# 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 ambiguosly and imprecisely labeled training data in which 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*—multiple instance learning, remote sensing, target detection, classifier fusion, Choquet integral, multiple instance regression.**

## I. Introduction

Classifier 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 classifiers (or regressors) 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 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 (HSI) imagery 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 the obtained HSI imagery is $1m$. This is a challenging target detection problem in which many targets are occluded (hidden under a tree, for example) and many are sub-pixel (target size less than $1m^2$). 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 the GPS information but we cannot pin-point the exact target locations.

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. Email: azare@ece.ufl.edu.

This material is based upon work supported by the National Science Foundation under Grant IIS-1723891-CAREER: Supervised Learning for Incomplete and Uncertain Data.

Figure 1 shows an illustration of targets in this scene. Figure 1a shows the RGB image over a brown target (the brown colored pixels in the scene). The red cross marks the target coordinates collected by the (inaccurate) GPS device. As can be seen, the GPS location is visibly different than where the brown target actually is in the scene. Figure 1b shows a sub-pixel target in the scene hidden under a tree. As can be seen, the pixels nearby are all in green color (color of the leaves of the tree) and the target is not visible in the scene. It would be impossible for humans to visibly locate and manually label where the sub-pixel target is in the scene. For both cases, it is difficult, or impossible, to obtain the accurate target locations in the scene.

**(a)** **(b)** **(c)** **(d)**

**Fig. 1:** Illustration of a target application given remotely sensed data. (a) An inaccurate GPS location (marked by a red cross) that differs from the true brown target location in the scene. (b) A sub-pixel target that cannot be visibly seen from the imagery. (c)(d) Red rectangles mark the approximate region that contains the targets.

However, in this example, it is possible to identify a region or a set of pixels that contain the targets. Figure 1c and Figure 1d demonstrate an approximate region that contains the target in each scene (the red rectangles). We know the target is somewhere in the red rectangle region, but we do not know exactly which pixel(s) corresponds to the target. In this example, the accurate target location per pixel is not available, but the regions or sets of pixels that contain the targets can easily be obtained from the (uncertain) 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.

This paper proposes a Multiple Instance Choquet Integral (MICI) framework for both multi-sensor classifier fusion and regression that can learn from ambiguously and imprecisely labeled training data. 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, among many other classification tasks. This paper also proposes a Multiple Instance Choquet Integral 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.

## II. Related Work

This section introduces related work in multiple instance classification, classifier fusion, and multiple instance regression. 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

The Multiple Instance Learning (MIL) framework was first proposed in [9] to address uncertainty and inaccuracy in labeled data in supervised learning applications. 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. Figure 2 shows an illustration of MIL bags.

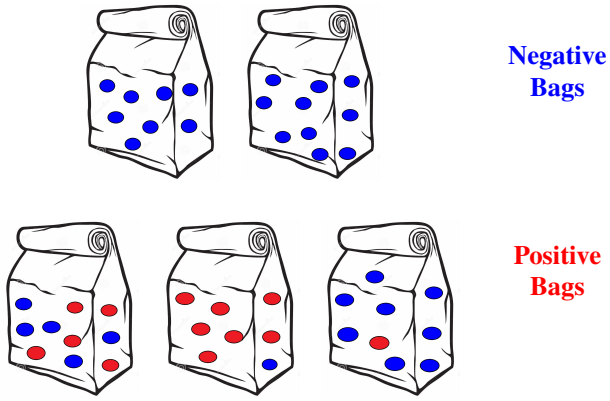


**Fig. 2:** Illustration of bags in multiple instance learning. Red color marks positive instances and blue color marks negative instances. The two bags on the top row are negative bags and the three bags on the bottom row are positive bags.

The EM-DD and mi-SVM algorithms are two widely cited multiple instance learning techniques for classification [10–13]. The two algorithms are used later in this paper as comparison MIL approaches. The EM-DD algorithm [14] combines the Expectation-Maximization (EM) [15] with the Diverse Density (DD) objective function [9, 16, 17] 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 [14]. 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 criteria is met.

The mi-SVM algorithm was proposed by Andrews et al. [18] 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 [18]:

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

such that

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

where $\mathbf{w}$ is the weights, $b$ is the bias, $y_i$ is the instance-level label, $\xi_i$ 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

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 (MHC-II) molecules [19], predicting aerosol optical depth from remote sensing data [20, 21], and predicting crop yield [21–23].

Ray and Page [24] first proposed an 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 et al. then investigated using MIR in predicting crop yield from real remote sensing data set [22, 23]. They provided an MI-ClusterRegress algorithm (or in some references, Cluster MIR algorithm) that mapped instances onto (hidden) cluster labels [23]. 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 cluster must be "relevant" to the bag-level labels. The algorithm assumes that the bag data follows a mixture of Gaussian distributions [23]. More recently, Trabelsi and Frigui [25] 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 Choquet integral (CI) has a long history of providing an effective framework for non-linear information fusion [26–30]. The CI is an aggregation operator based on the fuzzy measures [31]. 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 [32, 33]. The algorithms proposed in this paper rely on the Choquet integral and fuzzy measures and this subsection provides the definition of the Choquet integral and fuzzy measures.

Consider the case that there are $m$ sources, $C = \{c_1, c_2, \ldots, c_m\}$, for fusion. The "sources" refer to the outputs from the set of $m$ classifiers or regressors to be fused. The set $C$ contains $2^m - 1$ non-empty 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 [34–37]:

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 discrete Choquet integral is used 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 Choquet integral on instance $\mathbf{x}_n$ given $C$ is then computed as [31, 37, 38]:

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

where $C$ is sorted so that $h(c_1; \mathbf{x}_n) \geq h(c_2; \mathbf{x}_n) \geq \cdots \geq h(c_m; \mathbf{x}_n)$. 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 $A_k = \{c_1, \ldots, c_k\}$ is $g(A_k)$.

### *D. Previously Proposed MICI Noisy-or Model for Multiple Instance Classifier Fusion*

In this paper, we are proposing several Multiple Instance fusion algorithms that leverage the Choquet Integral. Previously, we proposed also a Multiple Instance Choquet Integral classifier fusion method that uses the Choquet integral and multiple instance learning framework with a noisy-or objective function [38]. 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. The noisy-or model was be used to express these assumptions [17, 38]:

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

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. The $\mu$ and $\sigma^2$ 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 (non-target) is marked label "0". The parameter $\mu$ can be set, in this case, to 1, to encourage the Choquet integral values of positive instances to be 1 and the Choquet integral values of negative instances to be far from 1. Here, the model parameter vector $\boldsymbol{\theta}$ consists of the variance of the Gaussian $\sigma^2$ and the fuzzy measure $\mathbf{g}$ values used to compute the Choquet integral.

However, this previously proposed noisy-or model was subject to user-defined parameter setting such as the variance $\sigma^2$ [38]. The variance parameter controls how sharply the Choquet integral 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. It would be interesting to develop an algorithm that can eliminate the variance parameter yet still able to perform effective classifier fusion and handle uncertain labels.

### *E. Summary of Related Work & Motivation for the Proposed Algorithms*

As discussed in Introduction, previous standard supervised fusion algorithms often require accurate and precise training labels. However, such accurate and precise training labels may be difficult, expensive, or infeasible to obtain in many applications. Therefore, a novel trained fusion method that can learn from ambiguously and imprecisely labeled training data is needed for both classification and regression applications.

In our previous work, the Multiple Instance Choquet Integral noisy-or model is able to perform classifier fusion while taking label uncertainty into consideration. An evolutionary algorithm based on the valid interval of each measure element was used to optimize the model. However, the previous noisy-or model only addresses two-class classification problems. It also is a slow algorithmic approach as takes a long time to update the fuzzy measures, since the algorithm needs to compute the valid interval for each fuzzy measure element at each iteration during optimization. It would, therefore, be useful to extend the previous work to both classifier fusion and regression, with improved efficiency.

This paper will propose two novel Multiple Instance Choquet Integral models for classifier fusion. This paper will also propose a novel Multiple Instance Choquet Integral Regression algorithm that works with regression applications where the predictions are real-valued, in addition to classification problems. All the newly proposed algorithms work under the multiple instance assumption and can handle uncertain and imprecise training labels. In addition, in order to learn the fuzzy measure to be used with the Choquet integral, a new optimization scheme is proposed in this paper to shorten the running time as compared to our previous approaches for MIL fusion.

## III. The Multiple Instance Choquet Integral Algorithms for Classifier Fusion

This section provides a detailed description on the newly proposed Multiple Instance Choquet Integral min-max and generalized-mean models. The proposed models can perform multi-sensor classifier fusion while handling uncertainties in training labels. A monotonic normalized fuzzy measure is learned to be used with the Choquet integral to perform two-classs classifier fusion given bag-level training labels. An optimization scheme based on an evolutionary algorithm is used to optimize the models proposed.

### A. 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:

$$J_M^- = \sum_{a=1}^{B^-} \max_{\forall \mathbf{x}_{ai}^- \in \mathscr{B}_a^-} \left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-) - 0\right)^2; \tag{5}$$

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:

$$J_M^+ = \sum_{b=1}^{B^+} \min_{\forall \mathbf{x}_{bj}^+ \in \mathscr{B}_b^+} \left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+) - 1\right)^2, \tag{6}$$

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. $C_{\mathbf{g}}$ is the Choquet integral output given measure $\mathbf{g}$ computed using (3), $\mathscr{B}_a^-$ is the $a^{th}$ negative bag, and $\mathscr{B}_b^+$ is the $b^{th}$ positive bag.

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

$$J_M = J_M^- + J_M^+. \tag{7}$$

By minimizing the objective function in (7), we encourage the Choquet integral of all the instances in the the negative bag to zero ("$J_M^-$" term) and encourage the Choquet integral 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.

### B. Generalized Mean Model

The "min" and "max" operators proposed in the above 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 non-zero real number, and $x_1$,...,$x_n$ are positive real numbers, then the generalized mean with the exponent $p$ of $x_1$,...,$x_n$ is defined as [39]:

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

The generalized mean has the following two properties:

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

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

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:

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

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

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

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 $b$, and $N_a^-$ 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. $C_{\mathbf{g}}$ is the Choquet integral given measure $\mathbf{g}$, $\mathscr{B}_a^-$ is the $a^{th}$ negative bag, and $\mathscr{B}_b^+$ 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:

$$J_G = J_G^- + J_G^+. \tag{13}$$

By minimizing the objective function in (13), we encourage the Choquet integral of all the points in the the negative bag to zero ("$J_G^-$" term) and encourage the Choquet integral 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 $p_1 \to \infty$ and $p_2 \to -\infty$, according to properties (9) and (10), the generalized mean model is equivalent to the min-max model in (7) and follows strictly the standard MIL assumptions.

## IV. The Multiple Instance Choquet Integral Regression Algorithm

A new Multiple Instance Choquet Integral Regression (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 [24]. 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 Choquet integral 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 $\mathbf{g}$:

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

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 $b$, $C_{\mathbf{g}}(\mathbf{x}_{bi})$ is the choquet integral output for the $i^{th}$ instance in bag $b$ given measure $\mathbf{g}$. The Choquet integral output is computed based on Equation (3). The objective function encourages the Choquet integral 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 re-constructed. The MIL assumes that a bag is labeled negative if all of the instances in the bag is 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".

## V. Optimization

The proposed MICI min-max and generalized-mean models and the proposed MICIR algorithm learn a fuzzy measure to be used within a Choquet integral for two-class classifier fusion and regression. The computed Choquet integral 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 [38]. The term "valid interval" was define 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 [38], 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 Choquet integral). 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 inital population, $I$ is the maximum number of iterations, $P$ is the measure population size, $\mathscr{G}$ refers to all measures in the current measure population, $\mathscr{G}\{p\}$ is the $p^{th}$ measure in measure population $\mathscr{G}$, $\eta$ 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 criteria 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

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

**TESTING**

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

$TestLabels \leftarrow$ 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 Choquet integral measures is generated and each measure in the population is intialized 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 towards 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" approaches samples measure element values from the bottom of the lattice up. First the singletons were sampled between 0 and 1 randomly. Then, the duples 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, 0.1 values, respectively, for a data point. To fuse these results using the CI, according to the definition of Choquet integral, $g_1$, $g_{12}$, 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 $\mathbf{v} = \{v_1, v_2, ...v_{2^m-1}\}$ 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 are 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

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

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 Choquet integral 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 truncated Gaussian sampling method can be seen in [40].

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 Equation (7) or (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 Equation (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 Equation (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 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.

## VI. Experimental Results

This section describes experiments and presents 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 HARVIST crop yield data set was used for real target detection and crop yield prediction applications.

### A. Synthetic Classification Data Set

A synthetic 5-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 the negative bags were "contaminated" in various degrees. Relative error [23] is used to evaluate the performance for the MICI classification models:

$$Error_{reg}(y, \hat{y}) = |y - \hat{y}|, \tag{16}$$

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

The relative error results 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 increases as the percentage of contamination increases, which is as expected. When the percentage of contamination goes towards 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. When the percentage of contamination increases towards 100%, the relative error also increases as the difficulty grows.

**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) |

### B. Synthetic Regression Data Set

A synthetic 5-source regression data set were constructed to investigate the performance of the proposed MICIR regression 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 5 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 [23] is used to evaluate the performance for the MICIR algorithm:

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

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 [24]. 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 proposed MICIR algorithm. The percentage of primary instances in the bag takes the value of 0% to 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 non-primary 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 signal-to-noise ratio (SNR). In this experiment, 100% of the points in the bag are primary instances and varying amount of noise is add to the entire data set, creating SNR value from 50dB to 0dB with an increment of -5dB. 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 1m.[1] The image from campus 1 is $325 \times 337$ pixels in size. The image from campus 3 is $329 \times 345$ pixels in size. The image from campus 4 is $333 \times 345$ pixels in size. All HSI data cubes contain 72 bands corresponding to wavelengths 367.7nm to 1043.4nm and were collected using the CASI hyperspectral camera [8, 41]. In this experiment, the first four and last four noisy bands were removed.

A total of sixty cloth panel targets were placed in the scene. The targets were cloth panels of five different colors: fifteen brown, fifteen dark green, twelve faux vineyard green (FVG),

[1]The 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".

**TABLE II:** Relative error versus percentage of primary instances for synthetic regression data set for MICI Regression 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 MICI Regression 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) | |

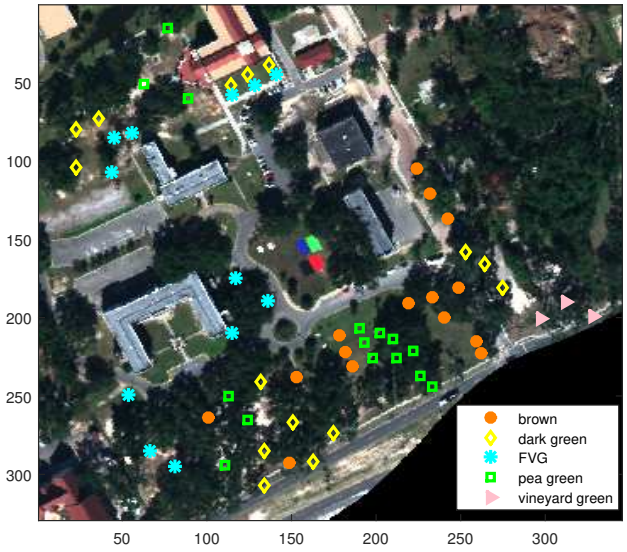


**Fig. 3:** The RGB image from MUUFL Gulfport "campus 3" data set. Orange circle marks the true brown target locations, yellow diamond marks the true dark green target locations, cyan asterisk marks the true FVG target locations, green square marks the true pea green target locations, and pink triangle marks the true vineyard green target locations.

fifteen pea green, and three 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 sub-pixel targets (at $0.25m^2$ corresponding to a quarter of a pixel in area) up to super-pixel targets (at $9m^2$) with varying levels of occlusion. For each target, a GPS ground truth location was collected using a Trimble Juno SB hand-held device. Figure 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 5m. Therefore, the groundtruth locations for each target are only accurate within a $5 \times 5$ 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 groundtruth labels.

The adaptive coherence estimator (ACE) detector [42–44] was applied to the imagery using spectral signatures of four of the target types (as spectral signatures for these targets were available from previous studies[2]). The background mean and background covariance for the ACE detector was estimated 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, $\mu_{tr}$ and $cov_{tr}$, were computed. Then, the ACE detection map for the training imagery, $ACE_{tr}$, was obtained using $\mu_{tr}$ 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 $5 \times 5$ window was put around each groundtruth 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 groundtruth 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 $\mu_{tr}$ and $cov_{tr}$. Then, the Choquet integral 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

[2]The 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.

**TABLE IV:** The AUC results at on un-normalized MUUFL Gulfport data across five runs. The standard deviation is noted in parentheses.

| *Notes* | Methods | **Train1Test3** | **Train1Test4** | **Train3Test1** | **Train3Test4** | **Train4Test1** | **Train4Test3** |
|---|---|---|---|---|---|---|---|
| *Sources: individual target types.* | 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 |
| *Instance-based fusion methods in comparison.* | 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 | 0.345 | 0.329 | 0.459 | 0.339 | **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 |
| *MIL comparison methods.* | 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)** | 0.331(0.001) | 0.461(0.000) | **0.340(0.000)** | **0.349(0.000)** | 0.329(0.000) |
| *Proposed MICI models.* | MICI Min-Max | 0.345(0.001) | 0.331(0.008) | 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)** |

with a standard Support Vector Machine (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 [26], mi-SVM [18], EM-DD [14], and the previously proposed MICI noisy-or methods. The CI-QP approach learns a fuzzy measure for Choquet integral 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 multiple instance learning 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 curve (AUC) results with FAR up to $1 \times 10^{-3}/m^2$ (corresponding to a reasonable scale of 1 false alarm in $1000m^2$).

Table IV shows the AUC results at FAR up to $1 \times 10^{-3}/m^2$ for all fusion methods with un-normalized 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.

Figure 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. Figure 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 False Alarm Rate (FAR) between $[0, 0.001]$ and the Y-axis represents Positive Detection (PD). As can be seen from the ROC curve plot, the proposed MICI min-max and generalized-mean model yield favorable detection results especially in the lower FAR range as compared to comparing MIL methods and individual sources.

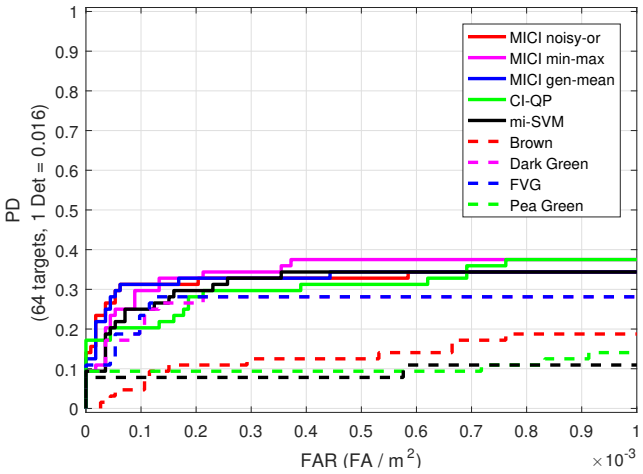


**Fig. 4:** The ROC curve for one run of train on campuses 1 and 4, test on campus 3. The red, magenta and blue solid lines mark the ROC curve of MICI noisy-or, MICI min-max, and MICI generalized mean model. The green solid line marks the CI-QP method and the black solid line marks the mi-SVM method results. The EM-DD method yield very low AUC and is not included in the plot. The dashed red, magenta, blue, and green lines mark the original ACE detector results for Brown, Dark Green, FVG, and Pea Green target types.

### *D. Crop Yield Data Set Experiments*

This section presents results on fusion with real-valued prediction values (regression) using the proposed Multiple Instance Choquet Integral Regression (MICIR) algorithm. Experiments were conducted on a real crop yield prediction application on a MODIS remote sensing data set provided from the HARVIST (Heterogeneous Agricultural Research Via Interactive, Scalable Technology) project [21–23].

The crop yield data set contains MODIS data observations of corn and wheat yield in the states of California and Kansas 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 timepoints (every 8 days across the year) and observations in infrared (IR) for 46 timepoints (every 8 days

**TABLE V:** RMSE error for CA and KS corn and wheat yield, Training on Years 2001-2004, Test on Year 2005. 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** |
|---|---|---|---|---|---|
| *Produces instance-level labels. Fusion sources.* | 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) |
| *Proposed regression method.* | MICI Regression Fusion | **1.57(0.27)** | 1.99(0.00) | **1.98(0.00** | 15.02(0.00 |
| *Fusion methods in comparison.* | 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) |
| *Direct bag-level label prediction.* | 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) |

**TABLE VI:** Number of counties (bags) with both corn and wheat yield in the crop yield data set [23].

| | **Training** | | | | **Testing** |
|---|---|---|---|---|---|
| **Year** | **2001** | **2002** | **2003** | **2004** | **2005** |
| **CA** | 17 | 16 | 18 | 15 | 13 |
| **KS** | 100 | 102 | 100 | 102 | 98 |

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 California and Kansas 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 Choquet integral works with values between zero and one, the corn and wheat yield values were normalized between zero and one by Equation (19):

$$Y_n = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}, \tag{18}$$

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 [45, 46], and Support Vector Regression (SVR) [47] 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 MICI Regression 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 multiple instance regression methods such as the Aggregate MIR, Cluster MIR, and Robust Fuzzy Clustering MIR (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:

$$RMSE = \sqrt{\frac{\sum_{b=1}^{B} \left(\hat{y_b} - y_b\right)^2}{B}}, \tag{19}$$

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 yield for the states of California (CA) and Kansas (KS), using crop yield training data from years 2001-2004 and testing on year 2005. The standard deviation is across three runs.

From the table, we can first of all see that all methods are able to predict the yields compariable to results from previous literature such as [21]. The proposed MICI Regression 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 MICI Regression, 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.

## VII. 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 VI-C. 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 "measure element" in the table refers to the newly proposed sampling according to the counts of measure element used. The "valid interval" refers to sample by sorting the valid intervals of the measure elements as previously proposed in [38]. 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. Besides, 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 are visibly very similar.

**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.

| | **Measure Element (new)** | | **Valid Interval (old)** | |
|---|---|---|---|---|
| | 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 | | $\mathbf{44.5(25.4)}$**s** | | 359.7(174.5)s |

## VIII. Conclusion

This paper proposed a Multiple Instance Choquet Integral (MICI) framework for classifier fusion and regression that can work with uncertain and imprecisely labeled training data. Two classifier fusion models, the min-max model and generalized-mean model, were proposed for classifier fusion and a MICIR regression algorithm was proposed for regression. The implementations of the algorithms are straight-forward and the min-max model and the MICIR algorithm, in particular, are parameter-free in their objective functions. All the proposed models can learn from ambiguously and imprecisely labeled training data while effectively performing classifier fusion and regression. In addition, a new optimization scheme was proposed to improve computation time. Experimental results show competitive performance of the proposed algorithms in real remote sensing applications, with shortened computation time.

Alternative optimization schemes can be explored in future work, such as sampling measure elements using fitness values or using message passing. 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 bi-capacity Choquet integrals [48]. Alternative features and remote sensing data sets as fusion sources can also be investigated.

## References


[1] J. Hackett and M. Shah, "Multi-sensor fusion: a perspective," in *IEEE Int. Conf. Robotics and Automation*, vol. 2, 1990, pp. 1324–1330.

[2] J. Zhang, "Multi-source remote sensing data fusion: status and trends," *Int. J. Image and Data Fusion*, vol. 1, no. 1, pp. 5–24, 2010.

[3] P. Gader, A. Mendez-Vasquez, K. Chamberlin, J. Bolton, and A. Zare, "Multi-sensor and algorithm fusion with the choquet integral: applications to landmine detection," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 3, Sept. 2004, pp. 1605–1608.

[4] B. Waske and J. 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 *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 4, July 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 Trans. Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 875–879, Oct. 2010.

[7] H. Frigui, L. Zhang, and P. Gader, "Context-dependent multisensor fusion and its application to land mine detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2528–2543, June 2010.

[8] P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell, "Muufl gulfport hyperspectral and lidar airborne data set," University of Florida, Gainesville, FL, Tech. Rep. 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, no. 1-2, pp. 31–71, Jan. 1997.

[10] 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, March 2017.

[11] 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.

[12] A. Zare, C. Jiao, and T. Glenn, "Discriminative multiple instance hyperspectral target characterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–12, 2017.

[13] L. Cao, F. Luo, L. Chen, Y. Sheng, H. Wang, C. Wang, and R. Ji, "Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning," *Pattern Recognition*, vol. 64, pp. 417 – 424, 2017.

[14] 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.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algo-

rithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[16] O. Maron, "Learning from ambiguity," AI Technical Report 1639, Massachusetts Institute of Technology, 1998.

[17] O. Maron and T. Lozano-Perez, "A framework for multiple-instance learning," in *Neural Inform. Process. Syst.*, vol. 10, 1998.

[18] S. Andrews, "Support vector machines for mulitple-instance learning," in *Ann. Conf. Neural Inf. Proc. Systems (NIPS)*, 2002.

[19] 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, July 2011.

[20] 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.

[21] 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, June 2012.

[22] K. L. Wagstaff and T. Lane, "Salience assignment for multiple-instance regression," in *Workshop on Constrained Optimization and Structured Output Spaces (ICML)*, June 2007.

[23] K. L. Wagstaff, T. Lane, and A. Roper, "Multiple-instance regression with structured data," in *IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2008, pp. 291–300.

[24] S. Ray and D. Page, "Multiple instance regression," in *Proc. 18th Int. Conf. Mach. Learn.(ICML)*, vol. 1, 2001, pp. 425–432.

[25] M. Trabelsi and H. Frigui, "Robust fuzzy clustering for multiple instance regression," *J. Pattern Recognition*, Submitted.

[26] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *European J. Operational Research*, vol. 89, no. 3, pp. 445–456, Mar. 1996.

[27] ——, "A new algorithm for identifying fuzzy measures and its application to pattern recognition," in *Int. Joint Conf. 4th IEEE Int. Conf. Fuzzy Systems and 2nd Int. Fuzzy Eng. Symp.*, vol. 1, Mar. 1995, pp. 145–150.

[28] C. Labreuche and M. Grabisch, "The choquet integral for the aggregation of interval scales in multicriteria decision making," *Fuzzy Sets and Systems*, vol. 137, no. 1, pp. 11 – 26, 2003.

[29] A. Mendez-Vazquez, P. D. Gader, J. M. Keller, and K. Chamberlin, "Minimum classification error training for choquet integrals with applications to landmine detection," *IEEE Trans. Fuzzy Systems*, vol. 16, no. 1, pp. 225–238, Feb 2008.

[30] A. Mendez-Vazquez and P. Gader, "Learning fuzzy measure parameters by logistic lasso," in *Proc. Annual Conf. North American Fuzzy Information Processing Society (NAFIPS)*, May 2008, pp. 1–7.

[31] J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of computational intelligence: Neural networks, fuzzy systems and evolutionary computation*, 1st ed., ser. IEEE Press Series on Computational Intelligence. John Wiley & Sons, Inc., 2016.

[32] D. 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*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6178, pp. 502–511.

[33] D. T. Anderson, S. R. Price, and T. C. Havens, "Regularization-based learning of the choquet integral," in *IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*, July 2014, pp. 2519–2526.

[34] G. Choquet, "Theory of capacities," in *Annales de l'institut Fourier*, vol. 5, 1954, pp. 131–295.

[35] M. Sugeno, "Theory of fuzzy integrals and its applications," Ph.D. dissertation, Tokyo Institute of Technology, 1974.

[36] M. Fitting and E. Orlowska, Eds., *Beyond Two: Theory and Applications of Multiple-Valued Logic*. Springer, 2003.

[37] A. Mendez-Vazquez, "Information fusion and sparsity promotion using choquet integrals," Ph.D. dissertation, University of Florida, 2008.

[38] X. Du, A. Zare, J. Keller, and D. Anderson, "Multiple instance choquet integral for classifier fusion," in *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, July 2016, pp. 1054–1061.

[39] P. S. Bullen, *Handbook of Means and Their Inequalities*. Kluwer Academic Publishers, 2003, ch. 3 The Power Means, p. 175.

[40] X. Du, "Multiple instance choquet integral for multiresolution sensor fusion," Ph.D. dissertation, University of Missouri, Columbia, MO, Dec. 2017.

[41] 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.

[42] L. L. Scharf and L. T. McWhorter, "Adaptive matched subspace detectors and adaptive coherence estimators," in *Proc. 30th Asilomar Conf. Signals Syst.* IEEE, Nov. 1996, pp. 1114–1117.

[43] S. Kraut, L. L. Scharf, and R. W. Butler, "The adaptive coherence estimator: a uniformly most-powerful-invariant adaptive detection statistic," *IEEE Trans. Signal Proc.*, vol. 53, no. 2, pp. 427–438, 2005.

[44] N. Pulsone and M. A. Zatman, "A computationally efficient two-step implementation of the glrt," *IEEE Trans. Signal Proc.*, vol. 48, no. 3, pp. 609–616, Mar 2000.

[45] M. E. Tipping, "The relevance vector machine," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 652–658.

[46] M. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Research (JMLR)*, vol. 1, pp. 211–244, 2001.

[47] 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.

[48] M. Grabisch and C. Labreuche, "Bi-capacitiesŮi: definition, möbius transform and interaction," *Fuzzy sets and systems*, vol. 151, no. 2, pp. 211–236, 2005.