A Generalization of the Bonferroni Mean Based on Partitions

Gleb Beliakov, Simon James
School of Information Technology
Deakin University
Burwood, Victoria 3125
Australia
Email: gleb@deakin.edu.au, sjames@deakin.edu.au

Radko Mesiar
Dept. Mathematics FCE STU
Slovak University of Technology, Slovakia, and
UTIA CAS Prague
Czech Republic
Email: radko.mesiar@stuba.sk

***Abstract*—The mean defined by Bonferroni in 1950 (known by the same name) averages all non-identical product pairs of the inputs. Its generalizations to date have been able to capture unique behavior that may be desired in some decision-making contexts such as the ability to model mandatory requirements. In this paper, we propose a composition that averages conjunctions between the respective means of a designated subset-size partition. We investigate the behavior of such a function and note the relationship within a given family as the subset size is changed. We found that the proposed function is able to more intuitively handle multiple mandatory requirements or mandatory input sets.**

***Index Terms*—Aggregation functions, Bonferroni mean, mandatory criteria, decision making.**

I. Introduction

The selection of an appropriate aggregation function is an important step in any context where a summarized representation or overall evaluation of an input set is required. Although standard functions such as the arithmetic mean or median often produce reasonable results, in some situations more complex aggregation functions are required in order to reflect the preferences of decision makers or the nuances of the problem at hand. There is no *best* aggregation function, so researchers continue to search for new methods and study aggregation behavior that could be useful in practical situations.

In the original paper where the Bonferroni mean was defined [1], of particular interest were the inequalities associated with different choices of the exponent parameters. It has gained interest today as a useful aggregation function for decision making, as generalizations of the Bonferroni mean proposed in [2], [3] allow users to model an arbitrary number of mandatory requirements whilst also taking into account weights and the contribution of all inputs.

Semantically, the generalized Bonferroni mean takes the average of the evaluations of each statement

$$x_i \; AND \text{ the average of the remaining } x_j, j \neq i.$$

Here we propose a construction that considers all partitions of the input set for a given size, i.e. we evaluate the statements,

$$\text{the average of } \; \{x_i : i \in E\}$$

$$AND$$

$$\text{the average of the remaining } \; \{x_j : j \in E^c\},$$

where E^c denotes the complement of E and each input $\mathbf{x} \in \mathcal{X} = E \cup E^c$, e.g. for five inputs we can take the conjunctions of the means of all combinations of 2 and 3 arguments.

This construction further generalizes that originally proposed, however we can now enforce mandatory requirements relating to coalitions of inputs. For instance, consider the following table showing job applicants with their scores for 5 criteria.

TABLE I
Job applicant evaluations against multiple criteria

	Logan	Betsy	Remy	Erik
Presentation	0	1	0.8	0.5
Communication	0	0.6	0.9	0.8
Leadership	0.9	0	0.5	1
Qualifications	1	0.3	0.7	0
Experience	1	1	0	0.5

Company X may decide that whilst presentation, communication and leadership are essential characteristics, formal qualifications and experience are desirable but not mandatory.

If they were to average the scores using a weighted arithmetic mean, Logan's scores of zero for presentation and communication (mandatory requirements) would be compensated for by his experience and qualifications. On the other hand, if they were to use a geometric mean (which will give an output of zero if *any* inputs are zero), Remy's score of zero for experience would make him unemployable, even though he scores quite well for the mandatory categories.

With the proposed operator, we can enforce that all scores for presentation, communication and leadership should be above zero for a non-zero output, however still take into consideration the scores for experience and qualifications.

On the other hand, if Company X decided that they just wanted at least 2 of the 3 essential characteristics to be satisfied, this can also be modeled with the proposed approach.

The paper will be structured as follows. In Section II, we will give an overview of aggregation functions and the generalized Bonferroni mean. In Section III we define our proposed operator which we will refer to as the *Bonferroni partition mean* (BPM). In Section IV we will outline some of the important properties of this operator and then we will provide some numerical examples in Section V before concluding.

II. Preliminaries

This section will give an overview of aggregation functions. In particular, we will focus on averaging aggregation functions and the generalized Bonferroni mean.

A. Aggregation Functions

Aggregation functions provide a single value that is usually intended to give an overall representation of the input set. Aggregation functions have become of increasing interest with a number of monographs appearing in recent years that detail their properties and how to construct them for particular situations [4]–[6]. Although aggregation functions can be defined over arbitrary real intervals, lattices and other spaces, we will consider the following definitions over the unit interval.

Definition 1: An aggregation function $f : [0,1]^n \to [0,1]$ is a function non-decreasing in each argument and satisfying $f(0,\ldots,0) = 0$ and $f(1,\ldots,1) = 1$.

Aggregation functions can be classed according to their behavior with respect to the minimum and maximum inputs.

Definition 2: An aggregation function is considered to be: *averaging* when $\min(\mathbf{x}) \le f(\mathbf{x}) \le \max(\mathbf{x})$, *conjunctive* when $f(\mathbf{x}) \le \min(\mathbf{x})$, *disjunctive* when $f(\mathbf{x}) \ge \max(\mathbf{x})$, and *mixed* otherwise.

Due to the monotonicity of aggregation functions, averaging behavior is equivalent to idempotency, i.e. $f(t,t,...,t) = t$.

An important generalized family of averaging functions are the weighted quasi-arithmetic means. We will refer to some of their special cases throughout.

Definition 3: For a strictly monotone continuous generating function $\phi : [0,1] \to [-\infty,\infty]$ and weighting vector $\mathbf{w}$, the weighted quasi-arithmetic mean is given by,

$$QAM_{\mathbf{w}}(\mathbf{x}) = \phi^{-1}\left(\sum_{i=1}^{n} w_i \phi(x_i)\right). \quad (1)$$

Special cases include weighted arithmetic means, $WAM(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i$ where $\phi(t) = t$, weighted power means $PM_q(\mathbf{x}) = \left(\sum_{i=1}^{n} w_i x_i^q\right)^{1/q}$ where $\phi(t) = t^q$ and weighted geometric means $G(\mathbf{x}) = \prod_{i=1}^{n} x_i^{w_i}$ if $\phi(t) = -\ln t$. The weights w_i are usually non-negative and sum to one.

This paper considers functions based on the Bonferroni mean. The Bonferroni mean was defined in 1950 [1] and generalizations have appeared since in [2], [3], [7], [8]. In its original form, it is defined as follows.

Definition 4: Let $p,q \ge 0$ and $x_i \ge 0, i = 1,\ldots,n$. The Bonferroni mean is the function

$$B^{p,q}(\mathbf{x}) = \left(\frac{1}{n(n-1)} \sum_{i,j=1, i\neq j}^{n} x_i^p x_j^q\right)^{\frac{1}{p+q}}. \quad (2)$$

The parameters p,q can take on any value and hence $q = 0$ (or $p = 0$) reduces the Bonferroni mean to a power mean (and therefore any of the special cases). Adjusting the ratio $\frac{p}{q}$ allows the Bonferroni mean to graduate between the geometric mean and the maximum operator. One can observe that there must exist at least one pair (i,j) such that $x_i, x_j > 0$, to obtain a non-zero output $B^{p,q}(\mathbf{x}) > 0$.

In [3], the Bonferroni mean was expressed as a composed aggregation function. We denote the vector in $[0,1]^{n-1}$ that includes the arguments from $\mathbf{x} \in [0,1]^n$ in each dimension except the i-th by $\{x_j : j \neq i\} = (x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$.

Definition 5: [3]. Let $\mathbb{M}$ denote a 3-tuple of aggregation functions $< M_1, M_2, C >$, with $M_1 : [0,1]^n \to [0,1]$, $M_2 : [0,1]^{n-1} \to [0,1]$ both averaging and $C : [0,1]^2 \to [0,1]$ conjunctive, with the diagonal of C denoted by $d_C(t) = C(t,t)$ and its inverse diagonal d_C^{-1}. The generalized Bonferroni mean is given by,

$$\begin{aligned} B_{\mathbb{M}}(\mathbf{x}) &= d_C^{-1}\Big(M_1\Big(C\big(x_1, M_2(\{x_j : j \neq 1\})\big),\ldots \\ &\quad \ldots, C\big(x_n, M_2(\{x_j : j \neq n\})\big)\Big)\Big). \end{aligned} \quad (3)$$

The original Bonferroni mean is returned where $M_1 = WAM(\mathbf{x})$, $M_2 = PM_q(\mathbf{x})$ and $C = x^p y^q$ (with all weights equal).

Although we usually choose C to be conjunctive (as it generalizes the product operation), in principle it can be any 2-variate function with an invertible diagonal. In [3] it was shown that changing C from conjunctive to disjunctive inversely affects the overall function behavior, i.e. if C is conjunctive then $B_{\mathbb{M}}$ tends toward the higher inputs while if it is disjunctive then it will tend toward lower inputs.

If M_1, M_2 are weighted aggregation functions, they will have weighting vectors of different dimension. Care should be taken so that the weights are consistent with the application and inputs. One option is to use stable weighting functions [9], obtained in the following way.

Given a weighting vector $\mathbf{u} \in [0,1]^n$, the vectors $\mathbf{u}^i \in [0,1]^{n-1}$, $i = 1, \ldots, n$ are defined by

$$u_j^i = \frac{u_j}{\sum_{k \neq i} u_k} = \frac{u_j}{1 - u_i}, \quad u_i \neq 1. \tag{4}$$

Note that for every i, $\mathbf{u}^i$ sums to one if and only if $\mathbf{u}$ sums to one.

This allows one to either use the same weighting vector or differing vectors if each stage of aggregation requires it. We can now look at adapting this construction so that partitions other than $x_i / \{x_j : j \neq i\}$ can be used.

III. The Bonferroni Partition Mean

With the formula given in Eq. (3), each component takes a single input with the average of those remaining. This enables us to use projections (i.e. weighting vectors with $w_i = 1$ for some i and 0 otherwise) on M_1 to enforce mandatory requirements. We now propose a function that allows us to pay attention to subsets based on all possible partitions of k-ary and $(n-k)$-ary subsets.

Definition 6: Fix $n \in \mathcal{N}/\{1\}$ and $k \in \{1, \ldots, n-1\}$. Let $M_\mu, M_k, M_{n/k}, C$ denote a 4-tuple of aggregation functions,

$M_\mu : [0,1]^p \to [0,1]$, where $p = \begin{pmatrix} n \\ k \end{pmatrix}$,
$M_k : [0,1]^k \to [0,1]$,
$M_{n/k} : [0,1]^{n-k} \to [0,1]$, and
$C : [0,1]^2 \to [0,1]$.

The Bonferroni partition mean is the function,

$$BPM(\mathbf{x}) = d_C^{-1}\left(M_\mu(\{C(M_k(\mathbf{x}_{E_i}), M_{n/k}(\mathbf{x}_{E_i^c}))\}_{i=1,\ldots,p})\right) \tag{5}$$

where E_i denotes the i-th subset of $\{1, \ldots, n\}$ when all k-sized subsets are arranged in lexicographical ordering, C is a function with a continuous strictly increasing (and invertible) diagonal d_C and inverse diagonal d_C^{-1}, and $M_\mu, M_k, M_{n/k}$ are averaging.

We can recover the previous generalization of the Bonferroni mean (Eq. (3)) with $k = 1$ and $M_k(x) = x$, and hence also the original Bonferroni mean.

In this case we have three averaging functions. Consider we use weighted functions such as quasi-arithmetic means, It is straightforward enough to determine weighting vectors for M_k and $M_{n/k}$ from an initial weighting vector $\mathbf{u}$ for consistency (i.e. analogously to the generalized Bonferroni mean defined in the previous section), however how to weight the resulting products aggregated by C requires a weight to be associated with each set E. We hence consider a framework for specifying all weights of the associated means from a single set function μ, where a weight is allocated to all subsets of the input vector $\{1, \ldots, n\}$, normalized for a given k. The following example helps illustrate this weighting convention.

Example 1: Consider a Bonferroni partition mean defined for $n = 4$ inputs. We denote the individual weights by

$$\mu(1) = 0.4, \mu(2) = 0.3, \mu(3) = 0.2, \mu(4) = 0.1.$$

For pairs we allocate the weights

$$\mu(1,2) = 0.5, \mu(1,3) = 0.2, \mu(1,4) = 0.1,$$
$$\mu(2,3) = 0, \mu(2,4) = 0.1, \mu(3,4) = 0.1,$$

and finally for 3-tuples we specify the weights

$$\mu(1,2,3) = 0.4, \mu(1,2,4) = 0, \mu(1,3,4) = 0.5$$
$$\text{and } \mu(2,3,4) = 0.1.$$

Note that for a given set size k, the weights add to 1.

If we set $k = 2$, then we can define the weighting vector for M_μ directly from the $p = 6$ pairs, $(0.5, 0.2, 0.1, 0, 0.1, 0.1)$. We then obtain each weighting vector for M_k and $M_{n/k}$ using the individual weights, so for $E_i = \{1,3\}$, M_k would use the weighting vector $(\frac{\mu(1)}{\mu(1)+\mu(3)}, \frac{\mu(3)}{\mu(1)+\mu(3)}) = (0.67, 0.33)$, while $M_{n/k}$ has the weights $(\frac{\mu(2)}{\mu(2)+\mu(4)}, \frac{\mu(4)}{\mu(2)+\mu(4)}) = (0.75, 0.25)$ and so on for other E_i.

Each μ then determines a family of BPM functions, $k = 1, \ldots, n-1$.

If M_μ, M_k and $M_{n/k}$ are weighted arithmetic means and C is the product operation, the weighting measure can be chosen such that any choice of k will yield the same output for any set of inputs.

Firstly, we can express the function as

$$BPM = \sqrt{\sum_{E \subset N, |E| = k} \mu(E) \frac{\sum\limits_{i \in E} \mu(i) x_i}{\sum\limits_{i \in E} \mu(i)} \cdot \frac{\sum\limits_{j \in E^c} \mu(j) x_j}{\sum\limits_{j \in E^c} \mu(j)}}.$$

As the two sets, E and E^c are complementary, it will be the case that

$$\sum_{j \in E^c} \mu(j) = 1 - \sum_{i \in E} \mu(i)$$

and therefore if we choose each $\mu(E)$ such that it is equal to

$$\mu(E) = \frac{\left(\sum\limits_{i \in E} \mu(i)\right)\left(1 - \sum\limits_{i \in E} \mu(i)\right)}{\sum\limits_{F \subset N, |F| = k} \mu(F)}, \tag{6}$$

then the normalizing denominator sums from M_k and $M_{n/k}$ will cancel out. Since we take the products

$$\sum_{i \in E} \mu(i) x_i \cdot \sum_{j \in E^c} \mu(j) x_j$$

for all combinations of E, E^c, there will then be an equal number of the $\mu(i)x_i \cdot \mu(j)x_j$ terms being added together, regardless of the choice of k. For other averaging functions, however, this will not be the case and so it will not generally be possible to choose the weights in this way.

As we will see in Section V, The value of k (along with the choice of means) can be used to control the number of mandatory requirements.

For interpreting the weights of M_μ, it should be noted that a high weight to $\mu(E)$ will be associated with the conjunction of $M_k(\mathbf{x}_E)$ and $M_{n/k}(\mathbf{x}_{E^c})$, thus simultaneously giving importance to E and E^c. Furthermore, in the previous example where $k = n - k = 2$, the pair $\{x_1, x_2\}$ is allocated a weight of $\mu(1,2) = 0.5$ when aggregated by M_k but a weight of $\mu(3,4) = 0.1$ when aggregated by $M_{n/k}$. The weight combination and behavior of each of the averaging operators hence will influence the overall function. We could, for instance, choose M_k and $M_{n/k}$ so that they favor higher or lower inputs respectively, after which the weights in M_μ would indicate which sets of inputs should given more importance when they are high or when they are low.

IV. Properties

There are a number of properties that can be established for the proposed operator which can help to guide its application. After first showing that the function is an aggregation operator, we will turn to results based on absorbing elements which can help us define functions to model mandatory requirements.

Theorem 1: The Bonferroni partition mean defined in Eq. (5) is an aggregation function.

Proof: The boundary conditions, $BPM(0,\ldots,0) = 0, BPM(1,\ldots,1) = 1$ and monotonicity follow from the properties of the aggregation functions, $M_\mu, M_k, M_{n/k}, C$. ■

For the following theorem and propositions we use the notation C_i, $C_i(M_k, M_{n/k})$ and so on to denote the i-th argument of M_μ corresponding with the set E_i.

Theorem 2: For $M_\mu, M_k, M_{n/k}$ averaging aggregation functions, the BPM is an averaging aggregation function, independently of C.

Proof: Since the BPM is an aggregation function, it is sufficient to show that idempotency holds. We can denote the vectors of length k and $n-k$ such that all the elements are the same by $\mathbf{t}_k$ and $\mathbf{t}_{n-k}$ respectively. We then have

$$\begin{aligned} &BPM(t,\ldots,t) = \\ &d_C^{-1}\left(M_\mu(\{C_i(M_k(\mathbf{t}_k), M_{n/k}(\mathbf{t}_{n-k}))\}_{i=1,\ldots,p})\right) \\ &\quad = d_C^{-1}\left(M_\mu(C_1(t,t),\ldots,C_p(t,t))\right) \\ &\quad = d_C^{-1}\left(C(t,t)\right) \\ &\quad = t. \end{aligned}$$

■

The following proposition establishes sufficient conditions for symmetry of the BPM.

Proposition 1: If the aggregation functions $M_\mu, M_k, M_{n/k}$ are symmetric, then the BPM is also symmetric, independently of C.

Proof: Consider the input vector $\mathbf{x} = (x_1,\ldots,x_n)$ and a permutation $\mathbf{x}_\alpha = (x_{\alpha(1)},\ldots,x_{\alpha(n)})$. It is clear that for each E_i there exists a corresponding set $E_{\alpha(j)}$ comprising the same elements so that $C_i(M_k(\mathbf{x}_{E_i}), M_{n/k}(\mathbf{x}_{E_i^c})) = C_{\alpha(j)}(M_k(\mathbf{x}_{E_{\alpha(j)}}), M_{n/k}(\mathbf{x}_{E^c_{\alpha(j)}}))$. From the symmetry of M_μ it follows that $M_\mu(C_1,\ldots,C_p) = M_\mu(C_{\alpha(1)},\ldots,C_{\alpha(p)})$ and hence that $BPM(\mathbf{x}) = BPM(\mathbf{x}_\alpha)$. ■

In some cases, it may be useful to be able to define the dual of the Bonferroni partition mean from its components. For instance, aggregation functions over Atanassov orthopairs (or intuitionistic fuzzy sets) can be defined in this way. The next proposition shows that this can be done by using the dual of each component in its construction.

Proposition 2: The dual of a Bonferroni partition mean is given by the Bonferroni partition mean defined with respect to the dual functions $M_\mu^d, M_k^d, M_{n/k}^d, C^d$.

Proof: Using the standard negation, $N(t) = 1 - t$ and omitting the $i = 1,\ldots,p$ from the argument notation of the mean M_μ, i.e. assume $\{C_i(.)\} = (C_1(.),\ldots,C_p(.))$, we have,

$$\begin{aligned} &BPM^d(\mathbf{x}) = 1 - BPM(1-\mathbf{x}) \\ &\quad = 1 - d_C^{-1}\left(M_\mu(\{C_i(M_k(1-\mathbf{x}_{E_i}), M_{n/k}(1-\mathbf{x}_{E_i^c}))\})\right) \\ &\quad = 1 - d_C^{-1}\left(M_\mu(\{C_i(1-M_k^d(\mathbf{x}_{E_i}), 1-M_{n/k}^d(\mathbf{x}_{E_i^c}))\})\right) \\ &\quad = 1 - d_C^{-1}\left(M_\mu(\{1-C_i^d(M_k^d(\mathbf{x}_{E_i}), M_{n/k}^d(\mathbf{x}_{E_i^c}))\})\right) \\ &\quad = 1 - d_C^{-1}\left(1-M_\mu^d(\{C_i^d(M_k^d(\mathbf{x}_{E_i}), M_{n/k}^d(\mathbf{x}_{E_i^c}))\})\right). \end{aligned}$$

It follows from $f^d(t) = 1 - f(1-t)$ that $f^{d^{-1}}(t) = 1 - f^{-1}(1-t)$, i.e. the inverse of the dual will be related to the inverse of the function in the same way (since over the unit interval the operations are equivalent to reflections and rotations respectively), and so finally we have

$$BPM^d(\mathbf{x}) = d_{C^d}^{-1}\left(M_\mu^d(\{C_i^d(M_k^d(\mathbf{x}_{E_i}), M_{n/k}^d(\mathbf{x}_{E_i^c}))\})\right).$$

■

We now turn to aggregation behavior associated with absorbing elements.

Definition 7: An element $a \in [0,1]$ is an absorbing element (or annihilator) of an aggregation function f if it follows that $f(\mathbf{x}) = a$ whenever $x_i = a$ for some i.

A typical example of an aggregation function with an absorbing element is the geometric mean with $a = 0$. Whenever any of the arguments of the geometric mean are zero, the output will be zero regardless. We have the following propositions for the BPM.

Proposition 3: If $M_\mu, M_k, M_{n/k}$ are averaging aggregation functions, if C and any two of $M_\mu, M_k, M_{n/k}$ have an absorbing element a, then a is an absorbing element of the resulting BPM.

Proof: We look at the two possible cases.

Case I: $M_k, M_{n/k}, C$ have an absorbing element a.
For each argument of M_μ, i.e. $C_i(M_k(\mathbf{x}_{E_i}), M_{n/k}(\mathbf{x}_{E_i^c}))$, if $a \in E_i$ or $a \in E_i^c$ then we have $C_i(a, M_{n/k}(\mathbf{x}_{E_i^c})) = a$ or $C_i(M_k(\mathbf{x}_{E_i}), a) = a$. Then it follows from the idempotency of M_μ that $BPM(\mathbf{x}) = d_C^{-1}(M_\mu(a, a, \ldots, a)) = d_C^{-1}(a)$. Since d_C^{-1} is defined such that $d_C^{-1}(C(t,t)) = t$, if $C(a,a) = a$, then we also have $d_C^{-1}(a) = a$ and hence a is an absorbing element of BPM.

Case II: M_μ, C and one of $M_k, M_{n/k}$ have an absorbing element a.
If M_k (or alternatively $M_{n/k}$) has an absorbing element a, then there exists an E_i such that $C_i(M_k(\mathbf{x}_{E_i}), M_{n/k}(\mathbf{x}_{E_i^c})) = C_i(a, M_{n/k}(\mathbf{x}_{E_i^c})) = a$. As M_μ has the same absorbing element, we again have $BPM(\mathbf{x}) = d_C^{-1}(a) = a$. ∎

The following situation will be particularly useful in specifying a minimum number of mandatory requirements.

Proposition 4: For M_μ averaging, if M_k and C have an absorbing element a, unless there are more than k values in the input vector such that $x_j \neq a$, then $BPM(\mathbf{x}) = a$.

Proof: For each $C_i(M_k(\mathbf{x}_{E_i}), M_{n/k}(\mathbf{x}_{E_i^c}))$ we require a set of k values $x_j \neq a$ so that $M_k(\mathbf{x}_{E_i}) \neq a$ and at least one additional value so that $M_{n/k}(\mathbf{x}_{E^c}) \neq a$ (from idempotency). If this requirement is not fulfilled for at least one E_i, then all C_i components will be a and $BPM(\mathbf{x}) = d_C^{-1}(M_\mu(a, \ldots, a) = a$. ∎

The upshot of this proposition is that we can use absorbing properties of M_k, C and adjust the size of k to enforce the desired number of mandatory requirements. The following example helps illustrate this.

Example 2: Let M_k be the geometric mean with absorbing element $a = 0$ and C the product operation which also has absorbing element 0. For $n = 7$ and $k = 4$, the resulting Bonferroni partition mean will require at least 5 non-zero values to give an output greater than 0.

We can also then use projections for the weighting vector in M_μ, allocating all the weight to a single subset E to ensure that all its elements are required to be non-zero for a zero output. Consider the following example.

Example 3: For $n = 6, k = 3$ We choose M_k as a weighted geometric mean G and C as the product, both with absorbing element $a = 0$. We assign the weights $\mu(1,4,5) = 1$ and $\mu(E_i) = 0$ otherwise. The resulting BPM can be expressed,

$$BPM(\mathbf{x}) = \sqrt{G(x_1, x_4, x_5) \cdot M_{6/3}(x_2, x_3, x_6)},$$

which will give an output of zero unless x_1, x_4 and x_5 are all greater than zero.

In the following section, we provide some numerical examples to help illustrate the behavior of the BPM based on different weights and components.

V. Numerical Examples

We now present some numerical examples to help illustrate the behavior of the Bonferroni partition mean. We implemented the composed function using the R programming language and the code is available from our website [1].

For the following examples, we first specify the means to be used for $M_k, M_{n/k}$. For calculating the output of a given input vector $\mathbf{x}$, it is supplied to the function environment as well as the weighting vector $\mathbf{u}$ (used for $M_k, M_{n/k}$), the value of k and optionally the position of the input set that is to be made mandatory (if a geometric mean is used for M_k). An array is then built of all possible subsets of size k in lexicographic order and this is then used to calculate the output.

We set M_k as the geometric mean, C as the standard product operation and $M_\mu, M_{n/k}$ as unweighted arithmetic means.

Table II shows some example input vectors and the output of this function as k is incremented. As soon as k is equal to the number of 0 values, the output will be zero.

TABLE II
EFFECT OF k IN ENFORCING A MINIMUM NUMBER OF MANDATORY REQUIREMENTS.

x	k=1	k=2	k=3	k=4	k=5
(1,1,1,0,0,0)	0.4472136	0.2236068	0	0	0
(1,1,1,1,0,0)	0.6324555	0.4472136	0.2581989	0	0
(1,1,1,1,1,0)	0.8164966	0.7071068	0.5773503	0.4082483	0

We can also introduce a weighting vector in order to determine M_k and $M_{n/k}$ while using equal weights for M_μ. We note that the minimum number of mandatory requirements is still controlled by k, however a much higher weight is allocated to x_1 and x_2. Table III shows output values where the weighting vector $\mathbf{u} = (0.4, 0.4, 0.05, 0.05, 0.05, 0.05)$ is used for $M_k, M_{n/k}$. At least k values must be non-zero, however the weights are also taken into account in the calculation.

Lastly, we can consider applying unequal weights for M_μ. In particular, we look at the specific case of $k = 3$ and use a projection for the subset $\{1, 2, 3\}$. Table IV shows the outputs

[1]http://aggregationfunctions.wordpress.com

TABLE III
WEIGHTING VECTOR $\mathbf{u} = (0.4, 0.4, 0.05, 0.05, 0.05, 0.05)$ USED FOR $M_k, M_{n/k}$. SHOWING EFFECT OF k FOR DIFFERENT INPUTS.

$\mathbf{x}$	k=1	k=2	k=3	k=4	k=5
(1,1,0,0,0,0)	0.4714045	0	0	0	0
(0,0,1,1,1,1)	0.3244428	0.2108185	0.1084652	0	0
(1,1,1,1,0,0)	0.758962	0.5574714	0.3366502	0	0

for the BPM with M_k an unweighted geometric mean and $M_{n/k}$ an unweighted arithmetic mean. If any of x_1, x_2, x_3 are zero, the output will be zero. If all three are above zero, we still require at least one of the other inputs to be greater than zero in order to avoid a zero output. Good scores for non-mandatory requirements still are able to compensate for lower scores in the mandatory subset.

TABLE IV
WITH $k = 3$ AND A PROJECTION USED FOR THE SUBSET $\{1, 2, 3\}$ IN THE WEIGHTING VECTOR OF M_μ.

$\mathbf{x}$	$k = 3$
(1,1,0,1,1,1)	0
(1,0,1,1,1,1)	0
(0,1,1,1,1,1)	0
(1,1,1,0,0,0)	0
(1,1,1,1,0,0)	0.5773503
(0.5,0.3,0.7,0.5,0.7,1)	0.5881872

VI. CONCLUSION

We have proposed a composed function which we refer to as the Bonferroni partition mean. This mean further generalizes the composed aggregation operator from our previous research, and is able to more intuitively handle multiple mandatory requirements or mandatory input sets.

ACKNOWLEDGEMENTS

The work of R. Mesiar on this contribution was supported by grants VEGA 1/0143/11 and GAČR P 402/11/0378. The work of G. Beliakov and S. James was supported by the Deakin University Central Research Grants Scheme, project number RM25299.

REFERENCES

[1] C. Bonferroni, "Sulle medie multiple di potenze," *Bollettino Matematica Italiana*, vol. 5, pp. 267–270, 1950.
[2] R. R. Yager, "On generalized Bonferroni mean operators for multi-criteria aggregation," *International Journal of Approximate Reasoning*, vol. 50, pp. 1279–1286, 2009.
[3] G. Beliakov, S. James, J. Mordelová, T. Rückschlossová, and R. R. Yager, "Generalized Bonferroni mean operators in multi-criteria aggregation," *Fuzzy Sets and Systems*, vol. 161, pp. 2227–2242, 2010.
[4] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Heidelberg: Springer, 2007.
[5] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. Cambridge: Cambridge University press, 2009.
[6] Y. Torra, V. Narukawa, *Modeling Decisions. Information Fusion and Aggregation Operators*. Berlin, Heidelberg: Springer, 2007.
[7] J. Mordelová and T. Rückschlossová, "ABC-aggregation functions," in *Proc. of the 5th Intl. Summer School on Aggregation Operators*, Palma de Mallorca, Spain, 2009.
[8] R. R. Yager, G. Beliakov, and S. James, "On generalized Bonferroni means," in *Proc. Eurofuse 2009 Conference*, Pamplona, Spain, 2009.
[9] G. Beliakov and S. James, "Stability of weighted penalty-based aggregation functions," *Fuzzy Sets and Systems*, vol. in press, doi: 10.1016/j.fss.2013.01.007, 2013.