Multiple Instance Choquet Integral for Classifier Fusion

Xiaoxiao Du, Alina Zare, James M. Keller
Department of Electrical and Computer Engineering
University of Missouri, Columbia, MO 65211
xdy74@mail.missouri.edu, {zarea, kellerj}@missouri.edu

Derek T. Anderson
Electrical and Computer Engineering Department
Mississippi State Universtiy, Miss. State, MS 39762
anderson@ece.msstate.edu

***Abstract*—The Multiple Instance Choquet integral (MICI) for classifier fusion and an evolutionary algorithm for parameter estimation is presented. The Choquet integral has a long history of providing an effective framework for non-linear fusion. However, previous methods to learn an appropriate measure for the Choquet integral required accurate and precise training labels. In many applications, data-point specific labels are unavailable and infeasible to obtain. The proposed MICI algorithm allows for training with uncertain labels in which class labels are provided for sets of data points (i.e., "bags") instead of individual data points (i.e., "instances"). The proposed algorithm is able to fuse multiple two-class classifier outputs by learning a monotonic and normalized fuzzy measure from uncertain training labels using an evolutionary algorithm. It produces enhanced classification performance by computing Choquet integral with the learned fuzzy measure. Results on both simulated and real hyperspectral data are presented in the paper.**

I. Introduction

Information fusion connects and combines multiple inputs into a single output. Data, features extracted from the data, and classifier outputs estimated from the data can all be used as sources for fusion [1, 2]. Classifier fusion or decision-level fusion takes multiple classifier outputs estimated from the data as sources of information and derives one fused output to provide a final classification of the data [2, 3].

Classifier fusion has wide applications including remote sensing and hyperspectral image classification [4–6], object detection [7], land mine detection [3, 8–11], handwriting recognition [12], and medical and fault diagnosis [13–15]. Previous fusion algorithms, however, require training labels for each data point, which are usually difficult, if not impossible, to obtain. In target detection applications given remote sensing data, for example, the GPS groundtruth locations of targets in the scene are often only accurate to the level of several pixels. This makes it difficult to pinpoint accurate pixel-level target locations. The fusion algorithm proposed in this paper is able to work with uncertainty of this type by formulating the problem as a Multiple Instance Learning (MIL) problem. In this approach, the class labels are associated with groups of data points ("bags") instead of individual data points ("instances"). Only bag-level labels are needed for training in the proposed algorithm. In this paper, only the two-class classification problem is considered; specifically, problems with only a positive class (labeled as "+1") and a negative class (labeled as "0") are considered.

This material is based upon work supported by the National Science Foundation under Grant IIS-1350078-CAREER: Supervised Learning for Incomplete and Uncertain Data. This effort was also partially sponsored by the Engineering Research and Development Center under Cooperative Agreement number W912HZ-15-2-0004. Material presented in this paper is also a product of the CREATE-GV Element of the Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office.

The Choquet integral is the basis for our proposed MICI fusion algorithm. The Choquet integral (CI) is an effective aggregation operator in decision making [16–19]. A crucial aspect of using CI for fusion is learning the fuzzy measures for the CI [20, 21]. This paper proposes an evolutionary algorithm to learn a monotonic and normalized fuzzy measure used for the Choquet integral to perform classifier fusion.

The remainder of the paper is organized as follows: Section II provides background in fuzzy measures and the Choquet integral; Section III describes the proposed MICI method; Section IV presents experimental results of the proposed algorithm on both simulated and real hyperspectral data sets; Finally, Section V provides conclusion and future work.

II. Fuzzy Measure and Choquet Integral

Consider the case that there are m classifiers, $C = \{c_1, c_2, \ldots, c_m\}$. The proposed MICI algorithm is used to fuse a set of outputs from these classifiers. The set of all (crisp) subsets of C is 2^C. A monotonic and normalized fuzzy measure, $\mathbf{g}$, is a real valued function that maps $2^C \rightarrow [0, 1]$. It satisfies the following properties [22–24]:

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*

All "fuzzy measures" in this paper refer to monotonic and normalized fuzzy measures.

The set C contains $2^m - 1$ non-empty subsets. Within the Choquet integral, each measure element in the fuzzy measure corresponds to each non-empty subset of C. In this paper, the measure elements 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).

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

Denote the classifier output value of k^{th} classifier, c_k, on n^{th} data point/instance, $\mathbf{x}_n$, as $h(c_k;\mathbf{x}_n)$. The Choquet integral on instance $\mathbf{x}_n$ given C is computed as [22]:

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

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

In a classifier fusion problem with training data $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, the $h(c_1;\mathbf{x}_n)$, $h(c_2;\mathbf{x}_n)$, $\ldots$, $h(c_m;\mathbf{x}_n)$ values for all n are known. The desired bag-level labels for sets of $C_{\mathbf{g}}(\mathbf{x}_n)$ values are known. The goal of the proposed MICI algorithm is to learn all the element values of the fuzzy measure $\mathbf{g}$ from training data of this form.

III. MICI Algorithm

The proposed MICI algorithm learns a fuzzy measure to be used within a Choquet integral for two-class classifier fusion. The learning approach relies on the MIL framework [25, 26] given bag-level training labels. The computed Choquet integral with the learned measure aims at matching as much as possible to the desired labels.

A. Objective Function

In 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 can be used to express these assumptions [26]:

$$\begin{aligned} P(\mathbf{X}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}) \\ &= \left(\prod_{a=1}^{B^-}\prod_{i=1}^{N_a^-} 1 - p(\mathbf{x}_{ai}^-|\boldsymbol{\theta})\right)\left(\prod_{b=1}^{B^+}\left(1 - \prod_{j=1}^{N_b^+} 1 - p(\mathbf{x}_{bj}^+|\boldsymbol{\theta})\right)\right) \end{aligned} \quad (2)$$

where N is the total number of data points, $\mathbf{x}_n$ is the n^{th} data point/instance, 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 $\mathbf{x}_n$ is either positive or negative, this is 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. $\boldsymbol{\theta}$ represents the model parameters.

For all $a = 1,\ldots,B^-$, we want:

$$C_{\mathbf{g}}(\mathbf{x}_{ai}^-) = 0, \forall \mathbf{x}_{ai}^- \in \mathscr{B}_a^-. \quad (3)$$

For all $b = 1,\ldots,B^+$, we want:

$$C_{\mathbf{g}}(\mathbf{x}_{bj}^+) = 1, \exists \mathbf{x}_{bj}^+ \in \mathscr{B}_b^+. \quad (4)$$

where $C_{\mathbf{g}}$ 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.

In this algorithm, we define

$$\begin{aligned} p(\mathbf{x}_n|\boldsymbol{\theta}) &= \mathscr{N}\left(C_{\mathbf{g}}(\mathbf{x}_n)|\mu,\sigma^2\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(C_{\mathbf{g}}(\mathbf{x}_n) - \mu\right)^2\right\} \end{aligned} \quad (5)$$

where $\mu = 1$ for all true positive instances, thus, the Choquet integral values of positive instances are pushed to 1 and the Choquet integral values of negative instances are pushed far from 1. Here, the model parameter vector $\boldsymbol{\theta}$ consists of the variance of the Gaussian σ^2 and the fuzzy measure $\mathbf{g}$ values used to compute the Choquet integral.

Then, after taking the logarithm, the objective function in (2) becomes:

$$\begin{aligned} \ln p(\mathbf{X}|\boldsymbol{\theta}) = &\sum_{a=1}^{B^-}\sum_{i=1}^{N_b^-} \ln\left(1 - \mathscr{N}\left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-)|1,\sigma^2\right)\right) \\ &+ \sum_{b=1}^{B^+} \ln\left(1 - \prod_{j=1}^{N_b^+} 1 - \mathscr{N}\left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+)|1,\sigma^2\right)\right). \end{aligned} \quad (6)$$

By maximizing the objective function in (6), we push the Choquet integral of all the points in the the negative bag to zero (first term) and push the Choquet integral of at least one of the points in the positive bag to one (second term). The variance σ^2 is a user-defined parameter. The variance parameter control how sharply the Choquet integral values are pushed to 0 and 1. 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.

B. Optimization

Psuedocode for the MICI algorithm for both training and testing stages can be seen in Algorithm 1. Table I lists the notation used in Algorithm 1. The right arrows in the pseudocode mark the corresponding subsections where each step will be explained in detail.

TABLE I: List of notations for Algorithm 1

Notation	Meaning
$\mathbf{F}_P^0$	Fitness values for all measures in the inital population
I	Maximum number of iterations
P	Measure population size
$\mathscr{G}$	All measures in the current measure population
$\mathscr{G}\{p\}$	The p^{th} measure in measure population $\mathscr{G}$
η	Rate of small-scale mutation
$\mathbf{F}_P^t$	Fitness values for all measures in Iteration t
$F*$	Best (highest) current fitness value
$\mathbf{g}^*$	Best current measure with the highest fitness value

1) 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.

2) Evaluation of Valid Intervals: A measure element, when updated, must still satisfy monotonicity. The term "valid interval" is used in this paper to define how large the element value can change without sacrificing monotonicity. The valid

Algorithm 1 MICI Algorithm

TRAINING

Require: Training Data, Training Labels, Parameters

Initialize a population of measures ▷ III-B1
$F^* = max(\mathbf{F}_P^0)$, $\mathbf{g}^* = \arg\max_{\mathscr{G}} \mathbf{F}_P^0$
for $t := 1 \rightarrow I$ **do**
  for $p := 1 \rightarrow P$ **do**
    Evaluate valid intervals of $\mathscr{G}\{p\}$ ▷ III-B2
    Randomly sample $z \in [0, 1]$
    if $z < \eta$ **then** ▷ III-B3
      Update $\mathscr{G}\{p\}$ by small-scale mutation
    else
      Update $\mathscr{G}\{p\}$ by large-scale mutation
    end if
  end for
  Evaluate fitness of updated measures using (6)
  Select measures ▷ III-B4
  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
end for
  return $\mathbf{g}^*$

TESTING

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

$TestLabels \leftarrow$ Choquet integral output computed based on Equation (1) using the learned $\mathbf{g}^*$ above
  return $TestLabels$

interval width for each measure element is set as the difference between the lower and upper bound for each 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. For example, when number of sources $m = 3$, g_1 and g_2 corresponds to the subsets of measure element g_{12}, while g_{123} corresponds to the superset of g_{12}. Therefore, the lower bound of g_{12} is equal to $max(g_1, g_2)$ and the upper bound of g_{12} is $g_{123} = 1$. The valid interval width for element g_{12} is thus $1 - max(g_1, g_2)$. In this step, the valid interval width for each measure element is computed.

3) Mutation: An evolutionary algorithm is used to learn the fuzzy measure in the MICI algorithm. Mutations of two different scales were designed in search of the optimal solution. The valid intervals computed above are used in determining which measure element to update during both types of mutations.

In the large-scale mutation, all the measure elements are sorted based on their valid interval widths in descending order. Then, all the measure elements are updated according to their sort order. That is to say, the measure element with the largest valid interval width (and has the largest room to change) will be updated first, the measure elements with the second largest valid interval width will be updated second, and so on. The new measure values are sampled from a truncated Gaussian (TG) distribution [27]. The lower and upper bounds of the truncated Gaussian are equal to the lower and upper bounds of the valid interval of the corresponding measure element.

In the small-scale mutation, only one measure element gets updated. The element to be updated is chosen by randomly sampling from a multinomial distribution based on the valid intervals of all the measure elements. The probability of sampling a particular measure element g_l is set to

$$P(g_l) = \frac{w_l}{\sum_{o=1}^{2^C-1} w_o}, \tag{7}$$

where w_l is the valid interval width for measure element g_l. The measure element with the largest valid interval width will, therefore, have the largest probability to be updated. As in the large-scale mutation, the new measure values are obtained by sampling from a truncated Gaussian (TG) distribution.

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

4) Selection: The measures retained for the next generation are selected based on their fitness function values computed using Equation (6). In each iteration, all measures in the population are updated, yielding a child measure population of size P. The measure population before updating 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 (6). 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 (7). 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.

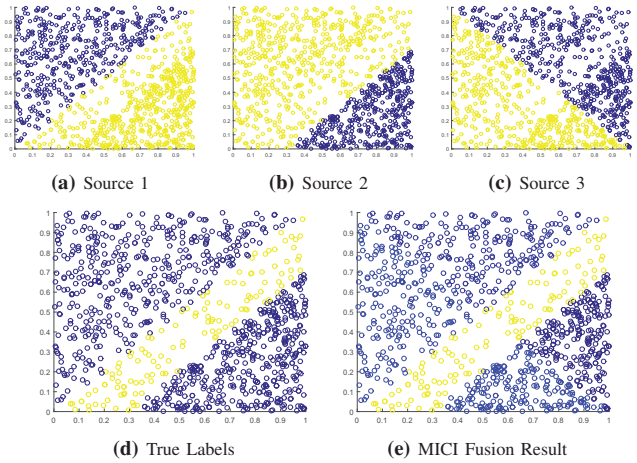
At the end of training process, the best measure $\mathbf{g}^*$ with the highest fitness value so far is returned as the learned measure and used for testing.

IV. Experiments

The proposed MICI algorithm is applied to both simulated classification and real hyperspectral target detection problems. Results are presented in this section to illustrate the effectiveness of the proposed MICI algorithm for classifer fusion. In the following experiments, the rate of small-scale mutation $\eta = 0.8$. The variance $\sigma^2 = 0.1$. The measure population size $P = 50$.

A. Simulated 3-Source Classification Data Set

A simulated classification data set was constructed with 1000 data points. Three individual classifiers were applied to the data set and produced classifier output of either "1" or "0" for each data point. Figure 1a to 1c show the scatter plot of the three classifer outputs for all the data points. The data points are plotted on the 2-D plane only for visualization, with the color indicating the classfier output source value.



(a) Source 1 **(b)** Source 2 **(c)** Source 3

(d) True Labels **(e)** MICI Fusion Result

Fig. 1: Simulated 3-source dataset and results. (a) classifier 1 output; (b) classifier 2 output; (c) classifier 3 output; (d) true labels; (e) MICI fusion result. The 1000 data points are scatter plotted in the 2-D space for visualization. The colors indicate the classification label for each data point according to the colorbar shown in Figure 2, bright yellow means "1" and deep blue means "0".

These data points are then grouped into 50 bags with 20 points per bag. Each positive bag contains 25% positive points and 75% negative points. MICI is used to perform classifier fusion of the three classifiers (three sources).

Figure 1d shows the desired true labels. As can be seen in Figure 1, source 1 and source 2 are useful in producing the true labels whereas source 3 does not provide useful information.

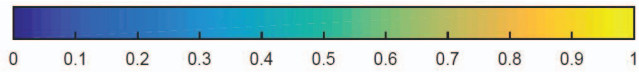


Fig. 2: Colorbar used in Sections IV.

The learned measure is presented in Table II. MICI successfully learns the interaction between the sources, i.e., putting high measure value on the intersection of source 1 and source 2 (g_{12}) and lower values otherwise. The proposed algorithm achieves 100% classification accuracy after fusing the three sources, consistently over multiple runs.

TABLE II: Mean and standard deviation of estimated and true measure element values learned for simulated 3-source classification data set over three runs.

	$\mathbf{g_1}$	$\mathbf{g_2}$	$\mathbf{g_3}$
True	0	0	0
Learned	0.065 ± 0.038	0.046 ± 0.046	0.096 ± 0.004
	$\mathbf{g_{12}}$	$\mathbf{g_{13}}$	$\mathbf{g_{23}}$
True	1	0	0
Learned	0.998 ± 0.001	0.117 ± 0.011	0.110 ± 0.008

B. Simulated Lane-Based Target Detection Data Set

A simulated 5-source classification data set is constructed based on the following scenario: five detector results were obtained on a target detection data set and MICI was used to perform fusion on the five detector outputs. Each detector result is one source for MICI. There are a total of ten true targets in the $120m \times 40m$ scene, where 1 pixel has a resolution of $1m$ in both directions. Five targets are metal and five targets are plastic. Among the five detectors, Detectors No. 1 and No. 2 can detect metal well and Detectors No. 3 and No. 4 can detect plastic well. Detector No. 5, however, performs very poorly on the targets and triggers high confidence values on non-target locations. There are 5 clutter (non-target) objects detected by each detector. The targets and clutter objects are generated with Gaussian filters with mean intensity value of 0.9. Figure 3 shows one data set example. Each $120m \times 40m$ lane in Figure 3 represents one detector result. The red boxes mark the true target locations. The above generation process was repeated five times with different true target and clutter locations each time. Now, five $120 \times 40 \times 5$ data sets were generated for the experiment.

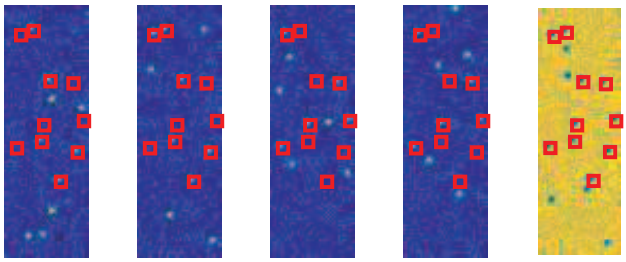


Fig. 3: One example of the simulated lane-based target detection data set. Each rectangular image represents one detector output - Detectors No. 1 to No. 5 from left to right. Each rectangular image has $120m$ (120 pixels) in the vertical direction and $40m$ (40 pixels) in the horizontal direction. The red boxes mark the true target locations.

The Reed-Xiaoli (RX) detector [28] was applied to the data set and connected component analysis was conducted on each RX detection result. RX is an anomaly detector. For each connected component found in each RX detection result, the centroid location was found. To construct training bags for the MICI algorithm, a 6×6 halo was put around each centroid obtained from the connected components. The number of training bags is the number of connected components and all the pixels within the halo size of each centroid are instances in the training bag. If any of the pixels (instances) in a training bag is within the halo of true target locations,that bag is labeled positive; Otherwise, the bag is labeled negative. Figure 4 shows one example of RX results. The squares marks the halo and all the points inside the halo belongs to a bag. The red squares form the true positive bags and the white squares are the negative bags.

The colorbar used in all images in this section is shown in Figure 2.

5-fold cross validation was conducted with four data sets as training and one as testing. One example of the measures learned after training on data sets 1, 2, 3, 4 and testing on data set 5 is shown in Table III. The MICI learns near-zero value on source 5 and relatively higher measure values on the

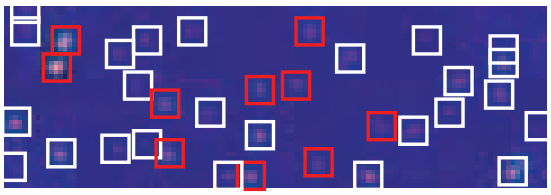


Fig. 4: RX output (plotted horizontally) of the simulated lane-based target detection data set shown in Figure 3. The squares marks the 6×6 halo and all the points inside the halo construct a bag. The red squares marks the true positive bags and the white squares are the negative bags.

combination of source 1 and 2 (g_{12}), for example. This fits the construction of the data set where detectors 1-4 provides information on both plastic and metal targets, while detector 5 does not provide positive reinforcement towards targets.

TABLE III: Mean and the standard deviation of estimated measure element values learned for simulated 5-source lane-based classification data set over five runs.

g_1	g_2	g_3
0.001 ± 0.001	0.001 ± 0.001	0.000 ± 0.000
g_4	g_5	g_{12}
0.001 ± 0.001	0.000 ± 0.000	0.009 ± 0.005
g_{13}	g_{14}	g_{15}
0.003 ± 0.001	0.004 ± 0.003	0.001 ± 0.001
g_{23}	g_{24}	g_{25}
0.007 ± 0.004	0.009 ± 0.011	0.003 ± 0.002
g_{34}	g_{35}	g_{45}
0.002 ± 0.003	0.001 ± 0.000	0.001 ± 0.001
g_{123}	g_{124}	g_{125}
0.013 ± 0.008	0.020 ± 0.014	0.010 ± 0.006
g_{134}	g_{135}	g_{145}
0.006 ± 0.002	0.004 ± 0.001	0.006 ± 0.004
g_{234}	g_{235}	g_{245}
0.017 ± 0.007	0.007 ± 0.004	0.010 ± 0.011
g_{345}	g_{1234}	g_{1235}
0.003 ± 0.004	0.050 ± 0.028	0.014 ± 0.007
g_{1245}	g_{1345}	g_{2345}
0.020 ± 0.014	0.008 ± 0.002	0.018 ± 0.007

Figure 5 shows the relationship between the fitness values versus the number of iterations. The fitness values in the first 1000 iterations over five runs are plotted. As can be seen in Figure 5, the fitness values in the first 300 iterations over five runs are increasing rapidly, and the fitness values are increasing more slowly beyond the first few hundred iterations.

Table IV shows the positive detection (PD) vs. false alarm rate (FAR) results after the five-fold cross validation across five runs. MICI algorithm is able to achieve 92.8% positive detection (PD) rate when $FAR = 0.002$. and continues to increase to near 100% positive detection with $FAR = 0.004$.

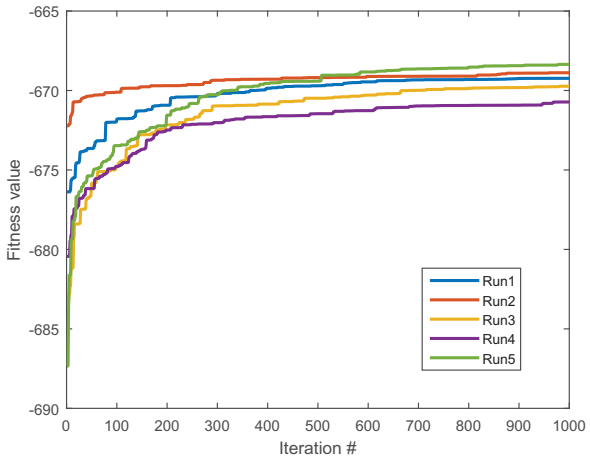


Fig. 5: Relationship of fitness values vs. number of iterations. The results in the first 1000 iterations are shown over five runs.

C. MUUFL Gulfport Hyperspectral Data Set

MUUFL Gulfport hyperspectral data set [29] was collected over the campus of University of Southern Mississippi-Gulfpark. The data set used in this experiment consists of two hyperspectral data cubes collected on two separate flights at an altitude of 3500' over the campus area. Both cubes have a ground sample distance of 1m.[1] The image from flight 1 is 325×337 pixels in size. The image from flight 2 is 329×345 pixels in size. Both data cubes contain 72 bands corresponding to wavelengths 367.7nm to 1043.4nm and were collected using the CASI hyperspectral camera [29, 30].

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), fifteen pea green, and three vineyard green. Figure 6 shows the RGB images of the two flights over the campus and the true target locations. These panels 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. The device has accuracy up to 5m. Thus, the groundtruth locations for each target are only accurate within a 5×5 pixel halo. Given the inaccuracy in the groundtruth, a multiple instance learning approach that can address this is required.

The adaptive coherence estimator (ACE) detector [31–33] 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

[1]The data set is available at http://engineers.missouri.edu/zarea/. Flight 1 corresponds to "muufl_gulfport_campus_w_lidar_1.mat", Flight 2 corresponds to "muufl_gulfport_campus_3.mat".

[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 positive detection and false alarm rate of the simulated lane-based target detection data set after five-fold cross validation across five runs.

		FAR ($/m^2$)						
		5×10^{-5}	1×10^{-4}	4×10^{-4}	1×10^{-3}	2×10^{-3}	4×10^{-3}	4.2×10^{-3}
PD (%)	Run1	18.0	26.0	54.0	70.0	94.0	100.0	100.0
	Run2	18.0	26.0	50.0	70.0	92.0	100.0	100.0
	Run3	20.0	24.0	54.0	70.0	92.0	98.0	100.0
	Run4	0.0	24.0	50.0	68.0	92.0	98.0	100.0
	Run5	18.0	26.0	50.0	68.0	94.0	98.0	100.0
	Total	14.8 ± 8.3	25.2 ± 1.1	51.6 ± 2.2	69.2 ± 1.1	92.8 ± 1.1	98.8 ± 1.1	100.0 ± 0.0

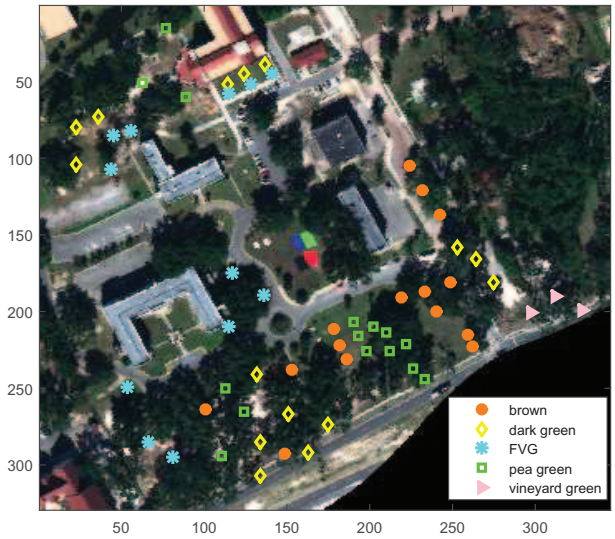


Fig. 6: The RGB image from Flight 2. Orange circle marks the true brown target locations, yellow diamond marks the true dark green target locations, cyan asterisk marks the true FVG target locations, green square marks the true pea green target locations, and pink triangle marks the true vineyard green target locations.

using the global mean and covariance of all the pixels in each image. Note that the signed ACE detector used yields confidence values between $[-1, 1]$. Therefore, all ACE results used in this experiment were normalized to be between zero and one by adding 1 to the original ACE results and divide by 2. All target signature and data pixel values were normalized to have norm 1.

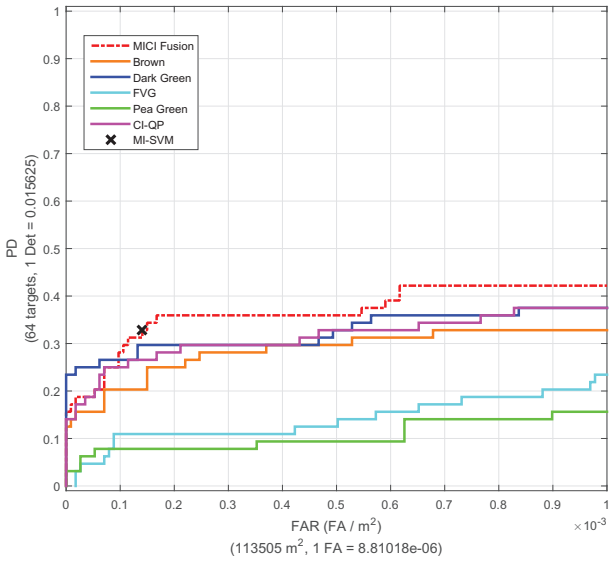
Two-fold cross validation is performed on this data set, i.e., training on Flight 1 and testing on Flight 2, and vice versa. First, the mean and covariance of the training imagery, μ_{tr} and cov_{tr}, were computed. Then, the ACE detection map for the training imagery, ACE_{tr}, was obtained using μ_{tr} and cov_{tr}. Each pixel in the detection map has four dimensions, each dimension corresponds to ACE confidence values for four target types. Each of the ACE results highlights different locations corresponding to different targets. The four dimensions of the ACE results will be the sources for the MICI to fuse.

To construct training bags for MICI, a 5×5 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". MICI was applied to the training data and a fuzzy measure $\mathbf{g}^*$ is learned.

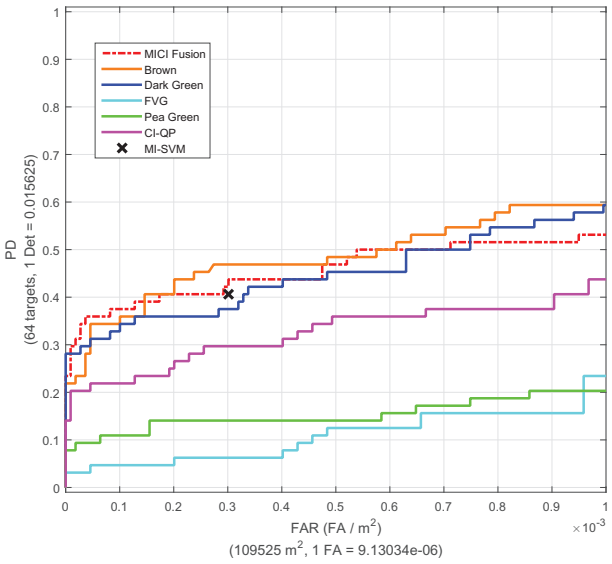
The ACE detection results for the test imagery, ACE_{te}, were computed using training mean and covariance μ_{tr} and cov_{tr}. Then, the Choquet integral given the learned measure $\mathbf{g}^*$ and ACE_{te} was computed. The Choquet integral result is the MICI fusion result.

The MICI fusion result is compared with the four ACE results using the four target signatures described above. The results are scored over all the targets in the scene with known groundtruth. The fusion result is also compared with two approaches: CI-QP [16] and MI-SVM [34]. CI-QP 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". MI-SVM is a Support Vector Machine classification approach with an extension for Multiple Instance Learning. Note that MI-SVM only yields "0" and "1" classification results as opposed to a confidence map. Therefore, a ROC curve for MI-SVM cannot be generated and the result for MI-SVM is shown as a single "x" in Figure 7. Figure 7 shows the ROC curve result of the two-fold cross validation.

The ROC curve plots the positive detection rate (PD, Y-axis) against the false alarm rate (FAR, X-axis). We only consider FAR up to $1 \times 10^{-3}/m^2$ (corresponding to a reasonable scale of 1 false alarm in $1000m^2$). As can be seen Figure 7a, for example, MICI achieves positive detection rate of 32.81% at $FAR = 1.41 \times 10^{-4}/m^2$, which is better than CI-QP approach (26.56%) and all the individual sources (25.00%, 26.69%, 10.94% and 7.81% for brown, dark green, FVG and pea green targets, respectively). MICI achieves similar PD with MI-SVM at $FAR = 1.41 \times 10^{-4}/m^2$. Similar results are observed in the second fold shown in Figure 7b, however, in this case the MICI outperforms both the CI-QP and the MI-SVM.



(a)



(b)

Fig. 7: ROC curve results for the two-fold cross validation on MUUFL Gulfport data when (a) training on Flight 1 and testing on Flight 2 (b) training on Flight 2 and testing on Flight 1. The red line marks the ROC curve after MICI fusion, the orange line marks brown target scoring result, the blue line marks dark green target scoring result, the cyan line marks FVG target scoring result, and the green line marks pea green target scoring result. The magenta line marks the CI-QP scoring result. The black "x" marks the MI-SVM scoring result. X-axis: False Alarm Rate (FAR) between $[0, 0.001]$. Y-axis: Positive Detection (PD).

V. Conclusion

This paper proposes a Multiple Instance Choquet Integral (MICI) algorithm for classifier fusion. MICI learns a fuzzy measure via an evolutionary algorithm based on uncertain labels from the training data. MICI then computes the Choquet integral with the learned measure to provide classification result on testing data. Results on both simulated and the MUUFL Gulfport hyperspectral data show that the proposed algorithm is able to perform fusion on multiple classifier inputs and yield competitive classification result.

It would be interesting to try alternative initialization of the fuzzy measure or vary the optimization scheme in the evolutionary algorithm and observe the change in the learning performance of the algorithm. The order of updating the measure elements in the current small-scale and large-scale mutations are dependent on the valid interval widths, but it would be possible to follow other sort orders such as sorting by fitness values.

The proposed MICI may be able to be extended towards regression applications in addition to the classifier fusion applications provided by this paper.

References

[1] J. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer, 1999.

[2] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.

[3] S. Auephanwiriyakul, J. Keller, and P. D. Gader, "Generalized choquet fuzzy integral fusion," *Information Fusion*, vol. 3, no. 1, pp. 69 – 85, 2002.

[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] F. Tabib Mahmoudi, F. Samadzadegan, and P. Reinartz, "Object recognition based on the context aware decision-level fusion in multiviews imagery," *IEEE J. Sel. Topics. Appl. Earth Observ.*, vol. 8, no. 1, pp. 12–22, Jan 2015.

[8] J. M. Keller, S. Auephanwiriyakul, and P. D. Gader, "New fuzzy set tools to aid in predictive sensor fusion," in *Proc. SPIE, Detection and Remediation Technologies for Mines and Minelike Targets V*, vol. 4038, Aug. 2000, pp. 1497–1507.

[9] J. Keller, S. Auephanwiriyakul, and P. Gader, "Experiments in predictive sensor fusion," in *Proc. SPIE, Detection and Remediation Technologies for Mines and*

Minelike Targets VI, vol. 4394, Oct. 2001, pp. 1047–1058.

[10] A. H. Gunatilaka, B. Baertlein *et al.*, "Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 577–589, 2001.

[11] 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.

[12] P. Gader, M. Mohamed, and J. Keller, "Fusion of handwritten word classifiers," *Pattern Recog. Lett.*, vol. 17, no. 6, pp. 577–584, 1996.

[13] Z. Liangbin, M. Wenjun, C. Li, and Z. Su, "Research on detection of prostate cancer mr images based on information fusion," in *12th Int. Conf. Signal Proc. (ICSP)*, Oct 2014, pp. 1094–1098.

[14] S. Khazendar, H. Al-Assam, H. Du, S. Jassim, A. Sayasneh, T. Bourne, J. Kaijser, and D. Timmerman, "Automated classification of static ultrasound images of ovarian tumours based on decision level fusion," in *6th Comp. Sci. Electron. Eng. Conf. (CEEC)*, Sept 2014, pp. 148–153.

[15] C. Zhang, Z.-Z. Yang, Z.-B. Liu, and D. Chen, "Research on intelligent fault diagnosis method for complex equipment based on decision-level fusion," in *Int. Conf. Mach. Learn. Cybernetics (ICMLC)*, vol. 1, July 2010, pp. 386–390.

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

[17] M.Grabisch, *Modelling data by the Choquet integral*. Springer Berlin Heidelberg, 2003, vol. 123.

[18] 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.

[19] A. Mendez-Vazquez, P. Gader, J. 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.

[20] D. Anderson, J. Keller, and T. 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.

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

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

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

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

[25] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intell.*, vol. 89, no. 1-2, pp. 31–71, Jan. 1997.

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

[27] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed. Wiley-Interscience, Oct. 1994, vol. 1.

[28] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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