# Multiple Instance Choquet Integral with Binary Fuzzy Measures for Remote Sensing Classifier Fusion with Imprecise Labels

Xiaoxiao Du*, Alina Zare†, and Derek T. Anderson‡
*Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA
†Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA
‡Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65201 USA
Email: xiaodu@umich.edu, azare@ece.ufl.edu, andersondt@missouri.edu

***Abstract*—Classifier fusion methods integrate complementary information from multiple classifiers or detectors and can aid remote sensing applications such as target detection and hyperspectral image analysis. The Choquet integral (CI), parameterized by fuzzy measures (FMs), has been widely used in the literature as an effective non-linear fusion framework. Standard supervised CI fusion algorithms often require precise ground-truth labels for each training data point, which can be difficult or impossible to obtain for remote sensing data. Previously, we proposed a Multiple Instance Choquet Integral (MICI) classifier fusion approach to address such label uncertainty, yet it can be slow to train due to large search space for FM variables. In this paper, we propose a new efficient learning scheme using binary fuzzy measures (BFMs) with the MICI framework for two-class classifier fusion given ambiguously and imprecisely labeled training data. We present experimental results on both synthetic data and real target detection problems and show that the proposed MICI-BFM algorithm can effectively and efficiently perform classifier fusion given remote sensing data with imprecise labels.**

***Index Terms*—Choquet integral, fuzzy measure, classifier fusion, hyperspectral, target detection**

## I. Introduction

Classifier fusion refers to the process of integrating information from multiple classifier outputs. In applications such as target detection given remote sensing data, confidence maps from multiple detectors or classifiers are usually fused into one decision output with the goal of improved detection or classification accuracy [1, 2].

Previous supervised fusion methods often require precise, pixel-level ground-truth labels for each training data point, which can be challenging to obtain for remote sensing data due to sensor noise, sub-pixel targets, or simply the large quantities of data [3–6]. In our experiments, for example, we collected hyperspectral imagery (HSI) on a flight over the University of Southern Mississippi-Gulfpark campus and we placed a variety of cloth panel targets in the scene [7]. The goal is to perform target detection given hyperspectral data. We measured target coordinates using a hand-held Global Positioning System (GPS) device during data collection, but the GPS device used was only accurate to 2-5 meters. Figure 1 shows an illustration of targets and their visual appearances in the RGB imagery from the HSI data.

Figure 1a shows a photograph of a dark green target on a lawn. As shown in Figure 1b, the measured GPS location of the target (red triangle) is visibly different than where the dark green target actually is in the scene (green circle). Figure 1c shows a sub-pixel brown target in the scene, partially occluded by trees. As shown in Figure 1d, the target is not visible in the HSI data amongst nearby tree canopy. As these two examples illustrate, it is difficult, and sometimes impossible, for human annotators to visually inspect, locate, and label accurate target locations in the HSI imagery to train a standard supervised fusion algorithm for target detection, given such remotely sensed data.

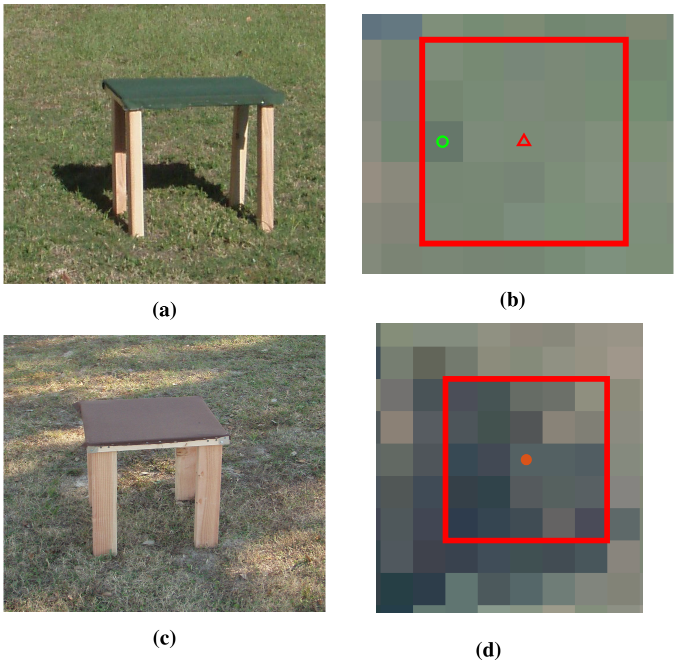
(a) (b) (c) (d)

**Fig. 1:** Illustration of imprecise labels of targets from a remotely sensed hyperspectral imagery. (a) A photo of a dark green target on a lawn. (b) An inaccurate GPS location (marked by a red triangle) that differs from the true dark green target location (green circle) in the scene. (c) A photo of a sub-pixel-size brown target. (d) The brown target is occluded by tree canopy and is invisible from the imagery. The brown dot marks the imprecise GPS location. Red rectangles in (b)(d) mark the approximate regions that contains the targets.

However, in this problem, annotators can easily circle the approximate locations of the targets (such as those marked by red rectangles in Figure 1) given the imprecise GPS information and identify that a certain area possibly contains a target. In this way, the tedious and expensive process of pinpointing exact target locations can be avoided. Instead, one can easily obtain approximate regions or sets of pixels that contain the targets from the imprecise GPS coordinates.

Previously, we proposed a Multiple Instance Choquet Integral (MICI) method for multi-sensor classifier fusion that can learn from such ambiguously and imprecisely labeled training data [6, 8]. The MICI approach is based on the Multiple Instance Learning (MIL) framework [9] and uses the Choquet integral (CI) [10] as the aggregation operator for fusion. The Choquet integral (CI) is a parametric function and can flexibly represent various relationships between fusion sources [11, 12], depending on the fuzzy measure (FM) it uses. In our previous work [6, 8], we showed that the MICI with a monotonic and normalized fuzzy measure can achieve state-of-the-art classifier fusion and detection results, given remote sensing data with imprecise labels.

Although effective, the previous MICI algorithms can be slow to train. The size of the fuzzy measure that needs to be optimized through an MICI algorithm is exponential to the number of classifier outputs ("sources") to be fused. We observed in our previous experiments [8, 12, 13] that the real-valued fuzzy measure learned by an MICI algorithm tends to approximate $\{0,1\}$ instead of any random number within $[0,1]$ for binary classification problems (such as target versus non-target detection). This suggests that a binary fuzzy measure (BFM) may be an efficient substitute to a regular normalized fuzzy measure to be used with MICI for two-class classification fusion. With such motivation, in this work, we propose the use of binary fuzzy measures (BFMs) with MICI to improve the efficiency while maintaining the effectiveness of MICI to solve classifier fusion problems given remote sensing data with imprecise labels.

## II. Preliminaries: Fuzzy Measure, Binary Fuzzy Measure, and Choquet Integral

The Choquet integral (CI) has a long history of providing an effective framework for non-linear information fusion [14–18]. The CI is an aggregation operator based on the fuzzy measures. Depending on the values of each element in the fuzzy measure, the CI can represent a variety of relationships and combinations among the fusion sources. Therefore, a crucial aspect of using the CI for information/sensor fusion is learning the fuzzy measures for the CI [19, 20]. This section provides description and definitions of the fuzzy measure and the (discrete) Choquet integral and reviews previous methods in learning the fuzzy measures, specifically within the CI. Table I provides a comprehensive list of notations for all symbols used in this paper. We proceed to describe the FM and CI as follows.

### A. Fuzzy Measure and Binary Fuzzy Measure

Consider the case that there are $m$ classifiers, $C = \{c_1, c_2, \ldots, c_m\}$, for fusion. The set of classifiers $C$ contains $2^m - 1$ non-empty subsets. The power set of all (crisp) subsets of $C$ is denoted $2^C$.

**TABLE I:** List of Notations

| Symbol | Description |
|---|---|
| $\mathbb{Z}^+$ | Positive integers. |
| $m$ | Total number of classifiers to be fused. |
| $\mathbf{x}_n$ | An instance (a data point). |
| $h(c_k; \mathbf{x}_n)$ | The $k^{th}$ classifier output on the $n^{th}$ instance. In this paper, all classifier outputs are normalized between $[0,1]$. |
| $C$ | The set of classifiers to be fused. In this paper, C is sorted so that $h(c_1;\mathbf{x}_n) \geq h(c_2;\mathbf{x}_n) \geq \cdots \geq h(c_m;\mathbf{x}_n)$. |
| $g$ | A (regular) fuzzy measure of length $2^m - 1$. |
| $\mathscr{G}$ | A binary fuzzy measure (BFM) of length $2^m - 1$. |
| $C_{\mathbf{g}}(\mathbf{x}_n)$ | The Choquet integral output for instance $\mathbf{x}_n$. |
| $g(A_k)$ | The fuzzy measure element value corresponding to the subset $A_k = \{c_1, \ldots, c_k\}$. |
| $k_0$ | The sorted index for the first encounter of 1 on a path in the lattice of BFM, $k_0 \in \mathbb{Z}^+$ and $k_0 \in [1, m]$. |
| $d_n$ | Desired labels for the $n^{th}$ instance. |
| $J$ | Fitness values. |
| $t$ | Iteration $t$. |
| $\eta$ | Rate of "small-scale mutation", where only one measure element is sampled and updated. |
| $U$ | A count threshold for the number of repeated samples. $u$ is the counter. |
| $Q$ | A count threshold for the number of times the new BFM samples do not improve over past iterations. $q$ is the counter. |
| $U_f$ | Binary flag (True or False) to signal whether the search has been exhaustive. |
| $\mathscr{G}_t$ | The new BFM sampled at iteration $t$. |
| $P$ | The set of unique BFMs that have been sampled during optimization. |
| $J^*$ | Optimal fitness value. |
| $\mathscr{G}^*$ | Optimal BFM value. |

**Definition 1.** *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 [10, 21–23]:*

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

Fuzzy measures model the relationship among the sources. Each measure element value represents the power/"worth" of a certain combination of the sources. In this paper, the measure elements within a fuzzy measure are denoted with a subscript matching its corresponding subset. For example, element $g_1$ corresponds to subset $\{c_1\}$, element $g_{12}$ corresponds to subset $\{c_1, c_2\}$, etc. Note that $\mathbf{g}$ has a total of $2^m - 1$ elements and $g_{123\ldots m}$ is always equal to 1 (Property 2). All other measure elements hold real values between $[0,1]$ and satisfy monotonicity (Property 3). Non-monotonic fuzzy measures have been studied in the literature [24–26], but this paper focuses on monotonic and normalized fuzzy measures, following the assumption of our previous MICI work [6, 8]. The term "(regular) fuzzy measures" in this paper always refers to such monotonic and normalized fuzzy measures. The word "regular" was added to differentiate with the binary fuzzy measure that we define as follows.

**Definition 2.** *[12, 13] A* ***binary fuzzy measure (BFM)****,* $\mathscr{G}$*, is a real valued function that maps* $2^C \rightarrow \{0,1\}$*. It satisfies the following properties:*

1. $\mathscr{G}(\emptyset) = 0$*;*
2. $\mathscr{G}(C) = 1$*;*

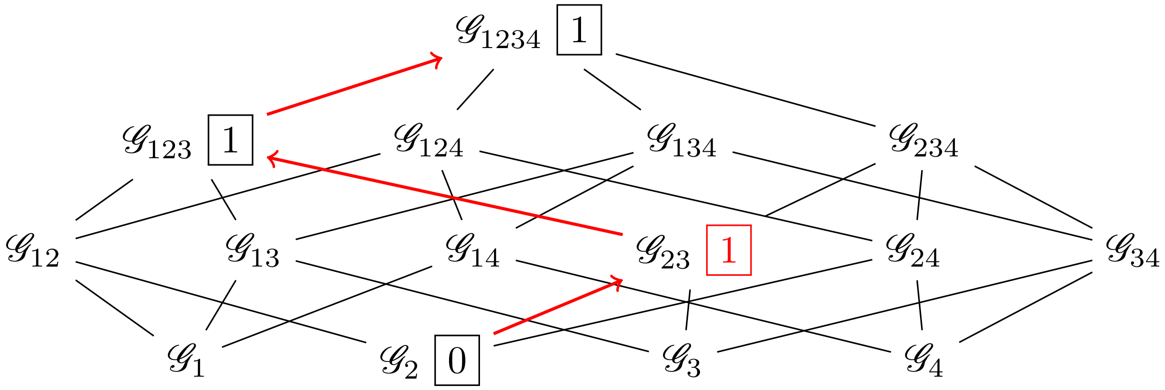

**Fig. 2:** An illustration for the subset and superset relationships between (binary) fuzzy measure elements given four sources. The red arrows describe one path for "climbing up the lattice" (the monotonicity property). Given $\mathscr{G}_{23} = 1$ (marked in red box), for example, it can be deduced from BFM definition that $\mathscr{G}_2 = 0$ and $\mathscr{G}_{123} = \mathscr{G}_{1234} = 1$.

*3. $\mathscr{G}(A) \leq \mathscr{G}(B)$ if $A \subseteq B$ and $A, B \subseteq C$. That is to say, if $\mathscr{G}(B) = 0$, $\mathscr{G}(A) \equiv 0$. Otherwise, if $\mathscr{G}(B) = 1$, $\mathscr{G}(A) \equiv 0$ or $1$. Similarly, if $\mathscr{G}(A) = 1$, $\mathscr{G}(B) \equiv 1$. Otherwise, if $\mathscr{G}(A) = 0$, $\mathscr{G}(B) \equiv 0$ or $1$.*

Figure 2 shows an illustration of a binary fuzzy measure for four sources. The red arrows show one path to "climb up the lattice" of the fuzzy measure elements. Suppose we learned that the "first encounter of 1" on the $\mathscr{G}_2 \rightarrow \mathscr{G}_{23} \rightarrow \mathscr{G}_{123} \rightarrow \mathscr{G}_{1234}$ path (marked in red) is the $\mathscr{G}_{23}$ element. Thus, we can automatically deduce that its subset, $\mathscr{G}_2 \equiv 0$ as $\mathscr{G}_2 \leq \mathscr{G}_{23}$ and $\mathscr{G}_2 \in \{0, 1\}$ ($\mathscr{G}_2 \neq 1$ as $\mathscr{G}_{23}$ is given as the "first encounter" of 1 on that path). On the other hand, the elements $\mathscr{G}_{123}$ and $\mathscr{G}_{1234}$ are the supersets of $\mathscr{G}_{23}$ and will have to take the value 1 as they can only be larger or equal to $\mathscr{G}_{23}$, according to the monotonicity property. The values for the remaining measure elements can be automatically deduced. This property of BFM leads to a simpler representation and more efficient computation, as the BFM only needs to optimize over $\{0, 1\}^{2^C}$ versus $[0, 1]^{2^C}$ for the regular fuzzy measures [12, 13].

### *B. Choquet Integral*

Fuzzy measures are used to define fuzzy integrals such as the Sugeno fuzzy integral (hereinafter referred to as "the Sugeno integral", or SI) [21] and the Choquet fuzzy integral (hereinafter referred to as "the Choquet integral", or CI) [10, 27] in the literature. There have been studies on the statistical properties of the Choquet and Sugeno Integrals [28–30] and we focus on the Choquet integral in this paper as it has shown better classification performance than the Sugeno integral in some applications in the literature [31, 32]. In fact, it has been proven in [13] that the CI and SI are equal for a BFM. The Choquet integral (CI) is a natural extension of the Lebesgue integral [33, 34] and has long been used as an effective aggregation operator for sensor fusion [23]. In this paper, we use the discrete Choquet integral to fuse discrete number of classifier outputs.

**Definition 3.** *The* ***discrete Choquet integral*** *on an instance $\mathbf{x}_n$ given classifiers $C$ with respect to a fuzzy measure $\mathbf{g}$ is computed as [8, 23, 34]:*

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

*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, $h(c_{m+1}; \mathbf{x}_n) := 0$. The fuzzy measure element value corresponding to the subset $A_k = \{c_1, \ldots, c_k\}$ is $g(A_k)$.*

To compute the CI of a BFM, one simply substitutes $\mathbf{g}$ with BFM $\mathscr{G}$ in (1). It has been proved in [12] that the Choquet integral on an instance $\mathbf{x}_n$ given classifiers C with respect to a binary fuzzy measure $\mathscr{G}$ is equal to the $k_0^{th}$ classifier output $h(c_{k_0}; \mathbf{x}_n)$, where $C$ is sorted so that $h(c_1; \mathbf{x}_n) \geq h(c_2; \mathbf{x}_n) \geq \cdots \geq h(c_m; \mathbf{x}_n)$, $h(c_{m+1}; \mathbf{x}_n) := 0$, $g(A_k)$ is the fuzzy measure element value corresponding to the subset $A_k = \{c_1, \ldots, c_k\}$ and $k_0$ is the (sorted) index such that $g(A_i) = 0 \ \forall i < k_0$ and $g(A_j) = 1 \ \forall j \geq k_0$. This way, the computation of CI with regard to a BFM is reduced to simply determining $k_0$ and is much more efficient to compute and store.

### *C. Learning The Fuzzy Measure*

A crucial aspect of using the CI to perform fusion is to learn all the element values of the fuzzy measure $\mathbf{g}$ from training data of this form [8]. In this subsection, we briefly review the least-square-based approaches and evolutionary-algorithm-based approaches for learning a fuzzy measure in the literature.

*1) Least-square based approaches:* As introduced in Section II-A, for $m$ input sources, there are $2^m - 1$ non-empty subsets and hence $2^m - 1$ fuzzy measure elements. Excluding $g_{123 \ldots m} \equiv 1$, there are $2^m - 2$ unknown fuzzy measure element values to be estimated. A quadratic programming (QP) approach to solve for the fuzzy measures based on the least-square criteria was discussed in [34, 35]. Given the discrete CI formula in equation (1) and assuming the desired labels for the $n^{th}$ data point/instance $\mathbf{x}_n$ is $d_n$, the goal of the the least-square criteria is to find the fuzzy measure $\mathbf{g}$ so that the squared error is minimized between the Choquet integral outputs of all training data points given $\mathbf{g}$ and their desired labels $d_n$ [34, 35]:

$$\min_{\mathbf{g}} E^2 = \sum_{n=1}^{N} \left(C_{\mathbf{g}}(\mathbf{x}_n) - d_n\right)^2 . \tag{2}$$

It has been shown in [34, 36] that the above least square criteria can be formulated into a quadratic programming (QP) problem given monotonicity constraints. The quadratic programming approach can then be solved by a QP solver such as the MATLAB built-in *quadprog()* function. This method is called the "CI-QP" method and will be used as a baseline comparison method later in our experiments. Note that for the CI-QP method, precise pixel-level label $d_n$ is required and assumed to be available for all $n$ data points.

*2) Evolutionary Algorithms:* The evolutionary algorithm (EA) has been used in the literature to determine fuzzy measure values [6, 8, 19, 37–46]. The EA algorithm usually considers the fuzzy measure as a chromosome and generates a population of potential measures. A fitness function is predefined to model and evaluate the measure. An example of the fitness function is a function that calculates misclassification rate or the least-squared error between the CI output and the actual class label, such as in (2). Measures are then selected from the measure population based on their fitness values. "Parent" measure element values from the last iteration are updated by mutation based on their fitness values and a "child" measure population is generated. This process continues until a stopping criterion is met (for example, maximum number of iterations or the misclassification rate is below a threshold). The measure that gives the best fitness value is returned as the optimal fuzzy measure solution.

Other methods, such as the gradient descent algorithm [17], Particle Swarm Optimization (PSO) [47–49], and neuron models and neural networks [50–52], have also been used in the literature to solve for fuzzy measures based on Choquet integrals.

## III. Multiple Instance Choquet Integral with Binary Fuzzy Measures

The binary fuzzy measure has been explored previously in [12, 13, 53]. However, these work mainly use hand-crafted, synthetic datasets only and cannot handle imprecise labels as commonly observed in remote sensing data. To our knowledge, no prior experiments on binary fuzzy measure have been conducted on real remote sensing applications considering imprecise labels. To address this challenge, we apply the binary fuzzy measure with the recently proposed, state-of-the-art Multiple Instance Choquet Integral (MICI) algorithm [6, 8] for classifier fusion applications in remote sensing. We use real-world hyperspectral data in our following experiments and demonstrate the effectiveness of the Choquet integral with binary fuzzy measure. We also propose a novel sampling-based optimization method for learning the optimal BFM in our applications.

### *A. The MICI-BFM Algorithm: Objective Function*

The MICI classifier fusion algorithm was proposed to address the label uncertainty problem commonly observed in remote sensing applications. As motivated in Section I, standard supervised Choquet integral fusion methods require accurate pixel-level training labels, which are often difficult or impossible to obtain in remote sensing applications, due to factors such as sensor noise, measurement inaccuracy, co-registration error, and the presence of sub-pixel and occluded targets in the scene. We previously proposed a MICI min-max approach [6] that can perform CI fusion with such label imprecision, and we observed that the learned fuzzy measure value of the CI tend to approximate binary values for two-class classifier fusion problems. Thus, we propose to use BFM with the MICI algorithm for more accurate classification results and more efficient computation.

The MICI model formulates the classifier fusion problem under the Multiple Instance Learning (MIL) framework [9] to address label uncertainty. It is assumed that labels are provided for a set (or, a *bag*) of data points, but pixel-level labels are not available or imprecise in training data. The MIL framework assumes that all the instances (pixels or data points) in a negative bag are negative (label "0"), and at least one instance in each positive bag should be positive (label "1"). The binary fuzzy measure $\mathscr{G}$ is used in the MICI-BFM algorithm for the CI fusion. The objective function of the MICI-BFM algorithm is written as

$$\begin{aligned} J = &\sum_{a=1}^{B^-} \max_{\forall \mathbf{x}_{ai}^- \in \mathscr{B}_a^-} \left(C_{\mathscr{G}}(\mathbf{x}_{ai}^-) - 0\right)^2 \\ &+ \sum_{b=1}^{B^+} \min_{\forall \mathbf{x}_{bj}^+ \in \mathscr{B}_b^+} \left(C_{\mathscr{G}}(\mathbf{x}_{bj}^+) - 1\right)^2 , \end{aligned} \tag{3}$$

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, $C_{\mathscr{G}}$ is the Choquet integral output given binary fuzzy measure $\mathscr{G}$ computed using (1), $\mathscr{B}_a^-$ is the $a^{th}$ negative bag, and $\mathscr{B}_b^+$ is the $b^{th}$ positive bag. By minimizing the objective function in (3), we encourage the CI value of all instances in the negative bags to be 0 (the *max* term) and encourage the CI value of at least one instance in the positive bags to be 1 (the *min* term), which satisfies the MIL assumption. Compared with standard MICI models, the BFM $\mathscr{G}$ is used to here restrict the search space of the fuzzy measure to binary for CI fusion.

### *B. The MICI-BFM Algorithm: Measure Optimization*

As discussed in [13], the binary fuzzy measure only needs to optimize the space of $\{0,1\}^{2^C}$ rather than $[0,1]^{2^C}$, which is significantly more efficient especially when the size of $C$ is large. Previously, when using regular fuzzy measures, the measure length is exponential to the number of sources $m$

(g is of length $2^m - 1$). For binary fuzzy measures, the number of fuzzy measure elements to be optimized is no longer exponential, but is linear ($O(m)$) [12]. Since BFM only consists of binary measure values, the search space is finite and the learning process is deterministic and can always converge if all possible combination of fuzzy measure elements are surveyed.

In this paper, we propose a novel optimization scheme inspired by the evolutionary algorithm to learn BFMs within the MICI-BFM framework. The pseudo-code for the proposed optimization scheme for both training and testing stages can be seen in Algorithm 1. We break down the algorithms by line numbers and describe each part in detail.[1] First, a BFM is randomly generated and used as an initial measure $\mathscr{G}_0$, and its fitness value $J_0$ is computed (A1L1). We set the current optimal measure and fitness to $\mathscr{G}_0$ and $J_0$, respectively. We maintain a set of unique BFMs that has been surveyed, $P$, and we add $\mathscr{G}_0$ as the first element $P_0$ (A1L2). The collection $P = \{P_0, P_1, ...\}$ can be viewed as a "look-up table" that contains all unique BFMs that have been searched in previous iterations. The variable $U_f$ is a binary logical flag to indicate the search has been deemed exhaustive (True) or not yet (False).

Then, at iteration $t$, we sample a new BFM to add to $P$, following Algorithm 2. A new BFM is sampled by either changing one measure element value in the existing BFM (A2L2-4) or simply generating a brand-new BFM (A2L5-7), which is parallel to the "small-scale mutation" and "large-scale mutation" steps in the evolutionary algorithm for sampling regular FMs in our previous work [8]. For a rate of small-scale mutation $\eta$, a new BFM is generated by updating only one measure element. we first evaluate the "valid intervals" of BFM $\mathscr{G}_{t-1}$ (A2L3). The term "valid interval" defines how much "wiggle room" a measure element can change values without sacrificing monotonicity. Previously, when using regular FMs, the valid interval is defined as the difference between the lower and upper bound for each measure element [8]. The lower and upper bounds of a measure element were computed as the largest value of its subsets and the smallest value of its supersets, respectively. In this paper, when using BFMs, there are only three options for the "valid interval" for each measure element. If both the upper and lower bounds of a measure element have the value 1, the measure element has no other option but to hold the value 1. If both the upper and lower bounds of a measure element have the value 0, the measure element has no other option but to hold the value 0. If the upper bound of a measure element has value 1 and the lower bound of a measure element has value 0, the measure element value can then choose from either 0 or 1. As an example, the $\mathscr{G}_2$ in Figure 2 has a valid interval of 1 but $\mathscr{G}_{123}$ has a valid interval of 0.

After evaluating the valid intervals, one measure element is selected in $\mathscr{G}_{t-1}$ by sampling from a multinomial distribution based on the valid intervals of all measure elements (A2L4). For BFM, the valid interval can only take values of 1 or 0, thus, this sampling process is equivalent to uniformly sampling from measure elements that can change values (you may think of them as the independent variables in [12]). Then, this measure element value is updated (from 0 to 1 or from 1 to 0) to form a new BFM. On the other hand, for a rate of large-scale mutation $1 - \eta$, a brand-new BFM is sampled randomly (A2L6). We set $\eta = 0.5$ in our experiments to balance between quickly updating a measure element with valid intervals and searching for new BFMs to avoid local minima.

**Algorithm 1** The optimization scheme of MICI-BFM

```
    TRAINING
Require: Training Data, Training Labels, Parameters
 1: Initialize a BFM $\mathscr{G}_0$. Compute fitness $J_0$ using (3).
 2: $J^* \leftarrow J_0$; $t, p, q \leftarrow 0$; $U_f \leftarrow False$; $P_0 \leftarrow \mathscr{G}_0$. $\mathscr{G}^* \leftarrow \mathscr{G}_0$.
 3: while True do
 4:     $t \leftarrow t + 1$.
 5:     $\mathscr{G}_t \leftarrow$ Sample a new BFM based on $\mathscr{G}_{t-1}$ using Algorithm 2 .
 6:     $u \leftarrow 0$.
 7:     while $\mathscr{G}_t \in P$ do
 8:         $\mathscr{G}_t \leftarrow$ Sample a new BFM based on $\mathscr{G}_t$ according to Algorithm 2.
 9:         $u \leftarrow u + 1$.
10:         if $u > U$ then break; $U_f \leftarrow True$;
11:         end if
12:     end while
13:     if $U_f$ then break;
14:     else
15:         $p \leftarrow p + 1$.
16:         $P_p \leftarrow \mathscr{G}_t$.
17:         Evaluate fitness $J_t$ of $\mathscr{G}_t$ using (3).
18:         if $J_t > J^*$ then
19:             $J^* \leftarrow J_t$, $\mathscr{G}^* \leftarrow \mathscr{G}_t$.
20:         else
21:             $q \leftarrow q + 1$.
22:         end if
23:         if $q > Q$ then break;
24:         end if
25:     end if
26: end while
        return $\mathscr{G}^*$

    TESTING
Require: Testing Data, $\mathscr{G}^*$
27: $TestLabels \leftarrow$ Choquet integral output computed based on Equation (1)
    using the learned $\mathscr{G}^*$ above.
        return $TestLabels$
```

**Algorithm 2** Sampling a new BFM

```
Require: a BFM $\mathscr{G}$, rate of small-scale mutation $\eta$
 1: Randomly generate $z \in [0, 1]$.
 2: if $z < \eta$ then
 3:     Evaluate valid intervals of $\mathscr{G}$.
 4:     $\mathscr{G}' \leftarrow$ Sample a measure element in $\mathscr{G}$ based on the valid intervals
    and update value.
 5: else
 6:     $\mathscr{G}' \leftarrow$ Randomly generate a new BFM.
 7: end if
        return Updated measure $\mathscr{G}'$
```

After sampling such a new measure (A1L5), we perform a check to see if this updated BFM already exists in $P$ (A1L7-12). If we already searched this new BFM, we re-sample another BFM using Algorithm 2 again. This sampling process is repeated until a new BFM is generated that has not been

[1] In the brackets, the number after "A" indicates which algorithm (1 or 2) and "L" indicates the line number in the corresponding algorithm.

previously searched in $P$. Then, the fitness of this new measure is computed. If the fitness is greater than the best fitness $J^*$ so far, the best BFM $\mathscr{G}^*$ is updated (A1L15-19). If a new BFM cannot be found after repeatedly sampling new measures more than $U$ times (A1L10), or if the fitness remains not updated after $Q$ times (A1L23), the algorithm assumes that the search has exhausted all possible BFMs and automatically stops. In our experiments, we set $U = 500$ and $Q = 100$ to ensure that possible BFM combinations can be sufficiently explored and yet the algorithm does not waste time if no possible BFMs are left to be sampled. The best BFM $\mathscr{G}^*$ is then returned at the end of training and used in the testing stage to compute CI fusion results for testing data.

## IV. Experimental Results of MICI-BFM on Target Detection Given Remote Sensing Data

The proposed MICI-BFM was tested on a real target detection application using the MUUFL Gulfport hyperspectral data set. The MUUFL Gulfport hyperspectral data set [7] was collected over the University of Southern Mississippi-Gulf Park Campus in Long Beach, MS, USA in November 2010 and contains 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.[2] The image size for the three flights are $325 \times 337$, $329 \times 345$, and $333 \times 345$ pixels, respectively. All HSI data cubes contain 72 bands corresponding to wavelengths 367.7nm to 1043.4nm and were collected using the CASI hyperspectral camera [7]. In this experiment, the first four and last four bands were removed due to noise.

Figure 3 shows the RGB images over the scene and the "ground truth" locations for four types of cloth panel targets: brown (15), dark green (15), faux vineyard green (12), and pea green (15). The ground truth locations were determined by a Trimble Juno SB hand-held GPS (Global Positioning System) device and were only accurate up to several pixels. Thus, the MICI approach is needed to account for such imprecision and uncertainty during fusion. In this experiment, we generated detection maps for each of the four individual target types using the adaptive coherence estimator (ACE) detector [55–57] based on known target spectral signatures. The goal of this experiment is to detect all targets in the scene by fusing results from the four individual target detectors using the MICI classifier fusion approach. We use similar experimental set-up as in [6] and perform flight-based cross validation, i.e., training on flight 1 and testing on flight 3 and flight 4, and so on. In our previous work, we presented fusion results of MICI min-max model using regular FMs and showed that it achieves state-of-the-art detection results (see Table IV in [6]). In this work, we compare our newly proposed MICI-BFM algorithm to our previous MICI results with regular FMs and show the effectiveness and efficiency of using BFMs in the MICI framework for fusion. In addition, we compare our proposed MICI-BFM method to the CI-QP method as discussed in Section II-C1. The CI-QP approach requires accurate pixel-level labels for every training data point and, thus, cannot handle imprecise labels as does the MICI method. In our implementation of the CI-QP in this problem, all points in a positive bag were labeled "1" and all points in a negative bag were labeled "0". We implemented two versions of the CI-QP method, "CI-QP FM" using regular fuzzy measures and "CI-QP BFM" enforcing binary FMs.

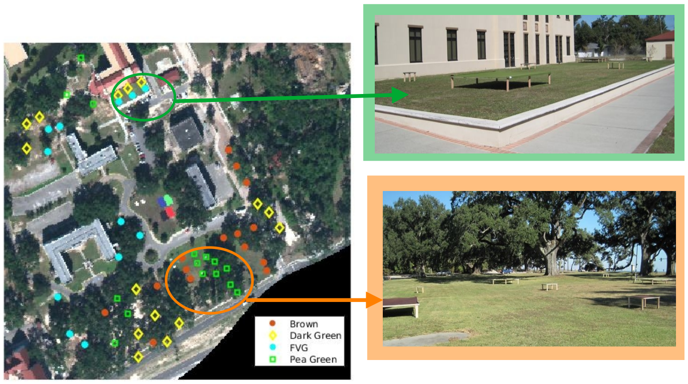

**Fig. 3:** The left column shows the RGB image from MUUFL Gulfport data set with ground truth target locations. Brown circle marks the true brown target locations, yellow diamond marks the true dark green target locations, cyan asterisk marks the true faux vineyard green (FVG) target locations, green square marks the true pea green target locations. The bottom right black area are invalid regions and are excluded in our experiments. The right column shows two sample photographs of the setup of cloth panel targets in the scene.

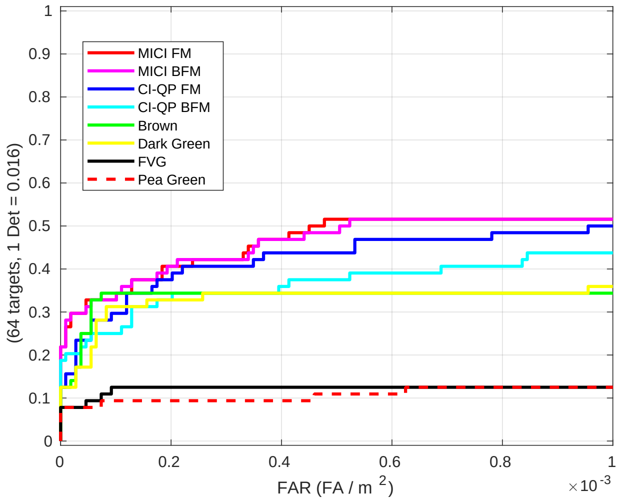

**Fig. 4:** A visual example of the ROC curve results, train on flight 3 data and test on flight 1 data.

We evaluated the target detection results using the receiver operating characteristic (ROC) curve. Figure 4 shows a visual ROC curve example for train on flight 3 and test on flight 1. The remaining cross validation experiments yield similar ROC curve results. The X-axis of the ROC curves represents the False Alarm Rate (FAR) up to $1\times10^{-3}/m^2$ (corresponding

[2] The data set is available at [54]. 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:** The AUC results of the MUUFL Gulfport data target detection experiment. The standard deviation across five runs is noted in parentheses. The higher the AUC, the better the detection performance. The best and second best results are marked in **bold** and underline, respectively.

| Methods | Train1Test3 | Train1Test4 | Train3Test1 | Train3Test4 | Train4Test1 | Train4Test3 |
|---|---|---|---|---|---|---|
| Brown | 0.262 | 0.263 | 0.334 | 0.267 | 0.308 | 0.265 |
| Dark Green | 0.260 | 0.260 | 0.328 | 0.257 | 0.297 | 0.267 |
| FVG | 0.118 | 0.116 | 0.122 | 0.107 | 0.135 | 0.102 |
| Pea Green | 0.000 | 0.000 | 0.107 | 0.000 | 0.100 | 0.091 |
| CI-QP FM | 0.320 | 0.325 | 0.421 | 0.307 | 0.261 | 0.279 |
| CI-QP BFM | 0.345 | 0.339 | 0.364 | 0.296 | 0.255 | 0.247 |
| MICI-FM | **0.348**(**0.000**) | 0.347(0.003) | **0.465**(**0.005**) | 0.339(0.001) | **0.345**(**0.001**) | **0.328**(**0.001**) |
| **MICI-BFM** | **0.348** | **0.348** | 0.460 | **0.340** | 0.344 | 0.326 |

**TABLE III:** Running time in seconds (s) on the MUUFL Gulfport data target detection experiments. Experiments were conducted in MATLAB using a desktop computer with Intel i7 CPU 3.60 GHz processor. The running times are provided for relative comparisons only. The standard deviation of running time across five runs is noted in parentheses. The better result (less computation time) is marked in **bold**.

| Methods | Train1Test3 | Train1Test4 | Train3Test1 | Train3Test4 | Train4Test1 | Train4Test3 | Overall Mean Run Time (s) |
|---|---|---|---|---|---|---|---|
| MICI-FM | 23.9(20.3) | 31.3(28.3) | 16.9(9.1) | 14.0(10.0) | 18.1(11.6) | 28.6(21.9) | 22.1 |
| **MICI-BFM** | 0.2(0.0) | 0.2(0.1) | 0.2(0.0) | 0.3(0.1) | 0.2(0.0) | 0.3(0.1) | **0.2** |

**TABLE IV:** RMSE and running time results on the synthetic data experiment varying number of sources $m$, with true binary measures. Experiments were conducted in MATLAB using a desktop computer with Intel i7 CPU 3.60 GHz processor. The running times are provided for relative comparisons only. The standard deviation across five runs is noted in parentheses.

| Metric | Method | $m$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Measure RMSE | MICI-FM | 0.002(0.001) | 0.021(0.010) | 0.183(0.069) | 0.080(0.021) | 0.090(0.020) | 0.273(0.372) | 0.202(0.052) | 0.213(0.034) |
| | MICI-BFM | 0 | 0 | 0 | 0 | 0 | 0.060(0.036) | 0.038(0.034) | 0.088(0.025) |
| Fusion RMSE | MICI-FM | 0.001(0.001) | 0.009(0.006) | 0.047(0.010) | 0.039(0.010) | 0.054(0.011) | 0.109(0.115) | 0.059(0.015) | 0.063(0.013) |
| | MICI-BFM | 0 | 0 | 0 | 0 | 0 | 0.032(0.023) | 0.013(0.013) | 0.026(0.008) |
| Run Time (s) | MICI-FM | 101.6(122.3) | 37.2(24.5) | 70.9(68.4) | 38.0(23.7) | 78.7(44.3) | 125.2(135.3) | 244.0(129.6) | 286.1(321.3) |
| | MICI-BFM | 0.1(0.0) | 0.1(0.0) | 0.2(0.2) | 0.3(0.1) | 0.9(0.2) | 1.2(0.2) | 2.0(0.5) | 4.0(1.4) |

**TABLE V:** RMSE and running time results on the synthetic data experiment varying number of sources $m$, with true non-binary measures. Experiments were conducted in MATLAB using a desktop computer with Intel i7 CPU 3.60 GHz processor. The running times are provided for relative comparisons only. The standard deviation across five runs is noted in parentheses.

| Metric | Method | $m$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| Measure RMSE | MICI-FM | 0.045(0.015) | 0.070(0.014) | 0.107(0.026) | 0.119(0.035) | 0.070(0.021) | 0.171(0.015) | 0.098(0.024) | 0.058(0.010) |
| | MICI-BFM | 0.349(0.134) | 0.271(0.013) | 0.470(0.106) | 0.262(0.098) | 0.151(0.012) | 0.305(0.310) | 0.173(0.058) | 0.261(0.383) |
| Fusion RMSE | MICI-FM | 0.015(0.005) | 0.039(0.010) | 0.051(0.012) | 0.038(0.012) | 0.036(0.013) | 0.068(0.007) | 0.053(0.012) | 0.043(0.011) |
| | MICI-BFM | 0.116(0.043) | 0.129(0.005) | 0.172(0.035) | 0.087(0.023) | 0.092(0.012) | 0.122(0.091) | 0.078(0.017) | 0.176(0.232) |
| Run Time (s) | MICI-FM | 23.9(9.0) | 28.0(22.6) | 27.7(9.1) | 37.5(15.5) | 32.1(24.3) | 59.5(40.2) | 183.2(110.9) | 217.6(243.3) |
| | MICI-BFM | 0.2(0.0) | 0.4(0.1) | 0.3(0.1) | 0.3(0.1) | 0.6(0.1) | 0.8(0.2) | 2.3(0.2) | 2.7(0.9) |

to a reasonable scale of 1 false alarm in $1000m^2$) and the Y-axis represents Positive Detection (PD). As shown from the plot, the two MICI methods yield top-performing ROC curves than comparison methods and MICI-BFM algorithm has slight improvement over MICI-FM especially at low FAR.

Table II shows quantitative results of the area under curve (AUC) statistics across all fusion methods. The higher the AUC, the better the detection results. The top four rows in Table II show the ACE detector results for the four individual target types. The bottom four rows show comparison fusion detection results between two CI-QP methods, the MICI method using regular FMs, and our proposed MICI-BFM algorithm. Table III reports the computation time comparison between the two MICI methods.

We obtain the following observations from the experimental results. First, we see from Table II that all fusion methods (the bottom four rows) outperform individual target detection results (the top four rows) in general, which shows the effectiveness and necessity of fusion. Second, the two MICI approaches outperform the two CI-QP methods regardless of regular or binary FMs, which shows that it is useful to incorporate label uncertainty and the MICI methods can perform effective fusion on remote sensing data with imprecise ground truth labels under the MIL framework. Third, the MICI fusion models are the two top-performing methods, with similar detection results using the regular and binary fuzzy measures. The proposed MICI-BFM algorithm did outperform slightly in half of the experiments. The measure values learned by the binary fuzzy measure method are all "1"s, which is as expected since the desired fusion output is the combination of all individual target types. The measure values learned by regular fuzzy measure are close to 1 but not exactly (approximately 0.99) due to the sampling process, which caused the slight performance deterioration. We also observed from Table III that the proposed MICI-BFM was able to converge in approximately 0.2 seconds, which was more than a hundred times faster than the previous MICI method using regular FMs. This is due to the fact that previous MICI-FM method samples a population of measure elements from $[0,1]^{2^C}$, while the MICI-BFM only needs to evaluate a limited amount of possible binary fuzzy measures.

## V. Further Analysis and Discussions of BFM versus FM with Varying Number of Classifiers

We conducted an additional set of experiments to further demonstrate the performance of MICI-FM and MICI-BFM when the number of classifier fusion sources vary. For $m$ between 3 and 10, a random FM or BFM is generated as the "true measure". We randomly generated instances (with dimension $m$) with values between $[0,1]$ and we compute their ground-truth CI output based on the true measures. If the CI output of an instance is larger than 0.5, the instance is classified as a positive instance and otherwise a negative instance. We keep generating such instances until a dataset is constructed with 100 bags with 20 instances in each bag. Half of the bags are labeled negative (only contains negative instances) and half of the bags are positive (contains one positive instance). We limit the number of positive instances in a bag to one to avoid possible influence of bag structure on the performance of fusion. The MICI-FM and MICI-BFM algorithms are applied on such dataset with the constructed bags and bag-level labels, and we compare the Root Mean Square Error (RMSE) between the estimated FM/BFMs with the true measures (measure RMSE), as well as the final fusion results compared with the ground-truth CI outputs of all instances (fusion RMSE). The lower the RMSE values, the better the performance. We also report the running time for all experiments. The RMSE and running time results are reported in Table IV when the given ground-truth measure is binary, and in Table V when the given ground-truth measure is non-binary.

As shown in Table IV, when the underlying true measure is binary, the MICI-BFM outperforms the MICI-FM algorithm in all cases tested. The MICI-BFM can learn exactly the true binary measure in most cases since it surveys as many BFMs combinations as possible. Our results in Table II for the real remote sensing fusion experiment also confirms this point. When $m$ increases, the length of BFM increases and it may take more iterations to search all possible measures depending on initialization, but it will usually converge to an optimal BFM that is close to the ground-truth, resulting in very small measure and fusion RMSE results. If we increase $U$ and $Q$ parameters so to allow more iterations to search for possible measures, the optimization process could learn a guaranteed optimal BFM, but the search time may be longer. Future work will include developing new optimization schemes to more intelligent and rapid search for BFM solutions given the monotonicity constraint.

When the true measure is non-binary, as shown in Table V, the MICI-FM algorithm outperforms MICI-BFM, simply because the MICI-BFM enforces the learned measure to be binary which is not true. The measure learned by the MICI-FM algorithm is usually very close to the true measure. It is also worth noting that the MICI-BFM can still converge to an optimal BFM that can achieve the best possible fitness value for a BFM and the fusion label RMSE is still quite small. For both cases, the MICI-BFM is much faster than the MICI-FM algorithm.

Both results from Sections IV and V suggest that the MICI-BFM algorithm may be used as an efficient alternative to learn the true BFM, especially when the underlying true measure is a BFM. Often, when the true underlying measure is unknown in real applications, the proposed MICI-BFM algorithm can provide an efficient method for a good initial estimation of the relationship among fusion sources.

## VI. Conclusion

This paper proposes the use of binary fuzzy measures with the Multiple Instance Choquet Integral framework for remote sensing classifier fusion with imprecise labels. The paper provides real-world applications and experimental results of MICI-BFM for target detection problems given remote sensing data and shows that our proposed BFM-MICI approach can perform accurate classifier fusion with much more efficient computation compared with previous methods.

Future work will include applying the BFMs to alternative types of integrals besides the CI, such as the Sugeno integral, for multiple instance fusion problems. Future work will also include investigating generalized fuzzy measures such as the bi-capacity fuzzy measures (where the fuzzy measure maps to $[-1,1]$) and bipolar fuzzy measures [58]. Using BFMs for multi-resolution and multi-modal fusion problems such as [46] can also be investigated.

## Acknowledgment

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

## References

[1] 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, 2007. 1

[2] S. Prasad and L. M. Bruce, "Decision fusion with confidence-based weight assignment for hyperspectral target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1448–1456, 2008. 1

[3] 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, 2010. 1

[4] 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, 2010.

[5] X. Xu, W. Li, Q. Ran, Q. Du, L. Gao, and B. Zhang, "Multisource remote sensing data classification based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 937–949, 2018.

[6] X. Du and A. Zare, "Multiple instance choquet integral classifier fusion and regression for remote sensing applications," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2741–2753, 2019. 1, 2, 4, 6

[7] 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, 2013. 1, 6

[8] X. Du, A. Zare, J. Keller, and D. Anderson, "Multiple instance choquet integral for classifier fusion," in *IEEE Congr. Evolutionary Computation*, Vancouver, BC, 2016, pp. 1054–1061. 2, 3, 4, 5

[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, 1997. 2, 4

[10] G. Choquet, "Theory of capacities," in *Annales de l'institut Fourier*, vol. 5, 1954, pp. 131–295. 2, 3

[11] T. C. Havens and A. J. Pinar, "Generating random fuzzy (capacity) measures for data fusion simulations," in *IEEE Symp. Series on Computational Intelligence*, Honolulu, HI, 2017, pp. 1–8. 2

[12] M. A. Islam, D. T. Anderson, X. Du, T. C. Havens, and C. Wagner, "Efficient binary fuzzy measure representation and choquet integral learning," in *Int. Conf. Info. Process. Management of Uncertainty in Knowledge-Based Systems*. Cham, Switzerland: Springer, 2018, pp. 115–126. 2, 3, 4, 5

[13] D. T. Anderson, M. Islam, R. King, N. Younan, J. Fairley, S. Howington, F. Petry, P. Elmore, and A. Zare, "Binary fuzzy measures and choquet integration for multi-source fusion," in *IEEE Int. Conf. Military Technologies*, 2017, pp. 676–681. 2, 3, 4

[14] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *European J. Operational Research*, vol. 89, no. 3, pp. 445–456, 1996. 2

[15] ——, "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, 1995, pp. 145–150.

[16] 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.

[17] 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, 2008. 4

[18] A. Mendez-Vazquez and P. Gader, "Learning fuzzy measure parameters by logistic lasso," in *Proc. Annual Conf. North American Fuzzy Information Processing Society*, 2008, pp. 1–7. 2

[19] 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. 2, 4

[20] D. T. Anderson, S. R. Price, and T. C. Havens, "Regularization-based learning of the choquet integral," in *IEEE Int. Conf. Fuzzy Systems*, 2014, pp. 2519–2526. 2

[21] M. Sugeno, "Theory of fuzzy integrals and its applications," Ph.D. dissertation, Tokyo Institute of Technology, 1974. 2, 3

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

[23] A. Mendez-Vazquez, "Information fusion and sparsity promotion using choquet integrals," Ph.D. dissertation, University of Florida, 2008. 2, 3

[24] T. Murofushi, M. Sugeno, and M. Machida, "Non-monotonic fuzzy measures and the choquet integral," *Fuzzy Sets and Systems*, vol. 64, no. 1, pp. 73–86, 1994. 2

[25] T. Murofushi and M. Sugeno, "Fuzzy measures and fuzzy integrals," *Fuzzy Measures and Integrals: Theory and Applications*, pp. 3–41, 2000.

[26] Y. Narukawa and V. Torra, "Non-monotonic fuzzy measures and intuitionistic fuzzy sets," in *Int. Conf. Modeling Decisions for Artif. Intell.*, 2006, pp. 150–160. 2

[27] T. Murofushi and M. Sugeno, "A theory of fuzzy measures: Representations, the choquet integral, and null sets," *J. Math. Analysis & Applications*, vol. 159, no. 2, pp. 532–549, 1991. 3

[28] S. Auephanwiriyakul and J. M. Keller, "A comparison of the linguistic choquet and sugeno fuzzy integrals," in *10th IEEE Int. Conf. Fuzzy Systems*, vol. 1, 2001, pp. 312–315. 3

[29] M. Grabisch and E. Raufaste, "An empirical study of statistical properties of the choquet and sugeno integrals," *IEEE Trans. Fuzzy Systems*, vol. 16, no. 4, pp. 839–850, 2008.

[30] E. P. Klement, R. Mesiar, and E. Pap, "A universal integral as common frame for choquet and sugeno integral," *IEEE Trans. Fuzzy Systems*, vol. 18, no. 1, pp. 178–187, 2010. 3

[31] 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.*, vol. 3, 2004, pp.

1605–1608. 3

[32] G. E. Martínez, O. Mendoza, P. Melin, and F. Gaxiola, "Comparison between choquet and sugeno integrals as aggregation operators for modular neural networks," in *IEEE Int. Conf. Fuzzy Systems*, 2016, pp. 2331–2336. 3

[33] D. Liginlal and T. T. Ow, "Modeling attitude to risk in human decision processes: an application of fuzzy measures," *Fuzzy Sets and Systems*, vol. 157, no. 23, pp. 3040–3054, 2006. 3

[34] 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. 3, 4

[35] M. Grabisch and J.-M. Nicolas, "Classification by fuzzy integral: performance and tests," *Fuzzy sets and systems*, vol. 65, no. 2-3, pp. 255–271, 1994. 3, 4

[36] X. Du, "Multiple instance choquet integral for multiresolution sensor fusion," Ph.D. dissertation, University of Missouri, Columbia, MO, Dec. 2017. 4

[37] K.-M. Lee and H. Leekwang, "Identification of $\lambda$-fuzzy measure by genetic algorithms," *Fuzzy Sets and Systems*, vol. 75, no. 3, pp. 301 – 309, 1995. 4

[38] D. Wang, X. Wang, and J. M. Keller, "Determining fuzzy integral densities using a genetic algorithm for pattern recognition," in *Annual Meeting of the North American Fuzzy Information Processing Society*, 1997, pp. 263–267.

[39] W. Wang, Z. Wang, and G. J. Klir, "Genetic algorithms for determining fuzzy measures from data," *J. Intelligent & Fuzzy Systems*, vol. 6, no. 2, pp. 171–183, 1998.

[40] Z. Wang, K. sak Leung, and J. Wang, "A genetic algorithm for determining nonadditive set functions in information fusion," *Fuzzy Sets and Systems*, vol. 102, no. 3, pp. 463 – 469, 1999.

[41] T.-Y. Chen, J.-C. Wang, and G.-H. Tzeng, "Identification of general fuzzy measures by genetic algorithms based on partial information," *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 4, pp. 517–528, 2000.

[42] T.-Y. Chen and J.-C. Wang, "Identification of $\lambda$-fuzzy measures using sampling design and genetic algorithms," *Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 321–341, 2001.

[43] Z. Wang and H.-F. Guo, "A new genetic algorithm for nonlinear multiregressions based on generalized choquet integrals," in *IEEE Int. Conf. Fuzzy Systems*, vol. 2, 2003, pp. 819–821.

[44] E. F. Combarro and P. Miranda, "Identification of fuzzy measures from sample data with genetic algorithms," *Computers & Operations Research*, vol. 33, no. 10, pp. 3046–3066, 2006.

[45] Y.-C. Hu, "Fuzzy integral-based perceptron for two-class pattern classification problems," *Information Sciences*, vol. 177, no. 7, pp. 1673 – 1686, 2007.

[46] X. Du and A. Zare, "Multi-resolution multi-modal sensor fusion for remote sensing data with label uncertainty," *arXiv preprint arXiv:1805.00930*, 2018. 4, 8

[47] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Int. Conf. Neural Networks*, vol. 4, 1995, pp. 1942–1948. 4

[48] T. Xiao, N. Zhang, and D.-M. Huang, "Solution to the model of nonlinear multiregression based on generalized choquet integral and its application in estimating forest volume," in *IEEE Int. Conf. Computer Application and System Modeling*, vol. 12, 2010, pp. V12–161–V12–165.

[49] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer, 2011, pp. 760–766. 4

[50] J. M. Keller and J. Osborn, "Training the fuzzy integral," *Int. J. Approximate Reasoning*, vol. 15, no. 1, pp. 1–24, 1996. 4

[51] J. Wang and Z. Wang, "Using neural networks to determine sugeno measures by statistics," *Neural Networks*, vol. 10, no. 1, pp. 183–195, 1997.

[52] H. Nemmour and Y. Chibani, "Neural network combination by fuzzy integral for robust change detection in remotely sensed imagery," *EURASIP J. Applied Signal Processing*, vol. 2005, pp. 2187–2195, 2005. 4

[53] M. A. Islam, D. T. Anderson, A. J. Pinar, and T. C. Havens, "Data-driven compression and efficient learning of the choquet integral," *IEEE Trans. Fuzzy Systems*, vol. 26, no. 4, pp. 1908–1922, 2018. 4

[54] A. Zare, P. Gader, J. Aitken, R. Close, G. Tuell, T. Glenn, D. Dranishnikov, and X. Du, "GatorSense/MUUFLGulfport: Release 01 (Version v0.1) [Data set]," https://github.com/GatorSense/MUUFLGulfport/tree/v0.1, 2018, DOI: https://doi.org/10.5281/zenodo.1186326. 6

[55] L. L. Scharf and L. T. McWhorter, "Adaptive matched subspace detectors and adaptive coherence estimators," in *Proc. 30th Asilomar Conf. Signals Syst.*, 1996, pp. 1114–1117. 6

[56] 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.

[57] 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, 2000. 6

[58] M. Grabisch and C. Labreuche, "Bi-capacities–II: the choquet integral," *Fuzzy sets and systems*, vol. 151, no. 2, pp. 237–259, 2005. 8