an alternative way, we calculate their total time by analyzing the complexity in solving each subproblem. Each iteration contains two steps. In step 1),we find the optimal discriminant and complete the data matrix. In step 2), we update the representative instance of each bag. In step 1), the time complexity to optimize $\hat{\boldsymbol{e}}$ is $O(Nd)$; the time complexity to optimize $\mathbf{X}$ is $O(nd)$; the time complexity to optimize $\boldsymbol{w}$ and b is $O(n\bar{d})$; the time complexity to optimize $\boldsymbol{s}$ is $O(nd)$; and the time complexity to optimize $\mathbf{M}$ is $O(nd^2)$. Therefore, the time complexity of step 1) is $O(nd^2)$. The time complexity of step 2) is $O(nd)$. As a result, the total time complexity of F-MI-SVM is $O(tnd^2)$, where t is the number of iterations. In each iteration, the completed data matrix $\mathbf{X}$ occupies the most memory space. Therefore, the memory complexity of F-MI-SVM is $O(nd)$.

VI. Experiments

A. Dataset

1) *MUSK* benchmark datasets are used in almost all the studies of MIL. Both MUSK1 and MUSK2 [3] datasets consist of descriptions of molecules using multiple low-energy conformations. Here, we regard each molecule as a bag, and regard its conformations as the instances in the bag. MUSK1 contains 47 positive and 45 negative molecules, and each molecule contains on average approximately 6 conformations. MUSK2 contains 39 positive and 63 negative molecules, and each molecule contains on average more than 60 conformations.
2) *COREL IMAGE* MIL datasets are generated by Andrews *et al.* [1] for an image annotation task. The datasets are from the Corel datasets preprocessed and segmented with the Blobworld method [35]. An image (bag) contains a set of blobs (instances), each represented by a 166-D feature vector. We utilize categories of elephant, fox, and tiger in our experiments. It has 100 positive and 100 negative images in each experiment and the negative images are drawn from the images of the other two animals randomly.
3) *COREL-2000* consists of 20 categories of COREL images. Each category contains 100 images. Each image represents a bag, and the regions of interest (ROIs) in the image represent instances, which are represented by a 9-D feature vector [36]. In our experiments, we use two pairs of categories ["Fashion"–"Sunsets" (F–S) and "Mountains and Glaciers"–"Food" (M–F)].
4) *SIVAL* contains 25 different objects in 10 scenes [37]. There are six different images taken for each object–scene pair. Thus, each object has 60 images. All the images have been segmented into regions. Each region is described by 30 visual features. We utilize objects "BlueScrunge" and "CandleWithHolder" for our experiments.
5) *PROCESS* is a text dataset [38], [39]. It is obtained as part of Task 2 of the BioCreative Text Mining Challenge. Given a name of a human protein and a full-text journal article, the task is to determine whether this protein–article pair can be annotated with a particular gene ontology (GO) term. For the MIL setting, each article represents a bag, and a paragraph in an article represents a member instance. The dataset contains 757 positive bags and 10961 negative bags, with 118 417 instance in total. Each paragraph instance is described by a set of word count and numerical features with 200 dimensions.

B. Comparison Between FIC Methods and Baselines

For each proposed model, we compare its classification performance with its baseline, that is, completing the data matrix and then learning the multi-instance classifier. For a fair comparison, we complete the data with the Robust Rank-k MC (RRMC) method [25], which is one of the most recent and advanced completion methods. Thus, the inputs can also be regarded as the fragmentary data.

Since the above datasets are all completed, to simulated the fragmentary case, we randomly choose a fixed missing percentage of locations in instance-feature matrix and hide their true values. Thus, the greater the missing proportion is, the less data and attributes that completion can depend on and it may result in greater error in completion and lower classification performance of the MIL classifiers. On each dataset, every category is randomly divided into half, with one subset for training and the other one for testing. We repeat each experiment for 10 random splits and record the average results.

As for the evaluation metrics, the area under the receiver-operating characteristic curve (AUC) [40], [41] and classification accuracy (ACC) are used in our experiments. The AUC represents the probability that a randomly chosen negative image will be ranked lower than a randomly chosen positive image. Different from the recall curve, it is insensitive to the class-imbalance. We vary the missing ratio (MR) from 10% to 50% with 10% as an interval on every dataset.

The comparisons results of average ACC and the standard deviations are given in Table III. The comparisons of average AUC are shown in Fig. 2. For each of the four MIL approaches, if our method performs better than its baseline method, its result is highlighted in boldface.

TABLE III
EXPERIMENTAL CLASSIFICATION ACCURACY (ACC, THE HIGHER THE BETTER) RESULTS [MEAN(STD)] ON 8 DATASETS. WE COMPARE OUR FOUR METHODS WITH THEIR CORRESPONDING BASELINES. IF OUR METHOD PERFORMS BETTER, ITS RESULT IS HIGHLIGHTED IN BOLDFACE

Data sets	Mr	MI-SVM	F-MI-SVM	EM-DD	F-EM-DD	CKNN	F-CKNN	aMILGDM	F-aMILGDM
MUSK1	0.1	.715(.079)	**.789(.061)**	.580(.067)	**.695(.058)**	.791(.082)	**.828(.059)**	.813(.039)	**.844(.010)**
	0.2	.726(.066)	**.767(.062)**	.576(.065)	**.680(.050)**	.806(.073)	**.826(.041)**	.809(.047)	**.817(.059)**
	0.3	.717(.063)	**.765(.055)**	.580(.101)	**.700(.043)**	.802(.089)	**.830(.042)**	.800(.050)	**.822(.036)**
	0.4	.708(.070)	**.735(.064)**	.558(.068)	**.657(.054)**	.787(.086)	**.819(.063)**	.800(.024)	**.813(.042)**
	0.5	.704(.085)	**.711(.076)**	.571(.057)	**.657(.062)**	.758(.094)	**.809(.054)**	.787(.039)	**.800(.028)**
MUSK2	0.1	.704(.059)	**.757(.052)**	.563(.210)	**.657(.042)**	.755(.061)	**.777(.052)**	.809(.096)	**.832(.031)**
	0.2	.698(.056)	**.752(.065)**	.563(.213)	**.659(.071)**	.761(.056)	**.772(.064)**	.804(.061)	**.855(.047)**
	0.3	.706(.077)	**.745(.046)**	.554(.203)	**.635(.049)**	.760(.069)	**.762(.038)**	.804(.038)	**.827(.047)**
	0.4	.712(.078)	**.727(.062)**	.563(.208)	**.641(.051)**	.755(.067)	**.771(.088)**	.800(.059)	**.805(.041)**
	0.5	.702(.041)	**.709(.065)**	.537(.205)	**.631(.054)**	.751(.058)	**.765(.056)**	.782(.055)	**.800(.054)**
ELEPHANT	0.1	.697(.043)	**.717(.029)**	.635(.034)	.633(.046)	.620(.069)	**.800(.034)**	.724(.011)	**.742(.034)**
	0.2	.678(.044)	**.707(.050)**	.604(.058)	**.651(.051)**	.559(.076)	**.753(.031)**	.728(.037)	**.758(.038)**
	0.3	.648(.065)	**.701(.057)**	.578(.058)	**.599(.063)**	.537(.065)	**.750(.032)**	.712(.040)	**.738(.019)**
	0.4	.634(.059)	**.673(.036)**	.594(.040)	**.615(.061)**	.526(.070)	**.752(.047)**	.712(.053)	**.736(.028)**
	0.5	.587(.086)	**.657(.073)**	.576(.051)	**.589(.067)**	.542(.088)	**.737(.049)**	.690(.047)	**.734(.021)**
FOX	0.1	.547(.066)	**.552(.069)**	.540(.048)	**.566(.044)**	.496(.061)	**.775(.029)**	.568(.048)	**.596(.067)**
	0.2	.521(.044)	**.565(.064)**	.550(.049)	**.571(.067)**	.506(.044)	**.755(.050)**	.590(.043)	**.606(.043)**
	0.3	.510(.047)	**.537(.044)**	.521(.059)	**.530(.042)**	.524(.061)	**.720(.036)**	.548(.074)	**.570(.042)**
	0.4	.520(.035)	**.553(.041)**	.524(.048)	**.539(.047)**	.513(.032)	**.724(.052)**	.540(.090)	**.556(.032)**
	0.5	.492(.042)	.486(.044)	.527(.027)	.523(.063)	.513(.043)	**.649(.069)**	.536(.051)	**.544(.049)**
TIGER	0.1	.710(.076)	**.712(.052)**	.586(.053)	**.673(.046)**	.653(.045)	**.727(.038)**	.700(.037)	**.762(.054)**
	0.2	.699(.068)	**.704(.058)**	.609(.052)	**.633(.063)**	.660(.038)	**.717(.045)**	.712(.016)	**.748(.011)**
	0.3	.694(.073)	**.714(.063)**	.633(.061)	.629(.063)	.666(.041)	**.710(.035)**	.686(.019)	**.702(.052)**
	0.4	.710(.051)	**.682(.061)**	.591(.084)	**.616(.075)**	.638(.042)	**.686(.027)**	.688(.038)	**.700(.090)**
	0.5	.690(.066)	.669(.055)	.577(.058)	.562(.092)	.602(.056)	**.627(.049)**	.676(.036)	**.698(.088)**
F-S	0.1	.755(.029)	**.842(.062)**	.844(.034)	**.865(.029)**	.913(.021)	.912(.034)	.886(.027)	**.906(.019)**
	0.2	.755(.028)	**.791(.052)**	.826(.050)	**.855(.050)**	.857(.041)	**.881(.037)**	.900(.037)	**.902(.039)**
	0.3	.758(.053)	**.779(.048)**	.831(.079)	.831(.041)	.837(.042)	**.870(.044)**	.860(.027)	**.862(.040)**
	0.4	.749(.059)	**.766(.037)**	.776(.198)	**.824(.060)**	.790(.025)	.790(.075)	.855(.036)	.850(.012)
	0.5	.711(.072)	**.766(.037)**	.785(.057)	**.820(.052)**	.666(.064)	**.749(.079)**	.860(.032)	.852(.041)
M-F	0.1	.708(.062)	**.753(.071)**	.841(.060)	**.881(.022)**	.890(.024)	.888(.024)	.928(.023)	**.962(.026)**
	0.2	.666(.051)	**.733(.069)**	.849(.050)	**.858(.031)**	.861(.034)	**.872(.034)**	.938(.022)	**.958(.019)**
	0.3	.699(.076)	**.709(.081)**	.812(.072)	**.861(.035)**	.825(.028)	**.845(.028)**	.914(.032)	**.940(.034)**
	0.4	.662(.064)	**.716(.108)**	.784(.069)	**.836(.055)**	.817(.052)	.780(.052)	.910(.049)	.882(.084)
	0.5	.658(.089)	.644(.093)	.746(.081)	**.788(.069)**	.761(.056)	**.777(.055)**	.840(.029)	**.848(.027)**
SIVAL	0.1	.600(.001)	.593(.021)	.922(.018)	.913(.026)	.715(.062)	**.763(.081)**	.947(.038)	**.960(.018)**
	0.2	.558(.032)	**.583(.036)**	.918(.031)	**.933(.025)**	.727(.059)	**.767(.053)**	.933(.033)	**.973(.018)**
	0.3	.560(.026)	**.587(.023)**	.860(.123)	**.937(.022)**	.678(.062)	**.712(.074)**	.927(.043)	**.953(.015)**
	0.4	.567(.027)	.565(.017)	.863(.098)	**.922(.052)**	.643(.063)	**.648(.069)**	.933(.033)	**.947(.015)**
	0.5	.565(.033)	**.568(.021)**	.835(.064)	**.863(.107)**	.633(.064)	**.642(.064)**	.926(.043)	.873(.140)
PROCESS	0.1	.774(.007)	**.809(.005)**	.414(.138)	**.749(.107)**	.910(.020)	**.926(.013)**	.912(.030)	**.928(.018)**
	0.2	.760(.009)	**.813(.004)**	.372(.057)	**.717(.138)**	.872(.029)	**.908(.013)**	.912(.033)	**.923(.033)**
	0.3	.759(.006)	**.794(.008)**	.381(.035)	**.736(.095)**	.839(.039)	**.853(.030)**	.904(.036)	**.911(.018)**
	0.4	.761(.006)	**.770(.010)**	.379(.050)	**.693(.145)**	.775(.038)	.754(.045)	.896(.022)	**.905(.018)**
	0.5	.759(.006)	**.767(.006)**	.409(.041)	**.668(.109)**	.796(.044)	.730(.116)	.880(.049)	.860(.032)

As seen from the experimental results in Table III and Fig. 2, we have the following observations. Although ACC and AUC are two different evaluation metrics, they both indicate the advantages of our methods. In almost all cases, our framework has improved the performances of their counterparts. It may be caused by the fact that our weight mechanism has taken into account the importance measurements of instances. Besides, the proposed weight function effectively facilitates the integration between completion and classifier learning. Besides, in a few cases, our methods perform slightly worse than their baselines. For example, when MR $= 0.1$ on ELEPHANT dataset, the ACC of F-EM-DD is 0.002 lower than that of EM-DD. It may be caused by that we use a fixed parameter δ for all MR values. This fixed δ may not be suitable for all the MR values.

C. Efficiency of RRMC Method

In this section, we compare the performances of the RRMC method and the mean completion method, that is, filling the missing attributes of each example with the mean of the observed attribute values on two datasets (MUSK1 and ELEPHANT), to verify the efficiency of RRMC. For intuition, we separate the matrix completing process from classifier learning process for each method, that is, first completing the fragmentary data matrix and then learning the classifier from the completed data. Table IV shows the comparison results for MR varying from 10% to 50% with 10% as interval. From Table IV, we observe that the RRMC method outperforms the mean completion method in almost all cases, which validates its efficiency.

D. Parameter Study

We tune δ for three different MR values, that is, 0.5, 0.7, and 0.9, in four different method–dataset pairs. As in the aforementioned experiments, on each dataset, we randomly divide each category into half, with one subset for training and the other one for testing. Then, an MR value of features of all the training data are randomly selected to be missing. Such process is repeated ten times. The effect of the parameter δ is shown in Fig. 3.

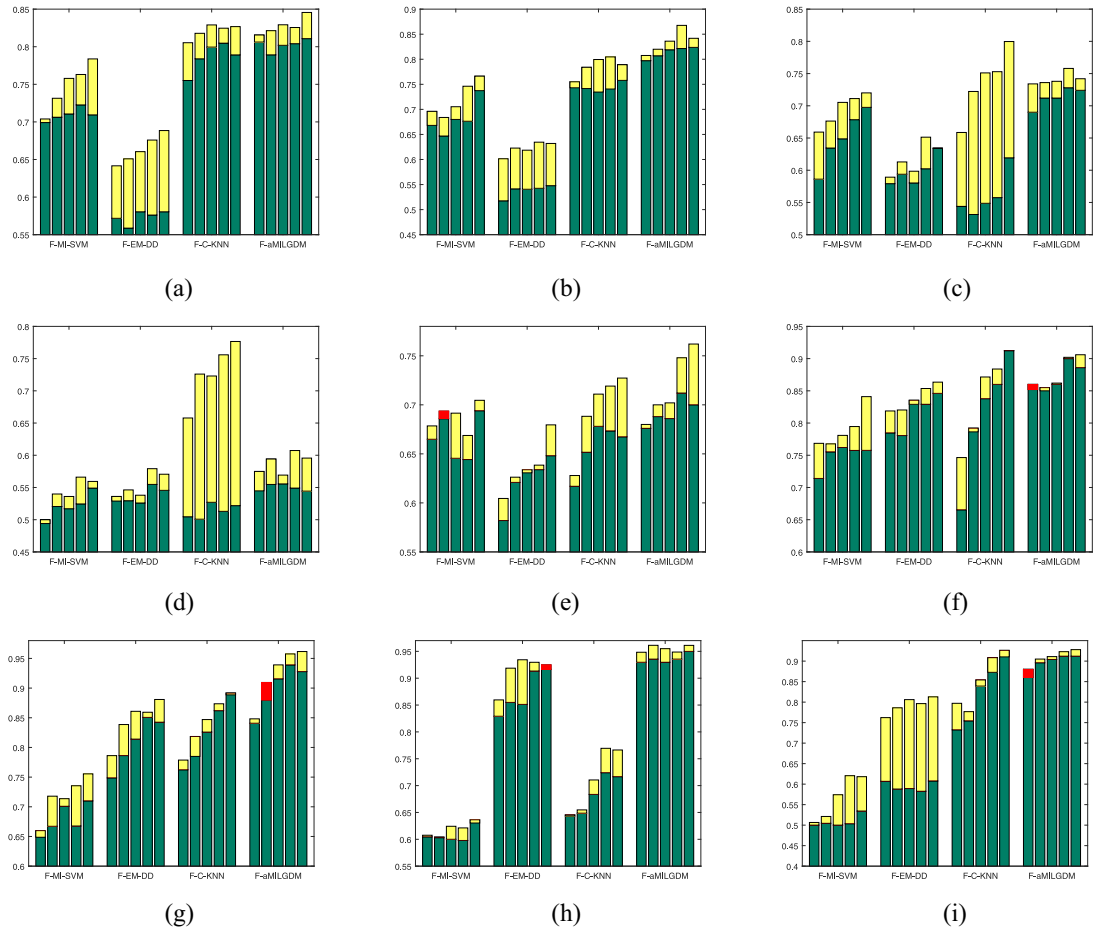


Fig. 2. AUC results for the nine datasets. For each method on each dataset, the bars from left to right represent the cases that $MR = 0.5$, $MR = 0.4$, $MR = 0.3$, $MR = 0.2$, and $MR = 0.1$, respectively. For each bar, the green part represents the performance of the baseline, the yellow part represents the improvement degree of our method, and the red part represents the performance level reduction of our method. (a) MUSK1. (b) MUSK2. (c) ELEPHANT. (d) FOX. (e) TIGER. (f) F-S. (g) M-F. (h) SIVAL. (i) PROCESS.

TABLE IV
AUC PERFORMANCES OF RRMC METHOD AND MEAN COMPLETION METHOD ON TWO DATASETS. IF RRMC METHOD PERFORMANCE BETTER THAN ITS BASELINE, ITS RESULT IS HIGHLIGHTED IN BOLDFACE

Datasets	MR	mean completion				RRMC			
		MI-SVM	EM-DD	C-KNN	aMILGDM	MI-SVM	EM-DD	C-KNN	aMILGDM
MUSK1	0.1	.681(.086)	.613(.078)	.720(.076)	.801(.067)	**.720(.090)**	**.720(.088)**	**.791(.072)**	**.816(.042)**
	0.2	.668(.065)	.561(.077)	.714(.045)	.780(.034)	**.714(.062)**	**.714(.085)**	**.806(.059)**	**.811(.043)**
	0.3	.691(.074)	.540(.071)	.727(.065)	.792(.031)	**.727(.065)**	**.727(.069)**	**.802(.064)**	**.804(.028)**
	0.4	.699(.090)	.547(.064)	.683(.084)	.776(.039)	.683(.083)	**.683(.061)**	**.787(.087)**	**.802(.056)**
	0.5	.659(.073)	.529(.055)	.695(.067)	.767(.028)	**.695(.071)**	**.695(.053)**	**.759(.070)**	**.789(.054)**
ELEPHANT	0.1	.594(.045)	.509(.043)	.692(.050)	.726(.021)	**.697(.044)**	**.509(.047)**	**.820(.045)**	**.727(.011)**
	0.2	.588(.031)	.542(.041)	.700(.031)	.717(.035)	**.678(.029)**	.540(.042)	**.769(.032)**	**.731(.035)**
	0.3	.588(.045)	.522(.029)	.628(.045)	.712(.051)	**.649(.045)**	**.543(.027)**	**.711(.045)**	**.716(.039)**
	0.4	.573(.033)	.519(.024)	.681(.036)	.701(.034)	**.634(.036)**	**.557(.024)**	**.751(.035)**	**.715(.059)**
	0.5	.565(.058)	.495(.023)	.632(.057)	.666(.058)	**.586(.058)**	**.579(.029)**	**.731(.058)**	**.693(.048)**

From Fig. 3, we can see that in all cases, our methods achieve stable and good performance with a very large variance of δ in all MR settings, which validates the robustness of our methods. On the whole, the optimal range of δ is $[10^{-4}, 10^{-2}]$ in F-MI-SVM (MUSK1) cases. Our experiments also indicate that on the whole, the optimal range of δ is $[10^{-5}, 10^{-1}]$ for F-EM-DD method, $[10^{-3}, 10^{-5}]$ for F-C-KNN method, and $[10^{-6}, 10^{-1}]$ for F-aMILGDM method. For each method, the optimal ranges of δ on different datasets are the same, which may be caused by the normalization of all datasets. On each dataset, however, different methods have different optimal δ ranges, which may be caused by the differences between the result scales of weight functions of different methods.

E. Convergence Behavior

To verify the convergence of Algorithm 2, we show the convergence behavior curves on two datasets MUSK1 and SIVAL with MR = 0.3 in Fig. 4. As shown in Fig. 4, the objective values are nonincreasing during the iterations and converge to a fixed value. In addition, it only takes around 20 rounds to converge.

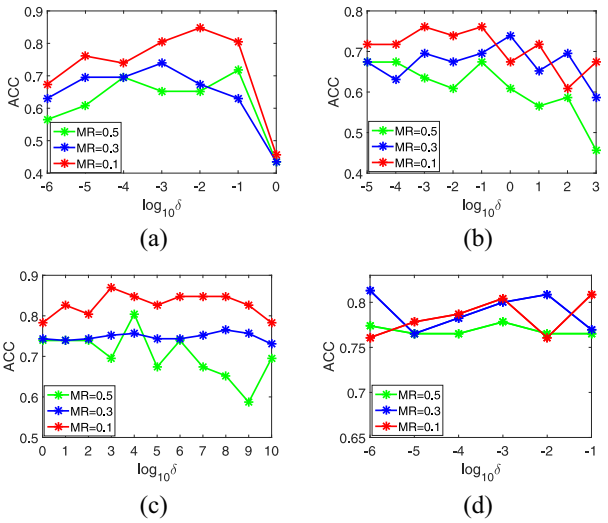


Fig. 3. Effect of the parameter δ in four cases of method-dataset pairs with different three MR values. (a) F-MI-SVM(MUSK1). (b) F-EM-DD(MUSK1). (c) F-C-KNN(MUSK1). (d) F-aMILGDM(MUSK1).

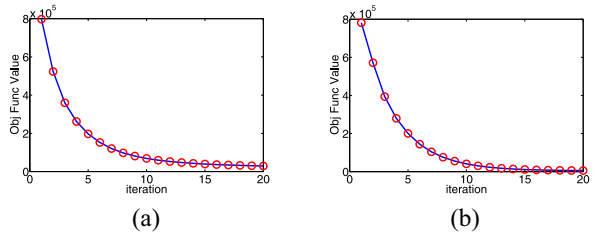


Fig. 4. Objective values of (21) with different numbers of iterations. (a) MUSK1. (b) SIVAL.

TABLE V
TRAINING TIME (MEASURED IN SECONDS) OF OUR ALGORITHMS AND CORRESPONDING BASELINES ON FIVE DATASETS

Methods	MUSK1	MUSK2	FOX	M-F	SIVAL	PROCESS
MI-SVM	1.2	14.3	0.8	0.9	0.7	944.0
F-MI-SVM	4.8	40.0	4.1	1.4	2.0	1905.1
EM-DD	64.6	177.3	216.7	14.6	5.3	7999.1
F-EM-DD	163.0	355.9	452.6	35.2	26.3	8718.4
C-KNN	2.9	425.1	69.6	18.3	3.4	16556.2
F-C-KNN	5.7	592.5	78.0	18.9	13.8	20741.6
aMILGDM	9.6	343.3	38.5	12.3	81.5	31768.2
F-aMILGDM	32.5	1415.45	120.6	34.7	238.6	94835.1

F. Runtime Comparison

Since the testing phase of each of our four algorithms is the same as that of the corresponding baseline, each of our algorithms and the corresponding baselines have the same running time in the testing phase. We only report the running time of the training phase. Table V shows the running time on five datasets.

As seen from Table V, we know that our methods have equivalent time spent by the other methods. In our four algorithms (F-MI-SVM, F-EM-DD, F-C-KNN, and F-aMILGDM), F-MI-SVM always has the best runtime performance, due to the efficiency of MI-SVM. The aMILGDM and F-aMILGDM often perform the worst because in each iteration, discriminative scores of all instances must be recalculated and all the bags must be remapped to instances. Besides, all the algorithms take the most time on PROCESS dataset because it has the largest data scales.

VII. CONCLUSION

In this article, we proposed probably the first framework to deal with multi-instance classification with fragmentary data. Our proposed FIC framework combines completion and classifier learning by the derived weight mechanism where more positive instances are given larger weights. Four MIL methods are embedded into the framework to verify the compatibility of our framework. As an example, the detailed algorithm of one model is provided. We further present the convergence guarantees. The experimental results on real datasets show that our framework can improve the performances of all the four MIL baselines, which validates the effectiveness and robustness of our framework. The first further work is to explore the feasibility of introducing other completion methods. Another one is to verify the compatibility of our framework with other MIL methods. For example, due to the great representation power of deep features, combining MIL with deep learning is an emerging topic. There are some challenging problems in this direction. For instance, how to incorporate the deep MIL algorithms into our framework or directly design a deep learning algorithm based on MIL with missing features. Since the instance is fragmentary, we should analyze the influence of incompleteness in constructing the neural networks, together with the consideration of instance correlation in MIL. They can extend the application scenario of the deep MIL.

REFERENCES

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. 15th Adv. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 561–568.
[2] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Proc. 14th Adv. Neural Inf. Process. Syst. (NIPS)*, 2001, pp. 1073–1080.
[3] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.
[4] R. Hong, M. Wang, Y. Gao, D. Tao, X. Li, and X. Wu, "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 669–680, May 2014.
[5] Y.-L. Zhang and Z.-H. Zhou, "Multi-instance learning with key instance shift," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 3441–3447.
[6] B. Li *et al.*, "Multi-view multi-instance learning based on joint sparse representation and multi-view dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2554–2560, Dec. 2017.
[7] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. 10th Adv. Neural Inf. Process. Syst. (NIPS)*, 1997, pp. 570–576.
[8] C. Yang, M. Dong, and F. Fotouhi, "Region based image annotation through multiple-instance learning," in *Proc. 13th ACM Int. Conf. Multimedia*, 2005, pp. 435–438.
[9] Z.-H. Zhou, K. Jiang, and M. Li, "Multi-instance learning based Web mining," *Appl. Intell.*, vol. 22, no. 2, pp. 135–147, 2005.
[10] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 1119–1126.
[11] R. A. Amar, D. R. Dooly, S. A. Goldman, and Q. Zhang, "Multiple-instance learning of real-valued data," in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*, 2001, pp. 3–10.
[12] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1065–1080, Jun. 2018.
[13] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognit.*, vol. 74, pp. 15–24, Feb. 2018.
[14] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 2132–2141.

[15] X. Liu *et al.*, "Deep multiple instance learning-based spatial–spectral classification for PAN and MS imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 461–473, Jan. 2018.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[17] P.-W. Lin and G.-L. Chang, "Modeling measurement errors and missing initial values in freeway dynamic origin–destination estimation systems," *Transport. Res. C Emerg. Technol.*, vol. 14, no. 6, pp. 384–402, 2006.

[18] N.-E. E. Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges—A survey," *Inf. Fusion*, vol. 12, no. 1, pp. 4–10, 2012.

[19] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Partial multi-view clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1968–1974.

[20] D. Pyle, *Data Preparation for Data Mining*. San Francisco, CA, USA: Morgan Kaufmann, 1999.

[21] W. Fan, H. Lu, S. E. Madnick, and D. W. Cheung, "DIRECT: A system for mining data value conversion rules from disparate data sources," *Decis. Support Syst.*, vol. 34, no. 1, pp. 19–39, 2002.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001.

[24] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Elect. Eng. Dept., Standford Univ., Standford, CA, USA, 2002.

[25] J. Huang, F. Nie, H. Huang, Y. Lei, and C. H. Q. Ding, "Social trust prediction using rank-*k* matrix recovery," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2647–2653.

[26] P. Li and S. Chen, "Shared Gaussian process latent variable model for incomplete multiview clustering," *IEEE Trans. Cybern.*, to be published.

[27] E. Garfield, *Citation Indexing Its Theory and Application in Science, Technology and Humanities* (Information Science). New York, NY, USA: Wiley, 1973.

[28] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *CoRR*, vol. abs/1009.5055, 2010. [Online]. Available: http://arxiv.org/abs/1009.5055

[29] Y. Liu, L. C. Jiao, and F. Shang, "A fast tri-factorization method for low-rank matrix recovery and completion," *Pattern Recognit.*, vol. 46, no. 1, pp. 163–173, 2013.

[30] H. Fan, Y. Luo, L. Qi, N. Wang, J. Dong, and H. Yu, "Robust photometric stereo in a scattering medium via low-rank matrix completion and recovery," in *Proc. 9th Int. Conf. Human Syst. Interact. (HSI)*, 2016, pp. 323–329.

[31] F. Nie, Y. Huang, X. Wang, and H. Huang, "New primal SVM solver with linear computational cost for big data classifications," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. 505–513.

[32] P. E. Gill and D. P. Robinson, "A primal-dual augmented Lagrangian," *Comput. Optim. Appl.*, vol. 51, no. 1, pp. 1–25, 2012.

[33] J. Wright, A. Ganesh, S. R. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. 23rd Annu. Conf. Adv. Neural Inf. Process.*, 2009, pp. 2080–2088.

[34] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[35] C. Carson, M. Thomas, S. J. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc. 3rd Int. Conf. Vis. Inf. Syst. (VISUAL)*, 1999, pp. 509–516.

[36] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, Dec. 2004.

[37] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts, "Localized content based image retrieval," in *Proc. 7th ACM SIGMM Int. Workshop Multimedia Inf. Retriev. (MIR)*, 2005, pp. 227–236.

[38] C. Blaschke, E. A. León, M. Krallinger, and A. Valencia, "Evaluation of biocreative assessment of task 2," *BMC Bioinformat.*, vol. 6, no. S1, p. S16, 2005.

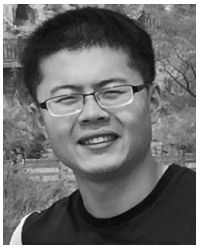
[39] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 697–704.

[40] W.-J. Li and D.-Y. Yeung, "Localized content-based image retrieval through evidence region identification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 1666–1673.

[41] Y. Jia and C. Zhang, "Instance-level semisupervised multiple instance learning," in *Proc. 23rd AAAI Conf. Artif. Intell. (AAAI)*, 2008, pp. 640–645.

Jie Wu received the B.S. degree from Zhejiang University, Hangzhou, China, in 2016, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2018.

His current research interests include data mining and machine learning.



Wenzhang Zhuge received the B.S. degree from Shandong University, Jinan, China, in 2015, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2017, where he is currently pursuing the Ph.D. degree.

His current research interests include machine learning, system science, and data mining.

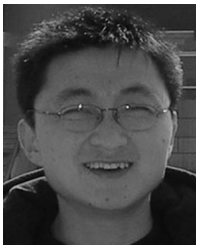
Xinwang Liu received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China.

He is currently an Associate Professor with the School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. He has published over 80 peer-reviewed papers, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, International Conference on Computer Version, IEEE Conference on Computer Vision and Pattern Recognition, AAAI Conference on Artificial Intelligence, and International Joint Conference on Artificial Intelligence.

Li Liu (SM'19) received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2012.

She is currently an Associate Professor with the College of System Engineering, National University of Defense Technology. From 2016 to 2018, she has visited Machine Vision Group with the University of Oulu, Oulu, Finland. Her current research interests include texture analysis, image classification, object detection, and scene understanding.

Dr. Liu was the Co-Chair of International Workshops at ACCV2014, IEEE Conference on Computer Vision and Pattern Recognition 2016, International Conference on Computer Version 2017, and ECCV2018. She was a Guest Editor of special issues for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Computer Vision*, and *Neurocomputing*.



Chenping Hou (M'12) received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2009.

He is currently a Full Professor with the Department of Systems Science, National University of Defense Technology. He has authored over 80 peer-reviewed papers in journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, International Joint Conference on Artificial Intelligence, and AAAI Conference on Artificial Intelligence. His current research interests include machine learning, data mining, and computer vision.