Fitting Symmetric Fuzzy Measures for Discrete Sugeno Integration

Marek Gagolewski[1,2] and Simon James[3]

[1] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland, gagolews@ibspan.waw.pl
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
[3] School of Information Technology, Deakin University,
221 Burwood Hwy, Burwood, Victoria 3125, Australia, sjames@deakin.edu.au

Abstract. The Sugeno integral has numerous successful applications, including but not limited to the areas of decision making, preference modeling, and bibliometrics. Despite this, the current state of the development of usable algorithms for numerically fitting the underlying discrete fuzzy measure based on a sample of prototypical values – even in the simplest possible case, i.e., assuming the symmetry of the capacity – is yet to reach a satisfactory level. Thus, the aim of this paper is to present some results and observations concerning this class of data approximation problems.

Keywords: Sugeno integral, aggregation functions, machine learning, regression, approximation

1 Introduction

An n-ary mean acting on elements in the unit interval is a function $\mathsf{M} : [0,1]^n \to [0,1]$ nondecreasing in each variable satisfying $\mathsf{M}(x, x, \ldots, x) = x$ for every $x \in [0,1]$, i.e., an idempotent aggregation function, see [3,5,9]. This definition embraces, e.g., the arithmetic mean, median, and their generalizations like quasi-arithmetic means or OWA operators [22,23] studied in fields like aggregation theory, decision making, probability and statistics, fuzzy sets theory, uncertainty modeling, and so forth.

Assume we observe m input sequences, each with n elements in the unit interval, $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}] \in [0,1]^{n \times m}$, together with m corresponding desired output values $\mathbf{Y} = [y^{(1)}, \ldots, y^{(m)}] \in [0,1]^{1 \times m}$. For some fixed class of means $\mathcal{M}$, the problem of fitting aggregation functions to data usually concerns the optimization task:

$$\text{minimize } \sqrt{\frac{1}{m} \sum_{k=1}^{m} \left| \mathsf{M}\left(\mathbf{x}^{(k)}\right) - y^{(k)} \right|^2} \text{ w.r.t. } \mathsf{M} \in \mathcal{M}$$

in the case of the root mean squared error (RMSE) criterion, or:

$$\text{minimize } \frac{1}{m}\sum_{k=1}^{m}\left|\mathsf{M}\left(\mathbf{x}^{(k)}\right)-y^{(k)}\right| \text{ w.r.t. } \mathsf{M}\in\mathcal{M}$$

in the case of the mean absolute error (MAE) criterion, which is known to be more robust in the presence of outliers. Assuming M is vectorized so that $\mathsf{M}(\mathbf{X})=\left[\mathsf{M}\left(\mathbf{x}^{(1)}\right),\ldots,\mathsf{M}\left(\mathbf{x}^{(m)}\right)\right]\in[0,1]^{1\times m}$, the two above error measures can be expressed as $m^{-1/p}\left\|\mathsf{M}(\mathbf{X})-\mathbf{Y}\right\|_p$ with $p=2$ and 1, respectively.

In this paper we shall focus on $\mathcal{M}$ consisting of particular discrete Sugeno integrals [19], see also, e.g., [3, Chap. 4], which have a wide range of applications, e.g., in decision making, preference modeling, and bibliometrics [12,14,21]. To recall, for a given fuzzy measure (normalized capacity or monotone measure) $\mu:2^{\{1,2,\ldots,n\}}\to[0,1]$, i.e., a set function such that $\mu(\emptyset)=0$, $\mu(\{1,\ldots,n\})=1$, $\mu(U)\leq\mu(V)$ for $U\subseteq V$, the corresponding discrete Sugeno integral of $\mathbf{x}\in[0,1]^n$ is defined as:

$$\begin{aligned}\mathsf{S}_\mu(\mathbf{x}) &= \bigvee_{j=1}^{n} x_{\sigma(j)}\wedge\mu\left(\{\sigma(1),\sigma(2),\ldots,\sigma(j)\}\right)\\ &= \max\left\{\min\left\{x_{\sigma(1)},\mu(\{\sigma(1)\})\right\},\ldots,\min\left\{x_{\sigma(n)},\mu(\{\sigma(1),\ldots,\sigma(n)\})\right\}\right\},\end{aligned}$$

where σ is a permutation of $\{1,2,\ldots,n\}$ such that $x_{\sigma(1)}\geq\cdots\geq x_{\sigma(n)}$.

The problem of fitting particular classes of means, like OWA operators, weighted quasi-arithmetic means, Bonferroni means, but also the Choquet integrals, etc., has already received some attention in the literature, see, e.g., [2,4,20] as well as [3] for a more extensive review. Yet, the case of the Sugeno integral fitting has not been covered sufficiently for practical use. To the best of our knowledge, there exist the following results. In [24] an approximate, neural network-based algorithm for fitting a Sugeno integral to an empirical data set based on the least squared error criterion has been proposed. On the other hand, in [1] a genetic algorithm has been developed for a more general case – fuzzy valued fuzzy measures. What is more, in [16] a family of Sugeno integrals compatible with a given data set (if it exists) has been studied in an algebraic framework.

Note that in the case of the Choquet integral, the main problem that arises in finding the optimum fuzzy measure is the exponentially increasing number of parameters. Since Sugeno integrals are also defined with respect to fuzzy measures, this will clearly be a problem in this domain too, however there is the further problem that the Sugeno integral cannot be expressed as a linear combination of the inputs and weights. This means that the standard approach of fitting according to RMSE or MAE between observed and predicted outputs will not work.

For the methods we develop and discuss in this preliminary study, we focus on symmetric fuzzy measures μ, i.e., such that if $|U|=|V|$, then $\mu(U)=\mu(V)$. Let us assume the inputs have been ordered nonincreasingly in advance, $1\geq$

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

$x_1 \geq x_2 \geq \cdots \geq x_n \geq 0$ so that we have:

$$\mathsf{S}_{\mathbf{h}}(\mathbf{x}) = \bigvee_{j=1}^{n} x_j \wedge h_j, \tag{1}$$

where $h_j = \mu(\{1, 2, \ldots, j\})$ giving us n weights with $0 \leq h_1 \leq h_2 \leq \cdots \leq h_n = 1$.

This contribution is set out as follows. In Sec. 2 we discuss the problem of data interpolation, i.e., fitting with zero error. Then in Sec. 3 we develop a branch-and-bound algorithm for the approximation problem, i.e., when the observed $\mathbf{Y}$ is subject to error. This shall enable us to compare the performance of some well-known nonlinear solvers in an extensive set of numerical experiments. In Sec. 4 we present an example application of the results in the field of bibliometrics. Finally, in Sec. 5 we conclude the paper.

2 Data Interpolation: Fitting with Zero Error

In the case where the observed outputs can be expressed exactly as the Sugeno integral values with respect to the inputs and given fuzzy measure, identifying $\mathbf{h}$ is reasonably straightforward (and independent of the choice of an error measure), see also [16]. A key aspect of Sugeno integration is that the output is either one of the inputs or one of the fuzzy measure weights.

Consider the following example $[\mathbf{X}^T \ \mathbf{Y}^T]$ instances ($n = 5, m = 3$):

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$	$x_5^{(k)}$	$y^{(k)}$
1	0.73	0.70	0.31	0.30	0.21	0.33
2	1.00	0.91	0.90	0.78	0.60	0.78
3	0.86	0.71	0.67	0.42	0.33	0.63

For a given input vector $\mathbf{x}^{(k)}$ and observed output $y^{(k)}$, let j be such that $x_j^{(k)} \geq y^k \geq x_{j+1}^{(k)}$, under the assumption $x_{n+1}^{(k)} = 0$. The relevant weight that can help obtain $y^{(k)}$ as the output is hence h_j.

For $k = 1$ in the above table, the observed output falls between the 2nd and 3rd inputs and so we can deduce that $h_2 = 0.33$. Since $h_2 \leq h_3$, we have:

$$h_2 \wedge x_2^{(1)} = 0.33, \qquad h_3 \wedge x_3^{(1)} = 0.31,$$

and therefore the output is indeed equal to 0.33.

We have slightly different reasoning for $k = 2$, because $y^{(4)}$ is *equal* to the 4th input. As $h_3 \leq h_4$, we necessarily have that $h_4 \geq 0.78$ as well as $h_3 \leq 0.78$. This gives:

$$h_3 \wedge x_3^{(2)} \leq 0.78, \qquad h_4 \wedge x_4^{(2)} = 0.78,$$

and so the output is 0.78.

In the case of $k = 3$, the input falls between the 3rd and 4th inputs, so (similar to the case of $k = 1$) we will have $h_3 = 0.63$.

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

In general, for $x_j^{(k)} > y^{(k)} > x_{j+1}^{(k)}$, it follows that $h_j = y^{(k)}$ and for $x_j^{(k)} \geq y^{(k)} = x_{j+1}^{(k)}$ we have $h_j \leq y^{(k)} \leq h_{j+1}$. Any symmetric fuzzy measure satisfying these constraints will hence interpolate the data. Note that in the above example we can only provide an upper bound for h_1 and a lower bound for h_4.

3 Data Approximation: Fitting with Error

In the vast majority of practical instances, the y-values are subject to some kind of error. In such cases, there is no fuzzy measure that generates the Sugeno integral interpolating $(\mathbf{X}, \mathbf{Y})$ and thus we deal with the problem of data approximation.

Our problem becomes:

$$\text{minimize} \quad \left(\frac{1}{m} \sum_{k=1}^{m} \left| \mathsf{S}_{\mathbf{h}} \left(\mathbf{x}^{(k)} \right) - y^{(k)} \right|^p \right)^{1/p} \quad \text{w.r.t. } \mathbf{h} \tag{2}$$

such that $0 \leq h_1 \leq h_2 \leq \cdots \leq h_n = 1$ for $p = 1$ in the case of MAE and $p = 2$ in the case of RMSE minimization.

Remark 1. For computational convenience, we may reformulate S so that the optimization problems have only bounding-box constraints. Setting $h_j := \bigvee_{i=1}^{j} h'_i$ (cumulative maximum) with arbitrary $h'_1, \ldots, h'_n \in [0, 1]$, $h'_n = 1$, we get $\mathsf{S}_{\mathbf{h}'}(\mathbf{x}) = \bigvee_{j=1}^{n} x_j \wedge \bigvee_{i=1}^{j} h'_i$. Note that as the $\vee, \wedge$ operations are distributive over each other, we have: $\mathsf{S}_{\mathbf{h}'}(\mathbf{x}) = \bigvee_{j=1}^{n} \bigvee_{i=1}^{j} x_j \wedge h'_i$. In practice, we may also consider $h_j = 1 \wedge \sum_{i=1}^{j} \delta'_i$ (bounded cumulative sum), $\delta'_1, \ldots, \delta'_n \in [0, 1]$, $\delta'_n = 1$, which yields $\mathsf{S}_{\boldsymbol{\delta}'}(\mathbf{x}) = \bigvee_{j=1}^{n} x_j \wedge \sum_{i=1}^{j} \delta'_i$.

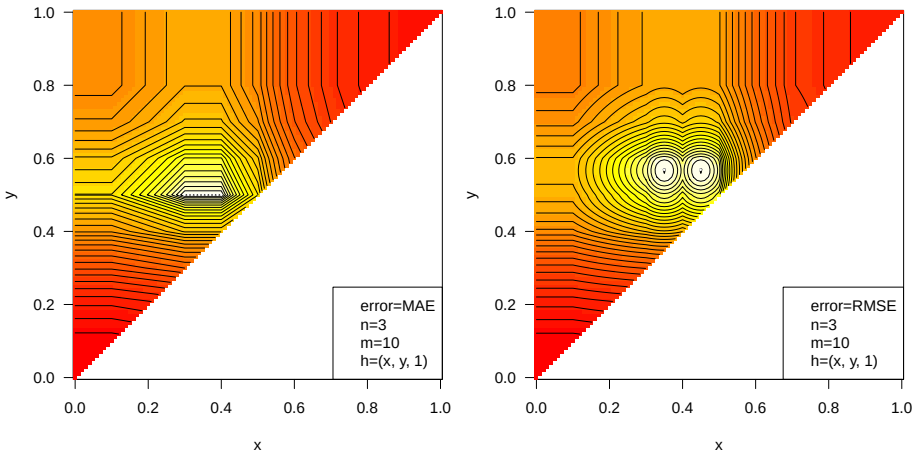


Fig. 1. Contour plots of MAE (left) and RMSE (right) functions for example data; $n = 3, m = 10$, fuzzy measure is given by $\mathbf{h} = (x, y, 1)$.

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

Unfortunately, both error measures are neither convex nor even quasi-convex in general. For instance, Fig. 1 gives example contour plots in the $n = 3$ case. Here the MAE function has multiple connected minima and RMSE has two quite different global minima. Studying more examples, we observe that it might also be the case that a local minimum is not a global minimum. Taking into account the fact that the error functions are not everywhere differentiable, such issues make the problem of fitting a Sugeno integral quite difficult, computationally speaking.

However, MAE and RMSE are Lipschitz continuous functions. More precisely, we have what follows.

Proposition 1. *For fixed* $\mathbf{X}, \mathbf{Y}$ *and any* $\mathbf{h}' \in [0,1]^n$*, the function*

$$\mathsf{E}_p(\mathbf{h}') = \left(\frac{1}{m} \sum_{k=1}^{m} \left| \bigvee_{j=1}^{n} \left(x_j^{(k)} \wedge \bigvee_{i=1}^{j} h'_i \right) - y^{(k)} \right|^p \right)^{1/p}$$

is a nonexpanding map for any $p \geq 1$*, i.e., for any* $\mathbf{h}' \in [0,1]^n$*,* $\boldsymbol{\delta} \in \mathbb{R}^n$ *such that* $\mathbf{h}' + \boldsymbol{\delta} \in [0,1]^n$ *it holds:*

$$|\mathsf{E}_p(\mathbf{h}' + \boldsymbol{\delta}) - \mathsf{E}_p(\mathbf{h}')| \leq \bigvee_{j=1}^{n} |\delta_j| \,.$$

Proof. First let us note that for each fixed $\mathbf{x}$, $\mathsf{S}_{\mathbf{h}}(\mathbf{x})$ as a function of h is itself nonexpanding:

$$\begin{aligned}
|\mathsf{S}_{\mathbf{h}'+\boldsymbol{\delta}}(\mathbf{x}) - \mathsf{S}_{\mathbf{h}'}(\mathbf{x})| &= \left| \bigvee_{j=1}^{n} \left| x_j \wedge \bigvee_{i=1}^{j} (h'_i + \delta_i) \right| - \bigvee_{j=1}^{n} \left| x_j \wedge \bigvee_{i=1}^{j} h'_i \right| \right| \\
&\leq \bigvee_{j=1}^{n} \left| x_j \wedge \bigvee_{i=1}^{j} (h'_i + \delta_i) - x_j \wedge \bigvee_{i=1}^{j} h'_i \right| \\
&= \bigvee_{j=1}^{n} \left| \left(x_j - \bigvee_{i=1}^{j} h'_i \right) \vee 0 - \left(x_j - \bigvee_{i=1}^{j} (h'_i + \delta_i) \right) \vee 0 \right| \\
&\leq \bigvee_{j=1}^{n} \left| \bigvee_{i=1}^{j} |\delta_i| \right| = \bigvee_{j=1}^{n} |\delta_j| \,.
\end{aligned}$$

And now:

$$\begin{aligned}
|\mathsf{E}_p(\mathbf{h}' + \boldsymbol{\delta}) - \mathsf{E}_p(\mathbf{h}')| &= m^{-1/p} \left| \|\mathsf{S}_{\mathbf{h}'+\boldsymbol{\delta}}(\mathbf{X}) - \mathbf{Y}\|_p - \|\mathsf{S}_{\mathbf{h}'}(\mathbf{X}) - \mathbf{Y}\|_p \right| \\
&\leq m^{-1/p} \, \|\mathsf{S}_{\mathbf{h}'+\boldsymbol{\delta}}(\mathbf{X}) - \mathsf{S}_{\mathbf{h}'}(\mathbf{X})\|_p = \left(\frac{1}{m} \sum_{k=1}^{m} \left| \mathsf{S}_{\mathbf{h}'+\boldsymbol{\delta}}(\mathbf{x}^{(k)}) - \mathsf{S}_{\mathbf{h}'}(\mathbf{x}^{(k)}) \right|^p \right)^{1/p} \\
&\leq \left(\frac{1}{m} \sum_{k=1}^{m} \left| \bigvee_{j=1}^{n} |\delta_j| \right|^p \right)^{1/p} = \bigvee_{j=1}^{n} |\delta_j|.
\end{aligned}$$

Hence, E_p is a nonexpansive function. □

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

1: Input: $\mathbf{X}, \mathbf{Y}, p$ and ε (precision);
2: Define struct $\mathcal{S} = \big\{(f, l, r, \mathbf{h})$, where $\mathbf{h} \in [0,1]^n$, $f = E_p(\mathbf{h})$, r – (Chebyshev) radius, l – lower bound for the estimate of $\min_{\boldsymbol{\delta}:\bigvee_{i=1}^n |\delta_i| \le r} E_p(\mathbf{h}+\boldsymbol{\delta})\big\}$;
3: Assume that $\mathbf{s} \prec \mathbf{s}'$, $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$, iff $s.l < s'.l$;
4: $q :=$ MinPriorityQueue() with elements in $\mathcal{S}$ ordered w.r.t. $\prec$;
5: $\mathbf{h}^* := (0.5, \ldots, 0.5)$; ▷ Start at the center of $[0,1]^n$
6: q.push($(E_p(\mathbf{h}^*), \text{EstimateLowerBound}(\mathbf{h}^*, 0.5), 0.5, \mathbf{h}^*)$);
7: **while** q is not empty **do**
8: $c := q$.pop(); ▷ Element with the lowest l
9: **if** $c.f < E_p(\mathbf{h}^*)$ **then**
10: $\mathbf{h}^* := c.\mathbf{h}$; ▷ New candidate solution
11: **end if**
12: **if** $E_p(\mathbf{h}^*) - c.l < \varepsilon$ **then**
13: break; ▷ Minimum found with precision ε
14: **end if**
15: $r := c.r/2$; ▷ Now we shall partition the n-cube $c.\mathbf{h} \pm c.r$ into subcubes:
16: **for** each possible $\mathbf{h} = (c.h_1 \pm r, \ldots, c.h_n \pm r)$ with $h_1 \le \cdots \le h_n$ **do**
17: $d = (E_p(\mathbf{h}), \text{EstimateLowerBound}(\mathbf{h}, r), r, \mathbf{h})$;
18: **if** $d.l < E_p(\mathbf{h}^*)$ **then**
19: q.push(d);
20: **end if**
21: **end for**
22: **end while**
23: return $\mathbf{h}^*$;

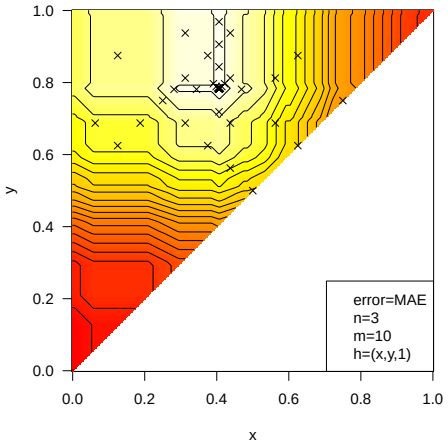
Fig. 2. A branch-and-bound algorithm for solving (2) with arbitrary precision ε. For simplicity we assume that h_n can be arbitrary, i.e., the capacity is not necessarily normalized.

3.1 Exact Algorithm: Branch-and-Bound

We can now develop a Branch-and-Bound-type algorithm, which enables us to locate the error function's minimum with arbitrary precision. Its pseudocode is given in Fig. 2. Please note that EstimateLowerBound$(\mathbf{h}, r)$ is used to estimate the lower bound of $\min_{\boldsymbol{\delta}:\bigvee_{i=1}^n |\delta_i| \le r} E_p(\mathbf{h}+\boldsymbol{\delta})$, i.e., the minimal error function value in the n-cube centered at $\mathbf{h}$ with Chebyshev radius of r. Based on Proposition 1, a crude (but inexpensive to compute) estimator to this value is $E_p(\mathbf{h}) - r$.

Remark 2. The lower bound may be slightly improved. Based on the fact that $\mathsf{S}_{\mathbf{h}}(\mathbf{x})$ is nondecreasing with respect to each element in $\mathbf{h}$, it holds $\mathsf{S}_{\mathbf{h}'-(r,\ldots,r)}(\mathbf{x}) \le \mathsf{S}_{\mathbf{h}'+\boldsymbol{\delta}}(\mathbf{x}) \le \mathsf{S}_{\mathbf{h}'+(r,\ldots,r)}(\mathbf{x})$ for each $\boldsymbol{\delta}$ such that $\bigvee_{i=1}^n |\delta_i| \le r$. Using a similar reasoning as in the proof of Proposition 1, we may approach:

$$\min_{\boldsymbol{\delta}:\bigvee_{i=1}^n |\delta_i| \le r} E_p(\mathbf{h}+\boldsymbol{\delta}) \ge \text{EstimateLowerBound}'(\mathbf{h}, r) := E_p(\mathbf{h}) - \sqrt[p]{\frac{1}{m}\sum_{k=1}^m \left|\left(\mathsf{S}_{\mathbf{h}}\left(\mathbf{x}^{(k)}\right) - \mathsf{S}_{\mathbf{h}-(r,\ldots,r)}\left(\mathbf{x}^{(k)}\right)\right) \vee \left(\mathsf{S}_{\mathbf{h}+(r,\ldots,r)}\left(\mathbf{x}^{(k)}\right) - \mathsf{S}_{\mathbf{h}}\left(\mathbf{x}^{(k)}\right)\right)\right|^p}.$$



k	1	2	3	4	5
$x_1^{(k)}$	0.414	0.494	0.695	0.001	0.383
$x_2^{(k)}$	0.475	0.708	0.799	0.005	0.534
$x_3^{(k)}$	0.485	0.914	0.968	0.057	0.737
$y^{(k)}$	0.475	0.708	0.788	0.057	0.534

k	6	7	8	9	10
$x_1^{(k)}$	0.987	0.629	0.128	0.292	0.302
$x_2^{(k)}$	0.992	0.690	0.209	0.591	0.385
$x_3^{(k)}$	0.995	0.707	0.295	0.721	0.593
$y^{(k)}$	0.987	0.690	0.288	0.591	0.385

Fig. 3. Contour plots of MAE (left) for an example data set $n = 3, m = 10$ (right). Points marked with "$\times$" represent centers of n-cubes visited by the branch-and-bound algorithm.

Let us study the example in Fig. 3, which concerns a data set with $n = 3$ and $m = 10$. The branch-and-bound algorithm converged in 73 iterations (precision of 10^{-10}): the minimum is $\mathbf{h}^* = (0.4069408, 0.7840181, 1.0)$ with $E_1(\mathbf{h}^*) = 0.0419531079$.

Unfortunately, the branch-and-bound algorithm usually needs many more iterations to converge: sometimes billions of points. Moreover, the amount of RAM needed strongly depends on n – in each iteration some n-cube is split into $O(2^n)$ subcubes. This causes the algorithm to be practically unusable for $n \gtrsim 5$.

3.2 Nonlinear Derivative-free Solvers

Taking the above observation on the performance of the branch-and-bound algorithm into account, we are in need of inspecting much faster and less memory demanding algorithms. From now on we shall focus on the case $p = 1$.

Let us then inspect the behavior of generic derivative-free nonlinear solvers (see, e.g., [11,15,18]) and verify whether they are useful in our approximation task. We take into account the Nelder–Mead (downhill simplex), COBYLA (Constrained Optimization BY Linear Approximation), and SUBPLEX algorithms. Moreover, for the sake of comparison, we also consider: the BOBYQA (Bound Optimization BY Quadratic Approximation), which constructs a quadratic approximation of the objective function, as well as the L-BFGS-B (limited memory Broyden–Fletcher–Goldfarb–Shanno) algorithm, which is a quasi-Newton method based on the approximation of the Hessian matrix of the objective function. The last two methods can possibly behave poorly as the function optimized for is not differentiable. The COBYLA algorithm allows inequality constraints, while with all the other methods we may only specify box constraints. In such

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

cases we deal with the objective function based on bounded cumulative sums (see Remark 1).

The objective function is expected to possess many plateaus. Each method may get stuck in a suboptimal solution and/or fail to converge. In order to increase the robustness of the solution, we restart each method from 100 randomly selected points and choose the least obtained value as the result. Yet, contrary to the branch-and-bound approach, none of these methods is guaranteed to find the exact solution with a given precision threshold.

3.3 Experiments

In order to verify the usefulness of the aforementioned generic nonlinear solvers, we consider the following scenarios. Firstly, we took $n = 3, 11, 101$ (2, 10, and 100 free parameters, respectively) and $m = 10, 100, 1000$. Secondly, there were three generation schemes for the elements of the $\mathbf{h}$ vectors:

- `o`) uniform distribution $U[0, 1]$,
- `f`) beta distribution $B(5, 5)$,
- `q`) beta distribution $B(0.25, 0.25)$.

Each element was sampled independently and the resulting vector was sorted nondecreasingly. Next, each of the m input vectors was constructed either by independently generating and then ordering:

- `u`) n random deviates $\sim U[0, 1]$,
- `r`) n random deviates $\sim B(u, v)$ with u and v sampled independently (for each vector separately) from $U[0.25, 5]$.

Finally, for each $\mathbf{x}^{(k)}$ the corresponding $y^{(k)}$ was set to:

- `n`) $0 \vee \left(1 \wedge \left(\mathsf{S}_{\mathbf{h}}\left(\mathbf{x}^{(k)}\right) + \varepsilon^{(k)}\right)\right)$, where $\varepsilon^{(k)}$ was sampled from the normal distribution with expected value 0 and standard deviation 0.1,
- `c`) $0 \vee \left(1 \wedge \left(\mathsf{S}_{\mathbf{h}}\left(\mathbf{x}^{(k)}\right) + \varepsilon^{(k)}\right)\right)$, where $\varepsilon^{(k)}$ was sampled from the Cauchy distribution with location 0 and scale 0.1,
- `p`) 0 with probability 0.5 and a random variate $\sim U[0, 1]$ otherwise (i.e., half of the outputs on average were not contaminated while the remaining were pure random noise).

We sampled 10 $(\mathbf{h}, \mathbf{X}, \mathbf{Y})$ triples in each of the 162 possible scenarios.

The COBYLA, SUBPLEX, and BOBYQA algorithms were implemented in the NLopt library [11]. For the other algorithms, we called the `optim()` function in R [17]. The relative convergence tolerance (`xtol`) was set to 10^{-14}.

Table 1 gives the relative error (i.e., $(E_1(\mathbf{h}^{(\text{computed})}) - E_1(\mathbf{h}^*))/E_1(\mathbf{h}^*)$) for each method as well as n and m. The branch-and-bound algorithm was only run for $n = 3$; in the $n = 11$ and $n = 101$ case we assumed that $E_1(\mathbf{h}^*)$ is the minimum of the 500 observed objective function values.

We observe that, overall, a method's worst-case performance decreases when n increases as well as when m decreases. The original $\mathbf{h}$ vectors (used to generate the data samples) are in many cases far from the optimal solutions. For

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

Table 1. Relative error for different n as well as m.

	quantile	0.00	0.25	0.50	0.75	0.90	0.99	1.00
n	method							
3	Nelder-Mead	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00159
	COBYLA	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00004
	SUBPLEX	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00004
	BOBYQA	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00015
	L-BFGS-B	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00159
	Original **h**	0.00000	0.00014	0.00387	0.03683	0.16077	0.41823	0.67732
11	Nelder-Mead	0.00000	0.00000	0.00000	0.00050	0.00493	0.06521	0.30351
	COBYLA	0.00000	0.00000	0.00000	0.00052	0.00434	0.06221	0.32084
	SUBPLEX	0.00000	0.00000	0.00000	0.00063	0.00480	0.06764	0.09748
	BOBYQA	0.00000	0.00000	0.00000	0.00041	0.00470	0.05210	0.17546
	L-BFGS-B	0.00000	0.00000	0.00000	0.00058	0.00566	0.05975	0.16225
	Original **h**	0.00000	0.00366	0.01971	0.10885	0.29507	0.80747	1.88920
101	Nelder-Mead	0.00000	0.00011	0.00108	0.00558	0.01629	0.09355	0.17555
	COBYLA	0.00000	0.00011	0.00090	0.00474	0.01782	0.09408	0.36617
	SUBPLEX	0.00000	0.00014	0.00115	0.00569	0.01666	0.09702	0.18982
	BOBYQA	0.00000	0.00003	0.00078	0.00474	0.01870	0.07507	0.15800
	L-BFGS-B	0.00000	0.00013	0.00096	0.00545	0.01950	0.08424	0.16859
	Original **h**	0.00000	0.00410	0.01569	0.06024	0.17737	0.53467	2.17816
	quantile	0.00	0.25	0.50	0.75	0.90	0.99	1.00
m	method							
10	Nelder-Mead	0.00000	0.00000	0.00000	0.00419	0.02313	0.13016	0.30351
	COBYLA	0.00000	0.00000	0.00000	0.00297	0.02714	0.09909	0.36617
	SUBPLEX	0.00000	0.00000	0.00000	0.00577	0.02536	0.09758	0.18982
	BOBYQA	0.00000	0.00000	0.00000	0.00425	0.02421	0.09083	0.17546
	L-BFGS-B	0.00000	0.00000	0.00000	0.00539	0.02549	0.09886	0.16859
	Original **h**	0.00000	0.04406	0.12079	0.24330	0.40664	1.13202	2.17816
100	Nelder-Mead	0.00000	0.00000	0.00000	0.00110	0.00432	0.01780	0.03576
	COBYLA	0.00000	0.00000	0.00000	0.00106	0.00396	0.01622	0.02994
	SUBPLEX	0.00000	0.00000	0.00000	0.00120	0.00398	0.01689	0.03082
	BOBYQA	0.00000	0.00000	0.00000	0.00086	0.00395	0.01763	0.02935
	L-BFGS-B	0.00000	0.00000	0.00000	0.00110	0.00380	0.01731	0.02710
	Original **h**	0.00000	0.00441	0.01542	0.03443	0.05758	0.10278	0.12463
1000	Nelder-Mead	0.00000	0.00000	0.00001	0.00034	0.00137	0.00972	0.01361
	COBYLA	0.00000	0.00000	0.00000	0.00032	0.00131	0.01070	0.01699
	SUBPLEX	0.00000	0.00000	0.00000	0.00034	0.00136	0.01116	0.01608
	BOBYQA	0.00000	0.00000	0.00000	0.00036	0.00145	0.01011	0.01550
	L-BFGS-B	0.00000	0.00000	0.00000	0.00034	0.00127	0.01142	0.01691
	Original **h**	0.00000	0.00002	0.00162	0.00466	0.00863	0.01517	0.01835

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

the nonlinear solvers, surprisingly, we observe no significant differences in their performance (as measured by the pairwise Wilcoxon test, all p-values ≥ 0.05). In the vast majority of cases, the minimum was found to a very acceptable degree of precision. However, solutions in the worst-case scenarios might be far from optimal. This happens for small m and large n, i.e., when the problems are "underdetermined".

4 Applications in Bibliometrics

For any continuous strictly increasing function $\varphi : [a, b] \to [0, 1]$ with $\varphi(a) = 0$ and $\varphi(b) = 1$ it holds that $\bigvee_{j=1}^{n} x_j \wedge h_j = \varphi^{-1}\left(\bigvee_{j=1}^{n} \varphi(x_j) \wedge \varphi(h_j)\right)$ for every $x_j, h_j \in [a, b]$. Therefore, the presented results may be straightforwardly generalized to the case of aggregation of elements on intervals other than $[0, 1]$.

The Hirsch h-index [10] is given by $\mathsf{H}(x_1, \ldots, x_n) = \max\{h : x_h \geq h\}$ for $x_1 \geq x_2 \geq \cdots \geq x_n$ with $x_1 \geq 1$. It may be shown, see [21], that for integer data, H is equivalent to the Sugeno integral on $[0, n]$ (values greater than n might be truncated) $\mathsf{S}_{\mathbf{h}}(x_1, \ldots, x_n) = \bigvee_{j=1}^{n} x_j \wedge j$, i.e., the underlying symmetric capacity is such that $h_j = j$, $j = 1, \ldots, n$.

Notably, some generalizations of the h-index exist in the literature, compare [8], many of which replace the "$\geq h$" term with some functions of h, e.g., h^2, ch for some c, etc. Such measures may be expressed as appropriate Sugeno integrals.

On September 30, 2016 from the Scopus database we fetched the metadata on all 3383 papers that cite Ronald R. Yager's 1988 article on OWA operators [22]. Based on this, we extracted the complete publication records of all the authors that published at least one work that cited [22]. For each publication record, we created a citation sequence, i.e., counted the number of citations to each paper authored by the corresponding scholar. The author name disambiguation algorithm in Scopus is imperfect, therefore we decided to cut off sequences of lengths greater than 1100 (among the researchers who authored more than 1000 papers we find Tasawar Hayat and Witold Pedrycz). Moreover, we filtered out citation sequences of lengths < 25. As the lengths vary (Min=25, 1st Qu.=40, Median=68, 3rd Qu.=127, Max=1092), we padded all the remaining $m = 1744$ sequences with zeroes so that they were of identical length $n = 1092$ (note that, among others, the Hirsch h- and many other bibliometric indices ignore the existence of papers with 0 citations entirely, see, e.g., [8]).

Our $\mathbf{X}$ data are integers in $\{0, 1, \ldots, 32028\}$ (the most cited paper in the set is by Lotfi A. Zadeh). Let us test the performance of the three derivative-free solvers while learning $\mathbf{h}$ such that $h_j = j$, $j = 1, \ldots, n$ in the case when $\mathbf{Y}$ is not subject to any error. Here, we consider the solutions leading to the least MSE over 50 experiment re-runs. Table 2 reports basic summary statistics for the empirical distribution of the differences between the obtained Sugeno integral values and the reference h-indices as well as the differences between the fitted fuzzy measures and the original $\mathbf{h}$. Unfortunately – perhaps because our data are on a discrete scale – the algorithms often fail to find the optimal solution.

Please cite this paper as: Gagolewski M., James S., Fitting symmetric fuzzy measures for discrete Sugeno integration, In: Kacprzyk J. et al. (Eds.), Advances in Fuzzy Logic and Technology 2017 (Advances in Intelligent Systems and Computing 642), Springer, 2018, pp. 104-116. doi:10.1007/978-3-319-66824-6_10

Table 2. Basic summary statistics for the empirical distribution of predicted vs true $\mathbf{Y}$ as well as estimated vs true $\mathbf{h}$ differences in the bibliometric data set example.

	predicted $\mathbf{Y}$ vs true $\mathbf{Y}$ difference			estimated $\mathbf{h}$ vs true $\mathbf{h}$ difference		
	Nelder-Mead	SUBPLEX	COBYLA	Nelder-Mead	SUBPLEX	COBYLA
Min.	−2.001	−4.199	−2.485	−35.820	−14.030	−35.090
1st Qu.	−0.004	−0.067	−0.011	−21.050	3.514	−19.940
Median	0.000	0.000	0.000	−5.494	12.940	−11.520
3rd Qu.	0.018	0.000	0.012	3.222	18.290	−3.860
Max.	2.532	1.899	2.112	21.460	31.680	8.914
MAE	0.164	0.192	0.202	12.570	13.043	13.042

5 Conclusions

Here we have observed that nonlinear derivative-free solvers are sometimes able to produce reasonable results in learning the simplest class of fuzzy measures for use with the Sugeno integral, however while they may work well on average – there is no guarantee of providing (even close to) an optimal solution. An open problem is hence how to construct algorithms that can obtain an arbitrary level of precision, which are more efficient than the branch-and-bound scheme-based approach we presented here.

In future research we can begin to approach this problem by considering paramaterized classes of fuzzy measures, i.e., generated via $h_i = f_{\mathbf{w}}(i)$ for some parameter vector $\mathbf{w}$, e.g., $f_{\mathbf{w}}(i) = w_1 i + w_2 i^2$ etc. While the use of non-symmetric measures will expand the difficulty of the problem, in general we can apply similar methods to those we develop for symmetric cases. From the optimization perspective, a more difficult problem lies in considering data and the elements of $\mathbf{h}$ given over a discrete chain, e.g., as we essentially observe in bibliometrics.

While here we approached the task as a pure optimization problem, in a statistical setting, we could be interested in the problem of constructing estimators of $\mathbf{h}$ enjoying some useful properties like unbiasedness, consistency, etc. To this end, results have already been obtained for Sugeno integrals (and other lattice polynomial functions) in, among others, [6,7,13].

Acknowledgments. This study was supported by the National Science Center, Poland, research project 2014/13/D/HS4/01700. Data provided by Scopus.com.

References

1. Anderson, D., Keller, J., Havens, T.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued Sugeno fuzzy integral. Lecture Notes in Artificial Intelligence 6178, 502–511 (2010)
2. Beliakov, G.: How to build aggregation operators from data. International Journal of Intelligent Systems 18, 903–923 (2003)

3. Beliakov, G., Bustince, H., Calvo, T.: A Practical Guide to Averaging Functions. Springer (2016)
4. Beliakov, G., James, S.: Using linear programming for weights identification of generalized Bonferroni means in R. Lecture Notes in Computer Science 7647, 35–44 (2012)
5. Bullen, P.: Handbook of means and their inequalities. Springer Science+Business Media, Dordrecht (2003)
6. Dukhovny, A.: Lattice polynomials of random variables. Statistics and Probability Letters 77, 989–994 (2007)
7. Gagolewski, M., Grzegorzewski, P.: S-statistics and their basic properties. In: Borgelt, C., et al. (eds.) Combining Soft Computing and Statistical Methods in Data Analysis, Advances in Intelligent and Soft Computing, vol. 77, pp. 281–288. Springer (2010)
8. Gagolewski, M., Mesiar, R.: Monotone measures and universal integrals in a uniform framework for the scientific impact assessment problem. Information Sciences 263, 166–174 (2014)
9. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: Aggregation functions. Cambridge University Press (2009)
10. Hirsch, J.E.: An index to quantify individual's scientific research output. Proceedings of the National Academy of Sciences 102(46), 16569–16572 (2005)
11. Johnson, S.G.: The NLopt nonlinear-optimization package (2017), http://ab-initio.mit.edu/nlopt
12. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic. Theory and applications. Prentice Hall PTR, New Jersey (1995)
13. Marichal, J.L.: Weighted lattice polynomials of independent random variables. Discrete Applied Mathematics 156, 685–694 (2008)
14. Mesiar, R., Gagolewski, M.: H-index and other Sugeno integrals: Some defects and their compensation. IEEE Transactions on Fuzzy Systems 24(6), 1668–1672 (2016)
15. Nocedal, J., Wright, S.: Numerical Optimization. Springer-Verlag, New York (2006)
16. Prade, H., Rico, A., Serrurier, M.: Elicitation of Sugeno integrals: A version space learning perspective. Lecture Notes in Computer Science 5722, 392–401 (2009)
17. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2017), http://www.R-project.org
18. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization 56, 1247–1293 (2013)
19. Sugeno, M.: Theory of fuzzy integrals and its applications. Ph.D. thesis, Tokyo Institute of Technology (1974)
20. Torra, V.: Learning weights for the quasi-weighted means. IEEE Transactions on Fuzzy Systems 10(5), 653–666 (2002)
21. Torra, V., Narukawa, Y.: The h-index and the number of citations: Two fuzzy integrals. IEEE Transactions on Fuzzy Systems 16(3), 795–797 (2008)
22. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man, and Cybernetics 18(1), 183–190 (1988)
23. Yager, R.R., Kacprzyk, J. (eds.): The ordered weighted averaging operators. Theory and applications. Kluwer Academic Publishers, Norwell (1997)
24. Yuan, B., Klir, G.J.: Constructing fuzzy measures: A new method and its application to cluster analysis. In: Proc. NAFIPS'96, pp. 567–571 (1996)