Visualization and Learning of the Choquet Integral With Limited Training Data

Anthony J. Pinar[a], Timothy C. Havens[a,b]
[a]Department of Electrical and Computer Engineering
[b]Department of Computer Science
Michigan Technological University
Houghton, Michigan, 49931 USA
email: {ajpinar, thavens}@mtu.edu

Muhammad Aminul Islam, Derek T. Anderson
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39759, USA
e-mail: mi160@msstate.edu,anderson@ece.msstate.edu

***Abstract*—The *fuzzy integral* (FI) is a nonlinear aggregation operator whose behavior is defined by the *fuzzy measure* (FM). As an aggregation operator, the FI is commonly used for evidence fusion where it combines sources of information based on the worth of each subset of sources. One drawback to FI-based methods, however, is the specification of the FM. Defining the FM manually quickly becomes too tedious since the number of FM terms scales as 2^n, where n is the number of sources; thus, an automatic method of defining the FM is necessary. In this paper, we review a data-driven method of learning the FM via minimizing the *sum-of-squared error* (SSE) in the context of decision-level fusion and propose an extension allowing knowledge of the underlying FM to be encoded in the algorithm. The algorithm is applied to real-world and toy datasets and results show that the extension can improve classification accuracy. Furthermore, we introduce a visualization strategy to simultaneously show the quantitative information in the FM as well as the FI.**

***Keywords*—Choquet fuzzy integral, fuzzy measure, quadratic programming, support vector machine**

I. Introduction

In many fields, we are often faced with the task of making decisions based on a set of feature-vector data $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \subset \mathbb{R}^d$. This data is typically accompanied by a set of training labels for each feature-vector, giving the pair $(\mathbf{y}, X)$, where $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ is a vector of labels such that y_i is the label of feature-vector $\mathbf{x}_i$. This problem can be considered a classification task, and is typically tackled by training a classifier such that it can accurately predict the class label of a new sample of data where the label is not known. More concretely, the data $(\mathbf{y}, X)$ are used to learn some prediction function f such that we can accurately predict the label of feature vectors as $y = f(\mathbf{x})$.

Linear classifiers are typically nothing more than a hyperplane in the feature-space representing the decision boundary, and training these classifiers involves finding the hyperplane's parameters in some optimal way. A very popular hyperplane classifier is the *support vector machine* (SVM) because it is easy to train and computationally efficient. The drawback to linear SVMs (and other linear classifiers), however, is that they require the data to be linearly separable—a distribution very rarely encountered with real data. One way around this is to instead use their kernel-based variants where the data are non-linearly projected to a high-dimensional space where a suitable hyperplane is more likely to be found. While this appears to solve the problem of non-separable data, it has its own baggage: what kernel function should be used?

Multiple kernel learning (MKL) typically answers this question by learning a new kernel through the combination of predetermined kernels while maintaining symmetry and positive-semidefiniteness, an approach discussed in many works [1]–[7]. These approaches fall under the roof of feature-level fusion in that they combine different "looks" at the data (each represented by an individual kernel) and use a single classifier to determine the predicted class label. Another MKL technique uses multiple kernel-based classifiers, each utilizing a different kernel. The outputs of these classifiers is then combined at the decision-level using some aggregation function. This approach to decision-level fusion is the premise for the *decision-level fuzzy integral multiple kernel learning* (DeFIMKL) classifier discussed in Section III, where aggregation is performed via the Choquet *fuzzy integral* (FI) with respect to a *fuzzy measure* (FM). Once again though we have a roadblock: how do we specify the FM?

The task we investigate in this work is learning a FM. Many previous works [8]–[10] have shown that an underlying FM can be learned from training data, though here we show that only a subset of the FM is accurately learned from the training data and the remaining FM terms simply follow the constraints from the learning process. In other words, only a subset of the FM is learned in a data-driven manner. Thus when asked to classify a new sample of data using the Choquet FI, we risk utilizing terms from the FM that were not learned accurately from the training data, leading to an erroneous prediction. In this work, we propose a method to more accurately learn the FM terms that are not data-driven. The method assumes that some knowledge of the underlying FM structure is known and thus can be encoded in the learning process as discussed in Section IV.

The remainder of this paper is organized as follows. Section II discusses fuzzy measures and the Choquet fuzzy integral; it also introduces our strategy of simultaneously visualizing the FM and behavior of the Choquet integral. Section III reviews learning a fuzzy measure through minimizing the *sum-of-squared error* (SSE) via *quadratic programming* (QP)—the backbone of the DeFIMKL algorithm—as well as its behavior with insufficient training data. Section IV proposes an extension to the DeFIMKL algorithm, allowing knowledge of the underlying FM to be encoded into the QP, and Section V summarizes experiments with real-world and contrived datasets. Finally, Section VI concludes the paper and discusses

our future work.

II. Fuzzy Measures and Fuzzy Integrals

FIs and FMs are used for many applications and for many types of data, from simple numeric data to intervals and type-2 fuzzy sets [11]–[14]. While manual specification of the FM works for small sets of sources, manually specifying the values of the FM for large collections of sources is virtually impossible. Thus, automatic methods have been proposed, such as the Sugeno λ-measure [15] and the S-decomposable measure [16], which build the measure from the densities[1], and genetic algorithm [5], [17], Gibbs sampling [18] and other learning methods, which build the measure by using training data. Other works [19]–[21] have proposed learning FMs that reflect trends in the data and have been specifically applied to crowd-sourcing, where the worth of individuals is not known, and is thus extracted from the data.

A. Fuzzy measures

A measurable space is the tuple (X, Ω), where X is a set and Ω is an Ω-algebra or set of subsets of X such that

- P1. $X \in \Omega$;
- P2. For $A \subseteq X$, if $A \in \Omega$, then $A^c \in \Omega$;
- P3. If $\forall A_i \in \Omega$, then $\bigcup_{i=1}^{\infty} A_i \in \Omega$.

A FM is a set-valued function, $g : \Omega \to [0, 1]$, with the following properties:

- P4. (Boundary conditions) $g(\emptyset) = 0$ and $g(X) = 1$;
- P5. (Monotonicity) If $A, B \in \Omega$ and $A \subseteq B$, $g(A) \leq g(B)$.

If Ω is an infinite set, then there is also a third property to guarantee continuity; however, in practice and in this paper, Ω is finite and thus this property is unnecessary. While fuzzy measures provide a way for quantifying the worth of combinations of sources, fuzzy integrals can be used to aggregate the information from these sources.

B. Fuzzy integrals

There are many forms of the FI; see [15] for detailed discussion. In practice, FIs are frequently used for evidence fusion [17], [22]–[24]. They combine sources of information by accounting for both the support of the question (the evidence) and the expected worth of each subset of sources (as supplied by the FM g). Here, we focus on the fuzzy Choquet integral, proposed by Murofushi and Sugeno [25], [26]. Let $h : X \to \mathbb{R}$ be a real-valued function that represents the evidence or support of a particular hypothesis.[2] The discrete (finite Ω) fuzzy Choquet integral is defined as

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

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 $g(A_0) = 0$ [13], [27]. Detailed treatments of the properties of FIs can be found in [13], [27], [28].

[1]The FM values of the singletons, $g(\{x_i\}) = g^i$ are commonly called the *densities*.

[2]Generally, when dealing with information fusion problems it is convenient to have $h : X \to [0, 1]$, where each source is normalized to the unit-interval.

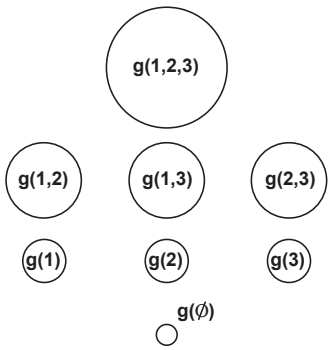


Fig. 1: Lattice of FM elements for $n = 3$. Monotonicity (P5) is illustrated by the size of each node, i.e., $g(\{x_1\}) \leq g(\{x_1, x_2\})$ as $\{x_1\} \subset \{x_1, x_2\}$. Note that shorthand notation is used where $g(1, 3)$ is equivalent to $g(\{x_1, x_3\})$.

C. Common Aggregations via the Choquet Integral

It is well known that the Choquet integral is a powerful aggregation operator parametrized by a FM, and thus can represent many aggregation functions [29]. For example, the Choquet integral acts as the maximum operator when the FM is all 1s (except $g\{\emptyset\} = 0$, due to boundary constraints), the minimum operator when the FM is all 0s (except $g\{X\} = 1$, due to boundary constraints), and the mean operator when $g(A_i) = |A_i|/n$, $\forall A_i \subset X$.

D. Visualizing the Fuzzy Integral

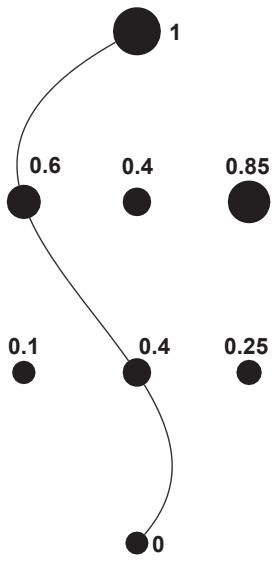
The FM lattice (Hasse diagram) is a convenient method to visualize a FM; Figure 1 illustrates the lattice of a FM for the case of $n = 3$. Note that the size of the individual nodes in the lattice indicates their relative magnitude, and monotonicity is apparent since nodes at higher levels in the lattice are larger—or at least as large—than those below.

The FM lattice alone, while useful for showing a FM, does not give insight into how the Choquet integral at (1) utilizes the lattice due to the π-permutation. Therefore, for a particular input we also show the path through the lattice followed by the Choquet integral. For example, suppose that a particular data sample x and hypothesis h gives rise to the permutation $\pi = \{2, 1, 3\}$. Then, for an arbitrary FM, the lattice visualization includes the path shown in Figure 2a. This visualization strategy allows us to summarize the FM as well as the Choquet integral's paths.

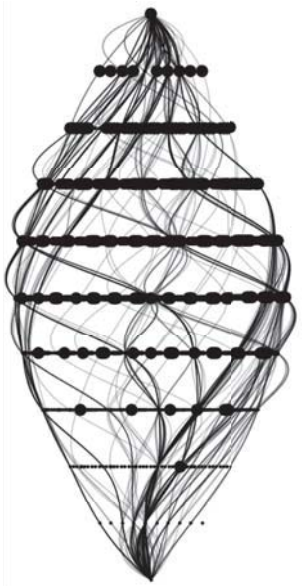
III. The DeFIMKL Algorithm

The DeFIMKL algorithm was introduced in [7] as a method of decision-level fusion in the context of classification, where a set of decisions from an ensemble of classifiers are non-linearly fused via the Choquet FI. To mathematically describe the algorithm, let the decision-value for feature-vector $\mathbf{x}_i$ from the kth classifier in an ensemble be $f_k(\mathbf{x}_i)$; the set of decisions from the ensemble comprise the evidence h for the Choquet integral. The evidence is then integrated with respect to the FM g, which encodes the relative worth of each classifier in the ensemble. This results in the ensemble decision $f_g(\mathbf{x}_i)$ for feature-vector $\mathbf{x}_i$ with respect to the FM g,

$$f_g(\mathbf{x}_i) = \sum_{k=1}^{m} f_{\pi(k)}(\mathbf{x}_i) \left[g(A_k) - g(A_{k-1}) \right], \quad (2)$$



(a) The path taken by the Choquet integral due to a single input inducing the permutation $\pi = \{2, 1, 3\}$. Note that the FM was arbitrarily defined in this example, and their distribution (ordering) follows that of Figure 1.



(b) Lattice of learned FM and paths for random training data from the Ionosphere data set using $m = 10$. Note there are numerous untouched nodes and their learned values are driven by the constraints in (9).

Fig. 2: Lattice visualization examples.

where $A_k = \{f_{\pi(1)}(\mathbf{x}_i), \ldots, f_{\pi(k)}(\mathbf{x}_i)\}$, such that $f_{\pi(1)}(\mathbf{x}_i) \geq f_{\pi(2)}(\mathbf{x}_i) \geq \ldots \geq f_{\pi(m)}(\mathbf{x}_i)$. This method has been explored in many previous works as a generalized classifier fusion method [24], [29]–[31].

The FM completely specifies the behavior of the Choquet integral. Thus, the next step in understanding the DeFIMKL algorithm is assigning a FM for the Choquet integral in (2), of which there are many methods. For example, the Sugeno λ-measure [15] may be naively used after specifying the FM values of the singletons; however, there is no guarantee that this choice of FM will yield acceptable results when used with (2) since it does not take training data into account. To address this, we suggested a data-driven method to learn the FM g through regularized *sum-of-squared error* (SSE) optimization in [32]. This method is summarized next.

Let the SSE be defined as

$$E^2 = \sum_{i=1}^{n} \left(f_g(\mathbf{x}_i) - y_i \right)^2. \tag{3}$$

It can be shown that (2), as a Choquet integral, can be reformulated as

$$f_g(\mathbf{x}_i) = \sum_{k=1}^{m} \left[f_{\pi(k)}(\mathbf{x}_i) - f_{\pi(k+1)}(\mathbf{x}_i) \right] g(A_k), \tag{4}$$

where $f_{\pi(m+1)} = 0$ [27]. We can then expand the SSE as

$$E^2 = \sum_{i=1}^{n} \left(H_{\mathbf{x}_i}^T \mathbf{u} - y_i \right)^2, \tag{5a}$$

where $\mathbf{u}$ is the lexicographically ordered FM g, i.e., $\mathbf{u} = (g(\{x_1\}), g(\{x_2\}), \ldots, g(\{x_1, x_2\}), g(\{x_1, x_3\}), \ldots, g(\{x_1, x_2, \ldots, x_m\}))$, and

$$H_{\mathbf{x}_i} = \begin{pmatrix} \vdots \\ f_{\pi(1)}(\mathbf{x}_i) - f_{\pi(2)}(\mathbf{x}_i) \\ \vdots \\ 0 \\ \vdots \\ f_{\pi(m)}(\mathbf{x}_i) - 0 \end{pmatrix}, \tag{5b}$$

where $H_{\mathbf{x}_i}$ is of size $(2^m - 1) \times 1$ and contains all the difference terms $f_{\pi(k)}(\mathbf{x}_i) - f_{\pi(k+1)}(\mathbf{x}_i)$ at the corresponding locations of A_k in $\mathbf{u}$. Finally, folding out the squared term in (5a) produces

$$\begin{aligned} E^2 &= \sum_{i=1}^{n} \left(\mathbf{u}^T H_{\mathbf{x}_i} H_{\mathbf{x}_i}^T \mathbf{u} - 2 y_i H_{\mathbf{x}_i}^T \mathbf{u} + y_i^2 \right) \\ &= \mathbf{u}^T D \mathbf{u} + \mathbf{f}^T \mathbf{u} + \sum_{i=1}^{n} y_i^2, \\ D &= \sum_{i=1}^{n} H_{\mathbf{x}_i} H_{\mathbf{x}_i}^T, \quad \mathbf{f} = -\sum_{i=1}^{n} 2 y_i H_{\mathbf{x}_i}. \end{aligned} \tag{6}$$

Since (6) is a quadratic function, we can add constraints on $\mathbf{u}$ such that it represents a FM, leading to a constrained QP. We can write the boundary and monotonicity constraints on $\mathbf{u}$ (see properties P4 and P5) as $C\mathbf{u} \leq 0$, where

$$C = \begin{pmatrix} \Psi_1^T \\ \Psi_2^T \\ \vdots \\ \Psi_{n+1}^T \\ \vdots \\ \Psi_{m(2^{m-1}-1)}^T \end{pmatrix} \tag{7}$$

and Ψ_1^T is a vector representation of the monotonicity constraint, $g\{x_1\} - g\{x_1, x_2\} \leq 0$. Hence, C is simply a matrix of $\{0, 1, -1\}$ values of size $(m(2^{m-1} - 1)) \times (2^m - 1)$ with the form

$$C = \begin{bmatrix} 1 & 0 & \cdots & -1 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \cdots & 0 & -1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}. \tag{8}$$

Thus, the full QP to learn the FM $\mathbf{u}$ is

$$\min_{\mathbf{u}} 0.5 \mathbf{u}^T \hat{D} \mathbf{u} + \mathbf{f}^T \mathbf{u}, \quad C\mathbf{u} \leq \mathbf{0}, \quad (\mathbf{0}, 1)^T \leq \mathbf{u} \leq \mathbf{1}, \tag{9}$$

where $\hat{D} = 2D$. Note that an additional regularization term can be included in the QP as

$$\min_{\mathbf{u}} 0.5 \mathbf{u}^T \hat{D} \mathbf{u} + \mathbf{f}^T \mathbf{u} + \lambda v_*(\mathbf{u}), \tag{10}$$

where λ is the regularization weight and $v_*(\cdot)$ is some regularization function. For example, ℓ_p-norm regularization is applied when $v_*(\mathbf{u}) = \|\mathbf{u}\|_p$. ℓ_1 and ℓ_2 regularization of this QP are discussed in [7], [32].

The QPs at (9) and (10) provide a method to learn the FM $\mathbf{u}$ (i.e., g) from training data, thus completing the requirements

for calculating the Choquet integral at (2). We now review how to use a kernel classifier to determine the decision-value $f_k(\mathbf{x}_i)$. Specifically, we will show how to use the SVM with this algorithm.

Suppose that each learner $f_k(\mathbf{x}_i)$ is a kernel SVM, each trained on a separate kernel K_k. The SVM classifier decision value is

$$\eta_k(\mathbf{x}) = \sum_{i=1}^{n} \alpha_{ik} y_i \kappa_k(\mathbf{x}_i, \mathbf{x}) - b_k, \tag{11}$$

which is interpreted as the distance of $\mathbf{x}$ from the hyperplane defined by the learned SVM model parameters, α_{ik} and b_k [33], [34]. The class label is typically computed as $\text{sgn}\{\eta_k(\mathbf{x})\}$,[3] which could be used as the training input to the FM learning at (6), however, we remap $\eta_k(\mathbf{x})$ onto the interval $[-1, +1]$ via the sigmoid function to create inputs for learning as

$$f_k(\mathbf{x}) = \frac{\eta_k(\mathbf{x})}{\sqrt{1 + \eta_k^2(\mathbf{x})}}. \tag{12}$$

Thus, the training data for DeFIMKL are $(\{K_k = [\kappa_k(\mathbf{x}_i, \mathbf{x}_j)], \mathbf{f}_k(X)\}, \mathbf{y})$, $k = 1, \ldots, m$, where K_k are the kernel matrices for each kernel function κ_k, $\mathbf{f}_k(X) = (f_k(\mathbf{x}_1), \ldots, f_k(\mathbf{x}_n))^T$ are the remapped SVM decision values, and $\mathbf{y} = (y_1, \ldots, y_n)$ are the ground-truth labels of $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, respectively; the output of the QP learner is the FM g. Algorithm 1 summarizes the training process. After training, a new feature vector $\mathbf{x}$—from a test data set—can be classified by via the procedure summarized in Algorithm 2.

Algorithm 1: DeFIMKL Classifier Training

Data: $(\mathbf{x}_i, y_i)$ - feature vector and label pairs; K_k - kernel matrices
Result: $\mathbf{u}$ - Lexicographically ordered fuzzy measure vector
for *each kernel matrix* **do**
  Compute the kernel SVM classifier decision values, η_k, as in (11).
  Remap the decision values onto the interval $[-1, +1]$ as f_k using (12).
Solve the minimization problem in (9) for the FM $\mathbf{u}$.

Algorithm 2: DeFIMKL Classifier Prediction

Data: $\mathbf{x}$ - feature vector; K_k - kernel matrices; $\mathbf{u}$ - learned fuzzy measure vector
Result: y - Predicted class label
Compute the SVM decision values $f_k(\mathbf{x})$ by using (11) and (12).
Apply the Choquet integral at (2) with respect to the learned FM $\mathbf{u}$.
Compute the class label as $y = \text{sgn}\{f_g(\mathbf{x})\}$.

A. FM Learning Behavior with Insufficient Training Data

Learning the entire FM for a DeFIMKL classifier utilizing m classifiers requires at least 2^m (or $2^m - 2$, observing the boundary conditions in property P4) *rank-independent* observations. Therefore, since so many rank-independent observations are rarely encountered in training data sets, there will likely be values of the FM that are not data-driven. Figure 2b shows an example of this in the wild, where the Ionosphere dataset [4] was used to train DeFIMKL with 10 classifiers. Note that there are many nodes in the lattice that are never "touched" by the training data; the learned values for these nodes is completely driven by the monotonicity constraints in the QP, the choice of regularization used, and the initialization used in the QP solver. It is therefore highly unlikely that the learned values at these nodes accurately represent the underlying FM, and if Algorithm 2 is applied to a new data point that utilizes one or more of the untouched nodes, prediction accuracy will suffer. The following contrived example demonstrates the behavior of the ℓ_2-regularized DeFIMKL algorithm with insufficient training data.

TABLE I: Underlying and learned FMs (excluding $g(\{\emptyset\})$ and $g(X)$ whose values are 0 and 1, respectively, due to the boundary conditions).

		Regularization		
FM Term	**Underlying**	**ℓ_2 (min)**	**max**	**mean**
$g(\{x_1\})$	0.14	0.14	0.19	0.14
$g(\{x_2\})$*	0.29	0.00017	0.93	0.33
$g(\{x_3\})$	0.43	0.43	0.44	0.43
$g(\{x_1, x_2\})$*	0.57	0.14	1	0.67
$g(\{x_1, x_3\})$	0.71	0.69	0.71	0.71
$g(\{x_2, x_3\})$	0.86	0.83	0.93	0.86

*FM terms marked with an asterisk are not addressed by the training data.

Example 1. *Learning an Underdetermined FM via ℓ_2-regularized DeFIMKL.* A three-SVM ℓ_2-regularized DeFIMKL algorithm (i.e., $m = 3$, however these results are also indicative of the behavior when $m > 3$) is trained with a synthetic dataset that purposefully avoids two nodes in the fuzzy lattice and was generated using the underlying FM shown in Table I; the underlying FM was arbitrarily assigned. The FM learned by the DeFIMKL algorithm is also shown in Table I. Note that two nodes in the lattice, corresponding to $g(\{x_2\})$ and $g(\{x_1, x_2\})$ were not driven by the training data, and thus are essentially driven by the monotonicity constraints.

What we see is that all nodes touched by the training data (i.e., nodes traversed by the Choquet integral) are learned successfully with minimal error (well within 5%). However, the two nodes untouched by the training data are assigned values based on monotonicity constraints. The node corresponding to $g(\{x_2\})$ gets a value of essentially 0, satisfying the monotonicity constraint that $g(\{\emptyset\}) \leq g(\{x_2\}) \leq min(g(\{x_1, x_2\}), g(\{x_2, x_3\}))$, and the node corresponding to $g(\{x_1, x_2\})$ gets a value of 0.14 to satisfy the constraint $max(g(\{x_1\}), g(\{x_2\})) \leq g(\{x_1, x_2\}) \leq g(\{X\})$. Note that in both of these cases, the learned FM value is essentially the minimum value permitted by the monotonicity constraints. This, as will be shown in the following section, is due to the ℓ_2-regularization of the DeFIMKL algorithm.

IV. FM Learning with a Goal

The standard DeFIMKL algorithm discussed in the previous section assumes that the structure of the underlying FM is not known, thus no information regarding the underlying FM

[3] Note that the $\text{sgn}(\cdot)$ function discards information about how well the kernel separates the classes of data.

[4] Retrieved from UCI Machine Learning Repository. Available online at http://archive.ics.uci.edu/ml

TABLE II: Classification Accuracy of Various Regularization Functions*

	Data Set						
Regularization	Sonar	Derm	Ecoli	Glass	Toy 3	Toy 5	Toy 8
None	**80.5 (5.63)**	94.3 (2.61)	**97.3 (1.90)**	91.2 (4.39)	84.7 (6.48)	**95.0 (3.39)**	**98.4 (1.76)**
ℓ_1	**78.4 (7.23)** $\lambda = 0.5$	89.6 (4.35) $\lambda = 0.5$	91.8 (2.79) $\lambda = 5$	82.7 (6.77) $\lambda = 0.5$	64.4 (7.23) $\lambda = 2.5$	91.0 (4.87) $\lambda = 0.5$	**96.9 (2.74)** $\lambda = 5$
ℓ_2 (min)	**80.0 (6.72)** $\lambda = 0.5$	91.9 (3.09) $\lambda = 0.5$	91.8 (2.79) $\lambda = 0.5$	85.9 (5.79) $\lambda = 0.5$	64.2 (7.05) $\lambda = 2.5$	92.4 (3.88) $\lambda = 0.5$	91.4 (4.36) $\lambda = 5$
max	72.0 (7.58) $\lambda = 0.5$	**97.4 (1.88)** $\lambda = 1.5$	**97.9 (1.91)** $\lambda = 4.5$	**94.2 (3.97)** $\lambda = 4.5$	88.5 (6.35) $\lambda = 1.5$	94.3 (2.84) $\lambda = 2$	**98.9 (1.61)** $\lambda = 2.5$
mean	**76.6 (7.17)** $\lambda = 0.5$	**97.7 (1.49)** $\lambda = 3$	**97.1 (2.14)** $\lambda = 1.5$	**95.2 (3.07)** $\lambda = 1$	**94.8 (4.50)** $\lambda = 1.5$	**96.7 (2.30)** $\lambda = 5$	**98.4 (1.89)** $\lambda = 1$

*Bold indicates best result according to a two-valued t-test at a 5% significance level.

is encoded in the QP. If, however, the FM is partially known, the QP at (10) should include that information. To this end, we propose the regularization function

$$v_*(\mathbf{u}) = \lambda \|\mathbf{u} - \mathbf{g}\|_p^2, \tag{13}$$

where $\mathbf{g}$ represents a goal of what we expect the FM to look like. Including this regularization function in the QP (with $p = 2$) gives

$$\min_{\mathbf{u}} 0.5\mathbf{u}^T \hat{D}\mathbf{u} + \mathbf{f}^T\mathbf{u} + \lambda\|\mathbf{u} - \mathbf{g}\|_2^2, \tag{14}$$

and the QP then also simultaneously minimizes the Euclidean distance between the learned FM $\mathbf{u}$ and the goal $\mathbf{g}$. Expanding the regularization term in (14) leads to

$$\min_{\mathbf{u}} 0.5\mathbf{u}^T \left(\hat{D} + \lambda I\right)\mathbf{u} + (\mathbf{f} - 2\lambda\mathbf{g})^T\mathbf{u}, \tag{15}$$

showing that the inclusion of this regularization function still results in a valid QP, though this comes as no surprise since the regularization function in (13) is quadratic in $\mathbf{u}$.

A. ℓ_2-regularization: Minimum Aggregation

It is interesting to note that when $\mathbf{g} = 0$, the regularization function in (13) reduces to that of ℓ_p-norm regularization of the FM vector. This is precisely why the learned FM's untouched nodes of last section's example "default" to lie at the lowest end of their allowable range as shown in Table I—we are essentially forcing the untouched FM values to be as close to zero as possible through our choice of ℓ_2-norm regularization. Tying this with the aggregation operators discussed in Section II-C, we recognize that when $\mathbf{g} = 0$ we are forcing the Choquet integral to aggregate like the minimum function.

B. Maximum Aggregation

Defining the goal as all 1s causes the untouched nodes to default to the maximum end of their allowable range, tuning the Choquet integral's behavior to that of maximum aggregation (see Section II-C). Rerunning the example in Section III-A with this goal yields the FM summarized in Table I, where it is obvious that the untouched nodes are assigned the maximum possible value permitted by the monotonicity constraints. Note that in this example the learned FM values for $g(\{x_1\})$ and $g(\{x_2, x_3\})$ have been pushed farther from the underlying FM, though they still lie fairly close. This discrepancy is due to the choice of λ, which essentially "tunes" where the error is incurred in the QP at (14)–a larger value of λ will force the learned FM to look like the goal $\mathbf{g}$ despite perturbing the data-driven nodes away from their underlying values. λ was arbitrarily set to 1 in these experiments.

C. Mean Aggregation

As a final example, we define the goal of the FM to be that of mean aggregation as explained in Section II-C. Doing so leads to the learned FM shown in Table I. Interestingly, the learned FM at the data-driven nodes is more accurate than that of the previous case of maximum aggregation. We attribute this to the fact that the goal of mean aggregation is more similar to the underlying FM than the goal of maximum aggregation.

V. Experiments

Experiments were performed using no regularization, ℓ_p-norm regularization, and the goal-based regularization function in (13) with the DeFIMKL algorithm on various datasets from the UCI Machine Learning repository as well as toy datasets generated to purposefully exclude 80% of the training of nodes in an arbitrarily generated fuzzy lattice (three toy datasets were generated using 3, 5, and 8 densities, respectively). Each experiment consists of 100 trials, where in each trial a random partition of 80% of the data is used for training and the remaining data is sequestered for testing; the results we report comprise the mean and standard deviation of classification accuracies. Finally, we vary the regularization parameter, λ, to explore its effect on classification accuracy and the results with the best λs are reported; so, essentially we are comparing the best from each algorithm.

A. Results

Table II summarizes the results of these experiments. The best algorithms for each dataset are shown in bold font; a two-sample t-test at a 5% significance level is used to determine the statistically best results—hence, more than one algorithm can be considered as best. In all experiments at least one goal-based regularization function emerges as a top performer. We also find that the max and mean goal-based regularization functions achieve superior results on the *Dermatology* and *Glass* datasets, suggesting that the data define an underlying FM that is most similar to mean or max aggregation. There is no clear trend in the results versus the regularization parameter λ, and not surprisingly the best selection of λ varies based on the dataset used.

VI. Conclusion

This paper first introduced a visualization technique that shows both the FM as well as the Choquet integral's path through the lattice. We also proposed and applied a new regularization function to our previously developed decision-level aggregation algorithm known as DeFIMKL. Including this new regularization function in the DeFIMKL algorithm allows knowledge of an underlying FM to be encoded into

the algorithm's training procedure; thus, the user can define a particular goal for the FM before learning. We discussed the application of the new regularization function and demonstrated its behavior using synthetic and real-world datasets where we found that tuning the Choquet integral's behavior to that of max or mean aggregation tended to do best across all datasets.

A. Future Work

The regularization extension proposed in this paper allows knowledge of an underlying FM to be encoded into the learning process of DeFIMKL, though we acknowledge the fact that the underlying FM is typically not known. We also previously demonstrated, however, that much of the underlying FM can be learned with very little error as long as the learning is data-driven, i.e., cases where there is sufficient training data [7]. Thus, future work is focused on extending the proposed idea of goal-based regularization to goal-based learning of the underlying FM at nodes not utilized by the Choquet integral during training, i.e., values of the underlying FM not attainable from the training data.

ACKNOWLEDGMENT

This work is funded in part by the Army Research Office grant W911NF-16-1-0017 and W911NF-16-1-0241. Dr. Anderson is partially funded by Army Research Office Grant W911NF-14-1-0673. Superior, a high performance computing cluster at Michigan Technological University, was used in obtaining some of the results presented in this publication.

REFERENCES

[1] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, "Non-sparse multiple kernel learning," in *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[2] Z. Xu, R. Jin, H. Yang, I. King, and M. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. Int. Conf. Machine Learning*, 2010, pp. 1175–1182.

[3] C. Cortes, M. Mohri, and A. Rostamizadeh, "ℓ_2 regularization for learning kernels," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 109–116.

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

[5] L. Hu, D. Anderson, T. Havens, and J. Keller, "Validity of different fuzzy integrals and representations for multiple kernel aggregation," in *Proc. Int. Conf. Info. Processing and Management of Uncertainty in Knowledge-Based Systems*, 2014.

[6] 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, pp. 1–1, 2016.

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

[8] M. Grabisch, I. Kojadinovic, and P. Meyer, "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*, vol. 186, no. 2, pp. 766 – 785, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221707002330

[9] 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.

[10] G. Beliakov, "Construction of aggregation functions from data using linear programming," *Fuzzy Sets and Systems*, vol. 160, pp. 65–75, 2009.

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

[12] C. Wagner, D. Anderson, and T. Havens, "Generalization of the fuzzy integral for discontinuous interval- and non-convex interval fuzzy set-valued inputs," in *IEEE International Conference on Fuzzy Systems*, July 2013, pp. 1–8.

[13] 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.

[14] D. Anderson, T. Havens, C. Wagner, J. Keller, M. Anderson, and D. Wescott, "Sugeno fuzzy integral generalizations for sub-normal fuzzy set-valued inputs," in *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, June 2012, pp. 1–8.

[15] M. Grabisch, *Fuzzy Measures and Integrals: Theory and Applications*, M. Sugeno and T. Murofushi, Eds. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2000.

[16] D. J. Dubois, *Fuzzy sets and systems: theory and applications*. Academic press, 1980, vol. 144.

[17] M. F. Anderson, D. T. Anderson, and D. J. Wescott, "Estimation of adult skeletal age-at-death using the sugeno fuzzy integral," *American journal of physical anthropology*, vol. 142, no. 1, pp. 30–41, 2010.

[18] A. Mendez-Vazquez and P. Gader, "Sparsity promotion models for the choquet integral," in *IEEE Symposium on Foundations of Computational Intelligence*. IEEE, 2007, pp. 454–459.

[19] C. Wagner and D. T. Anderson, "Extracting meta-measures from data for fuzzy aggregation of crowd sourced information," in *IEEE International Conference on Fuzzy Systems*. IEEE, 2012, pp. 1–8.

[20] T. C. Havens, D. T. Anderson, C. Wagner, H. Deilamsalehy, and D. Wonnacott, "Fuzzy integrals of crowd-sourced intervals using a measure of generalized accord," in *IEEE International Conference on Fuzzy Systems*. IEEE, 2013, pp. 1–8.

[21] T. Havens, D. Anderson, and C. Wagner, "Data-informed fuzzy measures for fuzzy integration of intervals and fuzzy numbers," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, pp. 1–1, 2014.

[22] M. Grabisch, "Fuzzy integral for classification and feature extraction," in *Fuzzy Measures and Integrals: Theory and Applications*. Springer-Verlag New York, Inc., 2000, pp. 415–434.

[23] J. Keller, P. Gader, and A. Hocaoglu, "Fuzzy integral in image processing and recognition," in *Fuzzy Measures and Integrals: Theory and Applications*. Springer-Verlag New York, Inc., 2000, pp. 435–466.

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

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

[26] T. Murofushi and M. Sugeno, "An interpretation of fuzzy measures and the choquet integral as an integral with respect to a fuzzy measure," *Fuzzy sets and Systems*, vol. 29, no. 2, pp. 201–227, 1989.

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

[28] M. Grabisch, "Fuzzy integral in multicriteria decision making," *Fuzzy sets and Systems*, vol. 69, no. 3, pp. 279–298, 1995.

[29] H. Tahani and J. M. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 3, pp. 733–741, 1990.

[30] X. Wang, A. Chen, and H. Feng, "Upper integral network with extreme learning mechanism," *Neurocomputing*, vol. 74, no. 16, pp. 2520–2525, 2011.

[31] J. Zhai, H. Xu, and Y. Li, "Fusion of extreme learning machine with fuzzy integral," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 21, no. 2, pp. 23–34, 2013.

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

[33] C. Cortes and V. N. Vapnik, "Support-vector networks," in *Machine Learning*, vol. 20, no. 3, 1995, pp. 273–297.

[34] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.