

Multiple Instance Choquet Integral Classifier Fusion and Regression for Remote Sensing Applications

Xiaoxiao Du[✉], Student Member, IEEE, and Alina Zare[✉], Senior Member, IEEE

Abstract—In classifier (or regression) fusion, the aim is to combine the outputs of several algorithms to boost overall performance. Standard supervised fusion algorithms often require accurate and precise training labels. However, accurate labels may be difficult to obtain in many remote sensing applications. This paper proposes novel classification and regression fusion models that can be trained given ambiguously and imprecisely labeled training data in which the training labels are associated with sets of data points (i.e., “bags”) instead of individual data points (i.e., “instances”) following a multiple-instance learning framework. Experiments were conducted based on the proposed algorithms on both synthetic data and applications such as target detection and crop yield prediction given remote sensing data. The proposed algorithms show effective classification and regression performance.

Index Terms—Choquet integral (CI), classifier fusion, multiple-instance learning (MIL), multiple-instance regression (MIR), remote sensing, target detection.

I. INTRODUCTION

CASSIFIER fusion methods aim to combine and integrate multiple classifier outputs while reducing uncertainties in the data, providing more detailed information and more accurate prediction [1], [2]. Each of the classifier (or regressor) outputs to be fused may provide complementary or reinforcing information that is helpful for a specific target detection, classification, or regression application [3].

Previous supervised fusion algorithms developed for remote sensing data often require precise labels for each training data point [4]–[7]. However, data-point specific labels are often either unavailable, difficult, or expensive to obtain in remote sensing applications. For example, consider the following target detection problem. We have collected hyperspectral imagery (HSI) over the University of Southern Mississippi–Gulfpark campus and the goal is to find targets emplaced within the scene [8]. The ground sample distance of

Manuscript received March 11, 2018; revised August 20, 2018; accepted October 14, 2018. This work was supported by the National Science Foundation through CAREER Project—Supervised Learning for Incomplete and Uncertain Data under Grant IIS-1723891. (Corresponding authors: Xiaoxiao Du; Alina Zare.)

X. Du is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: xdy74@mail.missouri.edu).

A. Zare is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: azare@ece.ufl.edu).

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

Digital Object Identifier 10.1109/TGRS.2018.2876687

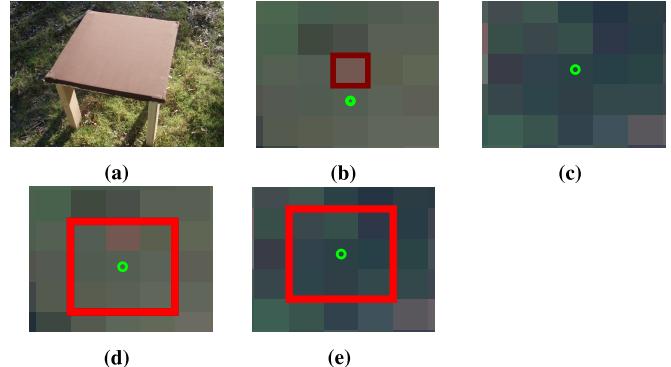


Fig. 1. Illustration of a target application given remote sensing data. (a) Brown target. (b) Inaccurate GPS location (green circle) that differs from the true brown target location (dark red rectangle) in the scene. (b) Subpixel target that cannot be visibly seen from the imagery (occluded by vegetation). (c) and (d) Red rectangles mark the approximate region that contains the targets.

the obtained HSI imagery is 1 m. This is a challenging target detection problem in which many targets are occluded (for example, hidden under a tree) and many are subpixel (target size less than 1 m^2). A global positioning system (GPS) device was used to obtain target coordinates during data collection. However, the GPS device used was only accurate to the level of several pixels. In this problem, we know the approximate locations of the targets given in the GPS information, but we cannot pinpoint the exact target locations. Fig. 1 shows an example of remote sensing data burdened with such imprecise and uncertain labels.

In the above-mentioned example, it is possible to identify an approximate region (a set of pixels) that contain the targets, as shown in Fig. 1(d) and (e). We do not know which pixel(s) exactly correspond(s) to the target, but we know that the target is somewhere in the red rectangle region. That is to say, the accurate target location per pixel is not available, but the regions or sets of pixels that contain the targets can be easily obtained from the (imprecise) GPS coordinates. It would be useful, therefore, to develop a trained classifier fusion and regression method that can be trained on labels with such uncertainty in order to better perform target detection and many other tasks given remote sensing data.

The multiple-instance learning (MIL) framework was first proposed in [9] to address uncertainty and inaccuracy in labeled data in supervised learning applications. Since then, the MIL has been used in various remote sensing applications such as spatial-spectral classification [10],

landmine detection [11], [12], seabed segmentation [13], [14], and target signature characterization [15], [16]. However, the majority of these studies were only applied to single-sensor classification problems. On the other hand, previous supervised fusion methods [4]–[7] often require accurate pixelwise training label and cannot handle uncertain/imprecise labels. There has been relatively little work on applying MIL to the label uncertainty problem in decision-level fusion. This paper aims at addressing the problem uncertain/imprecise training labels for multisensor/multiclassifier fusion applications by formulating the fusion problem as a MIL problem.

We use the Choquet integral (CI) as the aggregation operator for our proposed fusion framework. The CI has a long history of being an effective aggregation operator for nonlinear fusion [17]–[19]. Compared with commonly used aggregation operators such as weighted arithmetic means [20], the CI is able to model the relationship amongst the combinations of fusion sources and can flexibly represent a wide variety of aggregation operators [17], [21]. The Sugeno integral [22] is also a flexible, nonlinear aggregator. In this paper, we focus on the use of the CI since, in some cases in the literature, it has shown better classification performance compared with the Sugeno integral [3], [23]. However, certainly the future work will involve developing MIL Sugeno-based fusion approaches for comparison.

Standard CI fusion has been previously explored in remote sensing applications [3], [24]–[26]. However, these methods require accurate per-pixel training labels, which are difficult or impossible to obtain in remote sensing applications, as discussed earlier. We previously proposed a noisy-OR approach that can perform CI fusion with uncertain labels [27], but it only works specifically for two-class classification. It would be useful, therefore, to develop a CI fusion method that can handle uncertain labels for both classifier fusion and regression in remote sensing applications.

This paper proposes a multiple instance CI (MICI) framework for both multisensor classifier fusion and regression that can learn from ambiguously and imprecisely labeled training data. The proposed MICI fusion methods follow the MIL framework and perform fusion using the CI given uncertain and imprecise training labels. Two novel MICI classifier fusion models, the min–max model, and the generalized-mean model, are proposed. The classifier fusion models are useful for target detection given remote sensing data, among many other classification tasks. This paper also proposes a MICI regression (MICIR) model for regression problems where the desired prediction is real valued. The proposed model can fuse multiple sources with real-valued label as well as handling the uncertainties in the training labels.

This paper is organized as follows. Section I introduces the problem of uncertain labels in remote sensing data and describes our motivation in using the MIL framework and the CI fusion, which are the basis of the proposed algorithms. Section II describes the previous work in MIL and CI fusion, particularly in remote sensing applications. Section III introduces the proposed classifier fusion and regression models and Section IV describes the optimization approach. Section V presents fusion results on real target detection and crop yield

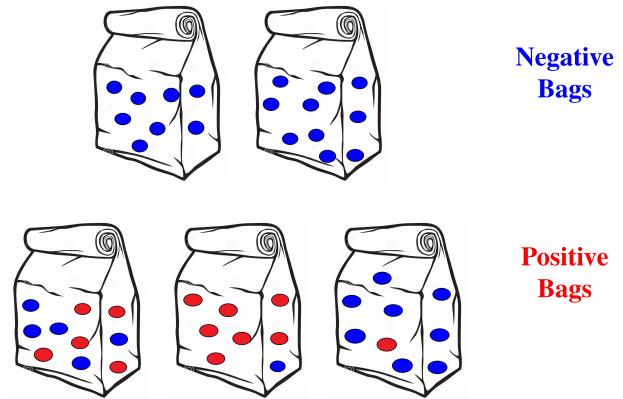


Fig. 2. Illustration of bags in MIL. Red marks: positive instances. Blue marks: negative instances. (Top row) Two negative bags. (Bottom row) Three positive bags.

prediction applications given remote sensing data. Section VII provides a summary and conclusion of this paper.

II. RELATED WORK

This section introduces related work in multiple-instance classification, classifier fusion, and multiple-instance regression (MIR). This section also describes in detail the previously proposed noisy-OR classifier fusion model to be compared with the newly proposed algorithms.

A. Multiple-Instance Classification

In the MIL framework, training labels are associated with sets of data points (“bags”) instead of each data point (“instance”). In the scenario of two-class classification, the standard MIL assumes that 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. Fig. 2 shows an illustration of MIL bags.

The expectation-maximization diverse density (EM-DD) and MI support vector machine (mi-SVM) algorithms are two widely cited MIL techniques for classification [16], [28]–[30]. The two algorithms are used later in this paper as comparison MIL approaches. The EM-DD algorithm [31] combines the expectation-maximization (EM) [32] with the diverse density (DD) objective function [9], [21], [33] to address MIL two-class classification problems. EM-DD views the relationship between all the instances in the bag and the label of the bag as a latent variable that can be estimated using the EM approach [31]. In the E-step, one instance is picked from each bag as the most influential instance for its bag-level label. In the M-step, the DD is maximized by a gradient ascent search. The process is iterated until a stopping criterion is met.

The mi-SVM algorithm was proposed by Andrews [34] as an MIL extension to support vector machine (SVM) learning approaches. The mi-SVM algorithm can work with bag-level labels in MIL. For two-class classification problems, the mi-SVM objective function is defined as [34]

$$\min_{\{y_i\}} \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \quad (1)$$

such that

$$\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \quad \forall i : y_i (< \mathbf{w}, \mathbf{x}_i > + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (2)$$

where \mathbf{w} is the weights, b is the bias, y_i is the instance-level label, and ξ_i are the slack variables (similar to that of a standard SVM). The label vector \mathbf{y} represents bag-leveled labels that satisfy the MIL assumption. The mi-SVM algorithm learns a linear discriminate function and maximizes the margin to separate the positive from the negative classes based on instance labels.

B. Multiple-Instance Regression

MIR addresses multiple instance problems where the prediction values are real valued. MIR has been used in the literature for applications such as predicting the ability of antigen peptides to bind to major histocompatibility complex class II molecules [35], predicting aerosol optical depth from remote sensing data [36], [37], and predicting crop yield [37]–[39].

Ray and Page [40] first proposed a MIR algorithm based on the “primary-instance” assumption, which assumes there is one primary instance in a bag that is responsible for the real-valued bag-level label. Wagstaff and Lane [38] and Wagstaff *et al.* [39] then investigated using MIR in predicting crop yield from real remote sensing data set. Wagstaff *et al.* [39] provided an MI-ClusterRegress algorithm (or in some references, Cluster MIR algorithm) that mapped instances onto (hidden) cluster labels. However, the Cluster MIR algorithm works under the assumption that the instances from a bag are drawn (with noise) from a set of underlying clusters and one of the clusters must be “relevant” to the bag-level labels. The algorithm assumes that the bag data follow a mixture of Gaussian distributions [39]. More recently, Trabelsi and Frigui [41] proposed a robust fuzzy clustering for MIR (RFC-MIR) algorithm that, similar to Cluster-MIR, clusters the training instances and learns multiple regression models for each cluster. Both Cluster-MIR and RFC-MIR pool all instances from all training bags in order to perform clustering.

C. Choquet Integral and Fuzzy Measures

The CI is an aggregation operator based on the fuzzy measures [42]. Depending on the values of each element in the fuzzy measure, the CI can represent a variety of relationships and combinations amongst the information sources. Therefore, a crucial aspect of using the CI for information/sensor fusion is learning the fuzzy measures for the CI [43], [44]. The algorithms proposed in this paper rely on the CI and fuzzy measures, and this section provides the definition of the CI and fuzzy measures.

Consider the case that there are m sources, $C = \{c_1, c_2, \dots, c_m\}$, for fusion. The “sources” refer to the outputs from the set of m classifiers or regressors to be fused. The set C contains $2^m - 1$ nonempty subsets. The power set of all (crisp) subsets of C is denoted 2^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 [22], [45]–[47].

- 1) $g(\emptyset) = 0$.
- 2) $g(C) = 1$; *normalized*.
- 3) $g(A) \leq g(B)$ if $A \subseteq B$ and $A, B \subseteq C$. *monotonic*.

In this paper, the monotonic normalized discrete CI is used as the aggregation operator for both classifier fusion and regression. Suppose we are fusing m classifier or regressor sources. Denote the classifier/regressor output of k th classifier/regressor, c_k , on n th data point/instance, \mathbf{x}_n , as $h(c_k; \mathbf{x}_n)$. The discrete CI on instance \mathbf{x}_n given C is then computed as [27], [42], [47]

$$C_{\mathbf{g}}(\mathbf{x}_n) = \sum_{k=1}^m [h(c_k; \mathbf{x}_n) - h(c_{k+1}; \mathbf{x}_n)] g(A_k) \quad (3)$$

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

Previously, we proposed an MICI classifier fusion method that also uses the CI and MIL framework with a noisy-OR objective function [27]. Recall that in standard MIL, a bag is labeled negative if all the instances in the bag are negative and a bag is labeled positive if there is at least one positive instance in the bag. We modeled the MIL assumption using a noisy-OR objective function [27], [33]

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{a=1}^{B^-} \sum_{i=1}^{N_a^-} \ln (1 - \mathcal{N}(C_{\mathbf{g}}(\mathbf{x}_{ai}^-) | \mu, \sigma^2)) \\ + \sum_{b=1}^{B^+} \ln \left(1 - \prod_{j=1}^{N_b^+} 1 - \mathcal{N}(C_{\mathbf{g}}(\mathbf{x}_{bj}^+) | \mu, \sigma^2) \right) \quad (4)$$

where B^+ is the total number of positive bags, B^- is 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 i th instance in the a th negative bag, and \mathbf{x}_{bj}^+ is the j th instance in the b th positive bag. μ and σ^2 are the mean and variance of the Gaussian function, respectively. For two-class classifier fusion problems, in this paper, the positive class (target) is marked label “+1” and the negative class (nontarget) is marked label “0.” The parameter μ can be set, in this case, to 1, to encourage the CI values of positive instances to be 1 and the CI values of negative instances to be far from 1. Here, the model parameter vector $\boldsymbol{\theta}$ consists of the variance of the Gaussian σ^2 and the fuzzy measure \mathbf{g} values used to compute the CI.

However, this previously proposed noisy-OR model was subject to the user-defined parameter setting such as the variance σ^2 [27]. The variance parameter controls how sharply the CI values are pushed to 0 and 1 and, thus, controls the weighting of the two terms separately. A larger variance parameter allows for more noise in the data by allowing

points in negative bags to have higher CI values and positive points to have lower CI values. In this paper, we propose new algorithms that can eliminate the variance parameter, yet still able to perform effective classifier fusion and handle uncertain labels.

Furthermore, the previously proposed noisy-OR model is a slow algorithmic approach. The noisy-OR model needs to compute the valid interval for each fuzzy measure element at each iteration during optimization and can be quite slow. In this paper, we propose a novel optimization technique that will improve the computation time significantly. In addition, the previous noisy-OR model only addresses two-class classification problems. In this paper, however, we shall address both two-class classification problems and real-valued regression problems given remote sensing data with uncertain training labels.

D. Contributions

The contributions of this paper are fourfold. First, all the newly proposed algorithms work under the multiple instance assumption and can learn from uncertain and imprecise training labels, which is a novelty in remote sensing data fusion. Second, this paper proposes two novel MICI models for classifier fusion, which successfully eliminated the variance parameter of the previously proposed noisy-OR model. Third, this paper proposes a novel MICIR algorithm, which extends for regression applications where the predictions are real valued. Finally, in order to learn the fuzzy measure to be used with the CI, a new optimization scheme is proposed in this paper which allows significantly improved computation time as compared to our previous approaches for MIL fusion.

III. PROPOSED ALGORITHMS

This section describes the proposed two MICI classifier fusion models and a novel MICIR algorithm for regression applications.

A. Multiple-Instance Choquet Integral Algorithms for Classifier Fusion

Two MICI models, the min–max model and generalized-mean model, are proposed for classifier fusion. The proposed models can perform multisensor classifier fusion while learning from uncertain and imprecise training labels. A monotonic normalized fuzzy measure is learned to be used with the CI to perform two-class classifier fusion given bag-level training labels. An optimization scheme based on an evolutionary algorithm is used to optimize the models proposed.

1) Min–Max Model: The min–max model uses “min” and “max” operators to follow the MIL assumption. The MIL framework assumes that for negative bags, all the instances in the bag are negative (label “0”). Thus, we write the objective function for the negative bags as

$$J_M^- = \sum_{a=1}^{B^-} \max_{\forall \mathbf{x}_{ai}^- \in \mathcal{B}_a^-} (C_g(\mathbf{x}_{ai}^-) - 0)^2. \quad (5)$$

For positive bags, at least one instance in the bag should be positive (label “1”), so we write the objective function for the positive bags as

$$J_M^+ = \sum_{b=1}^{B^+} \min_{\forall \mathbf{x}_{bj}^+ \in \mathcal{B}_b^+} (C_g(\mathbf{x}_{bj}^+) - 1)^2 \quad (6)$$

where B^+ is the total number of positive bags, B^- is the total number of negative bags, \mathbf{x}_{ai}^- is the i th instance in the a th negative bag, and \mathbf{x}_{bj}^+ is the j th instance in the b th positive bag. C_g is the CI output given measure g computed using (3), \mathcal{B}_a^- is the a th negative bag, and \mathcal{B}_b^+ is the b th positive bag.

Thus, the objective function for the MICI min–max model classifier fusion approach is written as follows:

$$J_M = J_M^- + J_M^+. \quad (7)$$

By minimizing the objective function in (7), we encourage the CI of all the instances in the negative bag to zero (“ J_M^- ” term) and encourage the CI of at least one of the points in the positive bag to one (“ J_M^+ ” term), which fits the MIL assumption. This objective function relies entirely on the training data and labels and does not require user-set variance parameters as needed previously in the noisy-OR model.

2) Generalized-Mean Model: The “min” and “max” operators proposed in the above-mentioned min–max model strictly follows the MIL assumption where only one point is selected for each of the positive bags. Here, in this second model, a generalized-mean model is proposed to allow more points to contribute to the class labels.

If p is a nonzero real number, and x_1, \dots, x_n are positive real numbers, then the generalized mean with the exponent p of x_1, \dots, x_n is defined as [48]

$$M_p(x_1, x_2, \dots, x_n) = \left(\frac{1}{n} \sum_{i=1}^n x_i^p \right)^{\frac{1}{p}}. \quad (8)$$

The generalized mean has the following two properties:

$$\begin{aligned} M_{-\infty}(x_1, x_2, \dots, x_n) &= \lim_{p \rightarrow -\infty} M_p(x_1, x_2, \dots, x_n) \\ &= \min(x_1, x_2, \dots, x_n). \end{aligned} \quad (9)$$

$$\begin{aligned} M_\infty(x_1, x_2, \dots, x_n) &= \lim_{p \rightarrow \infty} M_p(x_1, x_2, \dots, x_n) \\ &= \max(x_1, x_2, \dots, x_n). \end{aligned} \quad (10)$$

Therefore, by adjusting the p value, the term can act as varying aggregating operators.

For negative bags, all the instances in the bag are negative (label “0”). This assumption can be expressed using the generalized-mean model as

$$J_G^- = \sum_{a=1}^{B^-} \left[\frac{1}{N_a^-} \sum_{i=1}^{N_a^-} (C_g(\mathbf{x}_{ai}^-) - 0)^{2p_1} \right]^{\frac{1}{p_1}}. \quad (11)$$

Similarly, for positive bags, at least one instances in the bag should be positive (label “1”)

$$J_G^+ = \sum_{b=1}^{B^+} \left[\frac{1}{N_b^+} \sum_{j=1}^{N_b^+} (C_g(\mathbf{x}_{bj}^+) - 1)^{2p_2} \right]^{\frac{1}{p_2}} \quad (12)$$

where B^+ is the total number of positive bags, B^- is the total number of negative bags, N_b^+ is the total number of instances in positive bag b , N_a^- is the total number of instances in negative bag a , \mathbf{x}_{ai}^- is the i th instance in the a th negative bag, and \mathbf{x}_{bj}^+ is the j th instance in the b th positive bag. C_g is the CI given measure \mathbf{g} , \mathcal{B}_a^- is the a th negative bag, and \mathcal{B}_b^+ is the b th positive bag. p_1 and p_2 are the two exponent parameters of the generalized-mean functions.

Thus, the objective function for the proposed MICI generalized-mean classifier fusion is written as follows:

$$J_G = J_G^- + J_G^+. \quad (13)$$

By minimizing the objective function in (13), we encourage the CI of all the points in the negative bag to zero (“ J_G^- ” term) and encourage the CI of at least one of the points in the positive bag to one (“ J_G^+ ” term). The p term allows more points to contribute to the class labels. When $p_1 \rightarrow \infty$ and $p_2 \rightarrow -\infty$, according to properties (9) and (10), the generalized-mean terms become 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 [49].

B. Multiple-Instance Choquet Integral Regression Algorithm

A new MICIR model is proposed to solve regression problems where the desired prediction values are real valued. The proposed MICIR algorithm adopts the “primary-instance” assumption that there is one primary instance responsible for the label for each bag [40]. The MICIR algorithm fuses multiple sources with real-valued label as well as taking into account the uncertainties in the label.

In parallel with the proposed classifier fusion algorithm, given a set of training data and (real-valued) bag-level training labels, the proposed MICIR learns a fuzzy measure to be used with the CI and the CI value is used to perform regression on the testing data. The following objective function is proposed to optimize the fitness for regression given a measure \mathbf{g}

$$\min \sum_{b=1}^{N_b} \min_{\forall i, x_{bi} \in X_b} (C_{\mathbf{g}}(\mathbf{x}_{bi}) - d_b)^2 \quad (14)$$

where N_b is the total number of training bags, d_b is the desired training bag-level label for bag b , X_b is the set of all the instances in bag b , and $C_{\mathbf{g}}(\mathbf{x}_{bi})$ is the CI output for the i th instance in bag b given measure \mathbf{g} . The CI output is computed based on (3). The objective function encourages the CI value of one instance in the bag b to the desired real-valued label d_b .

The proposed MICIR algorithm is applicable to classification problems as well as regression problems with uncertain labels. For regression problems, the desired label d_b is real valued. The objective function selects the CI value of one instance in the bag b to match the desired real-valued label d_b . For two-class classification problems, we set the desired training bag label $d_b = 1$ for positive bags and $d_b = 0$ for

negative bags. In this case, the training data bags need to be reconstructed. The MIL assumes that a bag is labeled negative if all of the instances in the bag are negative. Therefore, each negative instance needs to form its own negative bag. The positive bag can stay the same, as the MIL assumes that bags are labeled positive if at least one instance in the bag is positive. In this case, the objective function encourages all the negative points to label “0” and encourages one of the instances in the positive bags to be “+1.”

IV. OPTIMIZATION

The proposed MICI min–max and generalized-mean models and the proposed MICIR algorithm learn a fuzzy measure to be used within a CI for two-class classifier fusion and regression. The computed CI with the learned measure is used as the estimated class labels. The goal is to learn the unknown measure from training data and known bag-level training labels and match the computed CI value as closely as possible to the desired labels. To learn the measure, a new optimization scheme based on the evolutionary algorithm is used to optimize the models proposed.

In previous studies, the valid interval was used to sample measure element values [27]. The term “valid interval” defines how large the element value can change without sacrificing monotonicity. The valid interval width for each measure element is set as the difference between the lower and upper bound for each measure element. The lower and upper bounds of a measure element are computed as the largest value of its subsets and the smallest value of its supersets, respectively. Essentially, the valid interval represents how much “wiggle room” each measure element has. The previous optimization scheme evaluates the valid intervals of each measure element and samples new element values according to the valid intervals. This method was proven effective in [27], yet it requires a long computation time as the valid interval changes for each measure element after each iteration, and the algorithm needs to evaluate all valid intervals again for the next sample.

In this section, a new optimization scheme based on the usage of measure element is proposed. First, for all the instances in the training bags, it is easy to obtain which measure element is used for which instance by sorting values from all sources (according to the definition of CI). Then, a new measure element is sampled according to a multinomial distribution that is based on the counts of how many times a measure element was used in all the training instances.

Pseudocode for the proposed optimization scheme for both training and testing stages for the proposed models can be seen in Algorithm 1. In the pseudocode, \mathbf{F}_P^0 is the fitness values for all measures in the initial population, I is the maximum number of iterations, P is the measure population size, \mathcal{G} refers to all measures in the current measure population, $\mathcal{G}\{p\}$ is the p th measure in measure population \mathcal{G} , η is the rate of small-scale mutation, \mathbf{F}_P^t is the fitness values for all measures in Iteration t , F_d is the difference of best fitness values between the current and last iteration, and F_T is the threshold of the difference of best fitness value. If $F_d \leq F_T$, the stopping criterion is met and the optimization process

stops. The value F^* is the best (minimum) current fitness value, and \mathbf{g}^* is the best current measure with the highest fitness value.

Algorithm 1 Proposed Optimization Scheme

TRAINING

Require: Training Data, Training Labels, Parameters

```

1: Initialize a population of measures. Set  $t = 0$ .
2:  $F^* = \max(\mathbf{F}_P^0)$ ,  $\mathbf{g}^* = \arg \max_{\mathcal{G}} \mathbf{F}_P^0$ .
3: Evaluate counts of usage of each measure element from
   training data.
4: while  $t < I$  do
5:   for  $p := 1 \rightarrow P$  do
6:     Randomly generate  $z \in [0, 1]$ .
7:     if  $z < \eta$  then
8:       Sample  $\mathcal{G}\{p\}$  by small-scale mutation.
9:     else
10:      Sample  $\mathcal{G}\{p\}$  by large-scale mutation.
11:    end if
12:   end for
13: Evaluate fitness of sampled measures using (7) or (13)
   or (14), depending on classification or regression
   problems.
14: Select measures.
15: Compute  $F_d = |\max(\mathbf{F}_P^t) - F^*|$ .
16: if  $\max(\mathbf{F}_P^t) > F^*$  then
17:    $F^* = \max(\mathbf{F}_P^t)$ ,  $\mathbf{g}^* = \arg \max_{\mathcal{G}} \mathbf{F}_P^t$ .
18: end if
19: if  $F_d \leq F_T$  then break;
20: end if
21:  $t \leftarrow t + 1$ .
22: end while
   return  $\mathbf{g}^*$ 

```

TESTING

Require: Testing Data, \mathbf{g}^*

```

23:  $TestLabels \leftarrow$  Choquet integral output computed based
   on Equation (3) using the learned  $\mathbf{g}^*$  above.
   return  $TestLabels$ 

```

A. Measure Initialization

In the algorithm, a population (size P) of the CI measures is generated, and each measure in the population is initialized randomly to a set of values between $[0, 1]$ that satisfies monotonicity.

Two types of initialization approaches, “top-down” and “bottom-up” approaches, were implemented. In the “top-down” initialization, the values of the measure elements were sampled from the top of the lattice toward the bottom. Suppose we have four sources to fuse ($m = 4$), for example, the $(m - 1)$ -tuple measure elements (g_{123} , g_{124} , g_{134} , and g_{234}) were first sampled randomly between 0 and 1. Then, the $(m - 2)$ -tuple measure elements (g_{12} , g_{13} , g_{14} , g_{23} , g_{24} , and g_{34}) were sampled between 0 and its corresponding superset. For example, g_{12} measure element value is sampled randomly between 0 and $\min(g_{123}, g_{124})$, due to the monotonicity

property of fuzzy measures. The process goes on until the singletons (g_1, g_2, g_3, g_4) were each sampled between 0 and their corresponding superset values.

Similarly, the “bottom-up” approach samples measure element values from the bottom of the lattice up. First, the singletons were sampled between 0 and 1, randomly. Then, the tuples were sampled between its corresponding subsets and 1. For example, in the four-source case, the initial value of g_{12} was sampled randomly between $\max(g_1, g_2)$ and 1. The process goes on until the $(m - 1)$ -tuple measure elements were sampled, thus initializing all the element values in the entire measure. Note that the measure element corresponding to the full set is always equal to 1 (e.g., $g_{1234} \equiv 1$ for four sources), according to the normalization property of the fuzzy measure.

In our experiments, the two initialization approaches yield different sets of measure element values but seem to have little impact on the final measure learned. In the following experiments, the measure is initialized by randomly flipping a coin and pick either “top-down” or “bottom-up” initialization approach.

B. Evaluation of Counts of Usage of Measure Elements

The counts of usage of measure elements can be obtained directly from training data by sorting values from source values in training. For example, three sensors yield 0.8, 0.2, and 0.1 values, respectively, for a data point. To fuse these results using the CI, according to the definition of CI, g_1 , g_{12} , and g_{123} will be used. This process is repeated for all training data points and the count of how many times each measure element will be used in the CI is recorded. The count value will be recorded as a vector $\mathbf{v} = \{v_1, v_2, \dots, v_{2^m-1}\}$ for m fusion sources. The count value vector has the same length as the fuzzy measure vector.

C. Mutation

In the optimization, mutations of two different scales were designed in search of the optimal solution. The measure element usage count obtained above is used in determining which measure element to sample during both types of mutations.

In the small-scale mutation, only one measure element is sampled. The element to be sampled is chosen by randomly sampling from a multinomial distribution based on the counts of how many times a measure element was used in all the training instances. The probability of sampling a particular measure element g_l is set to

$$P(g_l) = \frac{v_l}{\sum_{o=1}^{2^m-1} v_o} \quad (15)$$

where v_l is the number of times measure element g_l is used in training data. The measure element that was used most frequently by the training data to compute the CI will have the largest probability to be updated. In the large-scale mutation, all the measure elements are sorted in descending order based on the number of times it was used by the training data and all measure elements are updated according to the sort order. The new measure values are sampled from

a truncated Gaussian (TG) distribution. The details of TG sampling method can be seen in [50].

The rate of the small-scale mutation $\eta \in [0, 1]$ is defined by users. The rate of large-scale mutation is $1 - \eta$.

D. Selection

The measures retained for the next generation are selected based on their fitness function values computed using (7), (13) or (14), depending on using the min–max model, generalized-mean model, or the MICIR algorithm for classifier fusion or regression. In each iteration, all measures in the population are sampled, yielding a child measure population of size P . The measure population before sampling is regarded as the parent measure population (size P). Both the parent and child measure populations are pooled together (size $2P$) and their fitness values are computed using (7), (13), or (14). Then, $P/2$ measures with the top 25% fitness values are kept and carried over to the next iteration (elitism), and the remaining $P/2$ measures to be carried over are sampled according to a multinomial distribution based on their fitness values from the remaining 75% of the parent and child population pool, following a similar approach to (15). Among the new measure population, the measure with the highest fitness value is kept as the current best measure \mathbf{g}^* . The process continues until a stopping criterion is reached, such as when the maximum number of iterations is reached or the change in the objective function value from one iteration to the next is smaller than a fixed threshold.

At the end of the training process, the best measure \mathbf{g}^* with the best fitness value so far is returned as the learned measure and used for testing. Note that we are minimizing the objective function, so the minimum fitness value is regarded as the best fitness value.

V. EXPERIMENTAL RESULTS

This section describes the experiments and presents the results of the proposed algorithms on synthetic as well as real remote sensing data. Synthetic data sets were generated to evaluate the effect of parameters in both classification and regression. In addition, the MUUFL Gulfport HSI data set and the Heterogeneous Agricultural Research Via Interactive, Scalable Technology (HARVIST) crop yield data set was used for real target detection and crop yield prediction applications.

A. Synthetic Classification Data Set

A synthetic five-source classification data set was constructed to investigate the effect of “contamination” in the training bags. “Contamination” is defined as there are positive points mixed in the negative bags. The data set was constructed with 100 bags and 10 data points per bag. Half of the bags are positive (with label “1”) and half are negative (with label “0”). In the following experiments, $P = 30$, $\sigma^2 = 0.1$, $I = 5000$, $F_T = 0.0001$, $\eta = 0.8$, $\mu = 1$, $p_1 = 10$, and $p_2 = -10$.

Assume half of the training bags are positive and contain 100% positive points. Then, a varying percentage of positive points were added to the negative bags so that

TABLE I
RELATIVE ERROR VERSUS CONTAMINATION (DENOTED AS “CON.”,
IN PERCENTAGE) FOR SYNTHETIC CLASSIFICATION DATA SET
FOR MICI NOISY-OR MODEL, MIN-MAX MODEL, AND
GENERALIZED-MEAN MODEL ACROSS FIVE RUNS. THE
STANDARD DEVIATION IS NOTED IN PARENTHESES

Con. (%)	noisy-or	min-max	generalized-mean
0%	0.371(0.026)	0.288(0.025)	0.247(0.037)
10%	0.492(0.008)	0.514(0.003)	0.517(0.002)
20%	0.534(0.007)	0.536(0.011)	0.551(0.006)
30%	0.588(0.007)	0.570(0.017)	0.595(0.005)
40%	0.630(0.012)	0.655(0.005)	0.660(0.002)
50%	0.668(0.004)	0.686(0.005)	0.694(0.007)
60%	0.697(0.014)	0.255(0.171)	0.709(0.020)
70%	0.751(0.012)	0.217(0.191)	0.744(0.022)
80%	0.787(0.019)	0.359(0.304)	0.783(0.027)
90%	0.812(0.012)	0.579(0.255)	0.821(0.017)
100%	0.868(0.014)	0.345(0.366)	0.882(0.011)

the negative bags were “contaminated” in various degrees. Relative error [39] is used to evaluate the performance for the MICI classification models

$$\text{Error}_{\text{reg}}(y, \hat{y}) = |y - \hat{y}| \quad (16)$$

where y is the true label (“1” or “0”) and \hat{y} is the estimated label for each data point.

The relative error results (across all data points) with MICI noisy-OR model, min–max model, and generalized-mean model were presented in Table I. As can be seen, in all three models, the relative error generally increases as the percentage of contamination increases, which is as expected. When the percentage of contamination goes toward 100%, the error is nearly 1 (100% wrong) as well, which makes sense as the negative bags are now filled with positive points and it is not possible to distinguish positive and negative points. Note that the proposed min–max model optimizes only one instance in each bag based on the standard MIL assumption. In the min–max model, depending on the random generalization of this synthetic data set, the remaining instances (other than the max and min instance) in the bags can have varying CI values and will impact the overall relative error.

B. Synthetic Regression Data Set

A synthetic five-source regression data set was constructed to investigate the performance of the proposed MICIR algorithm. In this experiment, the percentage of primary instances in the training bags and signal-to-noise ratio (SNR) changes, and the performance of the proposed MICIR algorithm is observed. All data sets used in this section have 1000 data points. Each data point has five dimensions. The data points were grouped into 10 bags, each bag with 100 data points. The data points in this regression data set have real-valued labels between [0, 1]. Relative error [39] is used to evaluate the performance for the MICIR algorithm

$$\text{Error}_{\text{reg}}(y, \hat{y}) = \begin{cases} \left| \frac{y - \hat{y}}{y} \right|, & \text{if } y \in (0, 1] \\ |\hat{y}|, & \text{if } y = 0 \end{cases} \quad (17)$$

TABLE II

RELATIVE ERROR VERSUS PERCENTAGE OF PRIMARY INSTANCES FOR SYNTHETIC REGRESSION DATA SET FOR MICIR MODEL ACROSS FIVE RUNS. THE STANDARD DEVIATION IS NOTED IN PARENTHESES

	Percentage of primary instances					
	0%	10%	20%	30%	40%	50%
Relative Error	0.730(0.001)	0.488(0.001)	0.492(0.000)	0.451(0.001)	0.365(0.000)	0.301(0.001)
	60%	70%	80%	90%	100%	
Relative Error	0.345(0.000)	0.274(0.001)	0.104(0.001)	0.059(0.002)	0.002(0.002)	

TABLE III

RELATIVE ERROR VERSUS SNR FOR SYNTHETIC REGRESSION DATA SET MICIR MODEL ACROSS FIVE RUNS. THE STANDARD DEVIATION IS NOTED IN PARENTHESES

	SNR value					
	0dB	5dB	10dB	15dB	20dB	25dB
Relative Error	0.691(0.027)	0.443(0.029)	0.266(0.013)	0.173(0.025)	0.101(0.013)	0.061(0.012)
	30dB	35dB	40dB	45dB	50dB	
Relative Error	0.044(0.003)	0.021(0.003)	0.014(0.002)	0.008(0.002)	0.005(0.001)	

where y is the true label and \hat{y} is the estimated label for each data point.

The proposed MICIR algorithm operates under the assumption that only one primary instance is associated with the label of each bag [40]. First, we vary the percentage of primary instances in the bags to observe the relationship between the percentage of primary instances and the performance of the proposed MICIR algorithm. The percentage of primary instances in the bag takes the value of 0%–100% with an increment of 10%. For each bag, the “primary instances” have label values that are exactly the same as the bag-level training label. The nonprimary instances have randomly generated label values (different from the bag-level labels) that are generated from a completely random measure. Table II shows the relationship between the percentage of primary instances in the bag and the mean relative error over all the data points across five runs. As can be seen from the table, when the percentage of primary instances in the bag increases, the relative error decreases.

Second, the performance of MICIR is observed with varying levels of SNR. In this experiment, 100% of the points in the bag are primary instances, and varying the amount of noise adds to the entire data set, creating SNR value from 50 to 0 dB with an increment of –5 dB. Table III shows the relationship between the SNR values in the bag and the mean relative error over all the data points across five runs. As can be seen from the table, the relative error decreases when the SNR value increases, as expected.

C. MUUFL Gulfport Target Detection Experiments

The proposed MICI min–max model and generalized-mean model were tested on a real target detection application using the MUUFL Gulfport hyperspectral data set. The MUUFL Gulfport hyperspectral data set [8] was collected over the University of Southern Mississippi–Gulf Park Campus. The data set used in this experiment consists of three hyperspectral data cubes collected on three separate flights at an altitude of 3500' over the campus area. The HSI data cubes have a

ground sample distance of 1 m.¹ The image from campus 1 is 325 × 337 pixels in size. The image from campus 3 is 329 × 345 pixels in size. The image from campus 4 is 333 × 345 pixels in size. All HSI data cubes contain 72 bands corresponding to wavelengths 367.7–1043.4 nm and were collected using the CASI hyperspectral camera [8], [15]. In this experiment, the first four and last four noisy bands were removed.

A total of 60 cloth panel targets were placed in the scene. The targets were cloth panels of 5 different colors: 15 brown, 15 dark green, 12 faux vineyard green (FVG), 15 pea green, and 3 vineyard green (vineyard green targets are not considered in this experiment as the target signature was not available and target number too small). The goal is to find all (brown, dark green, FVG, and pea green) targets in the scene. These targets varied from subpixel targets (at 0.25 m² corresponding to a quarter of a pixel in area) up to superpixel targets (at 9 m²) with varying levels of occlusion. For each target, a GPS ground truth location was collected using a Trimble Juno SB handheld device. Fig. 3 shows the RGB images of the two flights over the campus and the GPS target locations. In this experiment, the GPS device has accuracy up to 5 m. Therefore, the ground truth locations for each target are only accurate within a 5 × 5 pixel halo. The MIL approach fits the problem well and the proposed MICI min–max and generalized-mean models were used to perform target detection given the inaccuracy in the ground truth labels.

The adaptive coherence estimator (ACE) detector [51]–[53] was applied to the imagery using spectral signatures of four of the target types (as spectral signatures for these targets were available from previous studies²). The background mean and background covariance for the ACE detector was estimated

¹The data set is available at <https://github.com/GatorSense>. The three flights used in this experiment corresponds to “muufl_gulfport_campus_w_lidar_1.mat,” “muufl_gulfport_campus_3.mat,” and “muufl_gulfport_campus_4.mat.”

²The target spectra used in this experiment come from “tgt_img_spectra.mat” in the data set. The four target types are: brown, dark green, FVG, and pea green.

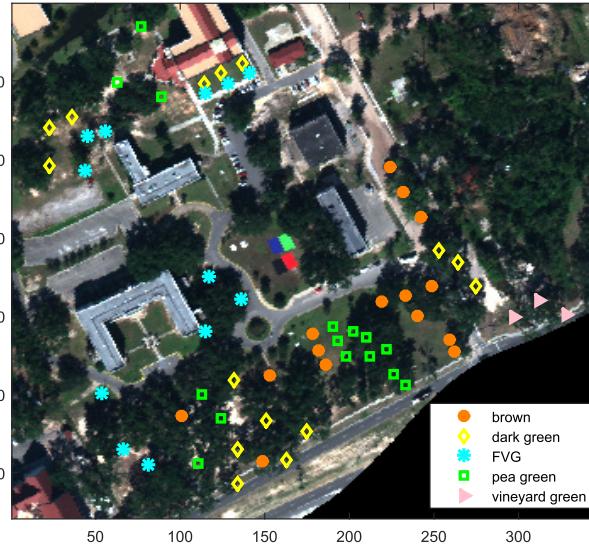


Fig. 3. RGB image from MUUFL Gulfport “campus 3” data set. Colored markers: true target locations for all target types.

using the global mean and covariance of all the pixels in each image. The ACE results used in this experiment were normalized between [0, 1].

Cross validation is performed on this data set, i.e., training on campus 1 and testing on campus 3 and campus 4, and so on. First, the mean and covariance of the training imagery, μ_{tr} and cov_{tr} , were computed. Then, the ACE detection map for the training imagery, ACE_{tr} , was obtained using μ_{tr} and cov_{tr} . Each pixel in the detection map has four dimensions; each dimension corresponds to ACE confidence values for four target types. Each of the ACE results highlights different locations corresponding to different targets. The four ACE confidence maps are the sources to be fused by the proposed MICI models.

To construct training bags, a 5×5 window was put around each ground-truth target location of the training imagery. Each window forms a positive bag and all the pixels in the windows are instances in the positive bag. The size of the positive bag corresponds to the accuracy of the GPS device used to collect the ground truth points. One negative bag was constructed by randomly picking 1600 background pixels that do not belong to any of the windows. The positive bags were labeled “1” and the negative bag was labeled “0.” The proposed models were applied to the training data and a fuzzy measure \mathbf{g}^* is learned.

In testing, the ACE detection results for the test imagery, ACE_{te} , were computed using training mean and covariance μ_{tr} and cov_{tr} . Then, the CI, given the learned measure \mathbf{g}^* and ACE_{te} , was computed as the test fusion result.

The proposed MICI min–max and MICI generalized-mean models were first compared with the four ACE results using the four individual target signatures (Brown, Dark Green, FVG, and Pea Green). The fusion results were then compared with a standard SVM on the four ACE maps, and taking the max, min, or mean over the four ACE results. The proposed models were also compared with CI-QP [54], mi-SVM [34],

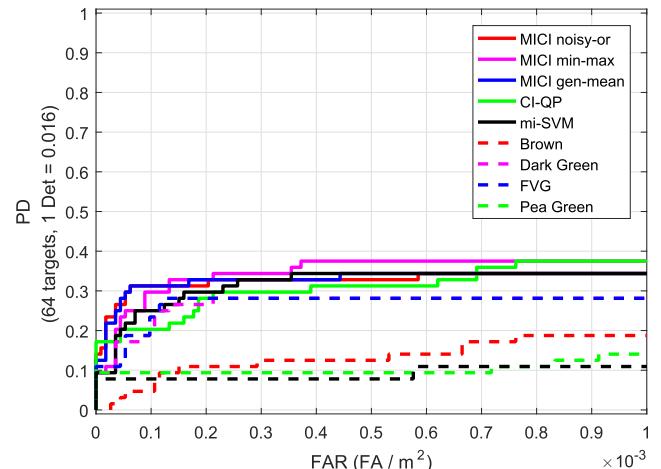


Fig. 4. ROC curve for one run of train on campuses 1 and 4, test on campus 3.

EM-DD [31], and the previously proposed MICI noisy-OR methods. The CI-QP approach learns a fuzzy measure for CI by optimizing a least-squares error objective using quadratic programming. The CI-QP approach assumes an accurate label for every training data point and, thus, does not inherently support MIL-type learning. In our application of CI-QP to this problem, we gave all points in a positive bag the label of “1” and all points in the negative bag as a label of “0.” The mi-SVM and EM-DD are both widely used MIL approaches as discussed in the literature review.

The receiver operating characteristic (ROC) curve is used to evaluate the target detection results. The ROC curve plots the positive detection rate (PD, y-axis) against the false alarm rate (FAR, x-axis). The performance of the algorithms was evaluated by computing the area under the curve (AUC) results with FAR up to $1 \times 10^{-3}/m^2$ (corresponding to a reasonable scale of 1 false alarm in 1000 m^2).

Table IV illustrates the AUC results at FAR up to $1 \times 10^{-3}/m^2$ for all fusion methods with unnormalized hyperspectral data. In the AUC table, the “best” performance was determined by comparing the mean value of the AUC results, and in the case where the mean is the same between methods, the one result with smaller standard deviation is preferred. The best two results were **bolded** and underlined, respectively. It can be seen from Table IV that the proposed MICI min–max and MICI generalized-mean models have mostly bolded and/or underlined (best and second best) results.

Fig. 4 shows the ROC curve plot after one run using unnormalized campuses 1 and 4 data as training data and test on campus 3 data. Fig. 4 provides a visual ROC curve example to complement the AUC table results. The remaining cross-validation experiments yield similar ROC curve results. The x-axis of the ROC curves represents the FAR between [0, 0.001] and the y-axis represents PD.

As can be seen from Table IV and Fig. 4, detection results of individual sources are relatively poor, which is understandable as the target types and signature vary greatly, and it is preferable to fuse detector outputs for all target types rather than use individual target detector. The CI-QP method

TABLE IV

AUC RESULTS AT ON UNNORMALIZED MUUFL GULFPORT DATA ACROSS FIVE RUNS. THE STANDARD DEVIATION IS NOTED IN PARENTHESES

Notes	Methods	Train1Test3	Train1Test4	Train3Test1	Train3Test4	Train4Test1	Train4Test3
<i>Sources: individual target types.</i>	Brown	0.265	0.264	0.334	0.267	0.307	0.263
	Dark Green	0.266	0.261	0.328	0.256	0.293	0.266
	FVG	0.114	0.106	0.122	0.107	0.136	0.109
	Pea Green	0.088	0.000	0.107	0.000	0.100	0.091
<i>Instance-based fusion methods in comparison.</i>	SVM	0.185	0.195	0.164	0.175	0.245	0.220
	min	0.000	0.000	0.046	0.073	0.026	0.023
	max	<u>0.345</u>	0.329	0.459	<u>0.339</u>	0.349	0.328
	mean	0.224	0.221	0.269	0.214	0.260	0.246
	CI-QP	0.328	0.325	0.399	0.330	0.260	0.272
<i>MIL comparison methods.</i>	mi-SVM	0.346	0.337	0.350	0.293	0.317	0.317
	EM-DD	0.062(0.014)	0.073(0.013)	0.002(0.003)	0.021(0.004)	0.005(0.008)	0.021(0.021)
	MICI Noisy-Or	0.346(0.000)	<u>0.331(0.001)</u>	0.461(0.000)	0.340(0.000)	0.349(0.000)	0.329(0.000)
<i>Proposed MICI models.</i>	MICI Min-Max	0.345(0.001)	<u>0.331(0.008)</u>	0.463(0.010)	0.333(0.013)	0.348(0.002)	0.329(0.001)
	MICI Generalized Mean	0.345(0.000)	0.331(0.005)	0.466(0.004)	0.339(0.001)	0.349(0.001)	0.330(0.002)

TABLE V

RMSE ERROR FOR CA AND KS CORN AND WHEAT YIELD. THE TWO RESULTS WITH THE LOWEST ERRORS WERE **BOLDED** AND UNDERLINED, RESPECTIVELY. THE UNIT IS BUSHELS PER ACRE. THE STANDARD DEVIATION ACROSS THREE RUNS IS NOTED IN PARENTHESES

Notes	Regression Methods	Wheat CA	Corn CA	Wheat KS	Corn KS
<i>Produces instance-level labels. Fusion sources.</i>	Linear Regress	8.32(0.00)	3.17(0.00)	2.08(0.00)	16.28(0.00)
	RVM	3.20(0.00)	0.22(0.00)	2.15(0.21)	13.48(0.75)
	SVR	8.60(0.14)	9.38(0.01)	3.35(0.37)	20.15(1.39)
<i>Proposed regression method.</i>	MICI Regression Fusion	1.57(0.27)	1.99(0.00)	1.98(0.00)	15.02(0.00)
<i>Fusion methods in comparison.</i>	Another layer of RVM/SVR	3.30(2.50)	4.14(1.25)	3.35(1.02)	26.73(10.54)
	Taking the max	11.92(0.67)	12.72(0.61)	2.85(0.16)	34.39(10.83)
	Taking the min	4.02(0.18)	4.23(0.06)	9.04(1.84)	20.77(7.12)
	Taking the mean	7.08(0.09)	4.81(0.12)	3.40(0.19)	20.71(1.48)
<i>Direct bag-level label prediction.</i>	Cluster MIR	10.17(0.00)	11.19(0.00)	8.09(0.00)	31.74(0.00)
	Aggregate MIR	10.05(0.00)	11.19(0.00)	6.87(0.00)	29.97(0.00)
	RFC-MIR	16.57(0.00)	15.63(0.00)	6.55(0.00)	33.20(0.00)

(which uses CI fusion but does not work with uncertain labels) and the mi-SVM method (which works under MIL but does not use CI for fusion) are both outperformed by the proposed MICI methods in general (which uses both MIL and CI techniques). Both proposed MICI models, the min-max and generalized-mean models, yield satisfactory results on target detection. The generalized-mean model has slightly higher overall AUC and higher detection rate compared with the min-max model in the lower FAR range. This is due to the fact that the min-max model only considers the minimum and maximum instance in the bag, while the generalized-mean model has a more relaxed constraint. In the case where there are more than one target pixels in a positive bag (which can be the case in the MUUFL Gulfport data set), the min-max model will likely only encourages one instance to be positive, while the generalized-mean model can assess more instances.

D. Crop Yield Data Set Experiments

This section presents results on fusion with real-valued prediction values (regression) using the proposed MICIR algorithm. Experiments were conducted on a real crop yield prediction application on a MODIS remote sensing data set provided from the HARVIST project [37]–[39].

The crop yield data set contains MODIS data observations of corn and wheat yield in the states of California (CA) and

TABLE VI

NUMBER OF COUNTIES (BAGS) WITH BOTH CORN AND WHEAT YIELD IN THE CROP YIELD DATA SET [39]

Year	Training				Testing
	2001	2002	2003	2004	
CA	17	16	18	15	13
KS	100	102	100	102	98

Kansas (KS) over 5 years (2001–2005). There are 100 randomly selected pixels included for each county. The surface reflectance values were reported for each pixel containing 92 values: observations in red for 46 time points (every 8 days across the year) and observations in infrared for 46 time points (every 8 days across the year). The zeros and “–32767” reflectance values were indicated as “bad values” in the original data set and are removed for this experiment. Only the counties that reported both corn and wheat yield values are considered and Table VI shows the number of counties that are considered in the states of CA and KS across the years.

This data set suits the multiple-instance framework well as each county can be naturally regarded as a bag with multiple data collections (instances). In this experiment, the corn and wheat yield values for each county are regarded as the

real-valued regression labels. As the CI works with values between zero and one, the corn and wheat yield values were normalized between zero and one by (19)

$$Y_n = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}} \quad (18)$$

where Y_n is the normalized corn or wheat yield value that will be used as the regression training labels. Y_{\min} and Y_{\max} are the min and max yield value in training, respectively.

Linear regression, relevant vector machine (RVM) regression [55], [56], and support vector regression (SVR) [57] were applied to the data. These three regressors operate on all instances and each give a set of instance-based labels (these are essentially instance-MIR methods with different regressors). We used Gaussian kernel for both RVM and SVR methods. Then, the MICIR model is applied to fuse these three regressors and compared to use another layer of RVM and/or SVR (whichever one gives better performance) or simply taking the max, min, or mean of the regression sources as a fusion method. The results are also compared with existing MIR methods such as the Aggregate MIR, Cluster MIR, and RFC-MIR.

The test error was computed by computing the root-mean-squared error (RMSE) between the predicted county-level (bag-level) yield values for the test year and the (known) actual county-level yield values for the test year, as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{b=1}^B (\hat{y}_b - y_b)^2}{B}} \quad (19)$$

where B is the number of (test) bags, \hat{y}_b is the predicted county-level yield values, and y_b is the (known) actual county-level yield values for the test year.

Table V presents the RMSE prediction error results for corn and wheat yields for the states of CA and KS, using crop yield training data from years 2001–2004 and testing on year 2005. From the table, we can, first of all, see that all methods are able to predict the yields comparable to results from previous literature such as [37]. The proposed MICIR method produces the lowest or the second lowest error across both states and both crop types. Naturally, the performance of MICI will depend on the performance of input sources (in this case, three instance MIR approaches linear regression, RVM, and SVR). The performance of MICIR, from Table V, has surpassed each of its sources as well as other fusion such as taking the max and min. RVM performs well, especially for corn, but RVM requires the computation of the kernel matrix, which can be hard when the input data set has higher dimensions. RVM is also highly dependent on the choice of kernel and the parameters such as the kernel width.

VI. DISCUSSION ON RUNNING TIME

Experiments were conducted to compare the computing time of the newly proposed optimization scheme based on the usage of measure elements in the training data and the previously proposed optimization schemes based on valid intervals. The experiment used here is the MUUFL Gulfport target detection experiment same as described in Section V-C.

TABLE VII
RUNNING TIME (IN SECONDS) AND NUMBER OF ITERATIONS UNTIL CONVERGENCE FOR OPTIMIZATION SCHEMES COMPARISON.
THE STANDARD DEVIATION ACROSS FIVE RUNS
IS NOTED IN PARENTHESES

	ME		VI	
	NumIteration	Run Time	NumIteration	Run Time
Run 1	213	79.7s	3488	491.1s
Run 2	61	9.2s	3536	490.4s
Run 3	256	37.0s	2012	280.4s
Run 4	55	49.3s	630	88.5s
Run 5	335	47.5s	3231	448.0s
Summary		44.5(25.4)s		359.7(174.5)s

The measures were initialized randomly and then updated using the two different sampling techniques in optimization. Experiments were conducted in MATLAB using a desktop PC with Intel Xeon CPU 2.40-GHz processor and 16-GB RAM. The code was trivially parallelized and the running times are provided for relative comparisons only.

Table VII lists the running times and the number of iterations until convergence for five runs of the two different optimization. The algorithm was considered reaching convergence if the change of fitness is below 10^{-4} . The “ME” in the table refers to the newly proposed sampling method that samples new measure element according to the counts of measure element used. The “VI” refers to the previously proposed method that samples new measure element by sorting the valid intervals of the measure elements [27]. It can be observed from Table VII that the “sampling according to measure element” approach is, in general, faster in running time and converges in less iterations than the “sort-by-valid-interval” approach, mainly because the valid interval approach has to go through and evaluate the valid interval for all measure elements in each iteration, while the newly proposed “measure element” approach simply counts for the times a measure element was used in training once before the optimization starts. In addition, the fitness of the training data depends on the measure elements used, and the measure element that was most frequently used in training is updated most frequently in the “measure element” approach, which encourages the optimization to converge faster. The ROC curve performance is visibly very similar.

VII. CONCLUSION

This paper proposed two novel MICI models for classifier fusion and a novel MICIR algorithm for remote sensing applications with uncertain and imprecise training labels. The newly proposed models successfully eliminated the variance parameter of the previously proposed noisy-OR fusion model and extends for regression applications where the predictions are real valued. In addition, the newly proposed optimization scheme was able to significantly improve the computation time. Experimental results show the competitive performance of the proposed algorithms in target detection and crop yield prediction applications given real remote sensing data.

The current optimization scheme is based on an evolutionary algorithm, which is effective but can be quite slow when the input image size increases. Alternative optimization schemes

can be explored in the future work, such as sampling measure elements more strategically using fitness values or develop a fast approximation approach using a message passing algorithm. The current regression model follows the standard “primary instance” assumption [40], which assumes that there is one primary instance in each bag that contributes to the regression label. However, this assumption may not necessarily be true in all data sets and it would be interesting to further relax the primary instance assumption. Currently, the normalized monotonic fuzzy measure is used in the proposed algorithms, yet it would be interesting to explore the binary fuzzy measure and/or bicapacity CIs [58] as well as alternative integral approaches such as the Sugeno integral. Alternative features and remote sensing data sets as fusion sources can also be investigated.

REFERENCES

- [1] J. K. Hackett and M. Shah, “Multi-sensor fusion: A perspective,” in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, May 1990, pp. 1324–1330.
- [2] J. Zhang, “Multi-source remote sensing data fusion: Status and trends,” *Int. J. Image Data Fusion*, vol. 1, no. 1, pp. 5–24, Nov. 2010.
- [3] P. Gader, A. Mendez-Vazquez, K. Chamberlin, J. Bolton, and A. Zare, “Multi-sensor and algorithm fusion with the Choquet integral: Applications to landmine detection,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 3, Sep. 2004, pp. 1605–1608.
- [4] B. Waske and J. A. Benediktsson, “Fusion of support vector machines for classification of multisensor data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3858–3866, Dec. 2007.
- [5] P. Du, W. Zhang, S. Zhang, and J. Xia, “Hyperspectral remote sensing image classification based on decision level fusion,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 4, Jul. 2009, pp. IV-940–IV-943.
- [6] H. Yang, Q. Du, and B. Ma, “Decision fusion on supervised and unsupervised classifiers for hyperspectral imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 875–879, Oct. 2010.
- [7] H. Frigui, L. Zhang, and P. D. Gader, “Context-dependent multisensor fusion and its application to land mine detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2528–2543, Jun. 2010.
- [8] P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell, “MUUFL Gulfport hyperspectral and LiDAR airborne data set,” Univ. Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570, Oct. 2013.
- [9] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, Jan. 1997.
- [10] X. Liu *et al.*, “Deep multiple instance learning-based spatial–spectral classification for PAN and MS imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 461–473, Jan. 2018.
- [11] S. E. Yuksel, J. Bolton, and P. Gader, “Multiple-instance hidden Markov models with applications to landmine detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6766–6775, Dec. 2015.
- [12] A. Manandhar, P. A. Torrione, L. M. Collins, and K. D. Morton, “Multiple-instance hidden Markov model for GPR-based landmine detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1737–1745, Apr. 2015.
- [13] X. Du, A. Zare, and J. T. Cobb, “Possibilistic context identification for SAS imagery,” *Proc. SPIE*, vol. 9454, p. 94541I, May 2015.
- [14] J. T. Cobb, X. Du, A. Zare, and M. Emigh, “Multiple-instance learning-based sonar image classification,” *Proc. SPIE*, vol. 10182, p. 101820H, May 2017.
- [15] C. Jiao and A. Zare, “Functions of multiple instances for learning target signatures,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4670–4686, Aug. 2015.
- [16] A. Zare, C. Jiao, and T. Glenn, “Discriminative multiple instance hyperspectral target characterization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2342–2354, Oct. 2018.
- [17] J. L. Marichal, “An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria,” *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 6, pp. 800–807, Dec. 2000.
- [18] C. Labreuche and M. Grabisch, “The Choquet integral for the aggregation of interval scales in multicriteria decision making,” *Fuzzy Sets Syst.*, vol. 137, no. 1, pp. 11–26, 2003.
- [19] A. Mendez-Vazquez, P. Gader, J. M. Keller, and K. Chamberlin, “Minimum classification error training for Choquet integrals with applications to landmine detection,” *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 225–238, Feb. 2008.
- [20] J. Fodor, J.-L. Marichal, and M. Roubens, “Characterization of the ordered weighted averaging operators,” *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 2, pp. 236–240, May 1995.
- [21] O. Maron, “Learning from ambiguity,” Dept. Elect. Eng. Comput., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 1639, 1998.
- [22] M. Sugeno, “Theory of fuzzy integrals and its applications,” Ph.D. dissertation, Tokyo Inst. Technol., Tokyo, Japan, 1974.
- [23] G. E. Martínez, O. Mendoza, P. Melin, and F. Gaxiola, “Comparison between Choquet and Sugeno integrals as aggregation operators for modular neural networks,” in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, Jul. 2016, pp. 2331–2336.
- [24] P. D. Gader, W.-H. Lee, and A. Mendez-Vasquez, “Continuous Choquet integrals with respect to random sets with applications to landmine detection,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 1, Jul. 2004, pp. 523–528.
- [25] G. J. Scott, K. C. Hagan, R. A. Marcum, J. A. Hurt, D. T. Anderson, and C. H. Davis, “Enhanced fusion of deep neural networks for classification of benchmark high-resolution image data sets,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1451–1455, Sep. 2018.
- [26] Y. Q. Zhao, P. Gong, and Q. Pan, “Object detection by spectropolarimetric imagery fusion,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 10, pp. 3337–3345, Oct. 2008.
- [27] X. Du, A. Zare, J. M. Keller, and D. T. Anderson, “Multiple instance Choquet integral for classifier fusion,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1054–1061.
- [28] L. Xu, D. A. Clausi, F. Li, and A. Wong, “Weakly supervised classification of remotely sensed imagery using label constraint and edge penalty,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1424–1436, Mar. 2017.
- [29] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, “Multiple-instance learning for medical image and video analysis,” *IEEE Rev. Biomed. Eng.*, vol. 10, pp. 213–234, 2017.
- [30] L. Cao *et al.*, “Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning,” *Pattern Recognit.*, vol. 64, pp. 417–424, Apr. 2017.
- [31] Q. Zhang and S. A. Goldman, “EM-DD: An improved multiple-instance learning technique,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2001, pp. 1073–1080.
- [32] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
- [33] O. Maron and T. Lozano-Pérez, “A framework for multiple-instance learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 10, 1998, pp. 570–576.
- [34] S. Andrews, I. Tsachanidis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 577–584.
- [35] Y. El-Manzalawy, D. Dobbs, and V. Honavar, “Predicting MHC-II binding affinity using multiple instance regression,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 4, pp. 1067–1079, Jul. 2011.
- [36] Z. Wang, V. Radosavljevic, B. Han, Z. Obradovic, and S. Vucetic, “Aerosol optical depth prediction from satellite observations by multiple instance regression,” in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 165–176.
- [37] Z. Wang, L. Lan, and S. Vucetic, “Mixture model for multiple instance regression and applications in remote sensing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 6, pp. 2226–2237, Jun. 2012.
- [38] K. L. Wagstaff and T. Lane, “Salience assignment for multiple-instance regression,” in *Proc. Workshop Constrained Optim. Struct. Output Spaces (ICML)*, Jun. 2007, pp. 1–6. [Online]. Available: <https://trs.jpl.nasa.gov/handle/2014/41349>
- [39] T. Lane, K. L. Wagstaff, and A. Roper, “Multiple-instance regression with structured data,” in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2008, pp. 291–300.
- [40] S. Ray and D. Page, “Multiple instance regression,” in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2001, pp. 425–432.
- [41] M. Trabelsi, “Robust fuzzy clustering for multiple instance regression,” M. S. thesis, Univ. Louisville, Louisville, KY, USA, 2018.

- [42] J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation* (IEEE Press Series on Computational Intelligence), 1st ed. Hoboken, NJ, USA: Wiley, 2016.
- [43] D. T. Anderson, J. M. Keller, and T. C. Havens, "Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral," in *Computational Intelligence for Knowledge-Based Systems Design* (Lecture Notes in Computer Science), vol. 6178. Berlin, Germany: Springer, 2010, pp. 502–511.
- [44] D. T. Anderson, S. R. Price, and T. C. Havens, "Regularization-based learning of the Choquet integral," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, Jul. 2014, pp. 2519–2526.
- [45] G. Choquet, "Theory of capacities," in *Proc. Annales de l'institut Fourier*, vol. 5, 1954, pp. 131–295.
- [46] M. Fitting and E. Orlowska, Eds., *Beyond Two: Theory and Applications of Multiple-Valued Logic*. Berlin, Germany: Springer-Verlag, 2003.
- [47] A. Mendez-Vazquez, "Information fusion and sparsity promotion using Choquet integrals," Ph.D. dissertation, Univ. Florida, Gainesville, FL, USA, 2008.
- [48] P. S. Bullen, *Handbook of Means and Their Inequalities*. Norwell, MA, USA: Kluwer, 2003, ch. 3, p. 175.
- [49] S. Rolewicz, *Functional Analysis and Control Theory: Linear Systems*, vol. 29. Dordrecht, The Netherlands: Springer, 2013.
- [50] X. Du, "Multiple instance Choquet integral for multiresolution sensor fusion," Ph.D. dissertation, Univ. Missouri, Columbia, MO, USA, Dec. 2017.
- [51] L. L. Scharf and L. T. McWhorter, "Adaptive matched subspace detectors and adaptive coherence estimators," in *Proc. 30th Asilomar Conf. Signals Syst.*, Nov. 1996, pp. 1114–1117.
- [52] S. Kraut, L. L. Scharf, and R. W. Butler, "The adaptive coherence estimator: A uniformly most-powerful-invariant adaptive detection statistic," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 427–438, Feb. 2005.
- [53] N. B. Pulsone and M. A. Zatman, "A computationally efficient two-step implementation of the GLRT," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 609–616, Mar. 2000.
- [54] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *Eur. J. Oper. Res.*, vol. 89, no. 3, pp. 445–456, Mar. 1996.
- [55] M. E. Tipping, "The relevance vector machine," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 652–658.
- [56] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001.
- [57] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 1997, pp. 155–161.
- [58] M. Grabisch and C. Labreuche, "Bi-capacities—I: Definition, Möbius transform and interaction," *Fuzzy Sets Syst.*, vol. 151, no. 2, pp. 211–236, 2005.



Xiaoxiao Du (S'13) received the B.Eng. degree (Hons.), with Chu Kochen Honors Certificate, from Zhejiang University, Hangzhou, China, in 2011, and the M.S. degree and Ph.D. degrees in electrical and computer engineering from the University of Missouri, Columbia, MO, USA, in 2013 and 2017, respectively.

She is currently a Research Assistant with the Department of Electrical and Computer Engineering, University of Missouri. Her research interests include machine learning, hyperspectral image analysis, remote sensing, pattern recognition, and signal and image processing.



Alina Zare (S'07–M'08–SM'14) received the Ph.D. degree in computer and information science and engineering from the University of Florida, Gainesville, FL, USA, in 2008.

She is currently an Associate Professor with the Department of Electrical and Computer Engineering and the Director of the Machine Learning and Sensing Laboratory, University of Florida. Her research interests include analysis of a variety of remote sensing data and the development of machine learning and pattern recognition algorithms for automated analysis of remote sensing data sets.

Dr. Zare is currently an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.