Fuzzy Integral for Rule Aggregation in Fuzzy Inference Systems

[1]Leary Tomlin, [1]Derek T. Anderson, [2]Christian Wagner, [3]Timothy C. Havens, and [4]James M. Keller

[1]Electrical & Computer Engineering, Mississippi State University, USA
[2]School of Computer Science, University of Nottingham, UK
[3]Elec. & Comp. Engineering, Comp. Science, Michigan Technological University, USA
[4]Elec. & Comp. Engineering, Comp. Science, University of Missouri, USA
{lt648@msstate.edu,anderson@ece.mstate.edu,christian.wagner@nottingham.ac.uk,thavens@mtu.edu,kellerj@missouri.edu}

Abstract. The *fuzzy inference system* (FIS) has been tuned and revamped many times over and applied to numerous domains. New and improved techniques have been presented for fuzzification, implication, rule composition and defuzzification, leaving one key component relatively underrepresented, rule aggregation. Current FIS aggregation operators are relatively simple and have remained more-or-less unchanged over the years. For many problems, these simple aggregation operators produce intuitive, useful and meaningful results. However, there exists a wide class of problems for which quality aggregation requires non-additivity and exploitation of interactions between rules. Herein, we show how the fuzzy integral, a parametric non-linear aggregation operator, can be used to fill this gap. Specifically, recent advancements in extensions of the fuzzy integral to "unrestricted" fuzzy sets, i.e., subnormal and non-convex, makes this now possible. We explore the role of two extensions, the gFI and the NDFI, discuss when and where to apply these aggregations, and present efficient algorithms to approximate their solutions.

Keywords: Fuzzy inference system, Choquet integral, fuzzy integral, gFI, NDFI, fuzzy measure

1 Introduction

In Lofti Zadeh's seminal 1965 paper on *fuzzy set* (FS) theory, a new philosophy was put forth to address uncertain data and/or information [1]. In 1973, Zadeh introduced fuzzy logic and he suggested that it might be a useful mechanism to model higher-level thought and reasoning in humans [2]. The first application selected was a steam engine and boiler control system and the rules were provided by the system operators [3]. The Mamdani-Assilian *fuzzy inference system* (FIS) is built on top of Zadeh's *compositional rule of inference* (CRI), a generalization of *modus ponens*, *modus tollens*, etc. The CRI is a way to calculate a FS-valued output based on crisp or FS-valued inputs and an implication function. Other well-known FISs that generalize the CRI are the *Takagi-Sugeno-Kang* (TSK) [4], Tsukamoto [5], and *single input rule modules* (SIRM) FISs [6, 7]. Nearly all FISs consist of some subset of fuzzification, CRI, rule firing, rule combination and defuzzification. Of particular importance to this paper is the aggregation step in an FIS, which is responsible for combining the output of different rules and deriving a final comprehensive decision. Specifically, we are concerned with the identification of functions that can take multiple FS-valued inputs and produce a FS-valued output. The Mamdani-Assilian FIS typically uses FS aggregation strategies such as maximum and summation per individual element in the discrete output domain. The aggregations used in most FISs share something in common, they are relatively simple and they do not model nor exploit interactions (when/if available) between rules. Herein, we investigate the role of the *fuzzy integral* (FI) [8], namely the *Choquet integral* (CI), for non-linear rule combining in fuzzy logic. In particular, we explore our two recent extensions, the *generalized FI* (gFI) and the *non-direct FI* (NDFI) [9], that are capable of aggregating any type of FS in contrast to prior extensions for interval-valued data and fuzzy numbers. The *fuzzy measure* (FM), introduced by Sugeno in 1974 [8], is used to model interactions (when/if available) between rules. Specifically, for the Mamdani-Assilian FIS, let X be the set of N rules, let $x_i \in X$ be the ith rule and $H(x_i)$, or H_i for short, be the ith FS-valued rule output. Herein, we address how to use the NDFI and gFI for fuzzy logic and when and where to use one or the other. Figure 1 illustrates the role of the FI in fuzzy logic.

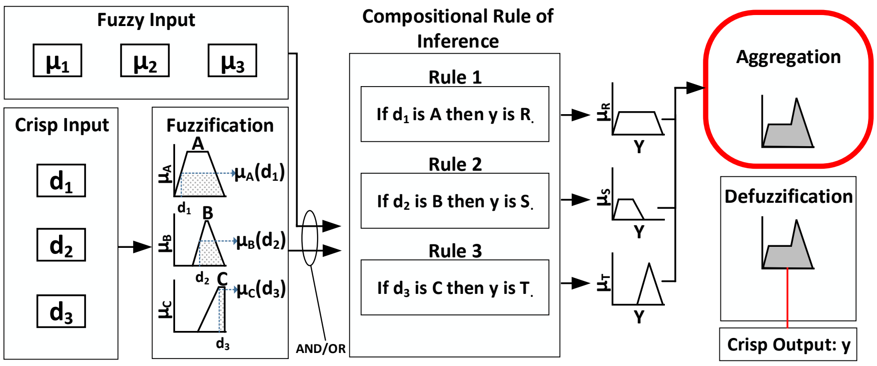


Fig. 1. Block diagram of a fuzzy inference system and where the FI fits in.

In 1992, Yager explored rule aggregation in fuzzy logic [10]. He considered two rule representations, the Mamdani model and the logical model, and obtained a general rule representation. For aggregation, he explored two soft classes of rule combining, *or-like* and *and-like*. Ultimately, Yager's work is subsumed by our current article. First, Yager's operators are performed per-element in a discrete output domain. This is how the NDFI operates, although not the gFI (which is based on the *extension principle* (EP)). Second, Yager's two classes of operators, and even his related *ordered weighted averages* (OWAs) [11], are special cases of the CI (with respect to particular FMs) [12]. Third, and related to the second point, the NDFI and gFI are not restricted to being additive (they only need to satisfy the more general rule of monotonicity) and they can model and exploit rich interactions between rules when/if available. This allows us to achieve a much wider, sophisticated and custom set of non-linear aggregation operators.

The organization of this paper is as follows. In Section 2, the FI, NDFI and gFI are reviewed and their role in FIS rule output aggregation is addressed. In Section 3, we provide an example of the proposed theory. The goal of this article is to make the necessary introduction to different relevant FI extensions, propose algorithms for computing the integrals, and present discussions about specifying or learning g in the context of fuzzy logic.

2 Fuzzy Measure and Integral

2.1 Fuzzy Measure

The concepts of a measure and integral are two of the most fundamental aspects of mathematics. A famous example is the Lebesgue measure and integral with respect to that measure. The Lebesgue measure is the typical way of assigning a value to subsets of a Euclidean space, i.e., length, area, volume. However, measures can capture a wide range of concepts outside of n-dimensional Euclidean spaces. In the context of the FI, the FM often is a subjective assessment of the *worth* or *importance* of a subset of data or information sources. In general, the FM requires the property of monotonicity with respect to set inclusion, a weaker property than the additive property of a probability measure. In [8], the initial focus was an integrand (h) and FM (g) in the interval $[0,1]$, $h : X \to [0,1]$ and $g : 2^X \to [0,1]$, where 2^X is all subsets of X. However, both integrand and FM have been defined more generally and used in the FI, e.g., $h : X \to [-\infty, \infty]$, $h : X \to \Re_0^+$, $g : 2^X \to \Re_0^+$, etc. Next, we briefly review the FM.

Definition 1. (Fuzzy measure) [8] For a finite set X, a FM is a function $g : 2^X \to \Re_0^+$, such that

1. (Boundary Condition) $g(\phi) = 0$ (and often $g(X) = 1$);
2. (Monotonicity) If $A, B \subseteq X$, $A \subseteq B$, then $g(A) \leq g(B)$.

Note, if X is an infinite set, then a third condition guaranteeing continuity is required. However, this is a moot point for finite X. In the FIS system, we only consider the finite set X of N rules.

Definition 2. (Sugeno λ-measure) [8] Let g be a FM. Furthermore, g is called a Sugeno λ-measure if it satisfies the following condition,

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B), \tag{1}$$

for some $\lambda > -1$. The Sugeno λ-measure is built using the singleton values (aka densities), $g^i = g(\{x_i\})$. Sugeno showed that λ can be found by solving

$$\lambda + 1 = \prod_{i=1}^{N} \left(1 + \lambda g^i\right), \lambda > -1, \tag{2}$$

where there exists one real solution that satisfies $\lambda > -1$.

Definition 3. (S-Decomposable Measure) Let S be a *triangular-conorm* (t-conorm). A FM g is called an S-decomposable measure if $g(\phi) = 0$, $g(X) = 1$, and for all A, B such that $A \cap B = \phi$,

$$g(A \cup B) = S(g(A), g(B)). \tag{3}$$

2.2 Choquet Integral

Next, we briefly review the real-valued discrete CI; see [8, 9, 13, 14] for additional information (proofs, properties, etc.).

Definition 4. (**$\Re$-Valued Discrete Choquet FI**) [8] Given a finite set X, FM g and function $h : X \to \Re$, the CI of h with respect to g is

$$\int h \circ g = C_g(h) = \sum_{i=1}^{N} h_{\Delta(i)} \left(g(A_{\Delta(i)}) - g(A_{\Delta(i-1)})\right), \tag{4}$$

where $h_{\Delta(i)} = h(\{x_{\Delta(i)}\})$ and Δ is a permutation on X such that $h_{\Delta(1)} \geq h_{\Delta(2)} \geq ... \geq h_{\Delta(N)}$, $A_{\Delta(i)} = \{x_{\Delta(1)}, ..., x_{\Delta(i)}\}$ and $g(A_{\Delta(0)}) = 0$.

2.3 Generalized Fuzzy Integral

In [9], we introduced the gFI, an extension of the FI based on Zadeh's *extension principle* (EP) for "unrestricted" (potentially non-convex and subnormal) FSs. Initially, the gFI was created for skeletal age-at-death estimation in forensic anthropology [15, 16]. Due to space considerations herein, see [9] for full mathematical detail, proofs and in-depth analysis. Algorithm 1 is an algorithmic description of how to calculate the gFI.

Note, that at a particular α-cut we obtain, for non-convex FSs, a *discontinuous interval*, e.g., an ${}^{\alpha}H_i$ that yields more than one interval, like $[0, 0.1]$ and $[0.7, 0.9]$, versus a single continuous interval, such as $[0, 0.9]$. The gFI plays off the fact that a discontinuous interval can be represented as the union of its continuous sub-intervals [9]. At each α-cut, we first decompose the discontinuous intervals into their corresponding continuous interval counterparts. Computationally, that is for a finite number of α-cut approximation, this only makes sense when we consider a finite number of continuous intervals. Next, we compute the continuous interval-valued FI on each combination ($\mathcal{M}_\alpha$ in total) of continuous intervals with respect to our different inputs/sources. For example,

Algorithm 1 Algorithm to calculate the *generalized FI* (gFI) [9]

```
Input FM $g$    ▷ e.g., Sugeno $\lambda$-FM, learn $g$ from data, specify $g$, etc.
Input FS-valued partial support function $H$    ▷ i.e., $H_i$ for $i = \{1, ..., N\}$
Calculate $\beta = \bigwedge_{i=1}^{N} Height(H_i)$    ▷ minimum height of partial support FSs
for each $\alpha \in (0, \beta]$ do    ▷ note, this step is discretized in practice
    $[\int H \circ g]_\alpha = \left(\int [^\alpha H]_1 \circ g\right)$    ▷ first gFI calculation; interval-valued FI
    for k = 2 to $\mathcal{M}_\alpha$ do    ▷ each continuous interval combo ($\mathcal{M}_\alpha$) at $\alpha$
        $[\int H \circ g]_\alpha = [\int H \circ g]_\alpha \bigcup \left(\int [^\alpha H]_k \circ g\right)$    ▷ kth gFI calculation
    end for
end for
```

for two inputs, if the first has three continuous intervals and the second has two then $\mathcal{M}_\alpha$ equals six. Let $[^\alpha H]_{k=1}$ be the first such combination, i.e., the first continuous interval from source one, first continuous interval from source two, etc. The gFI is simply the union of these continuous-valued interval FI results. At each α-cut, the beauty is that the gFI breaks down into a set of interval-valued FI calculations which breaks down further into two $\Re$-valued FIs [9, 17]—one for the left interval endpoints and one for the right interval endpoints.

2.4 Non-Direct Fuzzy Integral

Unlike the gFI, the NDFI does not extend the FI according to Zadeh's EP. It should be noted that the EP is not the true and only way to address extensions of the FI, it is just one well grounded and heavily studied approach. Due to space considerations herein, see [9] for full mathematical detail, proofs and indepth analysis of the NDFI. Algorithm 2 is a description of the NDFI.

Algorithm 2 Algorithm to calculate the *non-direct FI* (NDFI) [9]

```
Input FM $g$    ▷ e.g., Sugeno $\lambda$-FM, learn $g$ from data, specify $g$, etc.
Input the FS-valued partial support function $H$ ▷ i.e., $H_i$ for $i = \{1, ..., N\}$
Discretize the output domain, $D = \{d_1, ..., d_{|D|}\}$    ▷ e.g., $D = \{0, 0.01, ..., 1\}$
Initialize the (FS) result to $R[d_k] = 0$
for each $d_k \in D$ do    ▷ for each output domain location
    for each $i \in \{1, ..., N\}$ do    ▷ for each input
        Let $z_i = H_i(d_k)$    ▷ i.e., $\mathbf{z}$ is the vector of memberships at $d_k$
    end for
    $R[d_k] = \int \mathbf{z} \circ g$    ▷ $\Re$-valued FI of $\mathbf{z}$ with $g$
end for
```

Whereas the gFI decomposes the FI into a sequence of interval-based FI calculations across the membership domain, NDFI decomposes the FI into a sequence of $\Re$-valued FI calculations across the input/element domain. The gFI always produces FSs whose cardinality, i.e., number of elements of the set, is greater than zero, as long as the minimum input FS height is greater than zero.

However, the NDFI is not guaranteed to do this. For example, consider when the FI becomes the minimum operator, i.e., when $g(A) = 0, \forall A \subset X, g(X) = 1$. Furthermore, assume the input FSs are completely disjoint. The NDFI therefore produces a value of 0 everywhere.

2.5 FI for Combining Rules in a FIS

As the reader can clearly see, the NDFI and gFI are two ways to aggregate FS-valued inputs. Furthermore, they fit naturally into existing FISs. The first question we explore is, when to use the NDFI and when to use the gFI. Table 1 summarizes a few important properties of these two extensions.

Table 1. Summary of important properties of the NDFI and the gFI

FI extension	**domain**	**order**	**"types" of sets**
NDFI	discrete or continuous	no assumption	any
gFI	real-number domain	total order	any

In general, if the domain is naturally discrete and no total order exists then the gFI is not applicable but the NDFI is. The NDFI "aggregates in place"—it does not allow for interaction or cross-pollination between elements in the domain. On the other hand, the gFI allows for interaction between elements (thus why it works on $\Re$). Additionally, the gFI is based on the EP and it extends a function to FS-valued inputs. In the next section we provide numeric examples that illustrates these concepts.

Next, we explore common FMs encountered in practice (and FISs in the case of the NDFI). In Yager's approach [10], he explored *and-like* and *or-like* aggregation operators on a per-element basis. The NDFI yields Yager's *and-like* and *or-like* operators, OWA operators (specifically, all linear combinations of order statistics) and numerous other aggregation operators outside this set. It is well-known that the FI, specifically the CI, turns into a particular aggregation operator based on g (Table 2 provides an example for a few well-known FMs).

Table 2. Example FMs and aggregation operators induced

Resultant CI operator	FM properties
OWA	$g(\phi) = 0$, $g(X) = 1$, $g(A) = g(B)$ when $\|A\| = \|B\|, \forall A, B \subseteq X$
Maximum	$g(A) = 1, \forall A \subseteq X, A \neq \phi, g(\phi) = 0$
Minimum	$g(X) = 1$, $g(A) = 0, \forall A \subset X$
Mean	$g(A) = \frac{\|A\|}{N}, \forall A \subseteq X, A \neq \phi, g(\phi) = 0$
Sum	$g(A) = \|A\|, \forall A \subseteq X, A \neq \phi, g(\phi) = 0$

Of course, the FMs in Table 2 (minus "sum") are all OWAs and it is well-known that the CI can yield a class of custom non-linear aggregation operators outside of OWAs. The FM can be a number of things from a belief measure to a Sugeno λ-FM to much more. This is not counting the vast sea of custom and unique FMs that one might need and attempt to learn from data. The NDFI can also reproduce the *sum*, used by many FISs, if one discards the upper $g(X) = 1$ boundary condition. For example, the FM $g(A) = |A|$ makes the CI yield the summation operator, e.g., for $N = 4$, $h_{\Delta(1)}(1-0) + h_{\Delta(2)}(2-1) + h_{\Delta(3)}(3-2) + h_{\Delta(4)}(4-3) = h_{\Delta(1)} + h_{\Delta(2)} + h_{\Delta(3)} + h_{\Delta(4)}$.

Instead, if we do not want to use a standard aggregation operator then we can specify the full FM. However, this can be incredibly complex as g grows quickly in size—2^N terms for N inputs. If we know the "worth" of the singletons (aka densities) then one can refer to techniques such as the Sugeno λ-measure and S-decomposable measure to impute the remainder of the FM. We can also look to a technique such as the k-additive FI.

However, if one wants to learn the FM, densities or the full FM, from data then a number of methods can be used, e.g., an optimization algorithm such as a genetic algorithm [18], linear and quadratic programming [13, 14, 19], gradient descent [20], etc. See [21] for FI learning works prior to 2008. In the case of a FIS, learning depends on if one has labeled data that is the defuzzified or FS-valued target outputs. In the latter case, we must select a measure of distance or divergence between FSs, such as one that calculates the sum of differences of interval endpoints across α-cuts [22, 23]. While it is possible to learn the FM in fuzzy logic, we note that in practice it is likely not a trivial task as it has to be achieved in combination with FIS parameter specification.

Next, we explore the complexity of the NDFI versus the gFI. If the output domain has been discretized (or is already discrete) and is $D = \{d_1, d_2, ..., d_{|D|}\}$, then the NDFI requires $|D|$ FIs, each of which take $3N-1$ operations (specifically N multiplications, N subtractions and $N-1$ additions), therefore $3N|D| - |D|$ total calculations (not counting the H_i sorting step, which occurs for each element in D). The NDFI has more-or-less the same cost as current FIS operations, e.g., maximum, average, sum, etc. The only added complexity is determining the FM. Depending on the number of desired α-cuts, the gFI may or may not be more expensive than the NDFI (depends on the discretization level of the output domain). Assuming P α-cuts, we must first determine each continuous interval, all combinations of continuous intervals must be computed ($\mathcal{M}_\alpha$) and subjected to the interval-valued FI (which is the invocation of two $\Re$-valued FIs, thus N operations), and we must take the union of the results. The gFI cost formula is not as easily expressible (as the NDFI) because each α-cut on each FS can yield a different number of discontinuous intervals. We do not see complexity as the driving factor in selecting the NDFI or the gFI. Instead, it seems more important that a user prefers the EP (gFI) or "aggregation in place" (NDFI).

3 Numeric Example

In this section, we consider an augmented fuzzy logic tipping problem. We consider four arbitrary rules (Table 3) and an input of $(0.58, 0.75, 0.96)$, for (responsiveness, satisfaction, food), which results in four FSs to aggregate. Note, the particular vocabulary and their parameters is not as important as the trends of FSs. We want to show the case of one “outlier” (FS in extreme disagreement, i.e., H_1) and three overlapping (partially agreeing) FSs. Table 4 contains the terms and parameters used in the four rules. For comparison's sake, Table 5 contains five different FMs (whose full set of values are reported) used to aggregate our rule FS outputs. The particular FMs in Table 5 were selected in order to demonstrate the behavior and differences between the NDFI and gFI. In particular, we compute OWAs (maximum, minimum, average), the sum, the Sugeno λ-FM (for a set of densities) and an S-Decomposable FM (labeled possibility in Table 5), where S=max, for a given set of densities. In addition, we report two “binary FMs”—a FM (labeled $g_{2,3}$ in Table 5 for compactness sake) that is 0 except for $g(\{x_2, x_3\}) = 1$ (and satisfying monotonicity) and a FM (called $g_{1,4}$ in Table 5) that is 0 except for $g(\{x_1, x_4\}) = 1$ (and satisfying monotonicity).

Table 3. Example toy rules for tipping

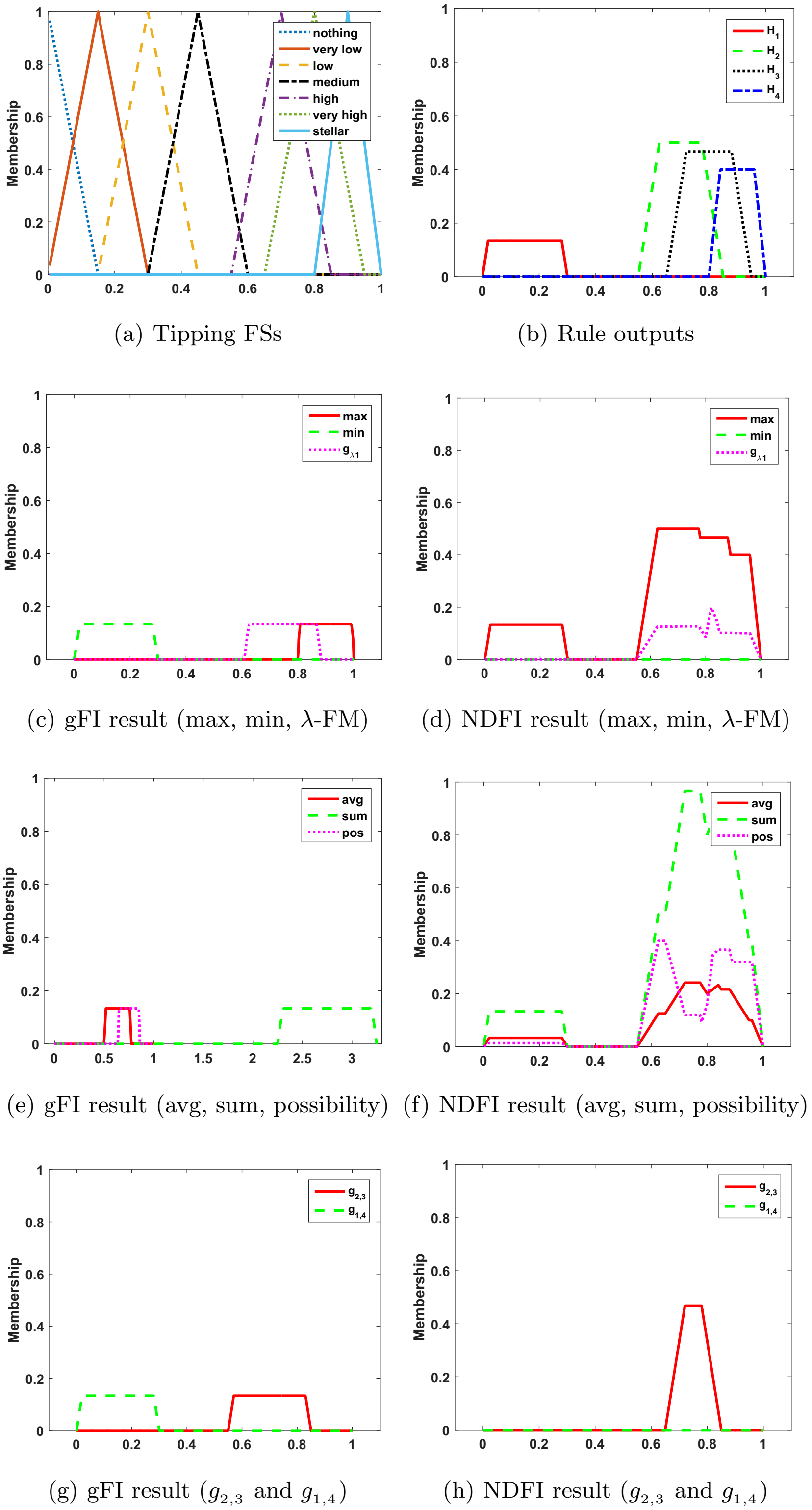
Rule	**If**	**Then**
1.	responsiveness is slow	tip is very low
2.	satisfaction is high	tip is high
3.	responsiveness is moderate AND satisfaction is high	tip is very high
4.	responsiveness is moderate AND food is delicious	tip is stellar

Table 4. Fuzzy set parameters for FIS example [1]

Variable	**Term**	**Tri. Membership**	**Variable**	**Term**	**Tri. Membership**
responsiveness	slow	(0.3,0.45,0.6)	Tip	very low	(0,0.15,0.3)
	moderate	(0.35,0.5,0.65)		high	(0.55,0.7,0.85)
satisfaction	high	(0.7,0.8,0.9)		very high	(0.65,0.8,0.95)
food	delicious	(0.8,0.9,1)		stellar	(0.8,0.9,1)

Our tipping example exposes a few interesting things. First, the gFI results look rather extreme as they are truncated according to the minimum height input FS (due to the formulation of the EP). Truncation is extremely frequent as each rule fires to some degree so the FSs are almost always subnormal. Next, the max, min and average look more like what we might expect, if our desire is fuzzy arithmetic. On the other hand, the NDFI aggregates in place—restricting fusion to just the set of available evidence at a particular output domain element. This produces different results for the same FM. Next, each FI extension and FM yields different FSs, which leads to different defuzzified results (when/if a

[1] note, this could be scaled to a level of tip, e.g. in [0,30]%



(a) Tipping FSs

(b) Rule outputs

(c) gFI result (max, min, λ-FM)

(d) NDFI result (max, min, λ-FM)

(e) gFI result (avg, sum, possibility)

(f) NDFI result (avg, sum, possibility)

(g) gFI result ($g_{2,3}$ and $g_{1,4}$)

(h) NDFI result ($g_{2,3}$ and $g_{1,4}$)

Fig. 2. Graphical illustration of the gFI and NDFI outputs for the FMs reported in Table 5, terms in Table 4 and rules in Table 3.

Table 5. List of FMs used for examples

FM	**max**	**min**	**sum**	**average**	**Sugeno λ-FM ($g_{\lambda 1}$)**	$\boldsymbol{g_{2,3}}$	$\boldsymbol{g_{1,4}}$	**possibility**
$g(\{x_1\})$	1	0	1	0.25	0.001	0	0	0.1
$g(\{x_2\})$	1	0	1	0.25	0.25	0	0	0.8
$g(\{x_3\})$	1	0	1	0.25	0.001	0	0	0.2
$g(\{x_4\})$	1	0	1	0.25	0.25	0	0	0.7
$g(\{x_1, x_2\})$	1	0	2	0.5	0.253	0	0	0.8
$g(\{x_1, x_3\})$	1	0	2	0.5	0.002	0	0	0.2
$g(\{x_1, x_4\})$	1	0	2	0.5	0.253	0	1	0.7
$g(\{x_2, x_3\})$	1	0	2	0.5	0.253	1	0	0.8
$g(\{x_2, x_4\})$	1	0	2	0.5	0.983	0	0	0.8
$g(\{x_3, x_4\})$	1	0	2	0.5	0.253	0	0	0.7
$g(\{x_1, x_2, x_3\})$	1	0	3	0.75	0.256	1	0	0.8
$g(\{x_1, x_2, x_4\})$	1	0	3	0.75	0.993	0	1	0.8
$g(\{x_1, x_3, x_4\})$	1	0	3	0.75	0.256	0	1	0.7
$g(\{x_2, x_3, x_4\})$	1	0	3	0.75	0.991	1	0	0.8
$g(X)$	1	1	4	1	1	1	1	1

crisp overall output is desired). The results of firing the rules with the given inputs and utilizing each measure in Table 5 can be found in Figure 2.

In the case of the sum FM and the gFI, if the output domain is restricted to $[0, 1]$ then the result is officially "out of range". However, the gFI and the NDFI yield results that do not exceed the min and max input FSs, which therefore is in the valid interval as long as the inputs are in the valid interval and the FM has at max $g(X) = 1$ [9]. Additionally, the NDFI also yields (due to boundedness property of the FI) results between $[0, 1]$ as long as the sets have memberships in $[0, 1]$ and the FM has at max $g(X) = 1$. Obviously, the sum FM violates these conditions. One rationalization of the usefulness of an NDFI operator that produces membership values greater than 1 is it still leads to a technically valid defuzzified result (but not a valid FS).

Next, we explore a binary FM. A binary FM is a FM whose values are restricted to $\{0, 1\}$. Consider the set of all maximal chains of the Hasse diagram $(2^X, \subseteq)$. A maximal chain in $(2^X, \subseteq)$ is a sequence ϕ, $\{x_{\pi(1)}\}$, $\{x_{\pi(1)}, x_{\pi(2)}\}$, ..., $\{x_{\pi(1)}, ..., \pi_{\pi(N)}\}$ (for a given permutation π). In the discrete case, the CI is nothing more than a weighted sum—the sorted integrands times the differences in their respective values in the maximal chain. For a binary FM, one of these difference-in-g's is 1 and the other values are 0 (trivial to prove due to monotonicity). Now, consider the FM in Figure 2 labeled $g_{2,3}$. This FM has value 0 for all subsets except 1 for $\{x_2, x_3\}$ and all respective subsets needed to ensure monotonicity. The result is the minimum of the two sets H_2 and H_3 (shown in Figure 2). The same argument holds for the FM labeled $g_{1,4}$ (the FM is computing the minimum of the two inputs H_1 and H_4). While this is a rather extreme (binary) example to illustrate the importance and resulting effect of specification and/or learning of higher-order tuple terms in the FM, the point is we can achieve a wealth of custom aggregation strategies for combinations of rules.

In closing, while it would be nice to definitively state that the NDFI or the gFI is better, we feel this is not the case. Instead, the ultimate decision of which extension to pick appears to break down into either what the problem demands (e.g., a discrete domain where elements cannot cross-pollinate) or the desire of what aggregation should do to the outputs, either aggregate in place or compute a function that has been extended to FS-valued inputs.

4 Conclusion and Future Work

Herein, we proposed and started to explore the NDFI and gFI extensions for rule aggregation in a FIS. We discussed how these extensions fit into existing FISs, reviewed efficient discrete algorithms and analyzed their computational complexities. Furthermore, we showed, via the toy tipping example, how to pick FMs that turn the CI into existing FIS aggregation operators, we discussed imputation methods (from the densities) and opened the door for learning. The example and Table 1 gave the reader a feel for the inner workings and differences between the gFI and NDFI and it helps with understanding when and where to use one extension over the other. We also highlighted that the NDFI "aggregates in place" (per-element) while the gFI is based on the EP. Ultimately, this makes a big difference, in terms of output, and it is a choice the user must make.

Due to space considerations, we had to restrict the focus of this article to different technical aspects of rule output aggregation using the NDFI and the gFI. However, in future work we feel it will be important to explore various applications, in particular in conjunction with learning rules from data and going beyond theories like the Wang-Mendel approach. While an important part of the gFI and NDFI is the fact that they can produce a wealth of different aggregation operators and extend the capabilities of a FIS, we will need to explore if there are any particular FMs (and therefore aggregation operators) that are "more appropriate" than others for different scenarios. The point is, a FIS user will need to either select or learn the FM. If no data is available then a big challenge they will have to face is what FM to select for their given application. Next, while we discussed type-1 fuzzy logic herein, in [24] we previously extended the FI for type-2 FS-valued integrands. We will explore the extension of NDFI and gFI, and thus FISs, for type-2 valued integrands. Last, we will investigate efficient methods (mathematics and algorithms) for learning the FM from both $\Re$-valued and FS-valued ground truth.

Acknowledgments. This work has been partially supported by the U.S. Army Research Office, the U.S. Army and RDECOM CERDEC NVESD via W911NF-16-1-0017, W911NF-14-1-0114, W911NF-14-1-0673, W909MY-13-C-0013 and 57940-EV. This work was also partially funded by RCUK's EP/M02315X/1.

Bibliography

[1] Zadeh, L.: Fuzzy sets. Inf. and control **8**(3) (1965) 338–353

[2] Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics **3**(1) (1973) 28–44

[3] Mamdani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. journal of man-machine studies **7**(1) (1975) 1–13

[4] Sugeno, M.: Industrial applications of fuzzy control. Elsevier Science Inc. (1985)

[5] Tsukamoto, Y.: An approach to fuzzy reasoning method. Advances in fuzzy set theory and applications **137** (1979) 149

[6] Yubazaki, N., Yi, J., Hirota, K.: Sirms (single input rule modules) connected fuzzy inference model. JACIII **1**(1) (1997) 23–30

[7] Seki, H., Mizumoto, M.: Sirms connected fuzzy inference method adopting emphasis and suppression. Fuzzy Sets and Systems **215** (2013) 112–126

[8] Sugeno, M.: Theory of fuzzy integrals and its applications. Tokyo Institute of Technology (1974)

[9] Anderson, D.T., Havens, T.C., Wagner, C., Keller, J.M., Anderson, M.F., Wescott, D.J.: Extension of the fuzzy integral for general fuzzy set-valued information. IEEE Transactions on Fuzzy Systems **22**(6) (2014) 1625–1639

[10] Yager, R.: A general approach to rule aggregation in fuzzy logic control. Applied Intelligence **2**(4) (1992) 333–351

[11] Yager, R.R., Filev, D.P.: Induced ordered weighted averaging operators. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **29**(2) (Apr 1999) 141–150

[12] Tahani, H., Keller, J.: Information fusion in computer vision using the fuzzy integral. IEEE Transactions System Man Cybernetics **20** (1990) 733–741

[13] Grabisch, M., Nguyen, H., Walker, E.: Fundamentals of uncertainty calculi, with applications to fuzzy inference. Kluwer Academic, Dordrecht (1995)

[14] Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners. 1st edn. Springer Publishing Company, Incorporated (2008)

[15] Anderson, M., Anderson, D.T., Wescott, D.: Estimation of adult skeletal age-at-death using the sugeno fuzzy integral. American journal of physical anthropology **142**(1) (2010) 30–41

[16] Anderson, D.T., Keller, J.M., Anderson, M.F., Wescott, D.J.: Linguistic description of adult skeletal age-at-death estimations from fuzzy integral acquired fuzzy sets. In: 2011 IEEE Int. Conf. on Fuzzy Systems (FUZZ), IEEE (2011) 2274–2281

[17] Grabisch, M., Murofushi, T., Sugeno, M.: Fuzzy Measures and Integrals: Theory and Applications. Physica-Verlag, Heidelberg (2000)

[18] Anderson, D.T., Keller, J.M., Havens, T.C.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. In: Int. Conf. on information processing and management of uncertainty. (2010) 502–511

[19] Beliakov, G.: Construction of aggregation functions from data using linear programming. Fuzzy Sets and Systems **160** (2009) 65–75

[20] Mendez-Vazquez, A., Gader, P., Keller, J., Chamberlin, K.: Minimum classification error training for choquet integrals with applications to landmine detection. IEEE Transactions on Fuzzy Systems **16**(1) (2008) 225 –238

[21] Grabisch, M., Kojadinovic, I., Meyer, P.: A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the kappalab r package. European Journal of Operational Research **186**(2) (2008) 766 – 785

[22] Anderson, D.T., Keller, J.M., Havens, T.C.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. In Hllermeier, E., Kruse, R., Hoffmann, F., eds.: Computational Intelligence for Knowledge-Based Systems Design. Volume 6178 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 502–511

[23] McCulloch, J., Wagner, C., Aickelin, U.: Analysing fuzzy sets through combining measures of similarity and distance. In: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). (July 2014) 155–162

[24] Havens, T.C., Anderson, D.T., Keller, J.M.: A fuzzy choquet integral with an interval type-2 fuzzy number-valued integrand. In: 2010 IEEE Int. Conf. on Fuzzy Systems (FUZZ). (July 2010) 1–8