Chapter 6
Addressing the Inevitable Imprecision: Multiple Instance Learning for Hyperspectral Image Analysis

Changzhe Jiao, Xiaoxiao Du and Alina Zare

Abstract In many remote sensing and hyperspectral image analysis applications, precise ground truth information is unavailable or impossible to obtain. Imprecision in ground truth often results from highly mixed or sub-pixel spectral responses over classes of interest, a mismatch between the precision of global positioning system (GPS) units and the spatial resolution of collected imagery, and misalignment between multiple sources of data. Given these sorts of imprecision, training of traditional supervised machine learning models which rely on the assumption of accurate and precise ground truth becomes intractable. Multiple instance learning (MIL) is a methodology that can be used to address these challenging problems. This chapter investigates the topic of hyperspectral image analysis given imprecisely labeled data and reviews MIL methods for hyperspectral target detection, classification, data fusion, and regression.

6.1 Motivating Examples for Multiple Instance Learning in Hyperspectral Analysis

In standard supervised machine learning, each training sample is assumed to be coupled with the desired classification label. However, acquiring accurately labeled training data can be time consuming, expensive, or at times infeasible. Challenges with obtaining precise training labels and location information are pervasive throughout many remote sensing and hyperspectral image analysis tasks. A learning methodol-

C. Jiao
Xidian University, Xi'an, China
e-mail: cjiao@xidian.edu.cn

X. Du
University of Michigan, Ann Arbor, USA
e-mail: xiaodu@umich.edu

A. Zare (✉)
University of Florida, Gainesville, USA
e-mail: azare@ufl.edu

S. Prasad and J. Chanussot (eds.), *Hyperspectral Image Analysis*,
Advances in Computer Vision and Pattern Recognition,
https://doi.org/10.1007/978-3-030-38617-7_6

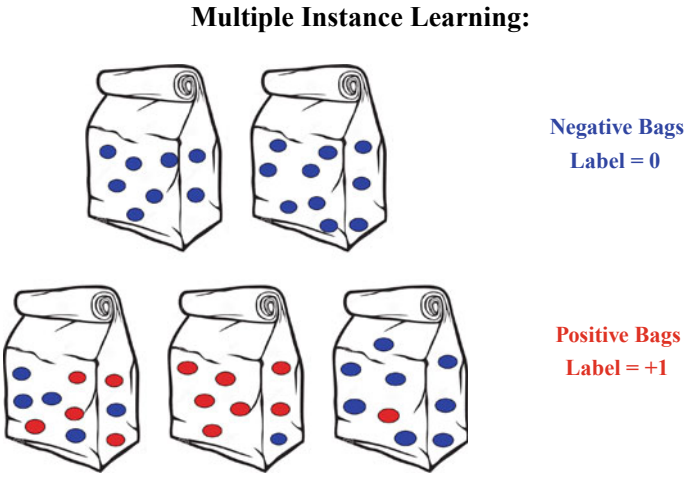

Fig. 6.1 In multiple instance learning, data is labeled at the bag level. A bag is labeled as a *positive* bag if it contains at least one target instance. The number of target versus nontarget instances in each positive bag is unknown. A bag is labeled as a *negative* bag if it contains only nontarget instances. In this figure, blue points correspond to nontarget instances where red points correspond to target instances. Source: © [2019] IEEE. Reprinted, with permission, from [62]

ogy to address imprecisely labeled training data is multiple instance learning (MIL). In MIL, data is labeled at the *bag* level where a bag is a multi-set of data points as illustrated in Fig. 6.1. In standard MIL, bags are labeled as "positive" if they contain any instances representing a target class whereas bags are labeled as "negative" if they contain only nontarget instances. Generating labels for a bag of points is often much less time consuming and aligns with the realistic scenarios encountered in remote sensing applications as outlined in the following motivating examples.

- *Hyperspectral Classification*: Training a supervised classifier requires accurately labeled spectra for the classes of interest. In practice, this is often accomplished by creating a ground truth map of a hyperspectral scene (scenes which frequently contain hundreds of thousands of pixels or more). Generation of ground truth maps is challenging due to labeling ambiguity that naturally arises due to relatively coarse resolution and compound diversity of the remotely sensed hyperspectral scene. For example, an area that is labeled as vegetation may contain both plants and bare soil, making the training label inherently ambiguous. Furthermore, labeling each pixel of the hyperspectral scene is tedious and annotator performance is generally inconsistent from person to person or over time. Due to these challenges, "ground-rumor" may be a more appropriate term than "ground-truth" for the maps that are generated. These ambiguities naturally map to the MIL framework by allowing an annotator to label spatial regions if it contains a class of interest (corresponding to positive bags) and negative bags for spatial regions known to exclude those classes. For instance, an annotator can easily mark (e.g., circle on a

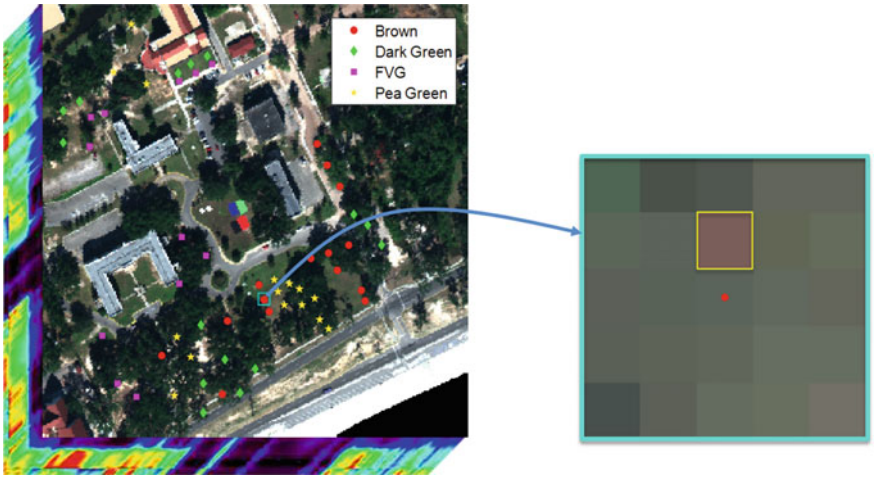

Fig. 6.2 Illustration of inaccurate coordinates from GPS: one target denoted as brown by GPS has one pixel drift. Source: © [2018] Elsevier Reprinted, with permission, from [45]

map) positive bag regions that contain vegetation and then mark regions of only bare soil and building/man-made materials for negative bags when vegetation is the class of interest.

- *Sub-pixel Target Detection*: Consider the hyperspectral target detection problem illustrated in Fig. 6.2. This hyperspectral scene was collected over the University of Southern Mississippi-Gulfpark Campus [1] and includes many emplaced targets. These targets are cloth panels of four colors (Brown, Dark Green, Faux Vineyard Green, and Pea Green) varying from 0.5 m × 0.5 m, 1 m × 1 m, and 3 m × 3 m in size. The ground sample distance of this hyperspectral data set is 1m. Thus, the 0.5 m × 0.5 m targets are, at best, a quarter of a pixel in size; the 1 m × 1 m targets are, at best, exactly one pixel in size; and the 3 m × 3 m targets cover multiple pixels. However, the targets are rarely aligned with the pixel grid, resulting in the 0.5 m × 0.5 m and 1 m × 1 m target responses often straddling multiple pixels and being sub-pixel. The scene also had heavy tree coverage and resulted in targets being heavily occluded by the tree canopy. The sub-pixel nature of the targets and occlusion by the tree canopy causes this to be a challenging target detection problem and one in which manual labeling of target location by visual inspection is impractical. Ground truth locations of the targets in this scene were collected by a GPS unit with 2–5 m accuracy. Thus, the ground truth is only accurate up to some spatial region (as opposed to the pixel level). For example, the region highlighted in Fig. 6.2 contains one brown target. From this highlighted region, one can clearly see that the GPS coordinate of this brown target (denoted by the red dot) is shifted one pixel from the actual brown target location (denoted by the yellow rectangle). This is a rare example where we can visually see the brown target. Most of the targets are difficult to distinguish visibly. Developing a classifier or extracting a

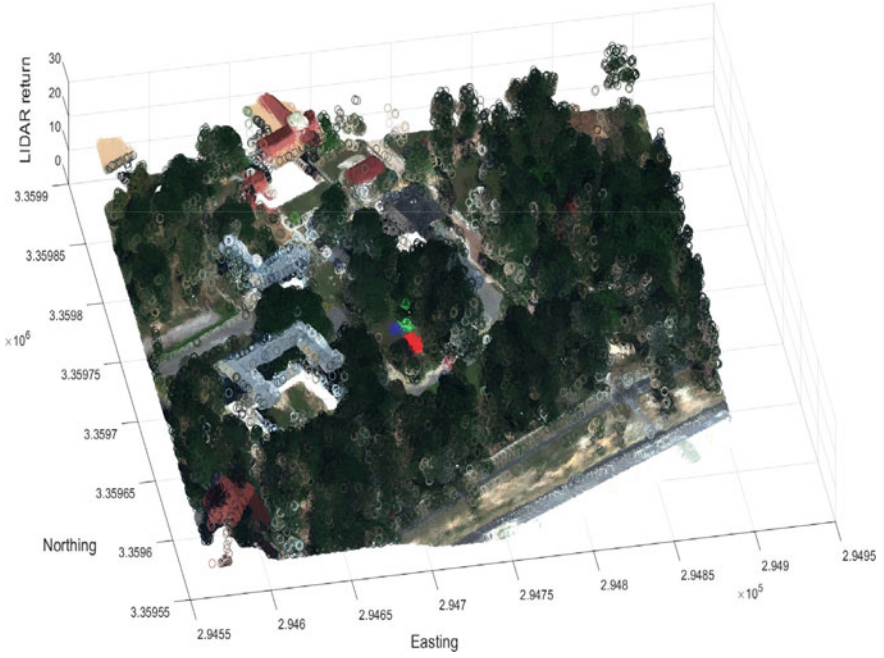

Fig. 6.3 An example of 3D scatterplot of LiDAR data over the University of Southern Mississippi-Gulfpark campus. The LiDAR points were colored by the RGB imagery provided by HSI sensors over the scene. Source: © [2020] IEEE. Reprinted, with permission, from [86]

pure prototype for the target class given incomplete knowledge of the training data is intractable using standard supervised learning methods. This also directly maps to the MIL framework since each positive bag can correspond to the spatial region associated with each ground truth point and its corresponding range of imprecision and negative bags can correspond to spatial regions that do not overlap with any ground truth point or its associated halo of uncertainty.

- *Multi-sensor Fusion*: When fusing information obtained by multiple sensors, each sensor may provide complementary information that can aid scene understanding and analysis. Figure 6.3 shows a three-dimensional scatter plot of the LiDAR (Light Detection And Ranging) point cloud data over the University of Southern Mississippi-Gulfpark Campus collected simultaneously with the hyperspectral imagery (HSI) described above. In this data set, the hyperspectral and LiDAR data can be leveraged jointly for scene segmentation, ground cover classification, and target detection. However, there are challenges that arise during fusion. The HSI and LiDAR data are of drastically different modalities and resolutions. HSI is collected natively on a pixel grid with a 1 m ground sample distance whereas the raw LiDAR data is a point cloud with a higher resolution of 0.60 m cross track and 0.78 m along track spot spacing. Commonly, before fusion, data is co-registered onto a shared pixel grid. However, image co-registration and rasterization may

introduce inaccuracies [2, 3]. In this example, consider the edges of the buildings with gray roofs in Fig. 6.3. Some of the hyperspectral pixels of the buildings have been inaccurately mapped to LiDAR points corresponding to the neighboring grass pixels on the ground. Similarly, some hyperspectral points corresponding to sidewalk and dirt roads have been inaccurately mapped to high elevation values similar to nearby trees and buildings. Directly using such inaccurate measurements for fusion can cause further inaccuracy or error in classification, detection, or prediction. Therefore, it is beneficial to develop a fusion algorithm that is able to handle such inaccurate/imprecise measurements. Imprecise co-registration can also be mapped to the MIL framework by considering a *bag* of points from a local region in one sensor (e.g., LiDAR) to be candidates for fusion in each pixel in the other sensors (e.g., hyperspectral).

These examples illustrate that remote-sensing data and applications are often plagued with inherent spatial imprecision in ground truth information. Multiple instance learning is a framework that can alleviate the issues that arise due to this imprecision. Therefore, although imprecise ground truth plagues instance-level labels, bags (i.e., spatial regions) can be labeled readily and analyzed using MIL approaches.

6.2 Introduction to Multiple Instance Classification

MIL was first proposed by Dietterich et al. [4] for the prediction of drug activity. The effectiveness of a drug is determined by how tightly the drug molecule binds to a larger protein molecule. Although a molecule may be determined to be effective, it can have variants called "conformations" of which only one (or a few) actually binds to the desired target binding site. In this task, the learning objective is to infer the correct shape of the molecule that actually has tight binding capacity. In order to solve this problem, Dietterich et al. introduced the definition of "bags." Each molecule was treated as a bag and each possible conformation of the molecule was treated as an instance in that bag. This directly induces the definition of multiple instance learning. A positively labeled bag contains at least one positive instance (but, also, some number of negative instances) and negatively labeled bags are composed of entirely negative instances. The goal is to uncover the true positive instances in each positive bag and what characterizes positive instances.

Although initially proposed for this drug activity application, the multiple instance learning framework is extremely relevant and applicable to a number of remote-sensing problems arising from imprecision in ground truth information. By labeling data and operating at the bag level, ground truth imprecision inherent in remote sensing problems are addressed and accounted for within a multiple instance learning framework.

6.2.1 Multiple Instance Learning Formulation

The multiple instance learning framework can be formally described as follows. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{n \times N}$ be training data instances where n is the dimensionality of an instance and N is the total number of training instances. The data are grouped into K bags, $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_K\}$, with associated binary bag-level labels, $L = \{L_1, \ldots, L_K\}$ where $L_i \in \{0, 1\}$ for two-class classification. A bag, $\mathbf{B}_i$, is termed *positive* with L_i=1 if it contains at least one positive instance. The exact number or identification of positive and negative instances in each positive bag is unknown. A bag is termed *negative* with L_i=0 when it contains only negative instances. The instance $\mathbf{x}_{ij} \in \mathbf{B}_i$ denotes the jth instance in bag $\mathbf{B}_i$ with the (unknown) instance-level label $l_{ij} \in \{0, 1\}$.

In standard supervised machine learning methods, all instance level labels are known for the training data. However, in multiple instance learning, only the bag-level labels are known. Given this formulation, the fundamental goal of an MIL method is to determine what instance-level characteristics are common across all positive bags and cannot be found in any instance in any negative bag.

6.2.2 Axis-Parallel Rectangles, Diverse Density, and Other General MIL Approaches

Many general MIL approaches have been developed in the literature. Axis-parallel rectangles (APR) [4] algorithms were the first set of MIL algorithms proposed by Dietterich et al. for drug activity prediction in the 1990s. An axis-parallel rectangle can be viewed as a region of true positive instances in the feature space. In APR algorithms, a lower and upper bound encapsulating the positive class is estimated in each feature dimension. Three APR algorithms, greedy feature selection elimination count (GFS elim-count), greedy feature selection kernel density estimation (GFS kde), and iterated discrimination (iterated-discrim) algorithms were investigated and compared in [4]. As an illustration, GFS elim-count APR refers to finding an APR in a greedy manner starting from a region that exactly covers all of the positive instances. Figure 6.4 shows the "all-positive APR" as a solid line bounding box of the instances, where the unfilled markers represent positive instances and filled markers represent negative instances. As shown in the figure, the all-positive APR may contain several negative examples. The algorithm proceeds by greedily eliminating all negative instances within the APR while maintaining as many positive instances as possible. The dashed box in Fig. 6.4 indicates the final APR identified by the GFS elim-count algorithm by iteratively excluding the "cheapest" negative instance, determined by requiring the minimum number of positive instances that need to be removed from the APR to exclude that negative instance.

Diverse density (DD) [5, 6] was one of the first multiple instance learning algorithms that estimated a *positive concept*. The positive concept is a representative of

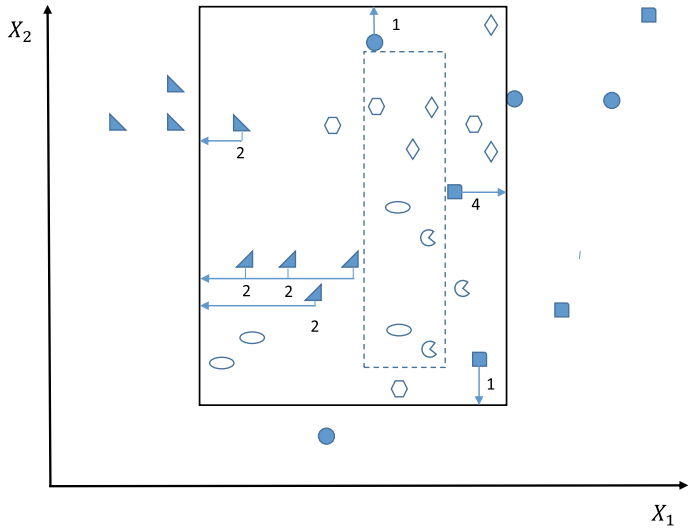

Fig. 6.4 Illustration of the GFS elim-count procedure for excluding negative instances. The "all-positive APR" is indicated by a solid box. The unfilled markers represent positive instances and filled markers represent negative instances. The final APR is indicated by the dashed box [4]

the positive class. This representative is estimated in DD by identifying a representative feature vector that is close to the intersection of all positive bags and far from every negative instance. In other words, the target concept represents an area that preserves both a high density of target points and a low density of nontarget points, called *diverse density*. This is accomplished in DD by maximizing the likelihood function in Eq. (6.1),

$$\arg\max_{d} \prod_{i=1}^{K^+} \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{B}_i^+) \prod_{i=K^++1}^{K^++K^-} \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{B}_i^-), \tag{6.1}$$

where $\mathbf{s}$ is the assumed true positive concept, $\mathbf{d}$ is the concept representative to be estimated, K^+ is the number of positive bags and K^- is the number of negative bags. The first term in Eq. (6.1), which is used for all positive bags, is defined by the noisy-or model,

$$\mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{B}_i^+) = \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iN_i}) = 1 - \prod_{j=1}^{N_i}(1 - \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{ij} \in \mathbf{B}_i^+)), \tag{6.2}$$

where $\mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{ij}) = exp(-\|\mathbf{x}_{ij} - \mathbf{d}\|^2)$. The term in (6.2) can be interpreted as requiring there be at least one instance in positive bag $\mathbf{B}_i^+$ that is close to the positive representative $\mathbf{d}$. This can be understood by noticing that (6.2) evaluates to 1 if

there is at least one instance in the positive bag that is close to the representative (i.e., $exp(-\|\mathbf{x}_{ij} - \mathbf{d}\|^2) \rightarrow 1$ which implies $1 - \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{ij} \in \mathbf{B}_i^+) \rightarrow 0$, resulting in $1 - \prod_{j=1}^{N_i}(1 - \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{ij} \in \mathbf{B}_i^+)) \rightarrow 1$). In contrast, (6.2) evaluates to 0 if all points in a positive bag are far from the positive concept.

The second term is defined by

$$\mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{B}_i^-) = \prod_{j=1}^{N_i}(1 - \mathbf{Pr}(\mathbf{d} = \mathbf{s}|\mathbf{x}_{ij} \in \mathbf{B}_i^-)). \tag{6.3}$$

which encourages positive concepts to be far from all negative points. The noisy-or model, however, is highly non-smooth and there are several local maxima in the solution space. This is alleviated in practice by performing gradient ascent repeatedly with starting points from every positive instance to maximize the proposed log-likelihood function. Alternatively, an expectation maximization version of diversity density (EM-DD) [7] was proposed by Zhang et al. in order to improve the computation time of DD [5, 6]. EM-DD assumes there exists only one instance per bag corresponding to the bag-level label and treats the knowledge of the key-point instance corresponding to the bag-level label as a hidden latent variable. EM-DD starts with an initial estimate of the positive concept $\mathbf{d}$ and iterates between an expectation step (E-step) that selects one point per bag as the representative point of that bag and then performs a quasi-newton optimization (M-step) [8] on the single-instance DD problem. In practice, EM-DD is much more computationally efficient than DD. However, the computational benefits are traded-off with potential inferior performance accuracy to DD [9].

Since the development of the APR and DD, many MIL approaches have been developed and published in the literature. These include prototype-based methods such as the dictionary-based multiple instance learning (DMIL) algorithm [10] and its generalization, generalized dictionaries for multiple instance Learning (GDMIL) [11] which propose to optimize the noisy-or model using dictionary learning approaches by learning a set of discriminative positive dictionary atoms to describe the positive class [12–14]. The Max-Margin Multiple-Instance Dictionary Learning (MMDL) methods [15] adopts the bag of words concept [16] and trains a set of linear SVMs as a codebook. The novel assumption of MMDL is that the positive instances could belong to many different categories. For example, the positive class "computer room" may have image patches containing a desk, a screen, and a keyboard. The MILIS algorithm [17] alternates between the selection of an instance per bag as a prototype that represents its bag and training a linear SVM on these prototypes.

Additional support vector machine-based methods include the MILES (Multiple-Instance Learning via Embedded Instance Selection) approach [18] which embeds each training and testing bag into a high-dimensional space and then performs classification in the mapping space using a one-norm support vector machine (SVM) [19]. Furthermore, the mi-SVM and MI-SVM methods model the MIL problem as a generalized mixed integer formulation of the support vector machine [20]. MissSVM

algorithm [21] solves the MIL problem using a semi-supervised SVM with the constraint that at least one point from each positive bag must be classified as positive. Hoffman et al. [22] jointly exploit the image-level and bounding box labels and achieve state-of-the-art results in object detection. Li and Vasconcelos [23] further investigate MIL problem with labeling noise in negative bags and use "top instances" as the representatives of "soft bags", then proceed with bag-level classification via latent-SVM [24].

Meng et al. [25] integrate the self-paced learning (SPL) [26] into MIL and propose SP-MIL for co-saliency detection. The Citation-*k*NN [27] algorithm adapts the *k* nearest neighbor (*k*NN) method [28] to MIL problems by using the Hausdorff distance [29] to compute distance between two bags and assigns bag-level labels based on the nearest neighbor rules. Extensions of Citation-*k*NN include Bayesian Citation-*k*NN [30] and Fuzzy-Citation-*k*NN [31, 32]. Furthermore, a large number of MIL neural network methods such as [33] (often called "weak" learning methods) have also been developed. Among the vast literature of MIL research, very few methods focus on remote sensing and hyperspectral analysis. These methods are reviewed in the following sections.

6.3 Multiple Instance Learning Approaches for Hyperspectral Target Characterization and Sub-pixel Target Detection

Hyperspectral target detection refers to the task of locating all instances of a target given a known spectral signature within a hyperspectral scene [34–36]. Hyperspectral target detection is challenging for a number of reasons: (1) *Class Imbalance*: The number of training instances from the positive target class is small compared to that of the negative training data such that training a standard classifier is difficult; (2) *Sub-pixel Targets*: Due to the relatively low spatial resolution of hyperspectral imagery and the diversity of natural scenes, one single pixel may also contain different ground materials, resulting in sub-pixel targets of interest; and (3) *Imprecise Labels*: As outlined in Sect. 6.1, precise training labels are often difficult to obtain. For these reasons, signature-based hyperspectral target detection [34] is commonly used as opposed to a two-class classifier. However, the performance of a signature-based detector depends on the target signature and obtaining an effective target signature is challenging. In the past, this was commonly accomplished by measuring target signatures for materials of interest in the lab or using point-spectrometers in the field. However, this approach may introduce error due to changing environmental and atmospheric conditions that impact spectral responses.

In this section, algorithms for multiple instance target characterization (i.e., estimation of target concepts) from training data with label ambiguity are presented. The aim is to estimate the target concepts from highly mixed training data that are effective for target detection. Since these algorithms extract target concepts from

training data assumed to have the same environmental context, influence from background materials, environmental and atmospheric conditions are addressed during target concept estimation.

6.3.1 Extended Function of Multiple Instances

The extended Function of Multiple Instances (*e*FUMI) approach [37, 38] is motivated by the linear mixing model in hyperspectral analysis. *e*FUMI assumes each data point is a convex combination of target and/or nontarget concepts (i.e., endmembers) and performs linear unmixing (i.e., decomposing spectra into endmembers and the proportion of each endmember found in the associated pixel spectra) to estimate positive and negative concepts. The approach also addresses label ambiguity by incorporating a latent variable which indicates whether each instance of a positively labeled bags is a true target.

More formally, the goal of *e*FUMI is to estimate a target concept, $\mathbf{d}_T$, nontarget concepts, $\mathbf{d}_k$, $\forall k = 1, \ldots M$, the number of needed nontarget concepts, M, and the abundances, $\mathbf{a}_j$, which define the convex combination of the concepts for each data point $\mathbf{x}_j$ from labeled bags of hyperspectral data. If a bag B_i is positive, there is at least one data point in B_i containing target,

$$\text{if } L_i = 1, \exists \mathbf{x}_j \in B_i \text{ s.t. } \mathbf{x}_j = \alpha_{jT}\mathbf{d}_T + \sum_{k=1}^{M} \alpha_{jk}\mathbf{d}_k + \boldsymbol{\varepsilon}_j, \alpha_{jT} > 0. \tag{6.4}$$

However, the exact number of data points in a positive bag with a target contribution (i.e., $\alpha_{jT} > 0$) and target proportions are unknown. Furthermore, if B_i is a negative bag, this indicates that none of the data in this bag contains target,

$$\text{if } L_i = 0, \forall \mathbf{x}_j \in B_i, \mathbf{x}_j = \sum_{k=1}^{M} \alpha_{jk}\mathbf{d}_k + \boldsymbol{\varepsilon}_j. \tag{6.5}$$

Given this framework, the *e*FUMI objective function is shown in (6.7). The three terms in this objective function were motivated by the sparsity promoting iterated constrained endmember (SPICE) algorithm [39]. The first term computes the squared error between the input data and its estimate found using the current target and nontarget signatures and proportions. The parameter u is a constant controlling the relative importance of various terms. The scaling value w, which aids in the data imbalance issue by weighting the influence of positive and negative data, is shown in (6.6),

$$w_{l(\mathbf{x}_j)} = \begin{cases} 1, & \text{if } l(\mathbf{x}_j) = 0; \\ \frac{\alpha N^-}{N^+}, & \text{if } l(\mathbf{x}_j) = 1. \end{cases}, \tag{6.6}$$

where N^+ is the total number of points in positive bags and N^- is the total number of points in negative bags.

The second term of the objective encourages target and nontarget signatures to provide a tight fit around the data by minimizing the squared difference between each signature and the global data mean, $\boldsymbol{\mu}_0$. The third term is a sparsity promoting term used to determine M, the number of nontarget signatures needed to describe the input data where $\gamma_k = \frac{\Gamma}{\sum_{j=1}^{N} a_{jk}^{(t-1)}}$ and Γ is a constant parameter that controls the degree sparsity is promoted. Higher values of Γ generally result in a smaller estimate M value. The $a_{jk}^{(t-1)}$ values are the proportion values estimated in the previous iteration of the algorithm. Thus, as the proportions for a particular endmember decrease, the weight of its associated sparsity promoting term increases.

$$F = \frac{1}{2}(1-u)\sum_{j=1}^{N} w_j \left\| (\mathbf{x}_j - z_j a_{jT}\mathbf{d}_T - \sum_{k=1}^{M} a_{jk}\mathbf{d}_k) \right\|_2^2 + \frac{u}{2}\sum_{k=T,1}^{M} \left\| \mathbf{d}_k - \boldsymbol{\mu}_0 \right\|_2^2 + \sum_{k=1}^{M}\gamma_k \sum_{j=1}^{N} a_{jk} \tag{6.7}$$

$$E[F] = \sum_{z_j \in \{0,1\}} \left[\frac{1}{2}(1-u)\sum_{j=1}^{N} w_j P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)}) \left\| \mathbf{x}_j - z_j a_{jT}\mathbf{d}_T - \sum_{k=1}^{M} a_{jk}\mathbf{d}_k \right\|_2^2 \right] + \frac{u}{2}\sum_{k=T,1}^{M} \|\mathbf{d}_k - \boldsymbol{\mu}_0\|_2^2 + \sum_{k=1}^{M}\gamma_k \sum_{j=1}^{N} a_{jk} \tag{6.8}$$

The difference between (6.7) and the SPICE objective is the inclusion of a set of hidden, latent variables, z_j, $j = 1, \ldots, N$, accounting for the unknown instance-level labels $l(\mathbf{x}_j)$. To address the fact that the z_j values are unknown, the expected values of the log likelihood with respect to z_j is taken as shown in (6.8). In (6.8), $\boldsymbol{\theta}^t$ is the set of parameters estimated at iteration t and $P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)})$ is the probability of individual points containing any proportion of target or not. The value of the term $P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)})$ is determined given the parameter set estimated in the previous iteration and the constraints of the bag-level labels, L_i, as shown in (6.9),

$$P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)}) = \begin{cases} e^{-\beta r_j}, & \text{if } z_j = 0, L_i = 1; \\ 1 - e^{-\beta r_j}, & \text{if } z_j = 1, L_i = 1; \\ 0, & \text{if } z_j = 1, L_i = 0; \\ 1, & \text{if } z_j = 0, L_i = 0; \end{cases} \tag{6.9}$$

where β is a scaling parameter and $r_j = \left\| \mathbf{x}_j - \sum_{k=1}^{M} a_{jk}\mathbf{d}_k \right\|_2^2$ is the approximation residual between $\mathbf{x}_j$ and its representation using only background endmembers. The definition of $P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)})$ in (6.9) indicates that if a point $\mathbf{x}_j$ is a nontarget point, it should be fully represented by the background endmembers with very small residual r_j; thus, $P(z_j = 0|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)}) = e^{-\beta r_j} \to 1$. Otherwise, if $\mathbf{x}_j$ is a target point, it may not be well represented by only the background endmembers, so the residual r_j must

be large and $P(z_j = 1|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)}) = 1 - e^{-\beta r_j} \rightarrow 1$. Note, z_j is unknown only for the positive bags; in the negative bags, z_j is fixed to 0. This constitutes the *E-step* of the EM algorithm.

The *M-step* is performed by optimizing (6.8) for each of the desired parameters. The method is summarized in Algorithm 6.1.[1] Please refer to [37] for detailed discussion of the optimization approach and derivation.

Algorithm 6.1 *e*FUMI EM algorithm

1: Initialize $\boldsymbol{\theta}^0 = \{\mathbf{d}_T, \mathbf{D}, \mathbf{A}\}$, $t = 1$
2: **repeat**
3: ***E-step***: Compute $P(z_j|\mathbf{x}_j, \boldsymbol{\theta}^{(t-1)})$ given $\boldsymbol{\theta}^{t-1}$
4: ***M-step***:
5: Update $\mathbf{d}_T$ and $\mathbf{D}$ by maximizing (6.8) wrt. $\mathbf{d}_T$, $\mathbf{D}$
6: Update $\mathbf{A}$ by maximizing (6.8) wrt. $\mathbf{A}$ s.t. the sum-to-one and non-negative constraints
7: Prune each $\mathbf{d}_k, k = 1, \ldots, M$ if $\max_j(a_{jk}) \leq \tau$ where τ is a fixed threshold (e.g. $\tau = 10^{-6}$)
8: $t \leftarrow t + 1$
9: **until** Convergence
10: **return** $\mathbf{d}_T$, $\mathbf{D}$, $\mathbf{A}$

6.3.2 *Multiple Instance Spectral Matched Filter and Multiple Instance Adaptive Coherence/Cosine Detector*

The *e*FUMI algorithm described above can be viewed as a semi-supervised hyperspectral unmixing algorithm, where the endmembers of the target and nontarget materials are estimated. Since *e*FUMI minimizes the reconstruction error of the data, it is a *representative* algorithm that learns target concepts that are representatives for (and have similar shape to) the target class. Significant challenges in applying the *e*FUMI algorithm in practice are the large number of parameters that need to be set and the fact that all positive bags are combined in the algorithm, neglecting the MIL concept that each positive bag contains at least one target instance.

In contrast, the multiple instance spectral matched filter (MI-SMF) and multiple instance adaptive coherence/cosine detector (MI-ACE) [41] learn *discriminative* target concepts that maximize the SMF or ACE detection statistics, which preserves bag structure and does not require tuning parameter settings. These goals are accomplished by optimizing the following objective function,

$$\arg\max_{\mathbf{s}} \frac{1}{K^+} \sum_{i:L_i=1} \Lambda(\mathbf{x}_i^*, \mathbf{s}) - \frac{1}{K^-} \sum_{i:L_i=0} \frac{1}{N_i^-} \sum_{\mathbf{x}_{ij} \in B_i^-} \Lambda(\mathbf{x}_{ij}, \mathbf{s}), \tag{6.10}$$

[1] The *e*FUMI implementation is available at: https://github.com/GatorSense/FUMI [40].

where $\mathbf{s}$ is the target signatures, $\Lambda(\mathbf{x}, \mathbf{s})$ is the detection statistics of data point $\mathbf{x}$ given target signature $\mathbf{s}$, and $\mathbf{x}_i^*$ is the selected representative instance from the positive bag B_i^+, K^+ is the number of positive bags and K^- is the number of negative bags.

$$\mathbf{x}_i^* = \arg\max_{\mathbf{x}_{ij} \in B_i^+} \Lambda(\mathbf{x}_{ij}, \mathbf{s}). \tag{6.11}$$

This general objective can be applied to any target detection statistics. However, consider the ACE detector, $\Lambda_{ACE}(\mathbf{x}, \mathbf{s}) = \frac{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}(\mathbf{x}-\boldsymbol{\mu}_b)}{\sqrt{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}\mathbf{s}}\sqrt{(\mathbf{x}-\boldsymbol{\mu}_b)^T \boldsymbol{\Sigma}_b^{-1}(\mathbf{x}-\boldsymbol{\mu}_b)}}$, where $\boldsymbol{\mu}_b$ is the mean of the background and $\boldsymbol{\Sigma}_b$ is the background covariance. This detection statistic can be viewed as an inner product in a whitened coordinate space

$$\begin{aligned}
\Lambda_{ACE}(\mathbf{x}, \mathbf{s}) &= \frac{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}(\mathbf{x} - \boldsymbol{\mu}_b)}{\sqrt{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}\mathbf{s}}\sqrt{(\mathbf{x} - \boldsymbol{\mu}_b)^T \boldsymbol{\Sigma}_b^{-1}(\mathbf{x} - \boldsymbol{\mu}_b)}} \\
&= \frac{\mathbf{s}^T \mathbf{U}\mathbf{V}^{-\frac{1}{2}}\mathbf{V}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}_b)}{\sqrt{\mathbf{s}^T \mathbf{U}\mathbf{V}^{-\frac{1}{2}}\boldsymbol{V}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{s}}\sqrt{(\mathbf{x} - \boldsymbol{\mu}_b)^T \mathbf{U}\mathbf{V}^{-\frac{1}{2}}\mathbf{V}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}_b)}} \\
&= \left(\frac{\hat{\mathbf{s}}}{\|\hat{\mathbf{s}}\|}\right)^T \left(\frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|}\right) \\
&= \hat{\hat{\mathbf{s}}}^T \hat{\hat{\mathbf{x}}},
\end{aligned} \tag{6.12}$$

where $\hat{\mathbf{x}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}_b)$, $\hat{\mathbf{s}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{s}$, $\mathbf{U}$ and $\mathbf{V}$ are the eigenvectors and eigenvalues of the background covariance matrix $\boldsymbol{\Sigma_b}$, respectively, $\hat{\hat{\mathbf{s}}} = \frac{\hat{\mathbf{s}}}{\|\hat{\mathbf{s}}\|}$, and $\hat{\hat{\mathbf{x}}} = \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|}$. It is clear from Eq. (6.12) that the ACE detector response is the cosine value between a test data point, $\mathbf{x}$, and a target signature, $\mathbf{s}$, after whitening. Thus, the objective function (6.10) for MI-ACE can be rewritten as

$$\arg\max_{\hat{\hat{\mathbf{s}}}} \frac{1}{K^+} \sum_{i:L_i=1} \hat{\hat{\mathbf{s}}}^T \hat{\hat{\mathbf{x}}}_i^* - \frac{1}{K^-} \sum_{i:L_i=0} \frac{1}{N_i^-} \sum_{\mathbf{x}_{ij} \in B_i^-} \hat{\hat{\mathbf{s}}}^T \hat{\hat{\mathbf{x}}}_{ij}, \text{ such that } \hat{\hat{\mathbf{s}}}^T \hat{\hat{\mathbf{s}}} = 1. \tag{6.13}$$

The l_2 norm constraint, $\hat{\hat{\mathbf{s}}}^T \hat{\hat{\mathbf{s}}} = 1$, is resulted from the normalization term in Eq. (6.12). The optimum for (6.13) can be derived by solving the Lagrangian optimization problem for the target signature

$$\hat{\hat{\mathbf{s}}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}, \text{ where } \mathbf{t} = \frac{1}{K^+} \sum_{i:L_i=1} \hat{\hat{\mathbf{x}}}_i^* - \frac{1}{K^-} \sum_{i:L_i=0} \frac{1}{N_i^-} \sum_{\mathbf{x}_{ij} \in B_i^-} \hat{\hat{\mathbf{x}}}_{ij}. \tag{6.14}$$

A similar approach can be applied for the spectral matched filter detector,

$$\Lambda_{SMF}(\mathbf{x}, \mathbf{s}) = \frac{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}(\mathbf{x} - \boldsymbol{\mu}_b)}{\sqrt{\mathbf{s}^T \boldsymbol{\Sigma}_b^{-1}\mathbf{s}}}, \tag{6.15}$$

resulting in the following update equation for MI-SMF:

$$\hat{\mathbf{s}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}, \text{ where } \mathbf{t} = \frac{1}{K^+}\sum_{i:L_i=1}\hat{\mathbf{x}}_i^* - \frac{1}{K^-}\sum_{i:L_i=0}\frac{1}{N_i^-}\sum_{\mathbf{x}_{ij}\in B_i^-}\hat{\mathbf{x}}_{ij}. \tag{6.16}$$

Algorithm 6.2 MI-SMF/MI-ACE

1: Compute $\boldsymbol{\mu}_b$ and $\boldsymbol{\Sigma}_b$ as the mean and covariance of all instances in the negative bags
2: Subtract the background mean and whiten all instances, $\hat{\mathbf{x}} = \mathbf{V}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu}_b)$
3: If MI-ACE, normalize: $\hat{\hat{\mathbf{x}}} = \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|}$
4: Initialize $\hat{\hat{\mathbf{s}}}$ using the instance in a positive bag resulting in largest objective function value
5: **repeat**
6: Update the selected instances, $\mathbf{x}_i^*$, for each positive bag, $\mathbf{B}_i^+$ using (6.11)
7: Update $\hat{\hat{\mathbf{s}}}$ using (6.14) for MI-ACE or (6.16) for MI-SMF
8: **until** Stopping Criterion Reached
9: **return** $\mathbf{s} = \frac{\mathbf{t}}{\|\mathbf{t}\|}$, where $\mathbf{t} = \mathbf{U}\mathbf{V}^{\frac{1}{2}}\hat{\hat{\mathbf{s}}}$

The MI-SMF and MI-ACE algorithms alternate between the two steps: (1) selecting representative instances from each positive bag and (2) updating the target concept $\mathbf{s}$. The MI-SMF and MI-ACE methods stop when there is no change in the selection of instances from positive bags across subsequent iterations. Similar to [7], since there exists a finite set of possible selection of positive instances given a finite training bags, the convergence of MI-SMF and MI-ACE is guaranteed. In the experiments shown in [41], MI-SMF and MI-ACE generally converged with less than seven iterations. The MI-SMF/MI-ACE algorithm is summarized in Algorithm 6.2.[2] Please refer to [41] for a detailed derivation of the algorithm.

6.3.3 Multiple Instance Hybrid Estimator

Both *e*FUMI and the MI-ACE/MI-SMF methods are limited in that they only estimate a single target concept. However, in many problems, the target class has significant spectral variability [43]. The Multiple Instance Hybrid Estimator (MI-HE) [44, 45] was developed to fill this gap and estimate multiple target concepts simultaneously.

[2]The MI-SMF and MI-ACE implementations are available at: https://github.com/GatorSense/MIACE [42].

The proposed MI-HE algorithm maximizes the responses of the hybrid sub-pixel detector [46] within the MIL framework. This is accomplished by maximizing the following objective function:

$$\begin{aligned} J &= \ln \prod_{i=1}^{K^+} \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \Pr(l_{ij} = 1 | \mathbf{B}_i)^b \right)^{\frac{1}{b}} \prod_{i=K^++1}^{K} \prod_{j=1}^{N_i} \Pr(l_{ij} = 0 | \mathbf{B}_i) \\ &= -\sum_{i=1}^{K^+} \frac{1}{b} \ln \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \exp\left(-\beta \frac{\|\mathbf{x}_{ij} - \mathbf{D}\mathbf{a}_{ij}\|^2}{\|\mathbf{x}_{ij} - \mathbf{D}^- \mathbf{p}_{ij}\|^2} \right)^b \right) \\ &\quad + \rho \sum_{i=K^++1}^{K} \sum_{j=1}^{N_i} \|\mathbf{x}_{ij} - \mathbf{D}^- \mathbf{p}_{ij}\|^2 \\ &\quad + \frac{\alpha}{2} \sum_{i=K^++1}^{K} \sum_{j=1}^{N_i} \left((\mathbf{D}^+ \mathbf{a}_{ij}^+)^T \mathbf{x}_{ij} \right)^2, \end{aligned} \tag{6.17}$$

where the first term corresponds to a generalized mean (GM) term [47], which can approximate the *max* operation as b approaches $+\infty$. This term can be interpreted as determining a representative positive instance in each positive bag by identifying the instance that maximizes the hybrid sub-pixel detector (HSD) [46] statistic, $\exp\left(-\beta \frac{\|\mathbf{x}_{ij} - \mathbf{D}\mathbf{a}_{ij}\|^2}{\|\mathbf{x}_{ij} - \mathbf{D}^- \mathbf{p}_{ij}\|^2}\right)$. In the HSD, each instance is modeled as a sparse linear combination of target and/or background concepts $\mathbf{D}$, $\mathbf{x} \approx \mathbf{D}\mathbf{a}$, where $\mathbf{D} = \left[\mathbf{D}^+ \ \mathbf{D}^-\right] \in \mathbb{R}^{d \times (T+M)}$, $\mathbf{D}^+ = [\mathbf{d}_1, \ldots, \mathbf{d}_T]$ is the set of T target concepts and $\mathbf{D}^- = \left[\mathbf{d}_{T+1}, \ldots, \mathbf{d}_{T+M}\right]$ is the set of M background concepts, β is a scaling parameter, and $\mathbf{a}_{ij}$ and $\mathbf{p}_{ij}$ are the sparse representation of $\mathbf{x}_{ij}$ given the entire concept set $\mathbf{D}$ and background concept set $\mathbf{D}^-$, respectively. The second term in the objective function is viewed as the background data fidelity term, which is based on the assumption that minimizing the least squares of all negative points provides a good description of the background. The scaling factor ρ is usually set to be smaller than one to control the influence of negative bags. The third term is the cross incoherence term (motivated by the Dictionary Learning with Structured Incoherence [48] and the Fisher discrimination dictionary learning (FDDL) algorithm [49, 50]) that encourages positive concepts to have distinct spectral signatures from negative points.

The initialization of target concepts in $\mathbf{D}$ is conducted by computing the mean of T random subsets drawn from the union of all positive training bags. The vertex component analysis (VCA) [53] method was applied to the union of all negative bags and the M cluster centers (or vertices) were set as the initial background concepts. The pseudocode of the MI-HE algorithm is presented in Algorithm 6.3.[3] Please refer to [44] for a detailed optimization derivation.

[3]The MI-HE implementation is available at: https://github.com/GatorSense/MIHE [54].

Algorithm 6.3 MI-HE algorithm

Input: MIL training bags $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_K\}$, MI-HE parameters

Initialize $\mathbf{D}^0$, $iter = 0$
repeat
  for $t = 1, \ldots, T$ **do**
    Solve $\mathbf{a}_{ij}$, $\mathbf{p}_{ij}$, $\forall i \in \{1, \ldots, K\}$, $j \in \{1, \ldots, N_i\}$ using the iterative shrinkage-thresholding algorithm [51, 52]
    Update $\mathbf{d}_t$ using gradient descent
    $\mathbf{d}_t \leftarrow \frac{1}{\|\mathbf{d}_t\|_2}\mathbf{d}_t$
  end for
  for $k = T + 1, \ldots, T + M$ **do**
    Solve $\mathbf{a}_{ij}$, $\mathbf{p}_{ij}$, $\forall i \in \{1, \ldots, K\}$, $j \in \{1, \ldots, N_i\}$ using the iterative shrinkage-thresholding algorithm [51, 52]
    Update $\mathbf{d}_k$ using gradient descent
    $\mathbf{d}_k \leftarrow \frac{1}{\|\mathbf{d}_k\|_2}\mathbf{d}_k$
  end for
  $iter \leftarrow iter + 1$
until Stopping criterion reached
return $\mathbf{D}$

6.3.4 *Multiple Instance Learning for Multiple Diverse Hyperspectral Target Characterizations*

The multiple instance learning of multiple diverse characterizations for SMF (MILMD-SMF) and ACE detector (MILMD-ACE) [55] is an extension of MI-ACE and MI-SMF that learns multiple target signatures for characterization of the variability in hyperspectral target concepts. Different from the MI-HE method explained above, the MILMD-SMF and MILMD-ACE methods do not model target and background signatures explicitly. Instead, the MILMD-SMF and MILMD-ACE methods focus on maximizing the detection statistics of the positive bags and capturing the characteristics of the training data using a set of diverse target signatures, as shown below:

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} \prod_i P(\mathbf{S}|B_i, L_i = 1) \prod_i P(\mathbf{S}|B_i, L_i = 0), \tag{6.18}$$

where $\mathbf{S} = \left\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots \mathbf{s}^{(\mathrm{K})}\right\}$ is the K assumed target signatures and $P(\mathbf{S}|B_i, L_i = 1)$ and $P(\mathbf{S}|B_i, L_i = 0)$ denote the probabilities given the positive and negative bags, respectively. The authors consider the following equivalent form of (6.18) for multiple target characterization can be shown as

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} \{C_1(\mathbf{S}) + C_2(\mathbf{S})\}, \tag{6.19}$$

$$C_1(\mathbf{S}) = \frac{1}{N^+} \sum_{i:L_i=1} \Omega(D,\ X_i^*, \mathbf{S}), \tag{6.20}$$

$$C_1(\mathbf{S}) = -\frac{1}{N^-}\sum_{i:L_i=0}\Upsilon(D,\ X_i, \mathbf{S}), \tag{6.21}$$

where $\Omega(\cdot)$ and $\Upsilon(\cdot)$ are defined to capture the detection statistics of the positive and negative bags, $D(\cdot)$ is detection response of the given ACE or SMF detectors and $\mathbf{X}_i^* = \{\mathbf{x}_i^{(1)*}, \mathbf{x}_i^{(2)*}, \ldots, \mathbf{x}_i^{(K)*}\}$ is the subset of the ith positive bag of selected instances with maximum detection responses corresponding to one of the target signatures $\mathbf{s}^k$ such that

$$x_i^{(k)*} = \underset{\mathbf{x}_n \in \mathbf{B}_i, L_i=1}{\arg\max}\ D(\mathbf{x}_n, \mathbf{s}^{(k)}). \tag{6.22}$$

The term $\Omega(D,\ X_i^*, \mathbf{S})$ is the global detection statistics term for the positive bags whose ACE form is shown in

$$\Omega_{ACE}(D,\ X_i^*, \mathbf{S}) = \frac{1}{K}\sum_k \hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{x}}}_i^{(k)*}. \tag{6.23}$$

Similar to [41], $\hat{\hat{\mathbf{s}}}^{(k)}$ and $\hat{\hat{\mathbf{x}}}^{(k)}$ are the transformed kth target signature and correspond instance after whitening using the background information and normalization. The global detection term $\Omega_{ACE}(D,\ X_i^*, \mathbf{S})$ provides an average detection statistics over the positive bags given a set of learned target signatures. Of particular note for this method, in contrast with MI-HE, is the approach assumes that each positive bag contains a representative for each variation of the positive concept.

On the other hand, the global detection term $\Upsilon_{ACE}(D,\ X_i, \mathbf{S})$ for negative instances should be small and thus suppresses the background as shown in Eq. (6.24). This definition means if the maximum responses of target signature set $\mathbf{S}$ over the negative instances are minimized, the estimated target concepts can effectively discriminate nontarget training instances

$$\Upsilon_{ACE}(D,\ X_i, \mathbf{S}) = \frac{1}{N_{i,L_i=0}}\sum_{\mathbf{x}_n \in \mathbf{B}_i, L_i=0}\max_k\ \hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{x}}}_n. \tag{6.24}$$

In order to explicitly apply the normalization constraint and encourage diversity in the estimated multiple target concepts, [55] also includes two terms, a normalization term by pushing the inner product of the estimated signatures to 1 and a diversity promoting term by maximizing the difference between estimated target concepts as shown in (6.25), and (6.26), respectively.

$$C^{div}(\mathbf{S}) = -\frac{2}{K(K-1)}\sum_{k,l,k\neq l}\hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{s}}}^{(l)}, \tag{6.25}$$

$$C^{con}(\mathbf{S}) = -\frac{1}{K}\sum_k\left|\hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{s}}}^{(k)} - 1\right|. \tag{6.26}$$

Combining the global detection statistics, the diversity promoting and normalization constraint terms, the final cost function is shown as (6.27).

$$C_{ACE} = \frac{1}{N^+}\sum_{i:L_i=1}\sum_{k}\frac{1}{K}\hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{x}}}_i^{(k)*} - \frac{1}{N^-}\sum_{i:L_i=0}\frac{1}{N_{i,L_i=0}}\sum_{\mathbf{x}_n\in\mathbf{B}_i,L_i=0}\max_k \hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{x}}}_n$$
$$-\frac{2\alpha}{K(K-1)}\sum_{k,l,k\neq l}\hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{s}}}^{(l)} - \frac{\lambda}{K}\sum_{k}\left|\hat{\hat{\mathbf{s}}}^{(k)T}\hat{\hat{\mathbf{s}}}^{(k)} - 1\right|. \tag{6.27}$$

The objective for SMF can be similarly derived, where the only difference is the use of training data without normalization. For the optimization of Eq. (6.27), gradient descent is applied. Since the $max(\cdot)$ and $|\cdot|$ operators are not differentiable at zero, the noisy-or function is adopted as an approximation for $max(\cdot)$ and a subgradient method is performed to compute the gradient of $|\cdot|$. Please refer to [55] for a detailed optimization derivation.

6.3.5 *Experimental Results for MIL in Hyperspectral Target Detection*

In this section, several MIL learning methods on both simulated and real hyperspectral detection tasks are evaluated to illustrate the properties of these algorithms and provide insight into how and when these methods are effective.

For the experiments conducted in this paper, the parameter settings of the comparison algorithms were optimized using a grid search on the first task of each experiment and then applied to the remaining tasks. For example, for mi-SVM classifier on the Gulfport Brown target task, the γ value of the RBF kernel was firstly varied from 0.5 to 5 at a step size of 0.5, and then a finer search around the current best value (with the highest AUC) at a step of 0.1 was performed. For algorithms with stochastic result, e.g., EM-DD, eFUMI, each parameter setting was run five times and the median performance was selected. Finally the optimal parameters that achieve the highest AUC for the brown target were selected and used for the other three target types.

6.3.5.1 Simulated Data

As discussed in Sect. 6.3.1, the *e*FUMI algorithm combines all positive bags as one big positive bag and all negative bags as one big negative bag and learns target concept from the big positive bag that is different from the negative bag. Thus, if the negative bags contain incomplete knowledge of the background, e.g., some nontarget concept appears only in the subset of positive bags, *e*FUMI will perform poorly. However, the discriminative MIL algorithms, e.g., MI-HE, MI-ACE, and MI-SMF, maintain bag structure and can distinguish the target.

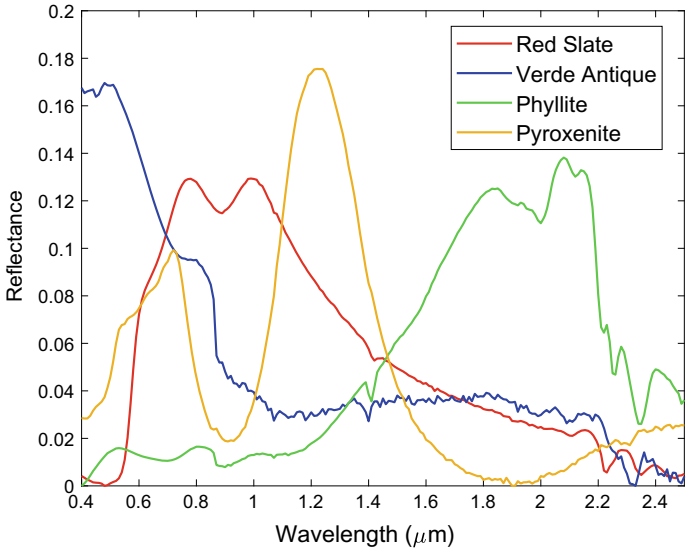

Fig. 6.5 Signatures from ASTER library used to generate simulated data

Given this hypothesis, simulated data was generated from four spectra selected from the ASTER spectral library [56]. Specifically, the Red Slate, Verde Antique, Phyllite, and Pyroxenite spectra from the rock class with 211 bands and wavelengths ranging from 0.4 to 2.5 μm (as shown in Fig. 6.5 in solid lines) were used as endmembers to generate hyperspectral data. Red Slate was labeled as the target endmember.

Four sets of highly mixed noisy data with varied mean target proportion value ($\boldsymbol{\alpha}_{t\_mean}$) were generated, a detailed generation process can be found in [37]. Specifically, this synthetic data has 15 positive and 5 negative bags with each bag having 500 points. If it is a positively labeled bag, there are 200 highly mixed target points containing mean target (Red Slate) proportion from 0.1 to 0.7, respectively, to vary the level of target presence from weak to high. Gaussian white noise was added so that signal-to-noise ratio of the data was set to 20 dB. To highlight the ability of MI-HE, MI-ACE and MI-SMF to leverage individual bag-level labels, we use different subsets of background endmembers to build synthetic data as shown in Table 6.1.

Table 6.1 List of constituent endmembers for synthetic data with incomplete background Knowledge

Bag no.	Bag label	Target endmember	Background endmember
1–5	+	Red slate	Verde Antique, Phyllite, Pyroxenite
6–10	+	Red slate	Phyllite, Pyroxenite
11–15	+	Red slate	Pyroxenite
16–20	−	N/A	Phyllite, Pyroxenite

Table 6.1 shows that the negatively labeled bags only contain two negative endmembers and there exists one confusing background endmember in the first 5 positive bags which is Verde Antique. It is expected that the discriminative MIL algorithms, MI-HE, MI-ACE, and MI-SMF, should be able to perform well in this experiment configuration.

The aforementioned MI-HE [44, 45], *e*FUMI [37, 38], MI-SMF and MI-ACE [41], DMIL [10, 11] and mi-SVM [9] are multiple instance target concept learning methods. The mi-SVM algorithm performs a comparison of MIL approach that does not rely on estimating a target signature. Figure 6.6a shows the estimated target signature from data with 0.3 mean target proportion value. It clearly shows that *e*FUMI is always confused with another nontarget endmember, Verde Antique, that exists in some positive bags but is excluded from the background bags. It also shows the other comparison algorithms can estimate a target concept close to the ground truth Red Slate spectrum. One thing need to be explained here is since MI-ACE and MI-SMF are discriminative concept learning methods that try to minimize the detection response of negative bags, they are not expected to recover the true target signature.

For simulated detection analysis, estimated target concepts from the training data were then applied to the test data generated separately following the same generating procedure. The detection was performed using the HSD [46] or ACE [57] detection statistic. For MI-HE and *e*FUMI, both methods were applied since those two algorithms can come out as a set of background concept from training simultaneously; for MI-SMF, both SMF and ACE were applied since MI-SMF's objective is maximizing the multiple instance spectral matched filter; for the rest multiple instance target concept learning algorithms, MI-ACE, DMIL, only ACE was applied. For the testing procedure of mi-SVM, a regular SVM testing process was performed using LIBSVM [58], and the decision values (signed distances to hyperplane) of test data determined from trained SVM model were taken as the confidence values. For the signature-based detectors, the background data mean and covariance were estimated from the negative instances of the training data.

For quantitative evaluation, Fig. 6.6b shows the receiver operating characteristic (ROC) curves using estimated target signature, where it can be seen that the *e*FUMI is confused with the testing Verde Antique data at very low PFA (probability of false alarms) rate. Table 6.2 shows the area under the curve (AUC) of proposed MI-HE and comparison algorithms. The results reported are the median results over five runs of the algorithm on the same data. From Table 6.2, it can be seen that for MI-HE and MI-ACE, the best performance on detection was achieved using ACE detector, which is quite close to the performance of using the ground truth target signature (denoted as values with stars). The reason that MI-HE's detection using HSD detector is a little worse is that HSD relies on knowing the complete background concept to properly represent each nontarget testing data, the missing nontarget concept (Verde Antique) makes the nontarget testing data containing Verde Antique maintain a relatively large reconstruction error, and thus large detection statistic.

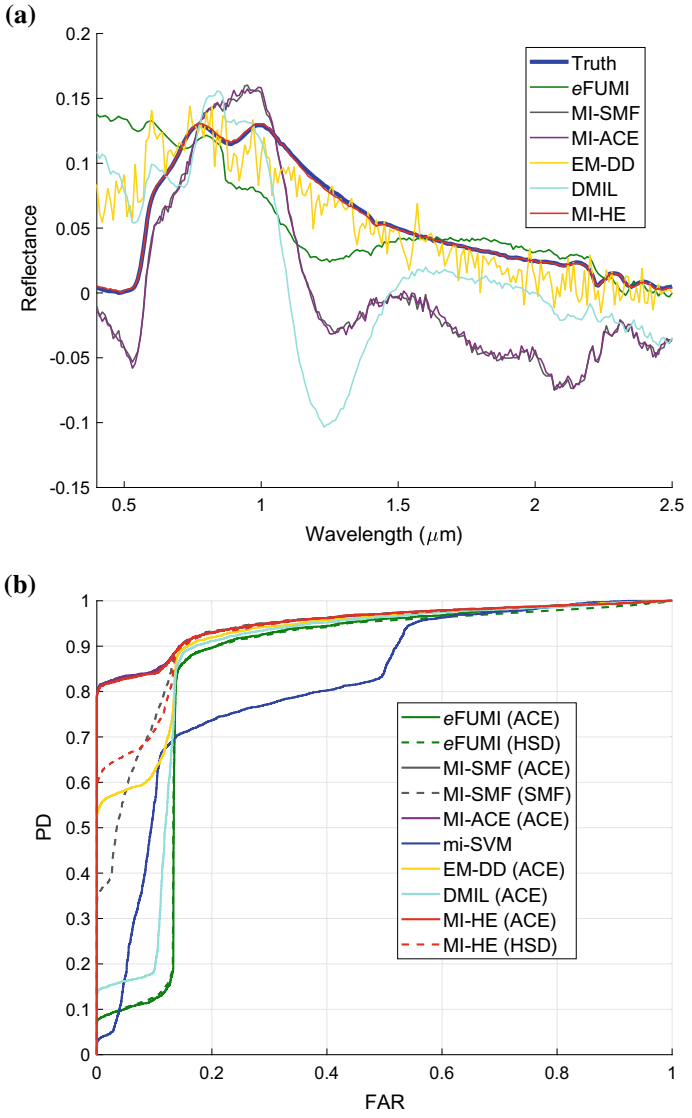

Fig. 6.6 MI-HE and comparisons on synthetic data with incomplete background knowledge, $\boldsymbol{\alpha}_{t\_mean} = 0.3$. MI-SMF and MI-ACE are not expected to recover the true signature. **a** Estimated target signatures for Red Slate and comparison with ground. **b** ROC curves cross validated on test data

Table 6.2 Area under the ROC curves for MI-HE and comparison algorithms on simulated hyperspectral data with incomplete background knowledge. Best results shown in bold, second best results underlined, and ground truth shown with an asterisk

Algorithm	α_{t_mean}			
	0.1	0.3	0.5	0.7
MI-HE (HSD)	0.743	<u>0.931</u>	0.975	0.995
MI-HE (ACE)	<u>0.763</u>	**0.952**	**0.992**	**0.999**
*e*FUMI [37] (ACE)	0.675	0.845	0.978	<u>0.998</u>
*e*FUMI [37] (HSD)	0.671	0.564	0.978	<u>0.998</u>
MI-SMF [41] (SMF)	0.719	0.923	0.972	0.993
MI-SMF [41] (ACE)	0.735	**0.952**	**0.992**	**0.999**
MI-ACE [41] (ACE)	**0.764**	**0.952**	**0.992**	**0.999**
mi-SVM [9]	0.715	0.815	0.866	0.900
DMIL [10, 11] (ACE)	0.687	0.865	0.971	0.996
Ground Truth (ACE)	0.765*	0.953*	0.992*	0.999*

6.3.5.2 MUUFL Gulfport Hyperspectral Data

The MUUFL Gulfport hyperspectral data set collected over the University of Southern Mississippi-Gulfpark Campus was used to evaluate the target detection performance across various MIL classification methods. This data set contains 325 × 337 pixels with 72 spectral bands corresponding to wavelengths from 367.7 to 1043.4 nm at a 9.5−9.6 nm spectral sampling interval. The ground sample distance of this hyperspectral data set is 1 m [1]. The first four and last four bands were removed due to sensor noise. Two sets of this data (Gulfport Campus Flight 1 and Gulfport Campus Flight 3) were selected as cross-validated training and testing data for these two data sets have the same altitude and spatial resolution. Throughout the scene, there are 64 man-made targets in which 57 were considered in this experiment which are cloth panels of four different colors: Brown (15 examples), Dark Green (15 examples), Faux Vineyard Green (FVGr) (12 examples), and Pea Green (15 examples). The spatial location of the targets are shown as scattered points over an RGB image of the scene in Fig. 6.7. Some of the targets are in the open ground and some are occluded by the live oak trees. Moreover, the targets also vary in size, for each target type, there are targets that are 0.25 m^2, 1 m^2, and 9 m^2 in area, respectively, resulting a very challenging, highly mixed sub-pixel target detection problem.

MUUFL Gulfport Hyperspectral Data, Individual Target Type Detection
For this part of the experiments, each individual target type was treated as a target class, respectively. For example, when "Brown" is selected as target class, a 5 × 5 rectangular region corresponding to each of the 15 ground truth locations denoted by GPS was grouped into a positive bag to account for the drift coming from GPS. This size was chosen based on the accuracy of the GPS device used to record the ground truth locations. The remaining area that does not contain a brown target was

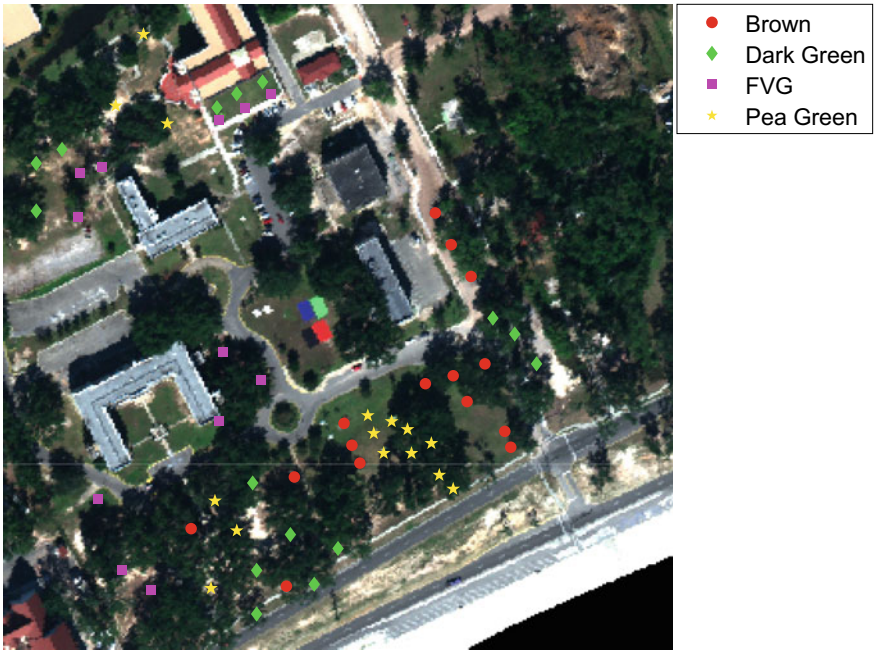

Fig. 6.7 MUUFL Gulfport data set RGB image and the 57 target locations

grouped into a big negative bag. This constructs the detection problem for "Brown" target. Similarly, there are 15, 12, and15 positive labeled bags for Dark Green, Faux Vineyard Green, and Pea Green, respectively.

The comparison algorithms were evaluated on this data using the Normalized Area Under the receiver operating characteristic curve (NAUC) in which the area was normalized out to a false alarm rate (FAR) of 1×10^{-3} false alarms/m^2 [59]. During detection on the test data, the background mean and covariance were estimated from the negative instances of the training data. The results reported are the median results over five runs of the algorithm on the same data.

Figure 6.8a shows the estimated target concept by all comparisons for Dark Green target type training on flight 3. We can see that the *e*FUMI and MI-HE are able to recover the target concept quite close to ground truth spectra manually selected from the scene. Figure 6.8b shows the detection ROCs given target spectra estimated on flight 3 and cross validated on flight 1. Table 6.3 shows the NAUCs for all comparison algorithms cross validated on all four types of target, where it can be seen that MI-HE generally outperforms the comparisons for most of the target types and achieves close to the performance of using ground truth target signatures. Since MI-HE is a discriminative target concept learning framework that aims to distinguish one target instance from each positively labeled bag, MI-HE had a lower performance for the pea green target because of the relatively large occlusion of those targets causing difficulty in distinguishing pea green signature from each of the positive bag.

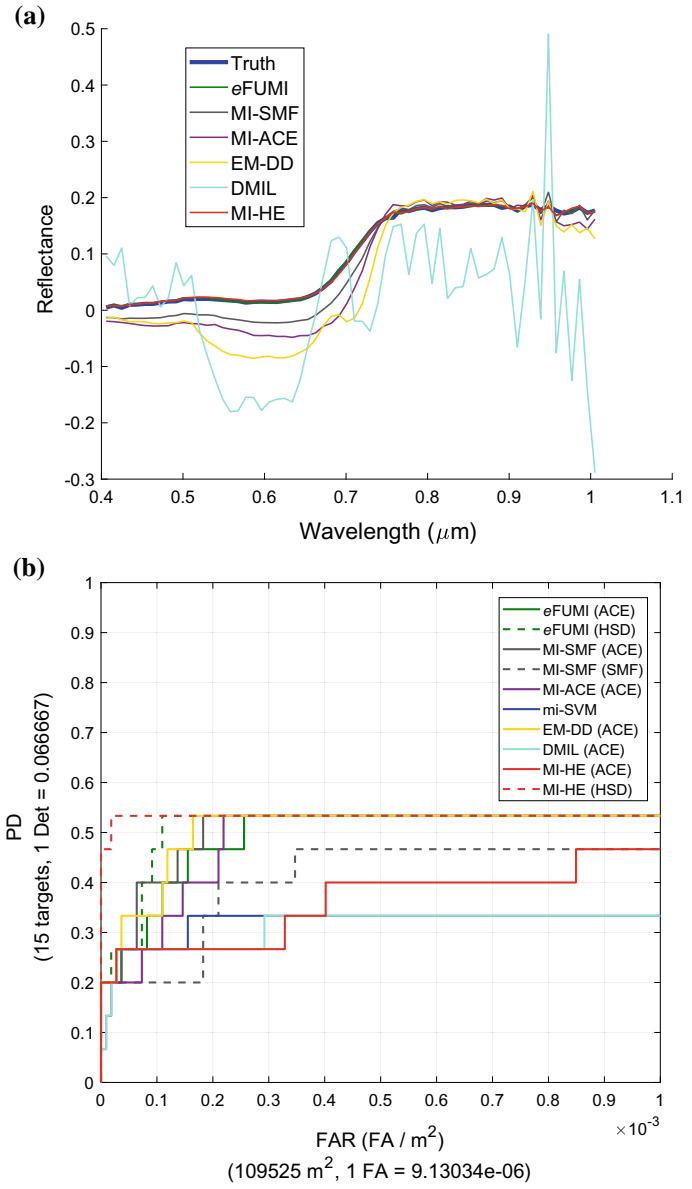

Fig. 6.8 MI-HE and comparisons on Gulfport Data Dark Green, training flight 3 testing flight 1. **a** Estimated target signatures from flight 3 for Brown and comparison with ground truth. **b** ROC curves cross validated on flight 1

Table 6.3 Area under the ROC curves for MI-HE and comparison algorithms on Gulfport data with individual target type. Best results shown in bold, second best results underlined, and ground truth shown with an asterisk

Alg.	Train on Flight 1; Test on Flight 3				Train on Flight 3; Test on Flight 1			
	Brown	Dark Gr.	Faux Vine Gr.	Pea Gr.	Brown	Dark Gr.	Faux Vine Gr.	Pea Gr.
MI-HE (HSD)	**0.499**	**0.453**	**0.655**	0.267	**0.781**	**0.532**	**0.655**	0.350
MI-HE (ACE)	0.433	0.379	0.104	0.267	0.710	0.360	0.111	0.266
*e*FUMI [37] (ACE)	0.423	0.377	<u>0.654</u>	0.267	0.754	0.491	0.605	<u>0.393</u>
*e*FUMI [37] (HSD)	0.444	<u>0.436</u>	0.653	0.267	0.727	<u>0.509</u>	0.500	0.333
MI-SMF [41] (SMF)	0.419	0.354	0.533	0.266	0.657	0.405	<u>0.650</u>	0.384
MI-SMF [41] (ACE)	0.448	0.382	0.579	<u>0.316</u>	<u>0.760</u>	0.501	0.613	0.388
MI-ACE [41] (ACE)	<u>0.474</u>	0.390	0.485	**0.333**	<u>0.760</u>	0.483	0.593	0.380
mi-svm [9]	0.206	0.195	0.412	0.265	0.333	0.319	0.245	0.274
EM-DD [7] (ACE)	0.411	0.381	0.486	0.279	<u>0.760</u>	0.503	0.541	**0.416**
DMIL [10, 11] (ACE)	0.419	0.383	0.191	0.009	0.743	0.310	0.081	0.083
Ground Truth (ACE)	0.528*	0.429*	0.656*	0.267*	0.778*	0.521*	0.663*	0.399*

MUUFL Gulfport Hyperspectral Data, All Four Target Types Detection

For training and detection for the four target types together, the positive bags were generated by grouping each of the 5×5 regions denoted by the ground truth that it contains any of the four types of target. Thus, for each flight there are 57 target points and 57 positive bags were generated. The remaining area that does not contain any target was grouped into a big negative bag. Table 6.4 summarizes the NAUCs as a quantitative comparison, which shows that the detection statistic by the proposed MI-HE using HSD is significantly better than the comparison algorithms.

Table 6.4 Area under the ROC curves for MI-HE and comparison algorithms on Gulfport data with all four target types. Best results shown in bold, second best results underlined, and ground truth shown with an asterisk

Alg.	Test Flight 3	Test Flight 1	Alg.	Test Flight 3	Test Flight 1
MI-HE (HSD)	**0.304**	**0.449**	MI-SMF [41] (ACE)	0.219	0.327
MI-HE (ACE)	<u>0.257</u>	0.254	MI-SMF [41] (SMF)	0.198	0.277
*e*FUMI [37] (ACE)	0.214	<u>0.325</u>	mi-SVM [9]	0.235	0.269
*e*FUMI [37] (HSD)	0.256	0.331	EM-DD [7] (ACE)	0.211	0.310
MI-ACE [41] (ACE)	0.226	0.340	DMIL [10, 11] (ACE)	0.198	0.225
Ground Truth (ACE)	0.330*	0.490*			

6.4 Multiple Instance Learning Approaches for Classifier Fusion and Regression

Although more extensively studied for the case of sub-pixel hyperspectral target detection, the Multiple Instance Learning approach can be used in other hyperspectral applications including fusion with other sensors and regression, in addition to two-class classification and detection problems discussed in previous sections. In this section, algorithms for multiple instance classifier fusion and regression are presented and their applications to hyperspectral and remote sensing data analysis are discussed.

6.4.1 Multiple Instance Choquet Integral Classifier Fusion

The multiple instance Choquet integral (MICI) algorithm[4] [61, 62] is a multiple instance classifier fusion method to integrate different classifier outputs with imprecise labels under the MIL framework. In MICI, the Choquet integral [63, 64] was used under the MIL framework to fuse outputs from multiple classifiers or sensors for improving the accuracy and accounting for imprecise labels for hyperspectral classification and target detection.

The Choquet integral (CI) is an effective nonlinear information aggregation method based on the fuzzy measure. Assume there exists m sources, $C = \{c_1, c_2, \ldots, c_m\}$, for fusion. These "sources" can be the decision outputs by different classifiers or data collected by different sensors. The power set of C is denoted as 2^C, which

[4]The MICI implementation is available at: https://github.com/GatorSense/MICI [60].

contains all possible (crisp) subsets of C. A monotonic and normalized fuzzy measure, $\mathbf{g}$, is a real valued function that maps $2^C \rightarrow [0, 1]$. It satisfies the following properties:

1. $g(\emptyset) = 0$; *empty set*
2. $g(C) = 1$; *normalization property*
3. $g(A) \leq g(B)$ if $A \subseteq B$ and $A, B \subseteq C$. *monotonicity property*.

Let $h(c_k; \mathbf{x}_n)$ denote the output of the kth classifier, c_k, on the nth instance, $\mathbf{x}_n$. The discrete Choquet integral of instance $\mathbf{x}_n$ given C (m sources) is computed using

$$C_{\mathbf{g}}(\mathbf{x}_n) = \sum_{k=1}^{m} \left[h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n) \right] g(A_k), \tag{6.28}$$

where the sources are sorted so that $h(c_1; \mathbf{x}_n) \geq h(c_2; \mathbf{x}_n) \geq \cdots \geq h(c_m; \mathbf{x}_n)$ and $h(c_{m+1}; \mathbf{x}_n)$ is defined to be zero. The fuzzy measure element value $g(A_k)$ corresponds to the subset $A_k = \{c_1, c_2, \ldots, c_k\}$.

In a classifier fusion problem, given training data and fusion sources, $h(c_m; \mathbf{x}_n)$ $\forall m, n$ are known. The desired bag-level labels for sets of $C_{\mathbf{g}}(\mathbf{x}_n)$ values are also known (positive label "+1", negative label "0"). Then, the goal of the MICI algorithm is to learn all the element values of the unknown fuzzy measure $\mathbf{g}$ from the training data and bag-level (imprecise) labels. The MICI method includes three variations to formulate the fusion problem under the MIL framework to address label imprecision. The variations include the noisy-or model, the min-max model, and the generalized-mean model.

The MICI noisy-or model follows the Diverse Density formulation (see Sect. 6.2.2) and uses a noisy-or objective function

$$\begin{aligned} J_N = & \sum_{a=1}^{K^-} \sum_{i=1}^{N_b^-} \ln \left(1 - \mathcal{N}\left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-) | \mu, \sigma^2\right)\right) \\ & + \sum_{b=1}^{K^+} \ln \left(1 - \prod_{j=1}^{N_b^+} 1 - \mathcal{N}\left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+) | \mu, \sigma^2\right) \right), \end{aligned} \tag{6.29}$$

where K^+ denotes the total number of positive bags, K^- denotes the total number of negative bags, N_b^+ is the total number of instances in positive bag b, and N_a^- is the total number of instances in negative bag a. Each data point/instance is either positive or negative, as indicated by the following notation: $\mathbf{x}_{ai}^-$ is the ith instance in the ath negative bag and $\mathbf{x}_{bj}^+$ is the jth instance in the bth positive bag. The $C_{\mathbf{g}}$ is the Choquet integral output given measure $\mathbf{g}$ computed using (6.28). The μ and σ^2 are the mean and variance of the Gaussian function $\mathcal{N}(\cdot)$, respectively. In practice, the parameter μ can be set to 1 or a value close to 1 for two-class classifier fusion problems, in order to encourage the CI values of positive instances to be 1 and the CI values of negative instances to be far from 1. The variance of the Gaussian σ^2 controls how

sharply the CI values are pushed to 0 and 1, and thus controls the weighting of the two terms in the objective function. By maximizing the objective function (6.29), the CI values of all the points in the negative bag are encouraged to be zero (first term) and the CI values of at least one instance in the positive bag are encouraged to be one (second term), which follows the MIL assumption.

The MICI min-max model applies the min and max operators to the negative and positive bags, respectively. The min-max model follows the MIL formulation without the need to manually set parameters such as the Gaussian variance in the noisy-or model. The objective function of the MICI min-max model is

$$J_M = \sum_{a=1}^{K^-} \max_{\forall \mathbf{x}_{ai}^- \in \mathbf{B}_a^-} \left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-) - 0\right)^2 + \sum_{b=1}^{K^+} \min_{\forall \mathbf{x}_{bj}^+ \in \mathbf{B}_b^+} \left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+) - 1\right)^2, \tag{6.30}$$

where $\mathbf{B}_a^-$ denotes the ath negative bag, and $\mathbf{B}_b^+$ denotes the bth positive bag. The remaining terms follow the same notation as in (6.29). The first term of the objective function encourages the CI values of all instances in the negative bag to be zero, and the second term encourages the CI values of at least one instance in the positive bag to be one. By minimizing the objective function in (6.30), the MIL assumption is satisfied.

Instead of selecting only one instance from each bag as a "prime instance" that determines the bag-level label as does the min-max model, the MICI generalized-mean model allows more instances to contribute toward the classification of bags. The MICI generalized-mean objective function is written as

$$J_G = \sum_{a=1}^{K^-} \left[\frac{1}{N_a^-} \sum_{i=1}^{N_a^-} \left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-) - 0\right)^{2p_1} \right]^{\frac{1}{p_1}} + \sum_{b=1}^{K^+} \left[\frac{1}{N_b^+} \sum_{j=1}^{N_b^+} \left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+) - 1\right)^{2p_2} \right]^{\frac{1}{p_2}}, \tag{6.31}$$

where p_1 and p_2 are the exponential factors controlling the generalized-mean operation. When $p_1 \to +\infty$ and $p_2 \to -\infty$, the generalized-mean terms becomes equivalent to the min and max operators, making the generalized-mean model equivalent to the min-max model. By adjusting the p value, the generalized-mean term can act as varying other aggregating operators, such as arithmetic mean ($p = 1$) or quadratic mean ($p = 2$). For another interpretation, when $p \geq 1$, the generalized-mean can be rewritten as the l_p norm [65].

The MICI models can be optimized by sampling-based evolutionary algorithms, where the element values of fuzzy measure $\mathbf{g}$ are sampled and selected through a truncated Gaussian distribution either based on valid interval (how much the element value can change without violating the monotonicity property of the fuzzy measure), or based on the counts of times a measure element is used in all training instances. A more detailed optimization process and psuedocode of the MICI models can be seen in [62, 66]. The MICI models have been used for hyperspectral sub-pixel target detection [61, 62] and were effective in fusing multiple detector inputs (e.g., the ACE detector) and can yield competitive classification results.

6.4.2 Multiple Instance Regression

Multiple instance regression (MIR) handles multiple instance problems where the prediction values are real-valued, instead of binary class labels. The MIR methods have been used in remote sensing literature for applications such as aerosol optical depth retrieval [67, 68] and crop yield prediction [62, 68–70].

Prime-MIR was one of the earliest MIR algorithms, proposed by Ray and Page in 2001 [71]. Prime-MIR is based on the "primary instance" assumption, which assumes there is only one primary instance per bag that contributes to the real-valued bag-level label. Prime-MIR assumes a linear regression hypothesis and the goal is to find a hyperplane $\mathbf{Y} = \mathbf{Xb}$ such that

$$\mathbf{b} = \arg\min_{b} \sum_{i=1}^{n} L\left(y_i, X_{ip}, \mathbf{b}\right), \tag{6.32}$$

where X_{ip} is the primary instance in bag i, and L is some error function, such as the squared error. An expectation–maximization (EM) algorithm was used to iteratively solve for the ideal hyperplane. First, a random hyperplane was initialized. For each instance j in each bag i, the error L of the instance X_{ij} to the hyperplane $\mathbf{Y} = \mathbf{Xb}$ was computed. In the E-step, the instance with the lowest error L was selected as the "primary instance." In the M-step, a new hyperplane was constructed by performing a multiple regression over all the primary instances selected in the E-step. The two steps were repeated until the algorithm converges and the best hyperplane solution was returned. In [71], Prime-MIR showed the benefits of using multiple instance regression over ordinary regression, especially when the non-primary instances in the bag were not correlated with the primary instances.

The MI k-NN approach and its variations [72] extends the Diverse Density, kNN, and Citation-kNN for real-valued multiple instance learning. The minimal Hausdorff distance from [27] was used to measure the distance between two bags. Given two sets of points $A = a_1, \ldots a_m$ and $B = b_1, \ldots, b_n$, the Hausdorff distance is defined as

$$H(A, B) = \max\{h(A, B), h(B, A)\}, \tag{6.33}$$

where $h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$, $\|a - b\|$ is the Euclidean distance between points a and b. In the MI k-NN algorithm, the prediction made for a bag B is the average label of the k closest bags, measured in Hausdorff metric. In the MI citation-kNN algorithm, the prediction made for a bag B is the average label of the R closest bag neighbors of B measured in Hausdorff metric and C-nearest citers, where the "citers" include the bags where B is a one of their C-nearest neighbors. It is generally recommended that $C = R + 2$ [72]. The third variant, a diverse density approach for the real-valued setting, maximizes

$$\prod_{i=1}^{K} Pr(r|B_i) \tag{6.34}$$

where $Pr(t|B_i) = (1 - |l_i - Label(B_i|t)|)/Z$, K is the total number of bags, t is the target point, l_i is the label for the ith bag, and Z is a normalization constant. The results in [72] showed good prediction performance of all three variants on a benchmark Musk Molecules data set [4], but the performance of both the nearest neighbor and diverse density algorithms were sensitive to the number of relevant features, as expected based on the sensitivity of the Hausdorff distance to outliers.

A real-valued multiple instance on-line model proposed by Goldman and Scott [73] uses MIR for learning real-valued geometric patterns, motivated by landmark matching problem in robot navigation and vision applications. This algorithm associates a real-valued label with each point and uses the Hausdorff metric to help classify a bag as positive, if the points in the bag are within some Hausdorff distance from target concept points. This algorithm differs from the supervised MIR in that the standard supervised MIR learns from a given set of training bags and bag-level training labels, while [73] applies an online agnostic model [74–76] where the learners make predictions as the bag $\mathbf{B}_t$ is presented at iteration t. Wang et al. [77] also used the idea of online MIR, i.e., to use the latest arrived bag with its training label to update the current predictive model. This work was also extended in [78].

A regularization framework for MIR proposed by Cheung and Kwok [79] defines a loss function that takes into consideration both training bags and training instances. The first part of the loss function computes the error (loss) between training bags label and its predictions and the second part considers the loss between the bag label prediction and all the instances in the bag. This work still adopted the "primary instance" assumption but simplified to assume the primary instance was the instance with the highest prediction output value. This model provided comparable or better performance on the synthetic Musk Molecules data set [72] as citation-kNN [27] and Multiple Instance kernel-based SVM [79, 80].

Most MIR methods discussed above only provided theoretical discussions or results on synthetic regression data sets. More recently, MIR methods have been applied to real-world hyperspectral and remote sensing data analysis. Wagstaff et al. in [69, 70] investigated using MIR to predict crop yield from remotely sensed data collected over California and Kansas. In [69], a novel method for inferring the "salience" of each instance was proposed with regard to the real-valued bag label. The salience of each instance, i.e., its "relevance" with respect to all other instances in the bag to predict the bag label, is the weight associated with each instance. The salience values were defined to be nonnegative and sum to one for all instances in each bag. Like Ray and Page [71], Wagstaff et al. followed the "primary-instance" assumption but their primary instance, or "exemplar" of a bag, is the weighted average of all the points in the bag instead of one single instance from the bag. Given training bags and instances, a set of salience values are solved based on a fixed linear regression model and given the estimated salience, the regressor is updated and the algorithm reiterates until convergence. This work did not intend to provide predic-

tions over new data, but instead focused on understanding the contents (the salience) of each training instance.

Wagstaff et al. then made use of the salience learned to provide predictions for new, unlabeled bags by proposing an MI-ClusterRegress algorithm (or sometimes referred to as the Cluster-MIR algorithm) [70] that mapped instances onto (hidden) cluster labels. The main assumption of MI-ClusterRegress is that the instances from a bag are drawn (with noise) from a set of underlying clusters and one of the clusters is "relevant" to the bag-level labels. After obtaining k clusters for each bag by EM-based Gaussian mixture models (or any other clustering method), a local regression model is constructed for each cluster. MI-ClusterRegress then selects the best-fit model and use it to predict labels for test bags. A support vector regression learner [81] is used for regression prediction. Results on simulated and predicting crop yield data sets show that modeling the bag structure when the structure (cluster) is present is effective for regression prediction, especially when the cluster number k is equal to or larger than what is actually present in the bags.

In Chap. 2, Moreno-Martínez et al. proposed a kernel distribution regression (KDR) model for MIR by embedding the bag distribution in a high-dimensional Hilbert space and performing standard least squares regression on the mean embedded data. This kernel method exploits the rich structure in bags by considering all higher order moments of the bag distributions and performing regression with the bag distributions directly. This kernel method also allows to combine bags with different number of instances per bag by summarizing the bag feature vectors with a set of mean map embeddings of instances in the bag. The KRD model was shown to outperform standard regression models such as the least squares regularized linear regression model (RLR) and the (nonlinear) kernel ridge regression (KRR) method for crop yield applications.

Wang et al. [67, 68] proposed a probabilistic and generalized mixture model for MIR based on the primary-instance assumption (sometimes referred to as the EM-MIR algorithm). It is assumed that the bag label is a noisy function of the primary instance, and the conditional probability $p(y_i|\mathbf{B}_i)$ for predicting label y_i for the ith bag is dependent entirely on the primary instance. A binary random variable z_{ij} is defined such that $z_{ij} = 1$ if the jth instance in the ith bag is the primary instance and $z_{ij} = 0$ if otherwise. The mixture model for each bag i is written as

$$p(y_i|\mathbf{B}_i) = \sum_{j=1}^{N_i} p(z_{ij} = 1|\mathbf{B}_i)\, p(y_i|\mathbf{x}_{ij}) \tag{6.35}$$

$$= \sum_{j=1}^{N_i} \pi_{ij}\, p(y_i|\mathbf{x}_{ij}), \tag{6.36}$$

where π_{ij} is the (pior) probability that the jth instance in the ith bag is the primary instance, $p(y_i|\mathbf{x}_{ij})$ is the label probability given the primary instance $\mathbf{x}_{ij}$ and N_i is the total number of instances in the ith bag $\mathbf{B}_i$. Therefore, the learning problem is transformed to learning the mixture weights π_{ij} and $p(y_i|\mathbf{x}_{ij})$ from training data and

an EM algorithm is used to optimize the parameters. This work discussed several methods to set the prior π_{ij}, including using deterministic function, or as a Gaussian function of prediction deviation, or as a parametric function (in this case a feed-forward neural network). It was discussed in [68] that several algorithms discussed above, including Prime-MIR [71] and Pruning-MIR [67], are in fact the special case of the mixture model. The mixture model MIR shows better performance on simulated data as well as for predicting aerosol optical depth (AOD) from remote sensing data and predicting crop yield applications, compared with the Cluster-MIR [70] and Prime-MIR [71] algorithms described above.

Two baseline methods for MIR have also been described in [68], Aggregate-MIR, and Instance-MIR. In Aggregate-MIR, a "meta-instance" is obtained for each bag by averaging all the instances in that bag, and a regression model can be trained using the bag-level labels and the meta-instances. In Instance-MIR, all instances in a bag are assumed to have the same label as the bag-level label, and a regression model can be trained by combining all instances from all bags. Then, in testing, the label for a test bag is the average of all the instance-level labels in that test bag. The Aggregate-MIR and Instance-MIR methods belong to the "input summary" and "output expansion" approaches as described in Chap. 2, Sect. 2.3.1. These two methods are straightforward and easy to implement, and have been used as basic comparison methods for a variety of MIR applications.

The robust fuzzy clustering for MIR (RFC-MIR) algorithm was proposed by Trabelsi and Frigui [82] to incorporate data structure in MIR. The RFC-MIR algorithm uses fuzzy clustering methods such as the fuzzy c-means (FCM) and possibilistic c-means (PCM) [83] to cluster the instances and fit multiple local linear regression models to the clusters. Similar to Cluster-MIR, the RFC-MIR method combines all instances from all training bags for clustering. However, Cluster-MIR performs clustering in an unsupervised manner without considering bag-level labels, while RFC-MIR uses instance features as well as labels in clustering. Validation results of RFC-MIR show improved accuracy on crop yield prediction and drug activity prediction applications [84], and the possibilistic memberships obtained from the RFC-MIR algorithm can be used to identify the primary and irrelevant instances in each bag.

In parallel with the multiple instance classifier fusion models described in Sect. 6.4.1, a Multiple Instance Choquet Integral Regression (MICIR) model[5] has been proposed to accommodate real-valued predictions for remote sensing applications [62]. The objective function of the MICIR model is written as

$$\min \sum_{i=1}^{K} \left[\min_{\forall j, x_{ij} \in \mathbf{B}_i} \left(C_g(\mathbf{x}_{ij}) - o_i \right)^2 \right], \tag{6.37}$$

where o_i is the desired training labels for bag $\mathbf{B}_i$. Note that MICIR is able to fuse real-valued outputs from regression models as well as from classifiers. When o_i is

[5]The MICIR implementation is available at: https://github.com/GatorSense/MICI [60].

binary, MICIR reduces to the MICI min-max model for two-class classifier fusion. The MICIR algorithm also follows the primary instance assumption by minimizing the error between the CI value of one primary instance and the given bag-level labels, while allowing imprecision in other instances. Similar to MICI classifier fusion models, an evolutionary algorithm can be used to sample the fuzzy measure **g** from the training data.

Overall, Multiple Instance Regression methods have been studied in the literature for nearly two decades and most studies are based on the primary-instance assumption proposed by Ray and Page in 2001. Linear regression models were used in most MIR methods if a regressor was used and experiments have shown effective results of using MIR on crop yield prediction and aerosol optical depth retrieval applications given remote sensing data.

6.4.3 *Multiple Instance Multi-resolution and Multi-modal Fusion*

Previous MIL classifier fusion and regression methods, such as the MICI and the MICIR models, can only be applied if the fusion sources have the same number of data points and the same resolution across multiple sensors. As motivated in Sect. 6.1, in remote sensing applications, sensor outputs often have different resolutions and modalities, such as rasterized hyperspectral imagery versus LiDAR point cloud data. To address multi-resolution and multi-modal fusion under imprecision, the multiple instance multi-resolution fusion (MIMRF) algorithm[6] was developed to fuse multi-resolution and multi-modal sensor outputs while learning from automatically generated, imprecisely labeled data [66, 86].

In multi-resolution and multi-modal fusion, there can be a set of *candidate* points from a local region from one sensor that corresponds to one point from another sensor, due to sensor measurement inaccuracy and different data resolutions and modalities. Take hyperspectral imagery and LIDAR point cloud data fusion, for example, for each pixel H_i in the HSI imagery, there may exist a set of $\{L_{i1}, L_{i2}, \ldots, L_{il}\}$ points from the LiDAR point cloud that corresponds to the area covered by the pixel H_i. The MIMRF algorithm first constructs such correspondences by writing the *collection* of the sensor outputs for pixel i as

$$\mathbf{S}_i = \begin{bmatrix} H_i & L_{i1} \\ H_i & L_{i2} \\ \vdots & \vdots \\ H_i & L_{il} \end{bmatrix}. \tag{6.38}$$

[6]The MIMRF implementation is available at: https://github.com/GatorSense/MIMRF [85].

This notation can extend to any number of correspondences l by row, and multiple sensors by column. The MIMRF assumes that, at least one point in all candidate LiDAR points is accurate, but it is unknown which one. One of the goals of the MIMRF algorithm is to automatically select the correct points with accurate measurement and correspondence information. To achieve this goal, the CI fusion for the collection of the sensor outputs of the ith negative data point is written as

$$C_{\mathbf{g}}(\mathbf{S}_i^-) = \min_{\forall \mathbf{x}_k^- \in \mathbf{S}_i^-} C_{\mathbf{g}}(\mathbf{x}_k^-), \tag{6.39}$$

and the CI fusion for the collection of the sensor outputs values of the jth positive data point is written as

$$C_{\mathbf{g}}(\mathbf{S}_j^+) = \max_{\forall \mathbf{x}_l^+ \in \mathbf{S}_j^+} C_{\mathbf{g}}(\mathbf{x}_l^+), \tag{6.40}$$

where $\mathbf{S}_i^-$ is the collection of sensor outputs for the ith negative data point and $\mathbf{S}_j^+$ is the collection of sensor outputs for the jth positive data point; $C_{\mathbf{g}}(\mathbf{S}_i^-)$ is the Choquet integral output for $\mathbf{S}_i^-$ and $C_{\mathbf{g}}(\mathbf{S}_j^+)$ is the Choquet integral output for $\mathbf{S}_j^+$. In this way, the min and max operators automatically select one data point (which is assumed to be the data point with correct information) from each negative and positive bag to be used for fusion, respectively.

Moreover, the MIMRF is designed to handle bag-level imprecise labels. Recall that the MIL framework assumes a bag is labeled positive if at least one instance in the bag is positive and a bag is labeled negative if all the instances in the bag are negative. Thus, the objective function for MIMRF algorithm is proposed as

$$\begin{aligned} J &= \sum_{a=1}^{K^-} \max_{\forall \mathbf{S}_{ai}^- \in \mathbf{B}_a^-} \left(C_{\mathbf{g}}(\mathbf{S}_{ai}^-) - 0\right)^2 + \sum_{b=1}^{K^+} \min_{\forall \mathbf{S}_{bj}^+ \in \mathbf{B}_b^+} \left(C_{\mathbf{g}}(\mathbf{S}_{bj}^+) - 1\right)^2 \\ &= \sum_{a=1}^{K^-} \boxed{\max_{\forall \mathbf{S}_{ai}^- \in \mathbf{B}_a^-}} \left(\min_{\forall \mathbf{x}_k^- \in \mathbf{S}_{ai}^-} C_{\mathbf{g}}(\mathbf{x}_k^-) - 0\right)^2 + \sum_{b=1}^{K^+} \boxed{\min_{\forall \mathbf{S}_{bj}^+ \in \mathbf{B}_b^+}} \left(\max_{\forall \mathbf{x}_l^+ \in \mathbf{S}_{bj}^+} C_{\mathbf{g}}(\mathbf{x}_l^+) - 1\right)^2, \end{aligned} \tag{6.41}$$

where K^+ is the total number of positive bags, K^- is the total number of negative bags, $\mathbf{S}_{ai}^-$ is the collection of ith instance set in the ath negative bag and similar for $\mathbf{S}_{bj}^+$. $C_{\mathbf{g}}$ is the Choquet integral given fuzzy measure $\mathbf{g}$, $\mathbf{B}_a^-$ is the ath negative bag, and $\mathbf{B}_b^+$ is the bth positive bag. The term $\mathbf{S}_{ai}^-$ is the collection of input sources for the ith pixel in the ath negative bag and $\mathbf{S}_{bj}^+$ is the collection of input sources for the jth pixel in the bth positive bag.

In (6.41), the min and max operators outside the squared errors (the boxed terms) are comparable to the MICI min-max model. The max operator encourages the Choquet integral of all the points in the negative bag to be 0 and the min operator encourages the Choquet integral of at least one point in the positive bag to be 1 (second term), which satisfies the MIL assumption. The min and max operators inside the squared error terms come from (6.39) and (6.40), which selects one correspondence

for each collection of candidates. By minimizing the objective function in (6.41), the first term encourages the fusion output of all the points in the negative bag to the desired negative label 0, and the second term encourages the fusion output of at least one of the points in the positive bag to the desired positive label +1. This satisfies the MIL assumption while addressing label imprecision for multi-resolution and multi-modal data. The MIMRF algorithm has been used to fuse rasterized hyperspectral imagery and un-rasterized LiDAR point cloud data over urban scenes and have shown effective fusion results for land cover classification [66, 86].

Here is a small example to illustrate the performance of the MIMRF algorithm using the MUUFL Gulfport hyperspectral and LiDAR data set collected over the University of Southern Mississippi-Gulfpark Campus [1]. An illustration of the rasterized hyperspectral imagery and the LiDAR data over the complete scene can be seen in Figs. 6.2 and 6.3 in Sect. 6.1. The task here is to fuse hyperspectral and LiDAR data to perform building detection and classification. The simple linear iterative clustering (SLIC) algorithm [87, 88] was used to segment the hyperspectral imagery. The SLIC algorithm is a widely used, unsupervised superpixel segmentation algorithm that can produce spatially coherent regions. Each superpixel from the segmentation is treated as a "bag" in our learning process and all pixels in each superpixel are all the instances in the bag. The bag-level labels in this data set were generated from OpenStreetMap (OSM), a third-party, crowd-sourced online map [89]. OSM provides map information for urban regions around the world. Figure 6.9c shows the map extracted from Open Street Map (OSM) over the study area based on the ground cover tags available, such as "highway", "footway", "building", etc. Information from Google Earth [90], Google Maps [91], and geo-tagged photographs from a digital camera taken at the scene were also be used as auxiliary data to assist the labeling process. This way, reliable bag-level labels can be automatically generated with minimal human intervention. These bag-level labels will then be used in the MIMRF objective function (6.41) to learn the unknown fuzzy measure $\mathbf{g}$ for HSI-LiDAR fusion. Figure 6.9 shows the RGB imagery, the SLIC segmentation, and the OSM map labels for the MUUFL Gulfport hyperspectral imagery.

Three multi-resolution and multi-modal sensor outputs were used as fusion sources, one generated from HSI imagery and two from raw LiDAR point cloud data. The first fusion source is the ACE detection map on buildings based on the mean spectral signature of randomly sampled building points from the scene. The ACE detection map for buildings is shown in Fig. 6.10a. As shown, the ACE confidence map highlights most buildings, but also highlights some roads which have similar spectral signature (similar construction material, such as asphalt). The ACE detector also failed to detect the top right building due to the darkness of the roof. Two other fusion sources were generated from LiDAR point cloud data according to the building height profile, with the rasterized confidence maps shown in Fig. 6.10b and Fig. 6.10c. Note that in MIMRF fusion, the LiDAR sources will be point clouds and Figs. 6.10b and c are provided for visualization and comparison purposes only.

As shown in Fig. 6.10, each HSI and LiDAR sensor output contains certain building information. The goal is to use MIMRF to fuse all three sensor outputs and perform accurate building classification. We randomly sampled 50% the bags (the

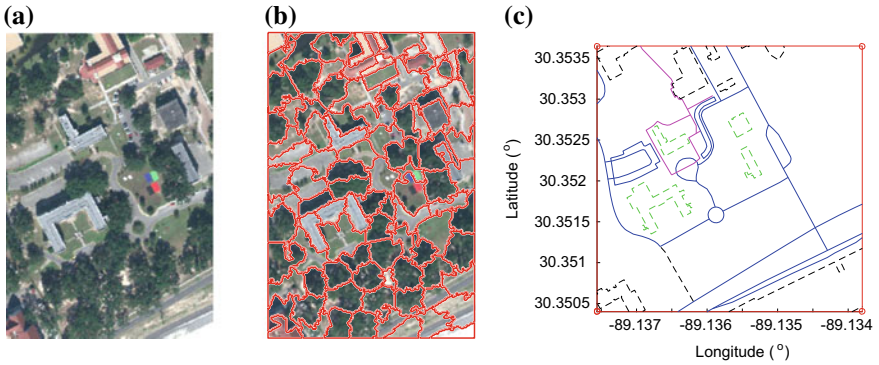

Fig. 6.9 The RGB image (**a**), SLIC segmentation (**b**), and the OSM map for the MUUFL Gulfport hyperspectral imagery (**c**). In the OSM map, the blue lines correspond to road and highway. The magenta lines correspond to sidewalk/footway. The green lines marks buildings. Here, the "building" tag is specific to the buildings with a grey (asphalt) roof. The black lines correspond to "other" tags. Source: © [2020] IEEE. Reprinted, with permission, from [86]

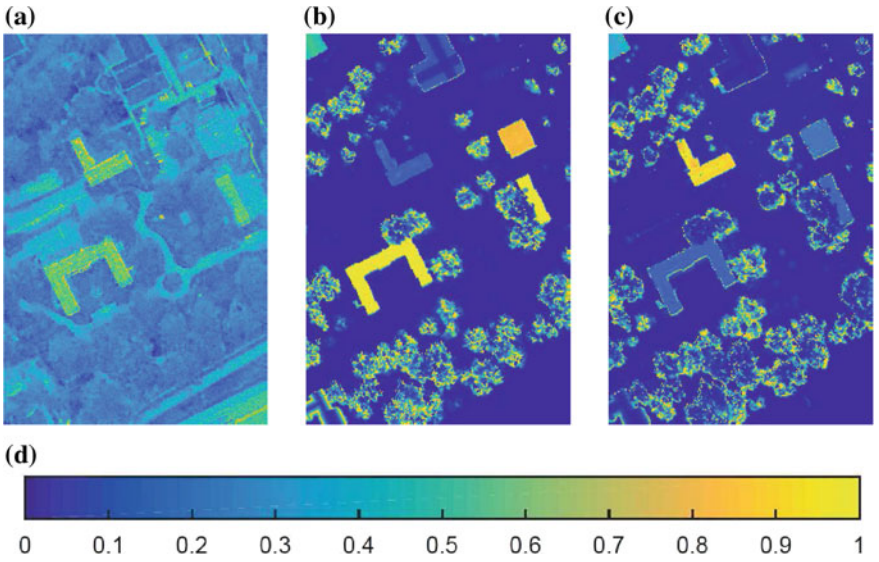

Fig. 6.10 The fusion sources generated from HSI and LiDAR data for building detection. **a** ACE detection map from HSI data. **b**, **c** LiDAR building detection map from two LiDAR flights. The colorbar can be seen in **d**. Source: © [2020] IEEE. Reprinted, with permission, from [86]

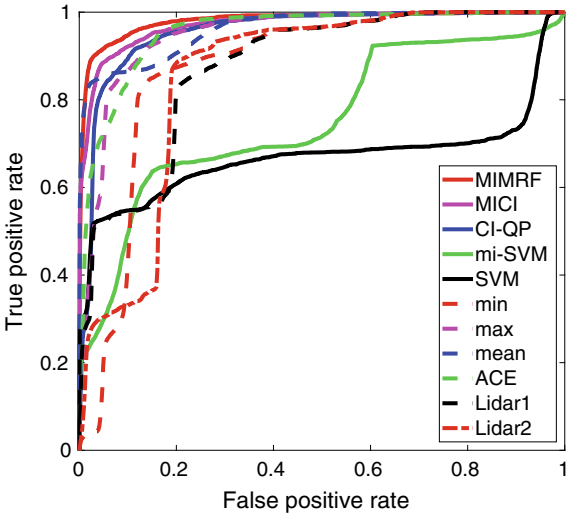

Fig. 6.11 An example of ROC curve results for building detection across all methods

superpixels) and use these to learn a set of fuzzy measures for the MIMRF algorithm. We conducted such random sampling three times by using the MATLAB *randperm()* function and call these the three random runs. The sampled bags are different at each random run. In each random run, the MIMRF algorithm is applied to learn a fuzzy measure from the randomly sampled 50% bags, and fusion results are evaluated on the remaining 50% data on a pixel level. Note that there will be two sets of results in each run—learn from the first sampled 50% bags (denoted "Half1") and perform fusion on the second half of data (denoted "Half2"), and vice versa. The fusion results of MIMRF were compared with previously discussed MIL algorithms such as MICI and mi-SVM and the CI-QP approach. The CI-QP (Choquet integral-quadratic programming) approach [64] is a CI fusion method that learns a fuzzy measure for the Choquet integral by optimizing a least squares error objective using quadratic programming. Note that these comparison methods only work with rasterized LiDAR imagery, while the MIMRF algorithm can directly handle raw LiDAR point cloud data. The fusion results of MIMRF were also compared with commonly used fusion methods, such as min, max, and mean operators and a support vector machine, as well as the ACE and LiDAR sensor sources before fusion.

Figure 6.11 shows an example of the receiver operating characteristic (ROC) curve results for building detection across all comparison methods. Table 6.5 shows the area under curve (AUC) results across all methods in all random runs. Table 6.6 shows the root mean square error (RMSE) results across all methods in all random runs. The AUC evaluates how well the method detects the buildings (the higher AUC the better) and the RMSE shows how the detection results on both the building and nonbuilding points differ from the ground truth (the lower the RMSE the better). We observed from the tables that the MIMRF method was able to achieve high AUC detection results and low RMSE compared to other methods, and the MIMRF is

stable across different randomizations. The MICI classifier fusion method also did well in detection (high AUC), but has higher RMSE compared to MIMRF, possibly due to MICI's inability to handle multi-resolution data. The min operator did well in RMSE due to the fact that it places low confidence everywhere, but was unable to have high detection results. The ACE detector did well in detection, which shows that the hyperspectral signature is effective at distinguishing building roof materials. However, it also places high confidence on other asphalt materials such as road, and thus yields a high RMSE value.

Figures 6.12 and 6.13 shows a qualitative comparison of our fusion performance. Figure 6.12 shows an example of our randomly sampled bags. All the semi-transparent bags marked by the red lines in Fig. 6.12a were used to learn a fuzzy measure in our method, and we evaluate pixel-level fusion results against the "test" ground truth shown in Fig. 6.12b. Note that the MIMRF is a self-supervised method that learns a fuzzy measure from bag-level labels and produces pixel-level fusion results. Although standard training and testing scheme does not apply here, this experiment is set up using cross validation to show that the MIMRF algorithm is able to utilize the fuzzy measure learned from one part of the data and apply that fuzzy measure to perform fusion on new test data, even when the learned bags were excluded from testing.

Table 6.5 The AUC results of building detection using MUUFL Gulfport HSI and LiDAR data across three random runs. (The higher the AUC the better.) The best two results with the highest AUC were **bolded** and underlined, respectively. "Half1" refers to the results of learning a fuzzy measure from the first 50% of the bag-level labels from campus 1 data and perform pixel-level fusion on the second half. "Half2" refers to the results of learning a fuzzy measure from the second 50% of the bag-level labels from campus 1 data and perform pixel-level fusion on the first half. The ACE, Lidar1, and Lidar2 rows show results from the individual HSI and LiDAR sources before fusion; the methods below the dotted line show fusion results for all comparison methods. The standard deviations of MICI and MIMRF methods are computed across three runs (three random fuzzy measure initializations) and are shown in parentheses. Same notation is applied for the RMSE table below as well

	First Random run		Second Random run		Third Random run	
	Half1	Half2	Half1	Half2	Half1	Half2
ACE	0.954	0.961	0.938	0.967	0.963	0.947
Lidar1	0.874	0.914	0.879	0.904	0.920	0.874
Lidar2	0.855	0.813	0.879	0.796	0.830	0.848
SVM	0.670	0.854	0.791	0.918	0.928	0.823
min	0.872	0.863	0.890	0.849	0.870	0.872
max	0.946	0.945	0.953	0.939	0.948	0.945
mean	0.963	0.952	0.969	0.947	0.959	0.960
mi-SVM	0.752	0.886	0.795	0.942	0.930	0.923
CI-QP	0.955	0.959	0.959	0.939	0.962	0.964
MICI	0.972(0.001)	**0.963(0.000)**	**0.976(0.000)**	0.960(0.000)	0.968(0.000)	**0.971(0.000)**
MIMRF	**0.978(0.003)**	**0.963(0.002)**	0.972(0.000)	**0.971(0.001)**	**0.973(0.000)**	**0.971(0.002)**

Table 6.6 The RMSE results of building detection using MUUFL Gulfport HSI and LiDAR data across three random runs. (The lower the RMSE the better.) The best two results with the highest AUC were **bolded** and underlined, respectively. "Half1" refers to the results of learning a fuzzy measure from the first 50% of the bag-level labels from campus 1 data and perform pixel-level fusion on the second half. "Half2" refers to the results of learning a fuzzy measure from the second 50% of the bag-level labels from campus 1 data and perform pixel-level fusion on the first half. The ACE, Lidar1, and Lidar2 rows show results from the individual HSI and LiDAR sources before fusion; the methods below the dotted line show fusion results for all comparison methods. The standard deviations of MICI and MIMRF methods are computed across three runs (three random fuzzy measure initializations) and are shown in parentheses

	First Random run		Second Random run		Third Random run	
	Half1	Half2	Half1	Half2	Half1	Half2
ACE	0.345	0.339	0.348	0.307	0.334	0.350
Lidar1	0.291	0.255	0.278	0.268	0.266	0.280
Lidar2	0.294	0.270	0.267	0.297	0.269	0.295
SVM	0.348	0.332	0.437	0.250	0.409	0.284
min	0.265	0.235	0.248	0.255	**0.240**	0.263
max	0.417	0.417	0.419	0.413	0.423	0.420
mean	0.307	0.291	0.296	0.298	0.298	0.302
mi-SVM	0.425	0.459	0.432	0.253	0.406	0.232
CI-QP	0.403	0.377	0.405	0.413	0.388	0.397
MICI	0.356(0.002)	0.348(0.002)	0.374(0.001)	0.336(0.001)	0.356(0.000)	0.350(0.000)
MIMRF	**0.238(0.024)**	**0.192(0.025)**	**0.244(0.002)**	**0.208(0.011)**	0.255(0.002)	**0.177(0.001)**

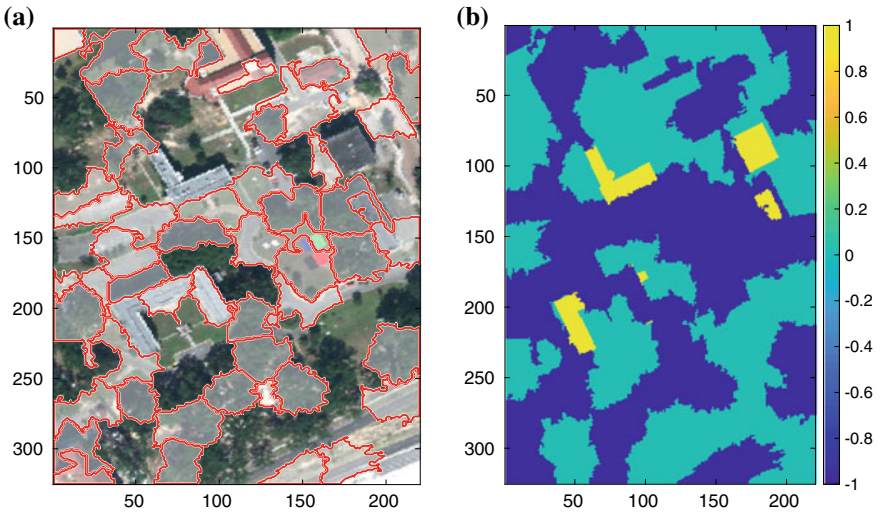

Fig. 6.12 **a** An illustration for the 50% randomly sampled bags from one of our random runs. The MIMRF algorithm learns a fuzzy measure from the red-labeled, transparent bags. **b** The ground truth for the the other 50% data [92]. The yellow and green regions are building and nonbuilding ground truth locations in the "test" data. The dark blue (labeled "–1") regions denote the 50% of the bags that were used in MIMRF learning and therefore not included in the testing process

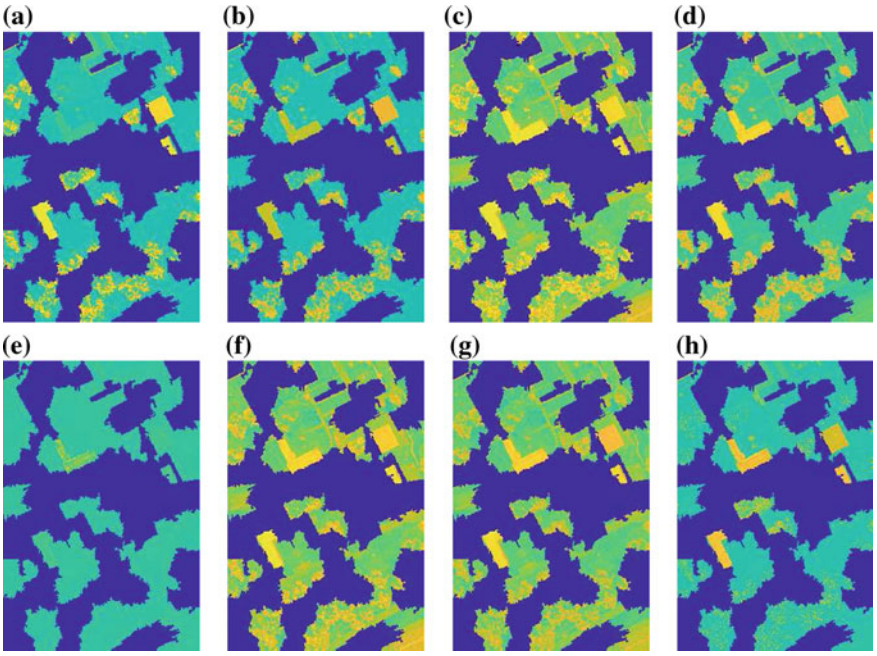

Fig. 6.13 The fusion results for building detection in the MUUFL Gulfport data set, learned from the randomly drawn bags shown in Fig. 6.12a and evaluated on the remaining regions against the ground truth shown in Fig. 6.12b. Note that the MIMRF method learns a set of fuzzy measures from bag-level data and produced per-pixel fusion results on the fusion regions. The subplots show fusion results by **a** SVM; **b** min operator; **c** max operator; **d** mean operator; **e** mi-SVM; **f** CI-QP; **g** MICI; **h** MIMRF. The yellow highlights where the fusion algorithm places high detection confidence and green indicates low confidence, and the dark blue indicates the regions not used in the evaluation. This plot uses the same color bar as in Fig. 6.10d. It is desirable that high confidence (yellow color) was placed on buildings for building detection. As shown, the MIMRF algorithm in **h** was able to detect all buildings (yellow color) in the regions that were evaluated and have low confidence (green color) on nonbuilding areas. The other comparison methods either missed some buildings, or have many more false positives in non-building regions, such as tree canopy

Figure 6.13 shows all fusion results on the test regions across all methods. As shown, the MIMRF algorithm in Fig. 6.13h was able to detect all buildings (yellow) in the evaluation regions well while having low confidence (green) on nonbuilding areas. The other comparison methods either missed some buildings, or have many more false positives in non-building regions. Other randomizations yielded similar effects.

To summarize, the above experimental results show that the MIMRF method was able to successfully perform detection and fusion with high detection accuracy and low root mean square error for multi-resolution and multi-modal data sets. This experiment further demonstrated the effectiveness of the self-supervised learning approach used by the MIMRF method at learning a fuzzy measure from one part of the data

(using only bag-level labels) and perform pixel-level fusion on other regions. Guided by publicly available crowd-sourced data such as the OpenStreetMap, the MIMRF algorithm is able to automatically generate imprecise bag-level labels instead of the traditional manual labeling process. Moreover, [86] has shown effective results of MIMRF fusion on agricultural applications as well, in addition to hyperspectral and LiDAR analysis. We envision the MIMRF as an effective fusion method to perform pixel-level classification and produce fusion maps with minimal human intervention for a variety of multi-resolution and multi-modal fusion applications.

6.5 Summary

This chapter introduced the Multiple Instance Learning framework and reviewed MIL methods for hyperspectral classification, sub-pixel target detection, classifier fusion, regression, and multi-resolution multi-modal fusion. Given imprecise (bag-level) ground truth information in the training data, the MIL methods are effective in addressing the inevitable imprecision observed in remote-sensing data and applications.

- Imprecise training labels are omnipresent in hyperspectral image analysis, due to unreliable ground truth information, sub-pixel targets, and occlusion, and heterogeneous sensor outputs. MIL methods can handle bag-level labels instead of requiring pixel-perfect labels in training, which enables easier annotation and more accurate data analysis.
- Multiple instance target characterization algorithms were presented, including *e*FUMI, MI-ACE/MI-SMF, and MI-HE algorithms. These algorithms can estimate target concepts from the data given imprecise labels, without obtaining target signature a priori.
- Multiple instance classifier fusion and regression algorithms were presented. In particular, the MICI method is versatile in that it can perform classifier fusion and regression with minor adjustments in the objective function.
- The MIMRF algorithm extends upon MICI to multi-resolution and multi-modal sensor fusion on remote sensing data with label uncertainty. To our knowledge, this is the first algorithm that can handle HSI imagery and LiDAR point cloud fusion without co-registration or rasterization, considering imprecise labels.
- Various optimization strategies exist to optimize an MIL problem, such as expectation maximization, sampling-based evolutionary algorithm, and gradient descent.

The algorithms discussed in this chapter covers the state-of-the-art MIL approaches and provides an effective solution to address the imprecision challenges

in hyperspectral image analysis and remote-sensing applications. There are several challenges in these current approaches that warrant future work. For example, current MI regression methods often rely on the "primary instance" assumption, which may not hold in all applications; or that MIL assumes no contamination (of positive points) in negative bags, but in practice this is often not the case. Future study in more flexible MIL frameworks (such as using kernel embedding as described in Chap. 2) can be conducted in relaxing these assumptions.

References

1. Gader P, Zare A, Close R et al (2013) MUUFL gulfport hyperspectral and lidar airborne data set. Technical report, University of Florida, Gainesville, FL, REP-2013-570. Data and code. https://github.com/GatorSense/MUUFLGulfport and Zenodo. https://doi.org/10.5281/zenodo.1186326
2. Brigot G, Colin-Koeniguer E, Plyer A, Janez F (2016) Adaptation and evaluation of an optical flow method applied to coregistration of forest remote sensing images. IEEE J Sel Topics Appl Earth Observ 9(7):2923–2939
3. Cao S, Zhu X, Pan Y, and Yu Q (2014) A stable land cover patches method for automatic registration of multitemporal remote sensing images. IEEE J Sel Topics Appl Earth Observ 7(8):3502–3512
4. Dietterich TG, Lathrop RH, Lozano-Pérez T et al (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 89(1–2):31–71
5. Maron O, Lozano-Perez T (1998) A framework for multiple-instance learning. In: Advances in neural information processing systems (NIPS), pp 570–576
6. Maron O, Ratan AL (1998) Multiple-instance learning for natural scene classification. In: International conference on machine learning, vol 98, pp 341–349
7. Zhang Q, Goldman SA (2002) EM-DD: an improved multiple-instance learning technique. In: Advances in neural information processing systems (NIPS), vol 2, pp 1073–1080
8. Press WH, Flannery BP, Teukolsky SA (1992) Numerical recipes in C: the art of scientific programming. Cambridge University Press, Cambridge
9. Andrews S, Tsochantaridis I, Hofmann T (2002) Support vector machines for multiple-instance learning. In: Advances in neural information processing systems (NIPS) 561–568
10. Shrivastava A, Pillai JK, Patel VM, Chellappa R (2014) Dictionary-based multiple instance learning. In: IEEE international conference on image processing (ICIP), pp 160–164
11. Shrivastava A, Patel VM, Pillai JK, Chellappa R (2015) Generalized dictionaries for multiple instance learning. Int J Comput Vis 114(2–3):288–305
12. Mallat SG, Zhang Z (1993) Matching pursuits with time-frequency dictionaries. IEEE Trans Signal Process 41(12):3397–3415
13. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322
14. Mairal J, Bach F, Ponce J (2012) Task-driven dictionary learning. IEEE Trans Pattern Anal Mach Intell 34(4):791–804
15. Wang X, Wang B, Bai X, Liu W, Tu Z (2013) Max-margin multiple-instance dictionary learning. In: International conference on machine learning, pp 846–854
16. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
17. Fu Z et al (2011) MILIS: multiple instance learning with instance selection. IEEE Trans Pattern Anal Mach Intell 33(5):958–977
18. Chen Y, Bi J, Wang JZ (2006) MILES: multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell 28(12):1931–1947

19. Zhu J, Rosset S, Hastie T, Tibshirani R (2004) 1-norm support vector machines. In: Advances in neural information processing systems (NIPS), vol 16, pp 49–56
20. Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press
21. Zhou Z, Xu J (2007) On the relation between multi-instance learning and semi-supervised learning. In: Proceedings of the 24th international conference on machine learning, pp 1167–1174
22. Hoffman J et al (2015) Detector discovery in the wild: joint multiple instance and representation learning. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2883–2891
23. Li W, Vasconcelos N (2015) Multiple instance learning for soft bags via top instances. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 4277–4285
24. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
25. Zhang D, Meng D, Han J (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39(5):865–878
26. Kumar MP, Packer B, Koller D (2010) Self-paced learning for latent variable models. In: Advances in neural information processing systems (NIPS), pp 1189–1197
27. Wang J (2000) Solving the multiple-instance problem: a lazy learning approach. In: Proceedings of the 17th international conference on machine learning, pp 1119–1125
28. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
29. Huttenlocher DP et al (1993) Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal Mach Intell 15(9):850–863
30. Jiang L, Cai Z, Wang D et al (2014) Bayesian citation-KNN with distance weighting. Int J Mach Learn Cybern 5(2):193–199
31. Ghosh D, Bandyopadhyay S (2015) A fuzzy citation-kNN algorithm for multiple instance learning. In: IEEE international conference on fuzzy systems, pp 1–8
32. Villar P, Montes R, Sánchez A et al (2016) Fuzzy-Citation-KNN: a fuzzy nearest neighbor approach for multi-instance classification. In: IEEE international conference on fuzzy systems, pp 946–952
33. Wang X, Yan Y, Tang P et al (2018) Revisiting multiple instance neural networks. Pattern Recognit 74:15–24
34. Nasrabadi NM (2014) Hyperspectral target detection: an overview of current and future challenges. IEEE Signal Process Mag 31(1):34–44
35. Manolakis D, Marden D, Shaw GA (2003) Hyperspectral image processing for automatic target detection applications. Linc Lab J 14(1):79–116
36. Manolakis D, Truslow E, Pieper M, Cooley T, Brueggeman M (2014) Detection algorithms in hyperspectral imaging systems: an overview of practical algorithms. IEEE Signal Process Mag 31(1):24–33
37. Jiao C, Zare A (2015) Functions of multiple instances for learning target signatures. IEEE Trans Geosci Remote Sens 53(8):4670–4686
38. Zare A, Jiao C (2014) Extended functions of multiple instances for target characterization. In: IEEE workshop hyperspectral image signal process: evolution in remote sensing (WHISPERS), pp 1–4
39. Zare A, Gader P (2007) Sparsity promoting iterated constrained endmember detection for hyperspectral imagery. IEEE Geosci Remote Sens Lett 4(3):446–450
40. Jiao C, Zare A (2019) GatorSense/FUMI: initial release (Version v1.0). Zenodo. https://doi.org/10.5281/zenodo.2638304
41. Zare Jiao C, Glenn T (2018) Discriminative multiple instance hyperspectral target characterization. IEEE Trans Pattern Anal Mach Intell 65(10):2634–2648
42. Zare A, Jiao C, Glenn T (2018). GatorSense/MIACE: version 1 (Version v1.0). Zenodo. https://doi.org/10.5281/zenodo.1467358
43. Zare A, Ho KC (2014) Endmember variability in hyperspectral analysis: addressing spectral variability during spectral unmixing. IEEE Signal Process Mag 31(1):95–104

44. Jiao C, Zare A (2017) Multiple instance hybrid estimator for learning target signatures. In: IEEE international geoscience and remote sensing symposium, pp 1–4
45. Jiao C et al (2018) Multiple instance hybrid estimator for hyperspectral target characterization and sub-pixel target detection. ISPRS J Photogramm Remote Sens 146:232–250
46. Broadwater J, Chellappa R (2007) Hybrid detectors for subpixel targets. IEEE Trans Pattern Anal Mach Intell 29(11):1891–1903
47. Babenko B, Dollár P, Tu Z, Belongie S (2008) Simultaneous learning and alignment: multi-instance and multi-pose learning. In: Workshop on faces in 'Real-Life' images: detection, alignment, and recognition
48. Ramirez I, Sprechmann P, Sapiro G (2010) Classification and clustering via dictionary learning with structured incoherence and shared features. In: IEEE conference on computer vision and pattern recognition, pp 3501–3508
49. Yang M, Zhang L, Feng X, Zhang D (2014) Sparse representation based fisher discrimination dictionary learning for image classification. Int J Comput Vis 109(3):209–232
50. Yang M, Zhang L, Feng X, Zhang D (2011) Fisher discrimination dictionary learning for sparse representation. In: International conference on computer vision, pp 543–550
51. Figueiredo MAT, Nowak RD (2003) An EM algorithm for wavelet-based image restoration. IEEE Trans Image Process 12(8):906–916
52. Daubechies I, Defrise M, De Mol C (2003) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun Pure Appl Math 57(11):1413–1457
53. Nascimento JMP, Dias JMB (2005) Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Trans Geosci Remote Sens 43(4):898–910
54. Jiao C, Zare A (2018) GatorSense/MIHE: initial release (Version 0.1). Zenodo. https://doi.org/10.5281/zenodo.1320109
55. Zhong P, Gong Z, Shan J (2019) Multiple instance learning for multiple diverse hyperspectral target characterizations. IEEE Trans Neural Netw Learn Syst 31(1): 246–258
56. Baldridge AM, Hook SJ, Grove CI, Rivera G (2009) The ASTER spectral library version 2.0. Remote Sens Environ 113(4):711–715
57. Kraut S, Scharf LL (1999) The CFAR adaptive subspace detector is a scale-invariant GLRT. IEEE Trans Signal Process 47(9):2538–2541
58. Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
59. Glenn T, Zare A, Gader P, Dranishnikov D (2013) Bullwinkle: scoring code for sub-pixel targets (Version 1.0) [Software]. https://github.com/GatorSense/MUUFLGulfport/
60. Du X, Zare A (2019) GatorSense/MICI: initial release (Version v1.0). Zenodo. https://doi.org/10.5281/zenodo.2638378
61. Du X, Zare A, Keller JM, Anderson DT (2016) Multiple Instance Choquet integral for classifier fusion. IEEE Congr Evol Comput 1054–1061
62. Du X, Zare A (2019) Multiple instance Choquet integral classifier fusion and regression for remote sensing applications. IEEE Trans Geosci Remote Sens 57(5):2741–2753
63. Choquet G (1954) Theory of capacities. Ann L'Institut Fourier 5:131–295
64. Keller JM, Liu D, Fogel DB (2016) Fundamentals of computational intelligence: neural networks, fuzzy systems and evolutionary computation. IEEE press series on computational intelligence, Wiley
65. Rolewicz S (2013) Functional analysis and control theory: linear systems. Springer Science & Business Media, Dordrecht, The Netherlands
66. Du X (2017) Multiple instance choquet integral for multiresolution sensor fusion. Doctoral dissertation, University of Missouri, Columbia, MO, USA
67. Wang Z, Radosavljevic V, Han B et al (2008) Aerosol optical depth prediction from satellite observations by multiple instance regression. In: Proceedings of the SIAM international conference on data mining, pp 165–176
68. Wang Z, Lan L, Vucetic S (2012) Mixture model for multiple instance regression and applications in remote sensing. IEEE Trans Geosci Remote Sens 50(6):2226–2237

69. Wagstaff KL, Lane T (2007) Salience assignment for multiple-instance regression. In: International conference on machine learn, workshop on constrained optimization and structured output spaces
70. Wagstaff KL, Lane T, Roper A (2008) Multiple-instance regression with structured data. In: IEEE international conference on data mining workshops, pp 291–300
71. Ray S, Page D (2001) Multiple instance regression. In: Proceedings of the 18th international conference on machine learning, vol 1, pp 425–432
72. Dooly DR, Zhang Q, Goldman SA, Amar RA (2002) Multiple-instance learning of real-valued data. J Mach Learn Res 3:651–678
73. Goldman SA, Scott SD (2003) Multiple-instance learning of real-valued geometric patterns. Ann Math Artif Intell 39(3):259–290
74. Haussler D (1992) Decision theoretic generalizations of the PAC model for neural net and other learning applications. Inf Comput 100(1):78–150
75. Kearns MJ, Schapire RE, Sellie LM (1994) Toward efficient agnostic learning. Mach Learn 17(2–3):115–141
76. Kivinen J, Warmuth MK (1997) Exponentiated gradient versus gradient descent for linear predictors. Inf Comput 132(1):1–63
77. Wang ZG, Zhao ZS, Zhang CS (2013) Online multiple instance regression. Chin Phys B 22(9):098702
78. Dooly DR, Goldman SA, Kwek SS (2006) Real-valued multiple-instance learning with queries. J Comput Syst Sci 72(1):1–5
79. Cheung PM, Kwok JT (2006) A regularization framework for multiple-instance learning. In: Proceedings of the 23rd international conference on machine learning, pp 193–200
80. Gärtner T, Flach PA, Kowalczyk A, Smola AJ. Multi-instance kernels. In: Proceedings of the 19th international conference on machine learning, vol 2, no 3, pp 179–186
81. Gunn SR (1998) Support vector machines for classification and regression. ISIS Tech Rep 14(1):5–16
82. Trabelsi M, Frigui H (2019) Robust fuzzy clustering for multiple instance regression. Pattern Recognit
83. Krishnapuram R, Keller JM (1993) A possibilistic approach to clustering. IEEE Trans Fuzzy Syst 1(2):98–110
84. Davis J, Santos Costa V, Ray S, Page D (2007) Tightly integrating relational learning and multiple-instance regression for real-valued drug activity prediction. In: Proceedings on international conference on machine learning, vol 287
85. Du X, Zare A (2019) GatorSense/MIMRF: initial release (Version v1.0). Zenodo. https://doi.org/10.5281/zenodo.2638382
86. Du X, Zare A (2019) Multiresolution multimodal sensor fusion for remote sensing data with label uncertainty. IEEE Trans Geosci Remote Sens, In Press
87. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2010) Slic superpixels. Ecole Polytechnique Fédéral de Lausssanne (EPFL). Tech Rep 149300:155–162
88. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
89. OSM contributors (2018) Open street map. https://www.openstreetmap.org
90. Google (2018) Google earth. https://www.google.com/earth/
91. Google (2018) Google maps. https://www.google.com/maps/
92. Du X, Zare A (2017) Technical report: scene label ground truth map for MUUFL gulfport data set. University of Florida, Gainesville, FL, Tech Rep 20170417. http://ufdc.ufl.edu/IR00009711/00001