Genetic Programming Based Choquet Integral for Multi-Source Fusion

Ryan E. Smith[1], *Student Member, IEEE,* Derek T. Anderson[1], *Senior Member, IEEE,* Alina Zare[2], *Senior Member, IEEE,* John E. Ball[1], *Senior Member, IEEE,* Brendan Alvey[3], *Student Member, IEEE*, Josh R Fairley[4], and Stacy E Howington[4]
[1]Electrical and Computer Engineering, Mississippi State University, MS, USA,
e-mail: *rs1045@msstate.edu,*{*anderson,jeball*}*@ece.msstate.edu*,
[2]Electrical and Computer Engineering, University of Florida, FL, USA,
e-mail: *azare@ece.ufl.edu*,
[3]Electrical and Computer Engineering, University of Missouri, MO, USA,
e-mail: *bja3md@mail.missouri.edu*,
[4]U.S. Army Engineer Research & Development Center, Geotechnical & Structures Laboratory, Vicksburg, USA,
e-mail: {*Josh.R.Fairley, Stacy.E.Howington*}*@usace.army.mil*.

***Abstract*—While the *Choquet integral* (ChI) is a powerful parametric nonlinear aggregation function, it has limited scope and is not a universal function generator. Herein, we focus on a class of problems that are outside the scope of a single ChI. Namely, we are interested in tasks where different subsets of inputs require different ChIs. Herein, a *genetic program* (GP) is used to extend the ChI, referred to as GpChI hereafter, specifically in terms of compositions of ChIs and/or arithmetic combinations of ChIs. An algorithm is put forth to learn the different GP ChIs via *genetic algorithm* (GA) optimization. Synthetic experiments demonstrate GpChI in a controlled fashion, i.e., we know the answer and can compare what is learned to the truth. Real-world experiments are also provided for the mult-sensor fusion of *electromagnetic induction* (EMI) and *ground penetrating radar* (GPR) for explosive hazard detection. Our mutli-sensor fusion experiments show that there is utility in changing aggregation strategy per different subsets of inputs (sensors or algorithms) and fusing those results.**

Index terms - Choquet integral, fuzzy integral, genetic program, genetic algorithm, multi-sensor fusion

I. Introduction

At the heart of numerous state-of-the-art challenges like Big Data, remote sensing, multi-sensor systems, computer vision, and machine learning, to name a few, is data and information fusion (referred to hereafter as data unless there is a reason to differentiate). However, the term “fusion” is a rather abstract and overloaded concept. In this paper, we focus on one aspect of fusion, aggregation. Generally, an aggregation operator is a functional mapping of N inputs to a single output. The hope is that by combining the N inputs we can achieve a “better” result than the inputs by themselves, where better is application specific, e.g., improved visualization, quality summarization, higher accuracy in machine learning, etc. Most often, a single aggregation operator is applied *uniformly* across all inputs. However, there are a class of problems that require us to use different aggregation strategies for different subsets of inputs (and ultimately their fusion). Herein, we investigate a new extension to the *Choquet integral* (ChI), a parametric aggregation function. Specifically, we explore a *genetic program* (GP) extension of the ChI, referred to as GpChI hereafter, that allows us to tailor different compositions and arithmetic combinations of ChIs to different subsets of inputs.

To further motivate the need for and utility of GpChI, we consider the multi-sensor challenge of *explosive hazard detection* (EHD). A serious threat to civilians and soldiers is buried and above ground EHs. Since 2008, these threats in Afghanistan alone are responsible for wounding or killing approximately 10,000 U.S. soldiers. Across the globe, on average there are 310 deaths and 833 wounded individuals per month [1]. The automatic detection of such threats is highly desired. Many approaches exist, e.g., *hand-held* (HH) based sensors, downward and forward looking vehicle mounted platforms, etc. Due to the complexity of EHD, e.g., environmental factors, target and emplacement variation, clutter, etc., multiple regions in the *electromagnetic* (EM) spectrum are usually required to achieve robust and accurate results. Herein, we demonstrate that the proposed GpChI provides better results for the fusion of HH-based *electromagnetic induction* (EMI), *ground penetrating radar* (GPR), and precise positioning sensors. Specifically, different ChIs are used within and across sensors.

The ChI, or more generally the *fuzzy integral* (FI), has seen great success in numerous areas from *multi-criteria decision making* (MCDM) to image processing, robotics, remote sensing, crowd sourcing, multi-sensor fusion, computer vision, machine learning, to name a few. We are not advocating that the ChI is of little-to-no use. Instead, we are highlighting the fact that the ChI has its place and limitations, like any function. While we consider a GP-based extension, others have extended the ChI for various purposes. Examples include, the k-additive ChI [2, 3], interval-valued ChI and set-valued ChI [4, 5], multiple instance learning ChI [6], type-2 fuzzy set ChI [7], regularization based ChI [8], multiple kernel learning based ChI [9–14], ChI for regularization, etc. In general, ChI extensions target the integrand (input representation), the *fuzzy* *measure* (FM) (representation of the worth of different input subsets) and/or the mechanics of the integral (e.g., constraints, variations in the operators, etc.).

To the best of our knowledge, there has been limited work on the aggregation of ChIs. Of that work, most has been purely mathematical in nature and/or applied to topics like multicriteria decision support. In [15], Mesiar et al. discussed the aggregation of ChIs via weighted arithmetic means and *ordered weighted averages* (OWAs) – or *linear combinations of order statistics* (LCOS), a well-known subset of the ChI.

Herein, we put forth the following four specific contributions. First, we introduce a GP extension to the ChI. Second, we introduce a hybrid GP-GA solver, where the GA is optimizing the ChIs within a single GP. Third, we study a few important mathematical outcomes of the impact of GpChI. Fourth, we demonstrate results for synthetic and a real data set from multi-sensor fusion to show the effectiveness of GpChI.

The remainder of the paper is organized as such. In Section II, the ChI, GP, GA and GpChI are discussed. Section III is properties of GpChI. Section IV discusses synthetic and real-world EMI-GPR multi-sensor fusion experiments.

II. Methods

A. Choquet Integral

As discussed in Section I, the ChI is a flexible and parametric aggregation operator. The ChI is defined with respect to the FM. Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of N inputs from sources like experts, algorithms and/or sensors. A FM is a monotonic set-valued function defined on the power set of X, 2^X, as $\mu : 2^X \to \mathbb{R}^+$ that satisfies the following two properties: (i) Boundary condition, $\mu(\emptyset) = 0$, and (ii) monotonicity, if $A, B \subseteq X$ and $A \subseteq B$, $\mu(A) \leq \mu(B)$. Often an additional constraint is imposed on the FM to limit the upper bound to 1, i.e., $\mu(X) = 1$. Throughout this paper, without loss of generality, we consider this condition for simplicity and convenience.

Let $h(x_i)$ (or h_i) be the information (beliefs, sensor measurements, etc.) provided by input i. The discrete ChI is

$$\int_C h \circ \mu = C_\mu(h) = \sum_{i=1}^{n} h(x_{\pi(i)}) \left[\mu(A_i) - \mu(A_{i-1})\right], \quad (1)$$

where π is a permutation of X, such that $h(x_{\pi(1)}) \geq h(x_{\pi(2)}) \geq \ldots \geq h(x_{\pi(n)})$, $A_i = \{x_{\pi(1)}, \ldots, x_{\pi(i)}\}$, and $g(A_0) = 0$. Based on selection of μ, the ChI turns into a specific aggregation operator. For example, when $\mu(A) = 0, \forall A \in 2^X \backslash X$, the ChI is equivalent to the minimum operator. When $\mu(A) = 1, \forall A \in 2^X \setminus \emptyset$, we obtain the maximum operator. More generally, when $\mu(A) = \mu(B) \forall A, B \in 2^X$ such that $|A| = |B|$, we recover the familiar class of *linear combinations of order statistics* (LCOS), e.g., min, max, soft min and max, mean, median, trimmed statistics, etc.

Before we delve into detail about how to extend the ChI, we first quickly review where the FM comes from. The FM can be specified by an expert, however since there are 2^N terms, this becomes intractable quickly. If one has training data, then a number of methods can be used to learn the FM (see [8]). However, another popular solution is to work with what we can obtain and to approximate the rest (under a set of assumptions). Often, the measure on the singletons, $\mu(x_i)$, are referred to as densities. If the densities are known or can be learned, then an “imputation function” can be used to fill in the remainder of the FM. A famous imputation function is Sugeno’s λ-FM [16]. The Sugeno λ-FM finds a unique solution for λ based on the densities and Sugeno’s famous characteristic polynomial function. Other well-known imputation functions are the S-Decomposable FM, of which S equal to the maximum operator is a pessimistic propagation of values up the FM lattice. The point is, a FM can be specified by a human, learned from data or imputed.

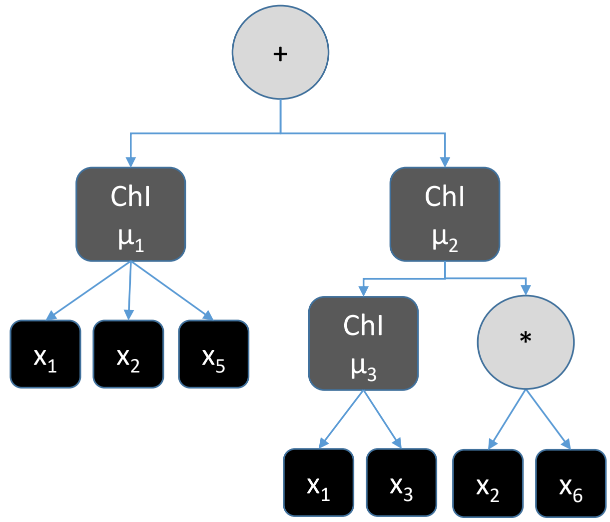

Fig. 1: Example of a GP-based ChI (GpChI) for $N = 6$ inputs. Note, we can have an arithmetic combination of inputs, ChIs or composition of ChIs. In addition to aggregation, the GpChI also performs input (feature) selection. The GpChI formula for this tree is $C_{\mu_1}(\{h_1, h_2, h_5\}) + C_{\mu_2}(\{C_{\mu_3}(h_1, h_3), h_2 * h_6\})$.

B. Genetic Programming-Based ChI

A GP is a way to represent and subsequently optimize a “program”, which can be a computer program, image or signal processing filter, mathematical formula, to name a few. In terms of representation, a GP is generally a tree data structure. In terms of optimization, a GP is a random search process guided by heuristics which mimics nature. Herein, we explore the GP as one way to extend the ChI, namely with respect to allowing compositions of ChIs and/or arithmetic combinations of ChIs on potentially different subsets of inputs. Figure (1) is an example GP ChI extension, hereafter called GpChI. In the remainder of this section we outline the GpChI algorithm (summarized in Algorithm 1).

Terminals and non-terminals: Our terminal set are the inputs, i.e., $\{x_1, ..., x_N\}$ and their corresponding values $\{h(x_1), ..., h(x_N)\}$. While it is possible to allow for any arbitrary constant, we do not herein (expanded on later). The non-terminal set are functions that operate on our terminals and/or the non-terminals. Herein, we restrict the non-terminal

Algorithm 1 High-level GpChI algorithm description
Note: steps in bold have been extended in this research

```
Initialize population of P individuals
Set iteration counter to zero, t = 0
while (not converged) and (t < T_max) do
    For each individual, optimize its ChIs using a GA
    Evaluate fitness of each individual
    Select next generation of individuals
    Perform crossover
    Perform mutation
    t = t + 1
end while
```

set to the ChI and elementary arithmetic operators; addition, subtraction, multiplication, and division.

Fitness function: In order to select individuals for the next generation, a fitness function is required. Since Algorithm 1 is a supervised learning problem, let D_{tr} be a set of training data, D_h is data held back for scoring during training and D_{te} is the test data. For each GpChI candidate solution, Θ_k, trained on D_{tr}, its *minimum squared error* (MSE) fitness is

$$E_1(D_h, \Theta_k) = \frac{1}{|D_h|} \sum_{i=1}^{|D_h|} \left(f_{\Theta_k}(o_i) - y_i\right)^2, \qquad (2)$$

where o_i is object i from the held back data set, $f_{\Theta_k}(o_i)$ is the kth GpChI $\Re$-valued output for object i, and y_i is the "label" ($\Re$-valued output) for object i.

Crossover: After selection, the genetic makeup from two parents (individuals) are swapped. Crossover promotes exploitation in a GP. The idea is to randomly select a subtree from parent one and two swap that sub-solution with a subtree from parent 2. Figure (2) illustrates GP crossover.

Mutation: Whereas crossover promotes exploitation, mutation is exploration. Mutation is the random alteration of values in a GP. Herein, we support the following mutation operators, based on they type of node:

- **Terminal node**
  - Flip its input index.
- **Non-terminal and non-ChI node**
  - (swap) replace with equivalent operator, e.g., binary multiplication for addition operator.
  - (prune) remove the subtree it represents and replace with input if necessary.
  - (grow) replace the subtree with a new random tree.
- **Non-terminal ChI node**
  - (variable flip) change a FM variable value – subject to monotonicity constraints.
  - (add inputs) add an input to the ChI
  - (remove inputs) remove an input to the ChI

Note, not all operators are applied to each individual at each iteration. Instead, an overall GP mutation rate (probability) is determined and sub-probabilities are given to each of the above operators, which sum to one. Thus, for each individual selected for mutation, one of the above operators is randomly selected and applied. The reader can refer to the literature for additional details surrounding general determination of GP mutation operator probability assignment. However, we do note that we adhered to the philosophy that the prune rate should be higher than the grow rate. Otherwise, one runs the risk of growing larger and larger GpChIs, which results in overly complex, and likely overfit, solutions that are also computationally taxing. Note, we use held back data to help mitigate overfitting. In future work we will explore adding a model complexity term to our fitness function.

Selection: In general, there are numerous schemes for selection, e.g., proportional, tournament, etc. Furthermore, no single method has been shown, theoretically or empirically, to be superior to all other methods. Herein, without loss of generality, we use proportional selection. In addition, we make use of elitism – meaning at each iteration the top (most fit) individuals are automatically passed onto the next generation "as is". Elitism is a useful strategy to ensure that we do not digress from iteration to iteration.

C. Genetic Algorithm for Learning a GpChI Individual

At each iteration, if a GpChI individual has any ChIs in it then we must learn the associated underlying FM(s). In general, there are numerous ways to learn the FM/ChI from data, each of which with associated benefits and drawbacks. The reader can refer to [8] for a recent method and review of related literature. Again, without loss of generality, we use a *genetic algorithm* (GA) to learn the FM(s)/ChI(s) [17]. The idea herein is to alternate between a GP for evolving GpChI candidate and a GA for optimizing each candidate.

III. Analysis of GpChI Properties

In this section we investigate mathematical consequences of the GpChI versus a single ChI. This is important because it informs us what we can expect from the GpChI or relative to restrictions that we might place on the GpChI algorithm.

Proposition 1: **(ChI with respect to the LCS of FMs is Equal to the LCS of ChIs)** For M ChIs, $\{C_{\mu_1}, ..., C_{\mu_M}\}$,

$$C_{\mu_*}(h) = LCS_w(C_{\mu_1}(h), C_{\mu_2}(h), ..., C_{\mu_M}(h)), \qquad (3)$$

where μ_* is the *linear convex sum* (LCS) of FMs, i.e., $\mu_* = w_1\mu_1 + ... + w_M\mu_M$, $w_i \in [0,1]$ and $\sum_{i=1}^{M} w_i = 1$.

Proof: Without loss of generality, let $M = 2$. First, we expand the RHS of Equation (3),

$$LCS_w(C_{\mu_1}(h), C_{\mu_2}(h)) = w_1\left(a_1^1 h_{\pi(1)} + a_2^1 h_{\pi(2)}\right) + w_2\left(a_1^2 h_{\pi(1)} + a_2^2 h_{\pi(2)}\right),$$

where $a_j^k = \mu_k(A_j) - \mu_k(A_{j-1})$ and $h_{\pi(i)}$ is shorthand for $h(x_{\pi(i)})$. Grouping similar terms, we obtain

$$h_{\pi(1)}\left(z_1 = \left[w_1 a_1^1 + w_2 a_1^2\right]\right) + h_{\pi(2)}\left(z_2 = \left[w_1 a_2^1 + w_2 a_2^2\right]\right).$$

Next, if we expand the LHS of Equation (3) we obtain

$$C_{\mu_*}(h) = b_1 h_{\pi(1)} + b_2 h_{\pi(2)},$$

where $b_j = \mu_*(A_j) - \mu_*(A_{j-1})$. The proof resides in analysis of b_j relative to z_j. Without loss of generality, we can expand

$$b_1 = \mu_*(A_1) = w_1\mu_1(A_1) + w_2\mu_2(A_1) = z_1,$$

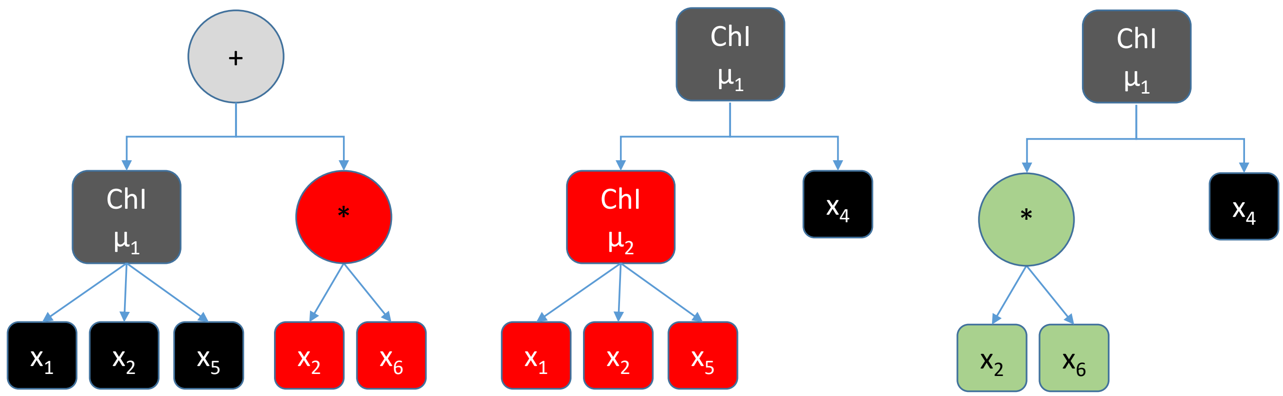

Fig. 2: Example of GpChI crossover – (left) tree one and (middle) tree two. Selected sub-trees are shown in red. (right) The result of crossover – crossed sub-solution is shown in green.

Furthermore,

$$b_j = \mu_*(A_j) - \mu_*(A_{j-1}) = (w_1\mu_1(A_j) + w_2\mu_2(A_j))$$

$$-(w_1\mu_1(A_{j-1}) + w_2\mu_2(A_{j-1})) = w_1(\mu_1(A_j) - \mu_1(A_{j-1}))$$

$$+w_2(\mu_2(A_j) - \mu_2(A_{j-1})) = z_j.$$

The above proof is relative to $M = 2$ for analytical tractability. However, these properties obviously hold for any M. ∎

Proposition 2: **(LCS of FMs is a FM)** The LCS of M FMs, $\mu_* = w_1\mu_1 + ... + w_M\mu_M$, is a FM, where $w_i \geq 0$, $\sum_{i=1}^{M} w_i = 1$, $w_i\mu_i$ is element-wise scalar multiplication, i.e., $w_i\mu_i(A)\ \forall A \in 2^X$, and $\mu_i + \mu_j$ is element-wise addition, i.e., $\mu_i(A) + \mu_j(A)$, $\forall A \in 2^X$.

Proof: First, $\mu_*(\emptyset) = 0$ and $\mu_*(X) = 1$ (FM boundary conditions) due to the LCSs idempotency property, i.e.,

$$LCS_w(a) = w_1 a + ... + w_M a = a\left[\left(\sum_{i=1}^{M} w_i\right) = 1\right] = a.$$

Next, it is trivial to prove that the LCS is bounded, $\bigwedge_i a_i \leq LCS_w(a) \leq \bigvee_i a_i$, and thus $\mu(A) \in [0,1]$, $\forall A \in 2^X$. Furthermore, monotonicity holds because if $c_i \geq d_i$, $\forall c_i, d_i \in \Re$,

$$LCS_w(c) \geq LCS_w(d),$$

$$w_1 c_1 + ... + w_M c_M \geq w_1 d_1 + ... + w_M d_M,$$

which after some algebraic manipulation,

$$w_1(c_1 - d_1) + ... + w_M(c_M - d_M) \geq 0,$$

as $c_i \geq d_i$, $w_i \geq 0, \sum_{i=1}^{M} w_i = 1$. In terms of monotonicity, let $c_i = \mu_i(A)$ and $d_i = \mu_i(B)$, for $A \in 2^X$ where $B = A \setminus z$ for our different $z \in A$. ∎

Remark 1. Proposition 1 tells us that, relative to LCS, we can aggregate the output of a set of ChIs or aggregate their FMs and use the resultant ChI (they are equal). Furthermore, Proposition 2 says we obtain a FM if we restrict the GpChI to LCS based aggregation of FMs. Together, these propositions inform us that the LCS of a set of ChIs is a ChI. Therefore, in the GpChI we can ensure that we obtain a ChI, if desired, by making the terminal set be inputs and scalars and the non-terminal set be multiplication, addition and ChI. However, to ensure a ChI we must monitor and guarantee that the scalars sum to one. All of these conditions can be met (via custom mutation, initialization and crossover operations).

Proposition 3: **(ChI of ChIs is Not Necessarily a ChI)** For M ChIs, $\{\mu_1, ..., \mu_M\}$, the ChI of ChIs, C_{μ_r}, is not always gaurenteed to be a ChI.

Proof: Consider the case of $M = 2$. Let μ_1 be "select the first input", i.e., $\mu_1(A) = 1$ if $x_i \in A$, zero otherwise. Next, let μ_2 be the mean, $\mu_2(A) = \frac{|A|}{N}$. Now, let C_{μ_r} be the maximum operator ($\mu_r(A) = 1$, $\forall A \in 2^X \setminus \emptyset$),

$$C_{\mu_r}(h) = \max\{C_{\mu_1}(h), C_{\mu_2}(h)\}.$$

Whereas the ChI of ChIs can model this, there is no single ChI (specifically FM) that can model this. Let $N = 3$. In order for a FM to select input one if its the largest, all measure terms except $\mu_r(\{x_2\})$, $\mu_r(\{x_3\})$ and $\mu_r(\{x_2, x_3\})$ must be one-valued–e.g., consider $h(x_1) = 0.2$, $h(x_2) = 0$, $h(x_3) = 0$. This is acceptable if $h(x_1)$ is greater than all other inputs. However, if $h(x_2) \geq h(x_1) \geq h(x_3)$, our third weight in the ChI is $\mu_r(X) - \mu_r(\{x_1, x_2\}) = 0$. We cannot compute the mean of our numbers, we need all weights to be $\frac{1}{3}$. This can be verified by $h(x_1) = 0.8$, $h(x_2) = 0.9$, $h(x_3) = 0$. ∎

Remark 2. Proposition 3 informs us that while LCS is "ChI preserving", meaning the result of aggregation is another ChI, the ChI of ChIs is not always. That is, we cannot always reduce the ChI of ChIs to a single ChI. Instead, we have obtained something outside the scope of a single ChI. Thus, the GpChI will often produce non-ChI operators.

Proposition 4: **(Boundedness of the GpChI)** Consider a set of M ChIs, $\{\mu_1, \mu_2, ..., \mu_M\}$, each defined on a subset of the avilable N inputs. A ChI, C_{μ_r}, of these ChIs is bounded by $[\bigwedge_i h(x_i), \bigvee_i h(x_i)]$.

Proof: This is easily verified by the boundedness property of the ChI, $\bigwedge_i h(x_i) \leq C_\mu(h) \leq \bigvee_i h(x_i)$, where C_μ is a ChI on all N inputs and $\wedge$ is the minimum and $\vee$ is the maximum. As each C_{μ_k} is the aggregation of a subset of our N inputs, i.e., $\bar{h}_k \subseteq h$, each are bounded by $\bigwedge_i \bar{h}_k(x_i) \leq C_{\mu_k}(h) \leq$

$\bigvee_i \bar{h}_k(x_i)$. Therefore, since each input to our ChI is bounded between the minimum and maximum of our N inputs, C_{μ_r} is obviously therefore bounded. ∎

In Proposition 4, if any C_{μ_k} has inputs that are themselves ChIs on a subset of h, or if an input is instead one of our h values, then the same logic applies, i.e., boundedness is recursively applicable.

Remark 3. Proposition 4 tells us about the range of a ChI of ChIs. Thus, if we restrict the GpChI to only ChI operators, i.e., no arithmetic operators, then we will not generate values higher or lower than the inputs.

Proposition 5: **(GpChI is Not Necessarily a ChI)** The aggregation of M ChIs, $\{\mu_1, \mu_2, ..., \mu_M\}$, with respect to operator set $\{*, -, +, /\}$ is not guaranteed to produce a ChI.

Proof: Without loss of generality, let $N = 3$, specifically $h(x_1) = 0.3$, $h(x_2) = 0.5$, and $h(x_3) = 0.9$. Also, let there be two ChIs ($M = 2$), specifically two maximum operators, i.e., $\mu(A) = 1$, $\forall A \in 2^X/\emptyset$ and $\mu(\emptyset) = 0$. Furthermore, let the GpChI be a minus root node and its two inputs are two maximum ChI operators, which results in

$$C_{\mu_1}(h) - C_{\mu_2}(h) = 0.$$

However, $\bigwedge_i h(x_i) \geq C_\mu(h) \geq \bigvee_i h(x_i)$. Clearly, 0 is outside the minimum, which is 0.3. ∎

Remark 4. Proposition 5 tells says an arithmetic combination of ChIs is not guaranteed to be a ChI. Thus, the GpChI can generate new aggregations outside the scope of the ChI.

In summary, this section informs us that the GpChI will very likely not generate a ChI unless algorithmic modifications are made, e.g., restriction to only trees that do LCS. How to interpret this is really application or context dependent.

IV. EXPERIMENTS

In this section, we experimentally explore the proposed GpChI. First, we start with synthetic data. This is important because we have the ground truth, i.e., we know the answer, and we can meaningfully evaluate the quality of the technique. Second, we apply the GpChI to a multi-sensor fusion task in explosive hazard detection for humanitarian demining.

A. Synthetic Example 1

In Experiment 1 we build a ground truth GpChI tree (shown in Figure (3)) for $N = 2$ (so we can plot the underlying surface) that has four ChIs, two minimum and two maximum ChIs, and three arithmetic operators (multiplication operators). The specific equation is

$$\left(C_{\mu_1}(h) * C_{\mu_2}(h)\right) * \left(C_{\mu_3}(h) * C_{\mu_3}(h)\right),$$

where μ_1 and μ_3 are minimum FMs and μ_2 and μ_4 are minimum FMs. The dataset used for training (shown in Figure (4)) was generated by uniformly sampling, with respect to a sampling delta of 1, a two dimensional grid of size 30 by 30 centered at $(0, 0)$. The non-terminal set for Experiment 1 is $\{+, -, *, /, \mathrm{ChI}\}$, where the ChI FM singletons have zero-value initially and are optimized with a GA.

Figure (3) tells the following story. When the ground truth capacities are minimum and maximum, the learned (middle) GpChI is a strictly polynomial result, meaning no ChIs are needed. Furthermore, the strictly polynomial solution is more succinct at that. The fact that our algorithm learns this solution is encouraging, as the ground truth and our learned solution are mathematically equivalent. This is the reason this ground truth was selected. In some cases, the underlying GpChI solution will possibly be strictly representable via a polynomial, which could be a more compact solution or an answer not expressible via a ChI or combination of ChIs. However, this is a two way road. Meaning, it is possible that the ground truth (underlying solution) cannot be expressed in terms of the non-terminal set if the ChI is removed. The non-terminal set would need to be expanded to include the range of the ChI. However, together they represent a small non-terminal set that can represent a wide variety of different functions.

B. Synthetic Example 2

In Experiment 2 we changed the capacities from maximum and minimum to two other LCOSs, "soft min" ($\mu(A) = 0.2$, $A \subset X, |A| = 1$) and "soft max" ($\mu(A) = 0.8$, $A \subset X, |A| = 1$) accordingly. The reason we selected these capacities is because we wanted to induce a nonlinear function which requires us to learn ChIs. The right GpChI in Figure (3) is the learned solution. This result highlights a well-known property of the GP. Whereas the learned solution yields the same output as our ground truth, a desirable property, they are not the exact same trees. In this initial work we have not included any mechanisms to learn low complexity solutions, i.e., smallest possible representable trees. This is often addressed via extending the cost function (adding a penalty term). The risk that we run when using a GP, even one that tries to learn low complexity solutions, is like most machine learning algorithms. In general, the longer, i.e., more complex, the tree the more overfit the solution. Meaning, it does wonderful on training data but fails to hold up as well on new test data.

C. Real World Data Set: Humanitarian Demining

In order to demonstrate that GpChI works outside of hand crafted experiments, we explore its use in the real world multi-sensor signal processing problem of buried explosive hazard detection. Specifically, we investigate a task in which there are two sensors, *electromagnetic induction* (EMI) and *ground penetrating radar* (GPR), on a *hand-held* (HH) platform. The platform used is called the *experimental HH demonstrator* (EHHD) and it is provided by the US Army RDECOM CERDEC NVESD.

In Experiment 3, two prescreeners–algorithms that identify alarms and return an associated *confidence* regarding explosive or not–is ran on EMI and two different prescreeners are ran on GPR. Specifics regarding these individual algorithms, the platform and data set can be found in [18, 19]. The GpChI is used here for decision-level fusion, as it is combining the $\Re$-valued output of prescreeners. For sake of completeness, we quickly summarize this experiment and data set. A total of six

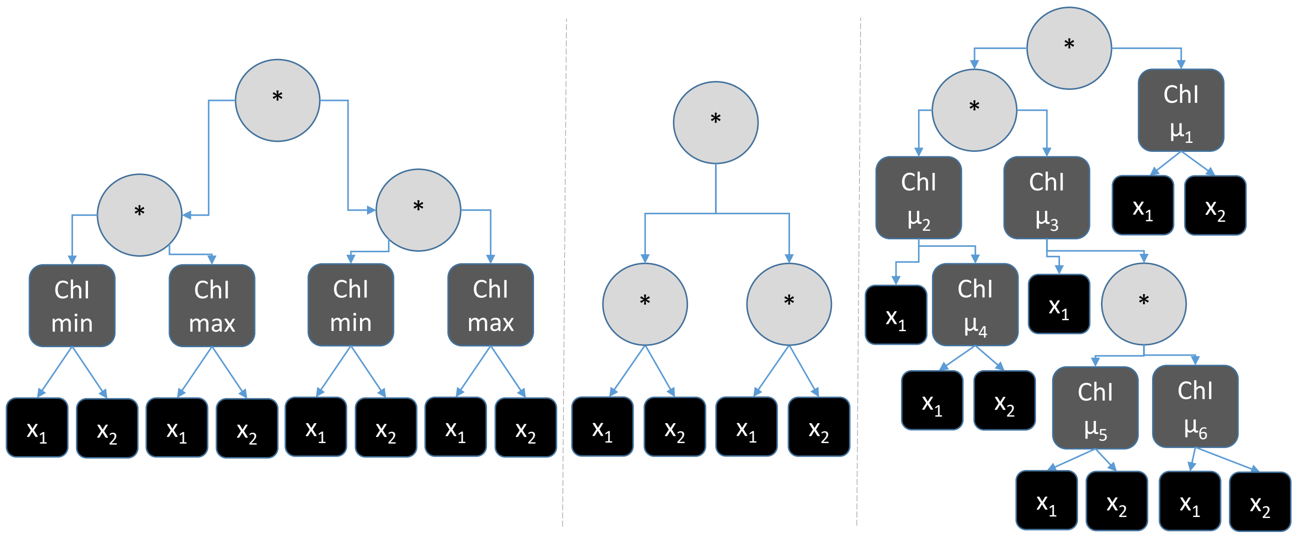

Fig. 3: (left) Ground truth (hand crafted) GpChI, (middle) learned GpChI solution with no ChI operators and (right) learned GpChI solution with ChI operators.

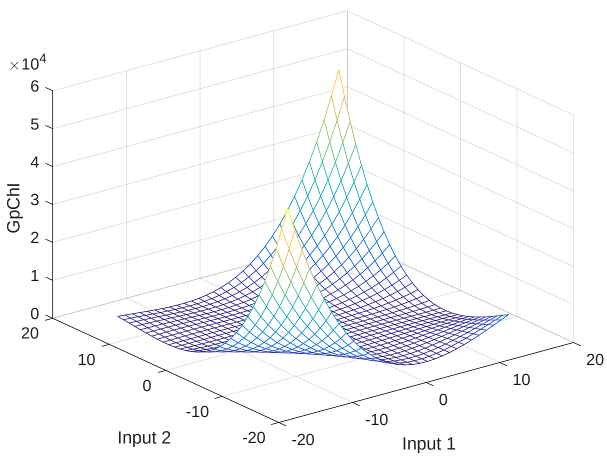

Fig. 4: Solution surface for Experiment 1 ground truth.

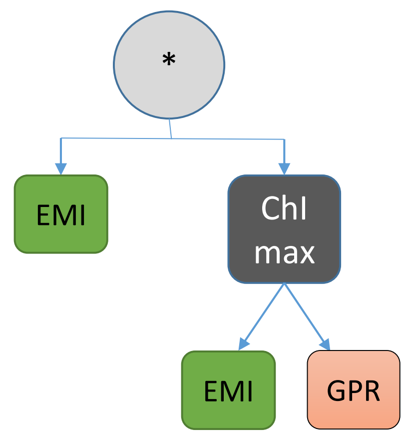

Fig. 5: GpChI EMI-GPR fusion solution learned.

runs were collected. Each run is a different lane with various metallic and non-metallic explosive hazards of varying size, materials and burial depths. In total, there are seventy targets in the six runs. The data sets were collected independently for the two sensors and then registered to created the input data. Two scripted platforms with precise positioning were used to scan the lanes. As such, the exact sweep pattern and region of coverage of the two sensors are not the same. In order to fuse these two sensors, we re-sampled (see [20] for full details) the output of four prescreeners ran on these sensors over a common area of interest at a sample distance rate of 1cm. At each point on this common sampled surface, we applied the GpChI. From that geo-spatial result, a simple extrema finder was used to generate a final set of alarms.

There is not enough data to run a comprehensive set of cross validation experiments. In part, this is due to the complexity and cost of collecting such data using two different sensors under development at a US Army test site. Each collection takes one to more weeks of teams of experts running hours each day. As such, we picked one run at random to train on and then we tested on all six runs. This is not technically resubstitution as we did not train and test on all six runs. One run will likely benefit, but the other five different runs are independent. Furthermore, as we show in a minute, the results obtained by training on just one run are positive. In the future, if/when more data is collected, then we will run additional experiments and build more comprehensive results. Regardless, the experimental results presented here are encouraging. Figure (5) shows the GpChI solution learned and Figure (6) is the quantitative result.

In Figure (6), we report *receiver operating characteristic* (ROC) curve performance. We report *positive detection rate* (PDR) bars, where each *false alarm rate* (FAR) is treated as a binomial experiment and the PDR is treated as probability of success. The 95% binomial confidence interval for each PDR

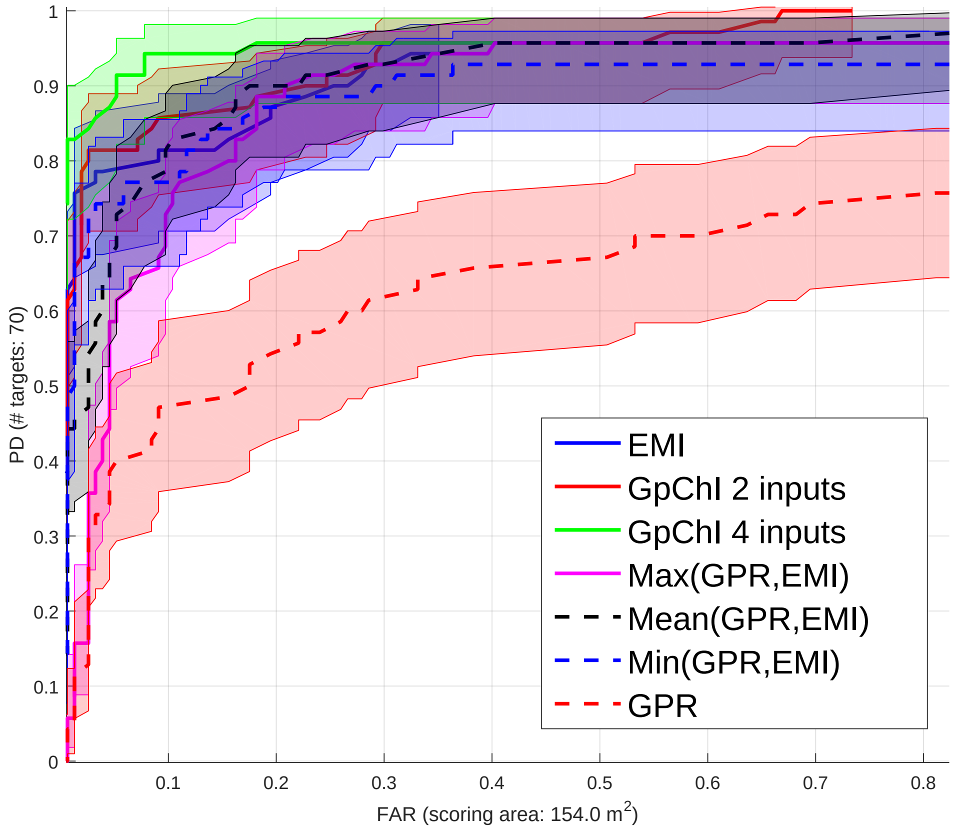

Fig. 6: ROC curve result for Experiment 3; x-axis is false alarms per meter squared and y-axis is positive detection rate.

is generated and its curve it plotted. In general, a ROC shows the PDR/FAR behavior of an algorithm with respect to some parameter. Here, that parameter is the output of the GpChI. The x-axis is the number of mistakes, described in terms of FAs per meter squared. The y-axis is the PDR. Ideally, we would like a 0 FAR and 1 PDR curve. However, in reality, as we change our threshold–the binary decision associated with calling anything above that value a target–we obtain a different PDR relative to some FAR. In some cases we are given a maximum FAR and we are asked what is the best PDR. In other instances, we are asked for a certain PDR and the question is what FAR is associated with that value? Typically, one looks at a ROC and searches for "runs". We want the ROC to climb fast, i.e., make few mistakes to get more detections. However, when we find long runs, horizontally flat or linearly increasing regions in the ROC, that means we have to accept a lot more FARs to get more detections or we are accepting as many mistakes as detections.

Figure (6) shows that EMI outperforms GPR by a good amount. We cannot show–it is too visually complex–each individual prescreener and different aggregation operators on combinations of these algorithms. Instead, we report the top performing GPR algorithm, the top performing EMI algorithm and the GpChI versus three different ChIs of these inputs–the minimum (intersection like), maximum (union like) and average (expected value). Furthermore, we report the GpChI on all four inputs. Figure (6) shows that a single ChI provides little-to-no benefit (over just EMI). However, the GpChI improves the PDR at each FAR and achieves an overall higher PDR (leads to more detections). At that, the GpChI of all four algorithms gives rise to the best ROC. Overall, these ROCs show that the GpChI has benefit and the fusion of GPR and EMI is advantageous.

Last, we analyze the two GpChIs. Figure (5) is the GpChI for two inputs–the top performing EMI algorithm and the top performing GPR algorithm. The tree has both arithmetic and ChI operators (non-terminals) and it uses both inputs (GPR and EMI). Furthermore, it is not overly complex, meaning the tree depth is relatively small. The learned result is not expressible as a single ChI. The ChI is optimistically looking for which sensor returns the highest confidence. If the largest return is GPR then we intersect (using a multiplication operator, a t-norm) that value with the EMI result. However, if the EMI result is the highest then its value drops off at a rate proportional to its square. The GpChI of all four inputs is

$$C_{\mu_1}(\{C_{\mu_2}(\{h_1, h_2\}), C_{\mu_3}(\{h_3, h_4\})\}),$$

where μ_1 is the maximum, μ_2 and μ_3 are arithmetic means, and $\{x_1, x_2\}$ are the two EMI algorithms and $\{x_3, x_4\}$ are the two GPR algorithms. Thus, the solution is an optimistic aggregation of the expected value of the different algorithms on each sensor.

V. Conclusion and Future Work

In summary, we put forth a *genetic programming* (GP) based extension of the *Choquet integral* (ChI) for problems that require different aggregation philosophies on different subsets of inputs. The GP allows for compositions and/or arithmetic combinations of ChIs. The GP also performs feature selection. Furthermore, we used a GA to solve the set of ChIs per candidate GpChI solution. In order to understand the impact of the GpChI, we formally studied different properties of the GpChI. It was found that GP operations that restrict GpChI to linear convex sum trees yield a ChI. However, we showed that the ChI of ChIs and an arithmetic combination of ChIs is not guaranteed to yield a ChI. However, the ChI of ChIs is bounded by the smallest and largest observation. Overall, in general the GpChI yields an aggregation operator outside the direct scope of a single ChI that can be tailored to different tasks. Last, synthetic and real world experiments were used to show the potential of the proposed method.

In future work, we will incorporate a mechanism to help ensure the learning of high quality and low complexity trees. This helps in the acquisition of quality solutions that are less prone to overfitting (and therefore are more robust). Furthermore, in terms of the fusion of EMI and GPR, we will continue to process new data collections as they become available. We will develop additional prescreeners and use the GpChI to fuse these inputs. In terms of explosive hazard detection, we will analyze the GpChI solutions and try to understand their aggregation philosophy and relate it to the physics of the problem at hand. Furthermore, we would like to find a way to include additional physics and/or contextual information into the calculation and learning of the GpChI.

VI. Acknowledgments

This material is based upon work supported by the U.S. Army TACOM Life Cycle Command under Contract No. W56HZV-08-C-0236. In addition, this research was also supported by Army Research Office grant W911NF-14-1-0673 in support of the US Army RDECOM CERDEC NVESD.

References

[1] JIEDDO COIC MID. (2012, August) Global IED monthly summary report.

[2] M. Grabisch, "K-order additive discrete fuzzy measures and their representation," *Fuzzy sets and systems*, vol. 92, no. 2, pp. 167–189, 1997.

[3] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[4] D. Anderson, T. Havens, C. Wagner, J. Keller, M. Anderson, and D. Wescott, "Extension of the fuzzy integral for general fuzzy set-valued information," *Fuzzy Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 1625–1639, Dec 2014.

[5] M. Grabisch, H. T. Nguyen, and E. A. Walker, *Fundamentals of uncertainty calculi with applications to fuzzy inference*. Springer Science & Business Media, 2013, vol. 30.

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

[7] T. C. Havens, D. T. Anderson, and J. M. Keller, "A fuzzy choquet integral with an interval type-2 fuzzy number-valued integrand," in *International Conference on Fuzzy Systems*, July 2010, pp. 1–8.

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

[9] A. J. Pinar, J. Rice, L. Hu, D. T. Anderson, and T. C. Havens, "Efficient multiple kernel classification using feature and decision level fusion," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, 2016.

[10] A. Pinar, T. C. Havens, D. T. Anderson, and L. Hu, "Feature and decision level fusion using multiple kernel learning and fuzzy integrals," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Aug 2015, pp. 1–7.

[11] S. R. Price, D. T. Anderson, and T. C. Havens, "Fusion of ieco image descriptors for buried explosive hazard detection in forward-looking infrared imagery," in *SPIE Defense+ Security*, vol. 9454, 2015, pp. 945 405–945 405–14.

[12] L. Hu, D. T. Anderson, and T. C. Havens, "Multiple kernel aggregation using fuzzy integrals," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2013, pp. 1–7.

[13] L. Hu, D. T. Anderson, T. C. Havens, and J. M. Keller, *Efficient and Scalable Nonlinear Multiple Kernel Aggregation Using the Choquet Integral*. Cham: Springer International Publishing, 2014, pp. 206–215.

[14] T. C. Havens, K. Stone, D. T. Anderson, J. M. Keller, K. C. Ho, T. T. Ton, D. C. Wong, and M. Soumekh, "Multiple kernel learning for explosive hazard detection in forward-looking ground-penetrating radar," in *SPIE Defense, Security, and Sensing*, vol. 8357, 2012, pp. 83 571D–83 571D–15. [Online]. Available: http://dx.doi.org/10.1117/12.920482

[15] R. Mesiar, L. Šipeky, and A. Šipošová, *Aggregation of Choquet Integrals*. Cham: Springer International Publishing, 2016, pp. 58–64. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-40596-4_6

[16] M. Sugeno, "Theory of fuzzy integrals and its applications," *Ph.D. thesis*, vol. Tokyo Institute of Technology, 1974.

[17] D. T. Anderson, J. M. Keller, and T. C. Havens, "Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral," in *International conference on information processing and management of uncertainty*, 2010, pp. 502–511.

[18] M. A. Lee, D. T. Anderson, J. E. Ball, and J. L. White, "Background adaptive division filtering for hand-held ground penetrating radar," in *SPIE Defense+ Security*, vol. 9823, 2016, pp. 98 230I–98 230I–11.

[19] J. L. White, D. T. Anderson, J. E. Ball, and B. Parker, "Curvelet filter based prescreener for explosive hazard detection in hand-held ground penetrating radar," in *SPIE Defense+ Security*. International Society for Optics and Photonics, 2016, pp. 98 230H–98 230H.

[20] A. Z. B. A. J. B. Ryan Smith, Derek Anderson, "Fusion of choquet integrals for explosive hazard detection in emi and gpr for handheld platforms," in *Accepted for publication in Detection and Sensing of Mines, Explosive Objects, and Obscured Targets*, 2017.