Beliakov, Gleb 2008, Fitting fuzzy measures by linear programming. Programming library fmtools, *in 2008 IEEE International Conference on Fuzzy Systems : proceedings : FUZZ-IEEE 2008*, IEEE, Piscataway, N.J., pp. 862-867.

Fitting fuzzy measures by linear programming. Programming library `fmtools`

Gleb Beliakov

***Abstract*— We discuss the problem of learning fuzzy measures from empirical data. Values of the discrete Choquet integral are fitted to the data in the least absolute deviation sense. This problem is solved by linear programming techniques. We consider the cases when the data are given on the numerical and interval scales. An open source programming library which facilitates calculations involving fuzzy measures and their learning from data is presented.**

I. Introduction

Aggregation functions play an important role in several areas, including fuzzy logic, decision making, expert systems, risk analysis and image processing. Recent books [1]–[5] provide a comprehensive overview of aggregation functions and methods of their construction. The purpose of aggregation functions is to combine several input values into a single output value, which in some sense represents all the inputs. Typically the inputs and outputs are real numbers from $[0, 1]$, although other choices are possible, e.g. discrete sets, intervals and linguistic labels. Notable examples are weighted means, medians, ordered weighted averaging (OWA) functions, discrete Choquet and Sugeno integrals, triangular norms and conorms, uninorms and nullnorms.

Fuzzy integrals (see, e.g. [6]) constitute a general class of aggregation functions, which comprises many popular families, such as weighted means, medians and OWA functions. Choquet and Sugeno integrals are the most widely used fuzzy integrals. They are defined with respect to a fuzzy measure, a set function with the properties of monotonicity and boundedness, see below. For aggregation of a finite number of inputs n, discrete fuzzy measures are used. They are defined by means of 2^n values, two of which are fixed at 0 and 1. Simplifying assumptions are often used, which lead to the classes of k-additive, p-symmetric, decomposable and Sugeno fuzzy measures.

Often aggregation functions need to be constructed based on specified mathematical properties and some data. There is a number of methods of fitting aggregation functions to empirical data [1], [3], [7]. They range from identifying the weights of weighted arithmetic means and OWA functions [8]–[11] to fitting additive generators of triangular norms, conorms and uninorms [12], to constructing fuzzy measures [6], [13]–[15]. In most studies the least squares criterion was optimized, subject to constraints on the variable parameters, which ensure consistency with a priori specified properties.

Gleb Beliakov is with the School of Engineering and Information Technology, Deakin University, 221 Burwood Hwy, Burwood, 3125, Australia (phone: +613 9251 74754; email: gleb@deakin.edu.au).

This typically translates into a quadratic programming problem, for which many numerically efficient algorithms are available.

This paper concentrates on fitting Choquet integrals to empirical data. The data are given as a set of input-output pairs $\{\mathbf{x}_k, y_k\}_{k=1}^K, x_k \in [0, 1]^n, y \in [0, 1]$. Extension to the case of interval-valued inputs and outputs is also possible, and is treated here as well. The goal is to construct a discrete fuzzy measure v, k-additive if required, such that the values of the Choquet integral $C_v(\mathbf{x}_k)$ match the desired outputs y_k as close as possible. Fitting is performed in the least absolute deviation sense (LAD). Our main objective is to show how methods of linear programming (LP) can be used to determine fuzzy measures. Other formulations of the fitting problem, e.g. in the least squares sense, are discussed elsewhere [1], [6], [15].

The next section gives the necessary background on Choquet integrals. Section III formulates the fitting problem as a linear programming problem. Section IV treats interval valued input and outputs. Section V presents a programming library `fmtools` which provides an open source implementation of the described methods. Section VI summarizes conclusions and areas of future research.

II. Basic definitions

Recent comprehensive overviews of aggregation functions are given in [1]–[3], [5], from which we took some relevant definitions, see also [16], [17].

Definition 1: An aggregation function is a function of $n > 1$ arguments $f : [0, 1]^n \to [0, 1]$, with the properties

(i) $f(\underbrace{0, 0, \ldots, 0}_{n-times}) = 0$ and $f(\underbrace{1, 1, \ldots, 1}_{n-times}) = 1$.

(ii) $x \le y$ implies $f(x) \le f(y)$ for all $x, y \in [0, 1]^n$.

The vector inequality is understood componentwise. Aggregation functions may possess various properties, which often classify them into special classes.

- An aggregation function f is called averaging if it is bounded (for all $x \in [0, 1]^n$) by
$$\min(x) = \min_{i=1,\ldots,n} x_i \le f(x) \le \max_{i=1,\ldots,n} x_i = \max(x).$$
- An aggregation function f is called conjunctive if it is bounded by $f(x) \le \min(x)$. An aggregation function is called disjunctive if it is bounded by $\max(x) \le f(x)$. An aggregation function f is called mixed if it is neither conjunctive, disjunctive or averaging.
- An aggregation function is called idempotent if $f(t, t, \ldots, t) = t$ for any $t \in [0, 1]$. Since aggregation

978-1-4244-1819-0/08/$25.00©2008 IEEE

functions are monotone, idempotency is equivalent to the averaging behaviour.

- An aggregation function is called symmetric (commutative) if $f(x) = f(x_P)$ for any $x \in [0,1]^n$ and any permutation P of $\{1,\dots,n\}$.

Weighted arithmetic means are the most common aggregation functions. Discrete Choquet integrals generalize both the weighted arithmetic means and OWA functions. These functions are defined with respect to a fuzzy measure, and can take into account not only the relative weightings of the individual inputs, but also their groups (coalitions). A discrete fuzzy measure allows one to assign importances to all possible groups of criteria, and thus offers a much greater flexibility for modeling aggregation. The weighted arithmetic means and OWA are special cases of Choquet integrals with respect to additive and symmetric fuzzy measures respectively. The uses of Choquet and Sugeno integrals as aggregation functions are documented, e.g. in [15], [18], [19].

Definition 2: Let $\mathcal{N} = \{1, 2, \dots, n\}$. A discrete fuzzy measure is a set function $v : 2^{\mathcal{N}} \to [0,1]$ which is monotonic (i.e. $v(\mathcal{A}) \le v(\mathcal{B})$ whenever $\mathcal{A} \subset \mathcal{B}$) and satisfies $v(\emptyset) = 0$ and $v(\mathcal{N}) = 1$.

In the Definition 2, a subset $\mathcal{A} \subseteq \mathcal{N}$ can be considered as a *coalition*, so that $v(\mathcal{A})$ gives us an idea about the importance or the weight of this coalition. The monotonicity condition implies that adding new elements to a coalition does not decrease its weight.

Definition 3: Let v be a fuzzy measure. The Möbius transformation of v is a function defined for every $\mathcal{A} \subseteq \mathcal{N}$ as

$$\mathcal{M}(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{A}} (-1)^{|\mathcal{A} \setminus \mathcal{B}|} v(\mathcal{B}).$$

The Möbius transformation is invertible, and one recovers v by using its inverse, called *Zeta transform*,

$$v(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{A}} \mathcal{M}(B) \quad \forall \mathcal{A} \subseteq \mathcal{N}.$$

The Möbius transformation is helpful in expressing various quantities, like the interaction indices discussed later, in a more compact form. It also serves as an alternative representation of a fuzzy measure, called Möbius representation. That is, one can either use v or $\mathcal{M}$ to perform calculations, whichever is more convenient. The conditions of monotonicity of a fuzzy measure, and the boundary conditions $v(\emptyset) = 0, v(\mathcal{N}) = 1$ are expressed, respectively, as

$$\sum_{\mathcal{B} \subseteq A | i \in \mathcal{B}} \mathcal{M}(\mathcal{B}) \ge 0, \text{ for all } \mathcal{A} \subseteq \mathcal{N} \text{ and } i \in \mathcal{A}, \quad (1)$$

$$\mathcal{M}(\emptyset) = 0 \text{ and } \sum_{\mathcal{A} \subseteq \mathcal{N}} \mathcal{M}(\mathcal{A}) = 1.$$

Definition 4: The discrete Choquet integral with respect to a fuzzy measure v is given by

$$C_v(\mathbf{x}) = \sum_{i=1}^{n} x_{(i)} [v(\{j | x_j \ge x_{(i)}\}) - v(\{j | x_j \ge x_{(i+1)}\})], \quad (2)$$

where $\mathbf{x}_{\nearrow} = (x_{(1)}, x_{(2)}, \dots, x_{(n)})$ is a non-decreasing permutation of the input $\mathbf{x}$, and $x_{(n+1)} = \infty$ by convention.

Choquet integral can be expressed with the help of the Möbius transformation as

$$C_v(\mathbf{x}) = \sum_{\mathcal{A} \subseteq \mathcal{N}} \mathcal{M}(\mathcal{A}) \min_{i \in \mathcal{A}} x_i = \sum_{\mathcal{A} \subseteq \mathcal{N}} \mathcal{M}(\mathcal{A}) h_{\mathcal{A}}(\mathbf{x}), \quad (3)$$

with $h_{\mathcal{A}}(\mathbf{x}) = \min_{i \in \mathcal{A}} x_i$.

For computational purposes it is convenient to store the values of a fuzzy measure v in an array $\mathbf{v}$ of size 2^n, and to use some indexing system, which provides a one-to-one mapping between the subsets $\mathcal{J} \subseteq \mathcal{N}$ and the set of integers $I = \{0, \dots, 2^n - 1\}$, which index the elements of v. The two most common indexing systems are the binary ordering and cardinality ordering. In the binary ordering the subsets of $\mathcal{N}$ are ordered as

$$\emptyset, \{1\}, \{2\}, \{1,2\}, \{3\}, \{1,3\}, \{2,3\}, \{1,2,3\}, \dots.$$

An alternative ordering is based on set cardinality

$$\emptyset, \underbrace{\{1\}, \{2\}, \dots, \{n\}}_{n \text{ singletons}}, \underbrace{\{1,2\}, \dots, \{1,n\}, \dots, \{n-1,n\}}_{\binom{n}{2} \text{ pairs}}, \dots.$$

Such an ordering is useful when dealing with $\mathcal{K}$-additive fuzzy measures (see Definition 5 and Proposition 1 below), as it allows one to group non-zero values $\mathcal{M}(\mathcal{A})$ (in Möbius representation) at the beginning of the array.

There are various types of fuzzy measures, like symmetric, additive, decomposable, sub- and supermodular, possibility and necessity, plausibility and belief, self-dual, balanced and Sugeno fuzzy measures to name a few [6]. In this contribution we are specifically interested in $\mathcal{K}$-additive fuzzy measures.

Definition 5: A fuzzy measure v is called $\mathcal{K}$-additive ($1 \le \mathcal{K} \le n$) if its Möbius transformation verifies

$$\mathcal{M}(\mathcal{A}) = 0$$

for any subset $\mathcal{A}$ with more than $\mathcal{K}$ elements, $|\mathcal{A}| > \mathcal{K}$, and there exists a subset $\mathcal{B}$ with k elements such that $\mathcal{M}(\mathcal{B}) \neq 0$.

When dealing with multiple criteria, it is often the case that these are not independent, and there is some interaction (positive or negative) among the criteria. For instance, two or more criteria may point essentially to the same concept, for example criteria such as “learnability” and “memorability” that are used to evaluate software user interface. If the criteria are combined by using, e.g., weighted means, their scores will be double counted. In other instances, contribution of one criterion to the total score by itself may be small, but sharply rise when taken in conjunction with other criteria (i.e., in a “coalition”).

To measure such concepts as the importance of a criterion and interaction among the criteria we will use the concepts of Shapley value, which measures the importance of a criterion i in all possible coalitions, and the interaction index, which measures the interaction of a pair of criteria i, j in all possible coalitions [6], [20].

Definition 6: Let v be a fuzzy measure. The Shapley index for every $i \in \mathcal{N}$ is

$$\phi(i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus \{i\}} \frac{(n-|\mathcal{A}|-1)!|\mathcal{A}|!}{n!} [v(\mathcal{A} \cup \{i\}) - v(\mathcal{A})].$$

The Shapley value is the vector $\phi(v) = (\phi(1), \ldots, \phi(n))$.

Definition 7: Let v be a fuzzy measure. The interaction index for every pair $i, j \in \mathcal{N}$ is

$$I_{ij} = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus \{i,j\}} \frac{(n-|\mathcal{A}|-2)!|\mathcal{A}|!}{(n-1)!} \times$$
$$\times \left[v(\mathcal{A} \cup \{i,j\}) - v(\mathcal{A} \cup \{i\}) - v(\mathcal{A} \cup \{j\}) + v(\mathcal{A}) \right].$$

The interaction indices verify $I_{ij} < 0$ as soon as i, j are positively correlated (negative synergy, redundancy). Similarly $I_{ij} > 0$ for negatively correlated criteria (positive synergy, complementarity). $I_{ij} \in [-1, 1]$ for any pair i, j.

Definition 8: Let v be a fuzzy measure. The interaction index for every set $\mathcal{A} \subseteq \mathcal{N}$ is

$$I(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{N} \setminus \mathcal{A}} \frac{(n-|\mathcal{B}|-|\mathcal{A}|)!|\mathcal{B}|!}{(n-|\mathcal{A}|+1)!} \sum_{\mathcal{C} \subseteq \mathcal{A}} (-1)^{|\mathcal{A} \setminus \mathcal{C}|} v(\mathcal{B} \cup \mathcal{C}).$$

Möbius transformation helps one to express the indices mentioned above in a more compact form [6], [20], [21]

$$\phi(i) = \sum_{\mathcal{B} |\, i \in \mathcal{B}} \frac{1}{|\mathcal{B}|} \mathcal{M}(\mathcal{B}),$$

$$I(\mathcal{A}) = \sum_{\mathcal{B} | \mathcal{A} \subseteq \mathcal{B}} \frac{1}{|\mathcal{B}| - |\mathcal{A}| + 1} \mathcal{M}(\mathcal{B}).$$

The next result [20] establishes a fundamental property of $\mathcal{K}$-additive fuzzy measures, which justifies their use in simplifying interactions between the criteria in multiple criteria decision making.

Proposition 1: Let v be a $\mathcal{K}$-additive fuzzy measure, $1 \leq \mathcal{K} \leq n$. Then

- $I(\mathcal{A}) = 0$ for every $\mathcal{A} \subseteq \mathcal{N}$ such that $|\mathcal{A}| > \mathcal{K}$;
- $I(\mathcal{A}) = \mathcal{M}(\mathcal{A})$ for every $\mathcal{A} \subseteq \mathcal{N}$ such that $|\mathcal{A}| = \mathcal{K}$.

Thus $\mathcal{K}$-additive measures acquire an interesting interpretation. These are fuzzy measures that limit interaction among the criteria to groups of size at most k. For instance, for 2-additive fuzzy measures, there are pairwise interactions among the criteria but no interactions in groups of 3 or more. By limiting the class of fuzzy measures to $\mathcal{K}$-additive measures, one reduces their complexity (the number of values) by imposing linear equality constraints. The total number of linearly independent values is reduced from $2^n - 1$ to $\sum_{i=1}^{\mathcal{K}} \binom{n}{i} - 1$. However for $k > 1$ the number of monotonicity constraints on fuzzy measures does not change.

The *measure of orness*, also called the *degree of orness, orness value* or *attitudinal character*, is an important numerical characteristic of averaging aggregation functions. Basically, the measure of orness measures how far a given averaging function is from the $\max$ function, which is the weakest disjunctive function. By using the Möbius transform one can calculate the orness of a Choquet integral C_v with respect to a fuzzy measure v as follows.

Proposition 2: [22] For any fuzzy measure v the orness of the Choquet integral with respect to v is

$$orness(C_v) = \frac{1}{n-1} \sum_{\mathcal{A} \subseteq \mathcal{N}} \frac{n - |\mathcal{A}|}{|\mathcal{A}| + 1} \mathcal{M}(\mathcal{A}),$$

where $\mathcal{M}(\mathcal{A})$ is the Möbius representation of $\mathcal{A}$. In terms of v the orness value is

$$orness(C_v) = \frac{1}{n-1} \sum_{\mathcal{A} \subseteq \mathcal{N}} \frac{(n-|\mathcal{A}|)!|\mathcal{A}|!}{n!} v(\mathcal{A}).$$

III. Fitting Choquet integrals

This section outlines the problem of fitting fuzzy measures to some sort of empirical data, the observed (or sometimes desired) pairs of input-output values. In the most typical case, the data comes in pairs $(\mathbf{x}, y)$, where $\mathbf{x} \in [0,1]^n$ is the input vector and $y \in [0,1]$ is the desired output. There are several pairs, which will be denoted by a subscript k: $(\mathbf{x}_k, y_k), k = 1, \ldots, K$.

When the data comes from an experiment, it will normally contain some errors, and therefore it is pointless to interpolate the inaccurate values y_k. In this case our aim is to stay close to the desired outputs without actually matching them.

The goal is to find a fuzzy measure v, such that the function $f = C_v$ approximates y_k, $f(\mathbf{x}_k) \approx y_k$. The satisfaction of approximate equalities $f(\mathbf{x}_k) \approx y_k$ is usually translated into the following minimization problem.

$$\text{minimize } ||\mathbf{r}|| \quad (4)$$
$$\text{subject to } f \text{ satisfies properties } \quad \mathcal{P}_1, \mathcal{P}_2, \ldots,$$

where $||\mathbf{r}||$ is the norm of the residuals, i.e., $\mathbf{r} \in R^K$ is the vector of the differences between the predicted and observed values $r_k = f(\mathbf{x}_k) - y_k$. There are many ways to choose the norm, and the most popular are the least squares norm and the least absolute deviation norm.

In the case when f is the Choquet integral with respect to a fuzzy measure v, C_v, our goal is to identify the values of v. Identification of the $2^n - 2$ values from the data (two are given explicitly as $v(\emptyset) = 0, v(N) = 1$) involves the least squares or least absolute deviation problems

$$\text{minimize } \sum_{k=1}^{K} \left(C_v(x_{1k}, \ldots, x_{nk}) - y_k \right)^2, \text{ or}$$

$$\text{minimize } \sum_{k=1}^{K} \left| C_v(x_{1k}, \ldots, x_{nk}) - y_k \right|,$$

subject to the conditions of monotonicity of the fuzzy measure (they translate into a number of linear constraints).

We concentrate on the least absolute deviation problem, because a) it is less sensitive to outliers, and b) it can be translated into a linear programming problem, which can be solved quickly and reliably even in the case of a very large number of parameters and constraints. Note that the main difficulty in fitting fuzzy measures is the large number of unknowns, and typically a much smaller number of data [15].

The interaction indices and the orness measure are all linear functions of the values of the fuzzy measure. One can specify given values of importance (Shapley value) and interaction indices $\phi(i)$, I_{ij} by adding linear equality constraints. Of course, these values may not be specified exactly, but as intervals, say, for Shapley value we may have $a_i \le \phi(i) \le b_i$. In this case we obtain a pair of linear inequalities.

Recall that $\mathcal{K}$-additive fuzzy measures satisfy $\mathcal{M}(\mathcal{A}) = 0$ for any subset $\mathcal{A}$ with more than $\mathcal{K}$ elements. Since Möbius transform is a linear combination of values of v, we obtain a set of linear equalities. By using interaction indices, we can express k-additivity as (see Proposition 1) $I(\mathcal{A}) = 0$ for every $\mathcal{A} \subseteq \mathcal{N}$, $|\mathcal{A}| > \mathcal{K}$, which is again a set of linear equalities.

However, these conditions on the fuzzy measures do not reduce the complexity of the least squares or least absolute deviation problems. They only add a number of equality and inequality constraints to these problems. It is possible to reduce the complexity of the problem when working in Möbius representation.

As the variables we will use $m_j = m_{\mathcal{A}} = \mathcal{M}(\mathcal{A})$ such that $|\mathcal{A}| \le \mathcal{K}$ in some appropriate indexing system based on cardinality ordering. This is a much reduced set of variables ($\sum_{i=1}^{\mathcal{K}} \binom{n}{i} - 1$ compared to $2^n - 1$). Monotonicity of a fuzzy measure, expressed as

$$v(\mathcal{A} \cup \{i\}) - v(\mathcal{A}) \ge 0, \quad \forall \mathcal{A} | i \notin \mathcal{A}, i = 1, \ldots, n,$$

converts into (1), and using $\mathcal{K}$-additivity, into

$$\sum_{\mathcal{B} \subseteq A | i \in \mathcal{B}, |\mathcal{B}| \le \mathcal{K}} m_{\mathcal{B}} \ge 0, \quad \text{for all } \mathcal{A} \subseteq \mathcal{N} \text{ and all } i \in \mathcal{A}.$$

The (non-redundant) set of non-negativity constraints $v(\{i\}) \ge 0, i = 1, \ldots, n$, is a special case of the previous formula when $\mathcal{A}$ is a singleton, which simply become

$$\sum_{\mathcal{B} = \{i\}} m_{\mathcal{B}} = m_{\{i\}} \ge 0, \; i = 1, \ldots, n.$$

Finally, condition $v(\mathcal{N}) = 1$ becomes $\sum_{\mathcal{B} \subseteq \mathcal{N} | |\mathcal{B}| \le \mathcal{K}} m_{\mathcal{B}} = 1$.

Then the least absolute deviation problem is translated into a simplified optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{K} \left| \sum_{\mathcal{A} | \, |\mathcal{A}| \le \mathcal{K}} h_{\mathcal{A}}(\mathbf{x}_j) m_{\mathcal{A}} - y_j \right|, & (5) \\ \text{s.t.} \quad & \sum_{\mathcal{B} \subseteq A | i \in \mathcal{B}, |\mathcal{B}| \le \mathcal{K}} m_{\mathcal{B}} \ge 0, \\ & \quad \text{for all } \mathcal{A} \subseteq \mathcal{N}, |\mathcal{A}| > 1, \text{ and all } i \in \mathcal{A}, \\ & m_{\{i\}} \ge 0, \; i = 1, \ldots, n, \\ & \sum_{\mathcal{B} \subseteq \mathcal{N} | |\mathcal{B}| \le \mathcal{K}} m_{\mathcal{B}} = 1, \end{aligned}$$

where $h_{\mathcal{A}}(\mathbf{x}) = \min_{i \in \mathcal{A}} x_i$. Note that only the specified $m_{\mathcal{B}}$ are non-negative, others are unrestricted. Note that the number of monotonicity constraints is the same for all $\mathcal{K}$-additive fuzzy measures for $\mathcal{K} = 2, \ldots, n$, regardless their representation [23].

The problem (5) is subsequently converted to a linear programming problem using the following technique. Let $r_j = f(x_j) - y_j$ be the $j-$ th residual. We represent it as a difference of a positive and negative parts $r_j = r_j^+ - r_j^-$, $r_j^+, r_j^- \ge 0$. The absolute value is $|r_j| = r_j^+ + r_j^-$. The problem (5) is converted into an LP problem with respect to $\mathbf{m}, \mathbf{r}^+, \mathbf{r}^-$

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{K} (r_j^+ + r_j^-), & (6) \\ \text{s.t.} \quad & \sum_{\mathcal{A} | \, |\mathcal{A}| \le \mathcal{K}} h_{\mathcal{A}}(\mathbf{x}_j) m_{\mathcal{A}} \quad -(r_j^+ - r_j^-) = y_j, \; j = 1, \ldots, K \\ & \text{other constraints from (5)}, \\ & r_j^+, r_j^- \ge 0. \end{aligned}$$

We note that as any quadratic or linear programming problem, (6) can be converted into a linear complementarity problem (LCP) [24], however we are unaware of the algorithms for solving such LCPs more efficient than the solution to (6) by the simplex method. Although emerging parallelization techniques (for LCPs) may change this view (we remind that the simplex method is serial in nature).

IV. Interval valued inputs and outputs

In the process of knowledge acquisition from domain experts it is quite common that they provide answers about the output values as intervals, "between a and b". We shall now adapt the LAD problem (5) for that case. Let $\mathcal{L}$ denote the set of nonempty intervals $\{[x^-, x^+]\}, x^-, x^+ \in [0,1]$. Then we could have either $y_k \in \mathcal{L}$ (the expert provides a range of equally acceptable outputs for a given input), or both $y_k \in \mathcal{L}$, $x_k \in \mathcal{L}^n$. Let us denote the output values by $y_k = [y_k^-, y_k^+]$. We formulate the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K} \max(y_k^- - C_v(\mathbf{x}_k), 0) & (7) \\ & + \max(C_v(\mathbf{x}_k) - y_k^+, 0) \\ \text{s.t.} \quad & \text{various constraints on } v \end{aligned}$$

By using auxiliary variables $r_k^-, r_k^+ \ge 0$ we translate the problem to the form similar to (6)

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K} r_k^+ + r_k^- & (8) \\ \text{s.t.} \quad & r_k^+ - \sum_{\mathcal{A} | \, |\mathcal{A}| \le \mathcal{K}} h_{\mathcal{A}}(\mathbf{x}_j) m_{\mathcal{A}} \ge -y_k^+, \\ & r_k^- + \sum_{\mathcal{A} | \, |\mathcal{A}| \le \mathcal{K}} h_{\mathcal{A}}(\mathbf{x}_j) m_{\mathcal{A}} \ge y_k^-, \\ & k = 1, \ldots, K, \\ & \text{other constraints from (5)} \\ & r_k^-, r_k^+ \ge 0. \end{aligned}$$

Note that setting $y_k^- = y_k^+$ results in problem formulation (6) (for each k the auxiliary variables satisfy $r_k^- r_k^+ = 0$).

Let us denote the components of $x_k \in \mathcal{L}^n$ by $x_{ki} = [x_{ki}^-, x_{ki}^+]$ and $x_k^- = (x_{k1}^-, \ldots, x_{kn}^-)$, $x_k^+ = (x_{k1}^+, \ldots, x_{kn}^+)$. We formulate the LAD fitting problem similar to (7), namely

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K} \max(y_k^- - C_v(\mathbf{x}_k^+), 0) \\ & + \max(C_v(\mathbf{x}_k^-) - y_k^+, 0) \\ \text{s.t.} \quad & \text{various constraints on } v \end{aligned} \tag{9}$$

Monotonicity of C_v implies that $C_v(\mathbf{x}_k^-) \leq C_v(\mathbf{x}_k^+)$. In Problem (9) the k-th term of the objective function is nonzero only if $C_v(\mathbf{x}_k^-) \geq y_k^+$ or $C_v(\mathbf{x}_k^+) \leq y_k^-$ and one of these inequalities excludes the other. Then we obtain the problem

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K} r_k^+ + r_k^- \\ \text{s.t.} \quad & r_k^+ - C_v(\mathbf{x}_k^-) \geq -y_k^+, \\ & r_k^- + C_v(\mathbf{x}_k^+) \geq y_k^-, \\ & k = 1, \ldots, K, \\ & \text{other constraints from (5)}, \\ & r_k^-, r_k^+ \geq 0. \end{aligned} \tag{10}$$

We have shown in this section that when the data are given as intervals, the LAD fitting problem is still translated into an LP problem, which is very convenient. This is not the case for the least squares fitting, in which case the convenience of quadratic programming is lost.

V. Description of `fmtools` package

The `fmtools` package is an open source set of subroutines, distributed under LGPL licence, facilitating standard operations on fuzzy measures, calculation of various quantities, such as Shapley and interaction indices, orness value, as well as fitting fuzzy measures to empirical data by solving a linear programming problem (6). The package is available from [25]. This section outlines basic features of this package.

A. Basic manipulations and tests

Fuzzy measures can be characterized by various indices, such as interaction indices, and can belong to specific classes, such as sub or super-additive, etc. `fmtools` implements a number of calculation routines and tests, in particular:

1) Calculation of Shapley values;
2) Calculation of Banzhaf indices;
3) Calculation of all interaction indices;
4) Calculation of all Banzhaf interaction indices;
5) Calculation of the dual fuzzy measure;
6) Calculation of the orness value of the Choquet integral;
7) Calculation of the entropy of the Choquet integral;
8) Tests whether a fuzzy measure is: Balanced; Self-dual; Subadditive; Superadditive; Additive; Submodlar; Supermodular; Symmetric.

Tests are performed with a given tolerance. For numerical efficiency reasons, certain quantities (like ordering conversion tables, tables of sets cardinalities and factorials) are pre-computed for a given n, at the initialization stage. `fmtools` uses the formulas presented in Section III in standard and Möbius representations interchangeably. For Sugeno fuzzy measures it also computes the value of λ (given the values of v at singletons). `fmtools` also implements an efficient calculation of the Choquet and Sugeno integrals in standard and Möbius representations.

B. Fitting Choquet inegrals to data

We are given a data set representing the values of an unknown function f For example, when $n = 4$ we have

x_1	x_2	x_3	x_4	y
x_{11}	x_{12}	x_{13}	x_{14}	y_1
x_{21}	x_{22}	x_{23}	x_{24}	y_2
x_{31}	x_{32}	x_{33}	x_{34}	y_3
$\vdots$				
x_{K1}	x_{K2}	x_{K3}	x_{K4}	y_K

The goal is to identify a fuzzy measure v, such that the corresponding Choquet integral $f = C_v$ predicts the outputs y_k as close as possible in the least absolute deviation sense. This is done by solving a linear programming problem (6). In addition, optional conditions on the bounds of interaction indices, Shapley values or orness value are also incorporated as linear constraints.

The problem is set in the Möbius representation. To obtain a standard LP formulation, equality constraints are represented by pairs of inequality constraints, and unconstrained variables are replaced with pairs of non-negative variables (the positive and negative parts of the unconstrained variable).

If the fuzzy measure is assumed to be $\mathcal{K}$-additive, then in Möbius representation values corresponding to subsets of cardinality greater than $\mathcal{K}$ are 0. These decision variables are explicitly excluded from the problem formulation, which is the key to reducing its complexity. For numerical efficiency purposes, a dual of this LP problem is actually solved in `fmtools`, because when the fuzzy measure is $\mathcal{K}$-additive, the number of variables is much less than that of constraints.

C. Programming interface

Current distribution `fmtools` package includes the source files and the executables for fitting fuzzy measures to empirical data (Windows and linux 386). There are two sample programs included with this distribution, which illustrate the major features of the library.

The subroutines in `fmtools` are implemented in C++ language. They reside in two files `fuzzymeasuretools.cpp` and `fuzzymeasurefit.cpp`. To perform basic operations with fuzzy measures, the user should simply add the line

```
#include "fuzzymeasuretools.h"
```

to the main program, and add `fuzzymeasuretools.cpp` to the project (makefile). There is an example C++ program which illustrates the use of the basic routines.

For fitting fuzzy measures to the data, the user would typically use the provided executables `fmfitting` (for Windows and linux 386). This program (whose source code is included) reads parameters from a configuration file, the empirical data from a data file, and prints the values of the fitted fuzzy measure and other computed quantities to an outputfile, which can later be read and used by other programs. However, the users who want to change the functionality of `fmfitting`, or compile it on a different platform, can also use `fuzzymeasurefit.cpp` part of the package, and the source `fmfitting.cpp` as a template. The user should add

```
#include "fuzzymeasuretools.h"
#include "fuzzymeasurefit.h"
```

and then compile both `fuzzymeasuretools.cpp` and `fuzzymeasurefit.cpp`.

D. Performance and limitations

Written in C++ language, `fmtools` favorably compares with an alternative package `kappalab` [15], [26] (in R language), in terms of numerical efficiency. In addition, it can handle interval-values inputs and outputs by using linear programming, whereas in `kappalab` fitting is performed by quadratic programming, which is harder to solve. On the other hand, `kappalab` includes a more comprehensive set of routines and fitting methods described in [15], and the two packages are complementary.

The main limitation of `fmtools` is the complexity of the fuzzy measure, exponential in n. We mentioned that the number of monotonicity constraints on fuzzy measures cannot be reduced by using $\mathcal{K}$-additivity. In our experience, n up to 16 can be handled by `fmtools` adequately (i.e., with solution time < 1h), but it depends much on the data.

Novel parallelization tools, as those based on graphics processor units (GPU) [27], as suggested by one referee, may be used for some procedures in `fmtools`, such as conversion from one representation to another (which is essentially a matrix-vector multiplication problem). However the main task of fitting fuzzy measures to the data, performed by using the simplex method, cannot be parallelized at this stage.

VI. Conclusions

In this contribution we outlined the problem of fitting $\mathcal{K}$-additive fuzzy measures to empirical data, subject to various constraints on interaction indices, orness value, etc., and briefly presented a practical solution to this problem, the package `fmtools`. Our main objective was to show how such a problem can be formulated as a linear programming problem, which is the key for numerical efficiency. We note that there is an alternative set of tools based on R language, called `kappalab` [15], [26]. `kappalab` fits fuzzy measures by several methods, most are based on solving a least squares problem (which is in turn solved by quadratic programming).

Using our linear programming formulation is an alternative which extends the range of applicability of the methods in [15] (larger numbers of inputs and data can be handled), and also allows one to impose further conditions, e.g. preservation of outputs ordering [1], to specify the inputs and outputs on the interval scale, and many others. It is hoped that `fmtools` package will be of value for many practitioners.

References

[1] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Heidelberg, Berlin, New York: Springer, 2007.

[2] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. to appear: Elsevier, 2008.

[3] Y. Torra, V. Narukawa, *Modeling Decisions. Information Fusion and Aggregation Operators*. Berlin, Heidelberg: Springer, 2007.

[4] T. Calvo, G. Mayor, and R. Mesiar, Eds., *Aggregation Operators. New Trends and Applications*. Heidelberg, New York: Physica-Verlag, 2002.

[5] H. Bustince, F. Herrera, and J. Montero, Eds., *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*. Heidelberg, New York: Springer, 2007.

[6] M. Grabisch, T. Murofushi, and M. Sugeno, Eds., *Fuzzy Measures and Integrals. Theory and Applications*. Heidelberg: Physica-Verlag, 2000.

[7] G. Beliakov, "How to build aggregation operators from data?" *Int. J. Intelligent Systems*, vol. 18, pp. 903–923, 2003.

[8] G. Beliakov and T. Calvo, "Identification of weights in aggregation operators," in *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, H. Bustince, F. Herrera, and J. Montero, Eds. Heidelberg: Springer, 2007.

[9] V. Torra, "Learning weights for the quasi-weighted means," *IEEE Trans. on Fuzzy Systems*, vol. 10, pp. 653–666, 2002.

[10] ——, "OWA operators in data modeling and reidentification," *IEEE Trans. on Fuzzy Systems*, vol. 12, pp. 652–660, 2004.

[11] D. Filev and R. Yager, "On the issue of obtaining OWA operator weights," *Fuzzy Sets and Systems*, vol. 94, pp. 157–169, 1998.

[12] G. Beliakov, "Monotone approximation of aggregation operators using least squares splines," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, pp. 659–676, 2002.

[13] G. Beliakov, R. Mesiar, and L. Valaskova, "Fitting generated aggregation operators to empirical data," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 12, pp. 219–236, 2004.

[14] Y. Narukawa and V. Torra, "Fuzzy measure and probability distributions: Distorted probabilities," *IEEE Trans. on Fuzzy Systems*, vol. 13, pp. 617–629, 2005.

[15] M. Grabisch, I. Kojadinovic, and P. Meyer, "A review of methods for capacity identification in Choquet integral based multi-attribute utility theory," *Europ. J. Operations Research*, vol. 186, pp. 766–785, 2008.

[16] D. Dubois and H. Prade, "A review of fuzzy set aggregation connectives," *Inform. Sci.*, vol. 36, pp. 85–121, 1985.

[17] ——, "On the use of aggregation operations in information fusion processes," *Fuzzy Sets and Systems*, vol. 142, pp. 143–161, 2004.

[18] M. Grabisch, "The applications of fuzzy integrals in multicriteria decision making," *Europ. J. Operations Research*, vol. 89, pp. 445–456, 1996.

[19] J.-L. Marichal, "On Choquet and Sugeno integrals as agregation functions," in *Fuzzy measures and integrals. Theory and Applications*, M. Grabisch, T. Murofushi, and M. Sugeno, Eds. Heidelberg: Physica-Verlag, 2000, pp. 247–272.

[20] M. Grabisch, "k-order additive discrete fuzzy measures and their representation," *Fuzzy Sets and Systems*, vol. 92, pp. 167–189, 1997.

[21] M. Grabisch, J.-L. Marichal, and M. Roubens, "Equivalent representations of set functions," *Mathematics of Oper. Res.*, vol. 25, pp. 157–178, 2000.

[22] J.-L. Marichal, "Tolerant or intolerant character of interacting criteria in aggregation by the Choquet integral," *Europ. J. Oper. Research*, vol. 155, pp. 771–791, 2004.

[23] M. Grabisch, H. Nguyen, and E. Walker, *Fundamentals of Uncertainty Calculi, with Applications to Fuzzy Inference*. Dordrecht: Kluwer, 1995.

[24] R. Cottle, J.-S. Pang, and R. Stone, *The Linear Complementarity Problem*. San Diego, CA: Academic Press, 1992.

[25] G. Beliakov, "`fmtools` package, version 1.0," 2007. [Online]. Available: http://www.deakin.edu.au/∼gleb/aotool.html

[26] I. Kojadinovic and M. Grabisch, "Non additive measure and integral manipulation functions, R package version 0.2," 2005. [Online]. Available: http://www.polytech.univ-nantes.fr/kappalab

[27] nvidia, "`CUDA Zone`," 2007. [Online]. Available: http://www.nvidia.com/object/cuda_home.html