Efficient and Scalable Nonlinear Multiple Kernel Aggregation Using the Choquet Integral

Lequn Hu[1], Derek T. Anderson[1], Timothy C. Havens[2], and James M. Keller[3]

[1] Electrical and Computer Engineering, Mississippi State University, USA
[2] Electrical and Computer Engineering and Computer Science, Michigan Technological University, USA
[3] Electrical and Computer Engineering, University of Missouri, USA
lh432@msstate.edu, anderson@ece.msstate.edu, thavens@mtu.edu, kellerj@missouri.edu

Abstract. Previously, we investigated the definition and applicability of the *fuzzy integral* (FI) for nonlinear *multiple kernel* (MK) aggregation in pattern recognition. Kernel theory provides an elegant way to map multi-source heterogeneous data into a combined homogeneous (implicit) space in which aggregation can be carried out. The focus of our initial work was the Choquet FI, a per-matrix sorting based on the quality of a base learner and learning was restricted to the Sugeno λ-*fuzzy measure* (FM). Herein, we investigate what representations of FMs and FIs are valid and ideal for nonlinear MK aggregation. We also discuss the benefit of our approach over the linear convex sum MK formulation in machine learning. Furthermore, we study the Möbius transform and k-additive integral for scalable *MK learning* (MKL). Last, we discuss an extension to our *genetic algorithm* (GA) based MKL algorithm, called FIGA, with respect to a combination of multiple *light weight* FMs and FIs.

Keywords: Fuzzy integral, fuzzy measure, Möbius transform, multiple kernel learning, heterogeneous data fusion.

1 Introduction

Explosive growth in sensing and computing has given rise to numerous technological and mathematical dilemmas. Two well-known examples are *big data* and data diversity. Herein, we focus on the latter but in the context of a framework that can address the former as well. Consider the humanitarian effort of demining; the automatic identification and removal of hazards such as landmines and *improvised explosive devices* (IEDs) [1, 2]. These devices are responsible for injuring and claiming the lives of thousands of soldiers and civilians. The problem is that no single sensor or algorithm solves this challenging task. Instead, multiple sensors, multiple features, multiple algorithms and even human interaction/input is often critical for robust detection in different environments. However, information arising from multiple sensors, algorithms and people often result in great diversity, such as mixed-data type, multi-resolution (e.g., spatial and temporal),

A. Laurent et al. (Eds.): IPMU 2014, Part I, CCIS 442, pp. 206–215, 2014.

mixed-uncertainty, etc. An important question is, what is a well-grounded (non ad-hoc) way to carry out pattern analysis in light of such heterogeneity? This challenge is in no-way restricted to humanitarian demining. Another engineering example is combining multi-sensor, multi-band, multi-algorithm and even high-level human knowledge for wide area motion image analysis or earth observations using unmanned aerial vehicles. The point is, numerous challenges require the fusion and subsequent analysis of multi-source disparate data.

In prior work, we investigated the definition and applicability of the *fuzzy integral* (FI) for nonlinear *multiple kernel* (MK) aggregation in pattern recognition [3]. Kernel theory provides an elegant way to map multi-source heterogeneous data into a combined homogeneous (implicit) space where well-defined aggregation can be performed. The focus of our initial work was the Sugeno and Choquet FIs, a per-matrix sorting based on the quality of a base learner, e.g., a *support vector machine* (SVM), and learning was restricted to a Sugeno λ-*fuzzy measure* (FM). However, numerous questions remain: what types or representations of FIs are valid; are some representations better than others; what is an effective way to learn the FM for problems involving a large number of inputs; and what is the actual benefit of the FI for MK aggregation in comparison to other state-of-the-art MK techniques? Herein, we investigate these challenges.

This article is organized as follows. First, the FM, FI and the Möbius transform are reviewed. Next, MK aggregation and our prior *fuzzy integral MK learning* (FIMKL) work is described. The selection and representation of FI for nonlinear MK aggregation is then explored. Next, we investigate importance differences between FIMK and the popular machine learning *linear convex sum* (LCS) MKL formulation. The Möbius transform and the k-additive integral are then explored for efficient and scalable MKL. Last, we discuss the utilization of a combination of different *light weight* FMs in the context of our prior FIMKL *genetic algorithm* (GA) MKL algorithm, called FIGA.

2 Fuzzy Measure and Integral

The fusion of information using the Sugeno or Choquet FI has a rich history [4–8]. Depending on the problem domain, the input can be experts, sensors, features, similarities, pattern recognition algorithms, etc. The FI is defined with respect to the FM, a monotone (and often normal) measure. With respect to a set of m information sources, $X = \{x_1, ..., x_m\}$, the FM encodes the (often subjective) *worth* of each subset in 2^X.

Definition 1 (Fuzzy Measure). For a finite set of sources, X, the FM is a set-valued function $g : 2^X \to [0, 1]$ with the following conditions:

1. (Boundary condition) $g(\phi) = 0$,
2. (Monotonicity) If A, B $\subseteq X$ with A $\subseteq$ B, then $g(A) \leq g(B)$.

Note, if X is an infinite set, there is a third condition guaranteeing continuity and we often assume $g(X) = 1$ as usual (although it is not necessary in general).

Numerous FI formulations have been proposed [4, 7, 9] for generalizability, differentiability and to address different types of uncertain data. Herein, we stick to the *conventional* (real-valued integrand and measure) Choquet integral.

Definition 2 (Difference-in-Measure form of Choquet FI). For a finite set of m sources, FM g, and integrand $h : X \to \Re^+$, the discrete Choquet FI is

$$\int h \circ g = \sum_{i=1}^{m} \omega_i h\left(x_{\pi(i)}\right), \tag{1}$$

where $\omega_i = \left(G_{\pi(i)} - G_{\pi(i-1)}\right)$, $G_{\pi(i)} = g\left(\{x_{\pi(1)}, ..., x_{\pi(i)}\}\right)$, $G_{\pi(0)} = 0$, and $\pi(i)$ is a sorting on X such that $h(x_{\pi(1)}) \geq \ldots \geq h(x_{\pi(m)})$.

Numerous measures exist, e.g., the Sugeno λ-FM, S-decomposable measures, possibility (and necessity) measures and k-additive FMs. The literature contains numerous ways to estimate their parameters from data, e.g., [10].

2.1 Möbius Transform

While the difference-in-measure and difference-in-integrand formulations are common, the FI can also be represented in terms of the Möbius transformation. The Möbius transform is of particular use for compactly expressing different formulas, e.g., Shapley and the k-additive measure/integral [11].

Definition 3 (Möbius Transform). The Möbius transformation of a FM g is

$$\mathcal{M}(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} g(B), \qquad \forall A \subseteq X. \tag{2}$$

Note, the Möbius transformation is invertible via the Zeta transform,

$$g(A) = \sum_{B \subseteq A} \mathcal{M}(B), \qquad \forall A \subseteq X. \tag{3}$$

Definition 4 (Möbius Transform Representation of the Choquet FI). The Möbius transformation representation of the Choquet FI is

$$\int h \circ g = \sum_{A \subseteq X} \mathcal{M}(A) \bigwedge_{x_i \in A} h(x_i). \tag{4}$$

Remark 1. Equation (4) does not require sorting like Equation (1).

2.2 Multiple Kernel

In this section, we review basic concepts and definitions of kernels [12] needed for FIMK. First, assume that from each source of information in X, we measure a feature vector $\mathbf{x}$, where $\mathbf{x}_i$ describes the source x_i.[1]

Definition 5 (Kernel). Suppose we have a feature mapping $\phi : \mathcal{R}^d \rightarrow \mathcal{R}^{\mathcal{H}}$, where d is the dimensionality of the input space and $\mathcal{R}^{\mathcal{H}}$ is a (higher-)dimensional space called the *Reproducing Kernel Hilbert Space* (RKHS). A kernel is the inner product function $\kappa : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$, which represents the inner-product in $\mathcal{R}^{\mathcal{H}}$,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle , \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}. \tag{5}$$

Definition 6 (Mercer Kernel). Assume a kernel function κ and finite data $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$. The $n \times n$ matrix $K = [K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)]$, $i, j \in \{1, ..., n\}$, is a Gram matrix of inner products. Thus, K is symmetric and *positive semi-definite* (PSD), $\mathbf{x}_i^T K \mathbf{x}_i \geq 0$, $\forall \mathbf{x}_i \in \mathcal{X}$. Since K is PSD, all eigenvalues are non-negative.

This framework is particularly attractive for heterogeneous data fusion. Specifically, measuring similarity can be tailored to each individual domain. For example, if an input is a graph then a graph-based similarity kernel can be used. If another input is a histogram, then a kernel like the intersection kernel can be used. In effect, kernel theory provides a convenient way to map heterogeneous data into a homogeneous (implicit) space. In terms of aggregation, this is ideal. Aggregation can be engaged in the resultant homogeneous space to fuse multiple inputs. Furthermore, for homogeneous spaces, such as feature-valued data, kernel fusion is an elegant way to help mitigate challenges like feature vector size imbalance and feature vector scaling for the common method of combining feature-valued data (concatenation or *arraying*).

For readability, from now on we will denote features that are projected into the RKHS $\mathcal{R}^{\mathcal{H}}$ as ϕ_i, where ϕ_i is equivalent to $\phi(\mathbf{x}_i)$. Furthermore, we will assume that each source in X has either multiple features extracted from it or multiple kernels computed from a feature. We will use a super-script in this situation, viz., $\mathbf{x}_i^k$ is the feature vector describing source x_i that is used to compute the kth kernel matrix K_k; thus, we use ϕ_i^k to denote $\phi(\mathbf{x}_i^k)$.

Remark 2. Let $K_1 = [\kappa_1(\mathbf{x}_i, \mathbf{x}_j)]$ and $K_2 = [\kappa_2(\mathbf{x}_i, \mathbf{x}_j)]$, $i, j \in \{1, ..., n\}$. The set of Mercer kernels is closed under the following (non-exhaustive) set of operations for $i, j \in \{1, ..., n\}$ [12]: $\mathcal{K}_{ij} = (K_1)_{ij} + (K_2)_{ij}$, $\mathcal{K}_{ij} = c(K_1)_{ij}, \forall c \geq 0$, $\mathcal{K}_{ij} = (K_1)_{ij} + c, \forall c \geq 0$, $\mathcal{K}_{ij} = (K_1)_{ij}(K_2)_{ij}$.

[1] Later, we will extend this idea to the scenario where one can have multiple feature vectors (really, multiple kernels) per source. This can manifest in one (or a combination) of two ways: i) different kernels can be computed from one feature type, or ii) a kernel can be computed from each of multiple heterogeneous feature types.

Kernel design takes a number of forms. In most scenarios, a single kernel is employed. Its parameters are generally specified manually or the selections of the kernel and its parameters are learned from data. However, the most widespread practice is to experimentally choose from a finite number of kernels and associated kernel parameters to pick a winner. In recent years, MK has been proposed to address the problem of finding a "best" kernel for a data set, and more recently, for focusing on the fusion and unique transformation of different sources (e.g., sensors and/or features). Specifically, MKL has emerged to find such kernel combinations automatically. Research has shown that the construction of a kernel from a number of base kernels allows for a more flexible encoding of domain knowledge from different sources or cues.

Linear aggregation is the strategy employed by most [13, 14]. It is based on a (potentially weighted) summation of base kernels, $\mathcal{K} = \sum_{k=1}^{m} \omega_k K_k$, where ω are weights, and $K_k = [\kappa_k(\mathbf{x}_i^k, \mathbf{x}_j^k)]$ is the kernel matrix produced by the kth feature extracted from the sources X. Many search for a single aggregate kernel to transform x_i and x_j, thus one kernel function κ is applied to all features extracted from x_i and x_j. However, better performance may be obtained by placing each feature in its own space, viz., the kth feature vector gets its own kernel function κ_k. Approaches to MK aggregation differ in the way that restrictions are placed on the weights ω. The most common categories include the linear sum ($\omega_k \in \mathcal{R}$), conic sum ($\omega_k \in \mathcal{R}^+$), and convex sum ($\omega_k \in \mathcal{R}^+$ and $\sum_{k=1}^{m} \omega_k = 1$). Compared to linear sum, conic and convex sums are appealing because they lead to weight (thus kernel and/or source) importance interpretation.

A few nonlinear aggregations methods have been proposed. However their representations and expressiveness are extremely limited and they are difficult to interpret. In [3], we investigated the Sugeno and Choquet FIs for MK. We proved that per-element aggregation is not theoretically valid. Semantically, per-element does make much sense, but mathematically—viz., in terms of production of a Mercer kernel—it is not valid (a counter proof was provided with respect to the maximum operator and negative eigenvalues). Instead, we proposed an alternative solution based on sorting at the *matrix level*. Assume each kernel matrix K_k has a numeric "quality." As we showed, this can be computed, for example, by computing the classification accuracy of a base-learner that uses kernel K_k (or by a learning algorithm such as a genetic algorithm). Let $\nu_k \in [0, 1]$ be the kth kernels *quality*. These qualities can be sorted, $\nu_{(1)} \geq \nu_{(2)} \geq ... \geq \nu_{(m)}$.

Definition 7 (Difference-in-Measure Choquet FI for MK Aggregation). Given m base Mercer kernels, $\{\kappa_1, \ldots, \kappa_m\}$, FM g and a sorting $\nu_{(1)} \geq \nu_{(2)} \geq ... \geq \nu_{(m)}$, the difference-in-measure Choquet FI

$$\mathcal{K}_{ij} = \sum_{k=1}^{m} (G_{\pi(k)} - G_{\pi(k-1)})(K_{\pi(k)})_{ij} = \sum_{k=1}^{m} \omega_k (K_{\pi(k)})_{ij}, \quad i, j \in \{1, ..., n\}, \quad (6)$$

produces a Mercer kernel as multiplication by positive scalar and addition are PSD preserving operations [3]. Since Equation (6) involves per-matrix sorting it can be compactly wrote in a simpler (linear algebra) form, $\mathcal{K} = \sum_{k=1}^{m} \omega_k K_{\pi(k)}$.

3 FIMK Insight and Scalable MKL

In this section, we address FIMKL expressibility and scalability. The first topic is how is FIMK different from the LCS MK formulation and why does it outperform other methods from machine learning? The second topic is a search for appropriate ways to represent FIMK to restrict its operation as the number of free parameters grow exponentially with respect to the number of inputs. This is significant in MKL as many explore the use of a relatively large number of kernels. A kernel can be applied to each sensor/source, feature index, or group (i.e., each bin in a histogram of gradients or the full descriptor). In addition, multiple kernels can be used for each of the above and different parameters are often explored. The point is, scalability is an important element of MKL.

3.1 Comparison of FIMK to Linear Convex Sum MK

The majority of machine learning MK research uses LCS form. It is often desired due to its advantage in optimization for MKL. One example is Bach's SMO-like algorithm for tackling convex and smooth minimization problems [15].

Remark 3. The weights in equations (1) and (6) are positive, $w_i \geq 0$, by definition (monotonicity) and their sum is 1, i.e.,

$$\sum_{k=1}^{m} w_k = \left(G_{\pi(m)} - G_{\pi(m-1)}\right) + ... + \left(G_{\pi(1)} - G_{\pi(0)}\right) = G_{\pi(m)} - G_{\pi(0)} = 1.$$

Both FIMK and LCS MK are type convex sum, i.e., $w_k \in \Re_+^m$ and $\sum_{k=1}^{m} w_k = 1$. However, one is linear, the other is not, and the weights are derived from the FM. The Choquet FI is capable of representing a much larger class of aggregation operators. For example, it is well known that the Choquet FI can produce, based on the selection of FM, the maximum, minimum, *ordered weighted average* (OWA), family of order statistics, etc. However, the machine learning LCS form is simply m weights anchored to the individual inputs. The LCS is a subset (one of the aggregation operators) of the Choquet FI.

In [3], we reported improved SVM accuracies and lower standard deviations over the state-of-the-art, *MKL group lasso* (MKLGL) [14], on publically available benchmark data. We put forth a *genetic algorithm* (GA), called FIGA, based on learning the densities for the Sugeno λ-FM. An important question not addressed in our initial work is why exactly does FIMK perform notably better than LCS and, more specifically, MKLGL? Herein, we consider two possibilities. First is the expressibility of FIMK in terms of aggregation. Simply put, FIMK is nonlinear and more expressive, i.e., it can represent a much wider class of aggregation operators that can be specified or learned from data. Second is the learning algorithm, i.e., FIGA versus MKLGL. This is a more difficult topic to tackle mathematically. These two optimization algorithms operate in extremely different ways. Group lasso is an advanced approach designed to work on constrained problems, LCS type formulations. While it is relatively efficient and

mathematically elegant, it is simple to envision problems for which LCS is a inferior solution to a nonlinear or higher-order polynomial solution. On the other hand, GAs are a family of meta-heuristic optimization techniques that can operate on extremely challenging optimization surfaces. They exist to help avoid, potentially, pitfalls of early convergence to local minimia relative to a given initialization. Arguments can be made for both, i.e., efficiency versus expressibility. Globally, it is not likely this question has an answer (one approach *better* than the other). At the end of this article we show preliminary results for reducing FIGA to learning a simpler LCS solution to (empirically) understand what impact optimization had on our previous SVM-based experiments. However, first we investigate an additional compact way of representing the FI for MKL.

3.2 k-additive Choquet Integral for FIMK

In this section, we explore a definition of the Choquet FI for MK aggregation under the Möbius transform. If we attempt to directly analyze Equation (4) in terms of a valid Mercer kernel producing aggregation operator then we run into the same problem as before of per-element sorting and subsequently the existance of a FM of all ones ($g(A) = 1, \forall A \subseteq X$ such that $A \neq \phi$) that results in the maximum (known to not preserve PSD). Furthermore, the Möbius transform values can be positive, zero or negative. We know multiplication by positive scalars results in preservation of PSD, but we cannot guarantee PSD preservation in general for multiplication by negative numbers. Note, based on our proposed matrix-level sorting with respect to a base learner, this condition also arises for the common difference-in-integrand form of the Choquet FI and MK aggregation. However, we proved that the difference-in-measure form does indeed guarantee PSD preservation. Furthermore, we know that the difference-in-measure and difference-in-integrand are equivalent, they can algebraically be rewritten in terms of one another. We also know the Möbius transform form is equilivant to the difference-in-measure form of the Choquet FI. Two representations do not make it clear if the Choquet FI is valid with respect to per-matrix sorting. However, we proved (see [3]) one of these three forms and the other two are valid as well as they are simply re-formulations of one another.

Specifically, Equation (4) is not guaranteed to be valid because minimum is performed on a per-element basis (counter proof is trivial). We consider a slight reformulation of the Möbius transform, in combination with k-additivity, for MK aggregation that preserves the PSD property of matrices. Here, k-additivity is explored as it limits interaction between subsets to size $|A| \leq k, A \subseteq X$.

Definition 8 (k-additive FI for MK). The k-additive form of the Choquet FI for MK aggregation on a per-matrix basis (in terms of linear algebra) is

$$\mathcal{K} = \sum_{A \subseteq X, |A| \leq k} \mathcal{M}(A) \bigwedge_{x_i \in A} K_i, \tag{7}$$

where $\bigwedge$ is a per-matrix operator with respect to the associated $\nu_{(i)}$ values.

The main reason for considering the Möbius transform, outside of academic curiosity, is MKL. In MKL, it is often the case that we consider a large number of kernels. Different kernels for different sensors, features, feature groups, sets of parameters for kernels, etc. This presents a serious challenge for FIMK. Namely, FIMK is driven by the FM and the FM is exponential in the number of inputs. For example, [14] considered 793 kernels (different kernels for each individual feature and different kernels for the entire feature vector). While some might consider this rather extreme, such an approach would result in an unsolvable (intractable) problem if the full lattice was desired in FIMK, i.e., $2^{793} - 2$ free parameters. Constraints need be imposed to make FIMK scalable in this domain. For example, the 2-additive FI only requires $m + \binom{m}{2}$ parameters, where $\binom{m}{2}$ is the number of $2-$combinations. Regardless, this problem is still constrained by the monotonicity constraints and boundary conditions.

Example 1 (Maximum Operator). Consider a FM in which $g(A) = 1, \forall A \subseteq X$. Furthermore, let $m = 3$. One obtains $\mathcal{M}(\phi) = 0$, $\mathcal{M}(x_i) = 1$, $\mathcal{M}(\{x_i, x_j\}) = -1$ and $\mathcal{M}(X) = 1$. With respect to Equation (7), we get

$$\begin{aligned}\mathcal{K} &= \mathcal{M}(x_1)K_1 + ... + \mathcal{M}(\{x_1, x_2\})\,(K_1 \wedge K_2) + ... + (K_1 \wedge K_2 \wedge K_3) \\ &= K_1 + K_2 + K_3 - (K_1 \wedge K_2) - (K_1 \wedge K_3) - (K_2 \wedge K_3) + (K_1 \wedge K_2 \wedge K_3)\,.\end{aligned}$$

Furthermore, for a base learner sorting like $\nu_3 \geq \nu_2 \geq \nu_1$,

$$\mathcal{K} = K_1 + K_2 + K_3 - K_1 - K_1 - K_2 + K_1 = K_3,$$

i.e., we get the *maximum* (with respect to our base learners) of our kernels.

Example 2 (Average). Consider m sources and a FM in which $g(\phi) = 0$, $g(X) = 1$, and $g(A) = \frac{|A|}{|X|}$ for $A \subset X \setminus \phi$. We obtain $\mathcal{M}(\phi) = 0$, $\mathcal{M}(x_i) = \frac{1}{|X|}$, $\mathcal{M}(A) = 0$ for $A \subseteq X, |A| > 1$, and thus $\mathcal{K} = \frac{1}{m}K_1 + ... + \frac{1}{m}K_m$.

4 Multi-measure FIGA and Preliminary Results

The primary purpose of this article is to explore different *compact* representations of the FM and FI for MKL. We experimentally investigate if there is any observable benefit for SVM-based classification in pattern classification. Our results are compared to the LCS form and MKLGL on public domain benchmark data. Specifically, we explore an extension to FIGA. Herein, we explore a combination of the k-additive integral, the possibility measure and an OWA for MKL. The Sugeno λ-FM is not used as λ becomes difficult to accurately calculate in a computer for high order polynomials ($(m-1)^{th}$ order). These are three different *compact* measures (i.e., they involve relatively few number of free parameters versus $2^m - 2$) that together provide a wide range of possibilities for learning an effective MK aggregation strategy. In the case of the possibility measure and an OWA FM, m values are learned (densities for the former and the weights in

the latter). To begin, a chromosome is assigned to one of these three *types*. We use standard one point crossover between the densities (weights in the case of the OWA). In the case of an OWA, the values are normalized so they sum to 1. When a k-additive FM is crossed with another type of FM, just the m densities are crossed and monotonicity constraints are checked. Any violated constraints are "repaired" by simply assigning the smallest possible valid value. We also use elitism so that the best solution is kept every generation.

Below, FIGA is also used to search for the weights in an LCS to understand the result of MKLGL versus the proposed GA framework. Specifically, what contribution do we observe with respect to the GA instead of linear versus nonlinear aggregation? In the case of the GA, 20 kernels are used on the full feature vector: the dot product, the RBF (different multiples of 10 relative to $\frac{1}{d}$, where d is the feature dimensionality) and the polynomial kernel (of second, third and fourth order). We also note that the MKLGL article used a much greater number of kernels, 117, 442 and 793 respectively.

Table 1. Comparison of FIGA to MKLGL and FIMK to LCS form

	Method	**Breast**	**Ionosphere**	**Sonar**
Classification Accuracy $[0, 100]\%$	MKLGL reported in [14]	96.6 ± 1.2	92.0 ± 2.9	82.0 ± 6.8
	FIGA: just LCS form	97.95 ± 0.17	94.23 ± 0.50	91.03 ± 2.70
	FIGA: combination of FMs/FIs	98.00 ± 0.14	95.40 ± 0.47	92.86 ± 1.17

Table 1 tells the following story. First, it is clear that the GA approaches are more effective than the MKLGL approach, even though the GA approaches use fewer component kernels. Note that the FIGA approaches achieve a mean improvement of about 10% over MKLGL on the Sonar data set. The performance of FIGA comes at a cost though, as MKLGL is much faster in terms of actual running time than FIGA. Second, we see that FIGA using a combination of FM/FIs is somewhat more effective than the FIGA LCS form. These findings are not surprising as our intuition tells us that the nonlinear aggregation allowed by the FM/FI formulation is more flexible than just the LCS aggregation; hence, these results reinforce our expectation. Furthermore, FIGA using the combination of different compact FMs and FIs leads to improved performance at no real additional cost over the FIGA using just an LCS aggregation. Overall, these results are not surprising as different data sets require different solutions, and while an LCS may be sufficient for a given problem, it may not be appropriate for a different problem. Also, it should be noted that the FM/FI formulation includes LCS aggregation as a subset of its possible solutions; hence, when LCS is appropriate the FM/FI aggregation can mimic the LCS. In summary, these experiments suggest that the learner (GA vs GL) appears to be the most important improvement factor, followed by a slight improvement by using the nonlinear FM/FI aggregation versus LCS.

5 Conclusion and Future Work

In summary, we explored different compact FMs and FIs and their combination for MKL in the context of a GA for pattern recognition. We compared this framework to the LCS formulation and MKLGL optimization approach from machine learning. These contributions led to performance benefit in terms of SVM-based classification on benchmark data sets. In future work, we will explore additional, (constrained) efficient ways of learning the FM for FIMKL. We will also explore more efficient non-GA solvers relative to a set of FMs.

References

1. Anderson, D.T., Stone, K., Keller, J.M., Spain, C.: Combination of Anomaly Algorithms and Image Features for Explosive Hazard Detection in Forward Looking Infrared Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 5(1), 313–323 (2012)
2. Havens, T.C., Keller, J.M., Stone, K., Anderson, D.T., Ho, K.C., Ton, T.T., Wong, D.C., Soumekh, M.: Multiple kernel learning for explosive hazards detection in forward-looking ground-penetrating radar. SPIE Defense, Security, and Sensing (2012), http://dx.doi.org/10.1117/12.920482
3. Hu, L., Anderson, D.T., Havens, T.C.: Fuzzy Integral for Multiple Kernel Aggregation. In: IEEE International Conference on Fuzzy Systems (2013)
4. Grabisch, M., Nguyen, E., Walker, E.: Fundamentals of uncertainty calculi with applications to fuzzy inference. Kluwer Academic, Dordrecht (1995)
5. Grabisch, M., Murofushi, T., Sugeno, M.: Fuzzy measures and integrals: theory and applications. STUDFUZZ. Physica-Verlag (2000)
6. Tahani, H., Keller, J.: Information fusion in computer vision using the fuzzy integral. IEEE Trans. on Systems, Man, and Cyber. 20, 733–741 (1990)
7. Anderson, D.T., Havens, T.C., Wagner, C., Keller, J.M., Anderson, M., Wescott, D.: Extension of the Fuzzy Integral for General Fuzzy Set-Valued Information. IEEE Trans. on Fuzzy Systems (2014), doi:10.1109/TFUZZ.2014.2302479
8. Grabisch, M., Labreuche, C.: A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. 4OR: A Quarterly Journal of Operations Research 6, 1–44 (2008)
9. Sugeno, M.: Theory of fuzzy integrals and its application, Ph.D. thesis, Tokyo Institute of Technology (1974)
10. Mendez-Vazquez, A., Gader, P.D.: Learning Fuzzy Measure Parameters by Logistic LASSO. In: Proceedings of the North American Fuzzy Information Processing Society Meeng, NAFIPS, New York, NY, pp. 1–7 (2008)
11. Grabisch, M.: k-order additive discrete fuzzy measures and their representation. Fuzzy Sets and Systems 92, 167–189 (1997)
12. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2001)
13. Gonen, M., Alpaydin, E.: Multiple kernel learning algorithms. Journal of Machine Learning Research 12, 2211–2268 (2011)
14. Xu, Z., Jin, R., Yang, H., King, I., Lyu, M.: Simple and Efficient Multiple Kernel Learning by Group Lasso. In: Int'l Conference on Machine Learning (2010)
15. Bach, F., Lanckriet, G., Jordan, M.: Multiple kernel learning conic duality, and the SMO algorithm. In: Int'l Conference on Machine Learning, pp. 41–48 (2004)