Theoretical Computer Science 239 (2000) 309–338

Theoretical Computer Science

www.elsevier.com/locate/tcs

A formula for incorporating weights into scoring rules ☆

Ronald Fagin *,[1], Edward L. Wimmers

IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA

Abstract

A "scoring rule" is an assignment of a value to every tuple (of varying sizes). This paper is concerned with the issue of how to modify a scoring rule to apply to the case where weights are assigned to the importance of each argument. We give an explicit formula for incorporating weights that can be applied no matter what the underlying scoring rule is. The formula is surprisingly simple, in that it involves far fewer terms than one might have guessed. It has three further desirable properties. The first desirable property is that when all of the weights are equal, then the result is obtained by simply using the underlying scoring rule. Intuitively, this says that when all of the weights are equal, then this is the same as considering the unweighted case. The second desirable property is that if a particular argument has zero weight, then that argument can be dropped without affecting the value of the result. The third desirable property is that the value of the result is a continuous function of the weights. We show that if these three desirable properties hold, then under one additional assumption (a type of local linearity), our formula gives the unique possible answer.

1. Introduction

A *scoring rule* is an assignment of a value to every tuple (of varying sizes). If the entries of each tuple are numbers, then one example of a scoring rule is to take the average of the entries. Information retrieval provides us with examples of scoring rules where the entries of the tuples are not necessarily numbers. In information retrieval, the entries in each tuple might be search terms, and the scoring rule might assign a relevance score telling how well these search terms match a given document.

A common example of a scoring rule arises in a situation where an object is somehow assigned several scores. These different scores may be ratings on different attributes, or

☆ An extended abstract of this paper appears in Proc. 1997 International Conference on Database Theory, pp. 247–261, under the title "Incorporating User Preferences in Multimedia Queries".

* Corresponding author.

E-mail addresses: fagin@almaden.ibm.com (R. Fagin), wimmers@almaden.ibm.com (E.L. Wimmers)

[1] Part of this research was done while the author was a Research Fellow at the IBM Haifa Research Laboratory.

0304-3975/00/$ - see front matter
PII: S0304-3975(99)00224-8

they may be ratings by different scorers on the same attribute. There is often a scoring rule for combining these scores into an overall score. An example where there are ratings by different scorers on the same attribute arises in competitive diving, where there are multiple judges who each assign a single score to each dive. [2] An overall score is then assigned to the dive, by eliminating the top and bottom scores, summing the remaining scores, and multiplying by the "degree of difficulty" of the dive. [3]

An important case where there are multiple attributes and a single score per attribute takes place in fuzzy logic. In particular, a score must be assigned to a conjunction $A_1 \wedge A_2$ that is a function of the scores of A_1 and A_2. In his original paper [35], Zadeh defined the score of $A_1 \wedge A_2$ to be the min of the scores of A_1 and A_2. Similarly, he defined the score of the disjunction $A_1 \vee A_2$ to be the max of the scores of A_1 and A_2. Zadeh's choices were later justified by a famous result of Bellman and Giertz [5], which was extended and simplified by Yager [32], Voxman and Goetschel [30], Dubois and Prade [12], and Wimmers [31]. They showed that min and max are the unique choices that should be assigned to the conjunction and disjunction, respectively, that fulfill certain natural conditions. There is a large literature on other possible choices for scoring rules in fuzzy logic: see, for example, the discussion in Zimmermann's textbook [37].

Another important case where there are multiple attributes and a single score per attribute arises in queries in multimedia database systems, which we now discuss. A database system faces the task of responding to queries. In a traditional database system, all queries deal with Boolean values, since a property is either true or false. As queries over multimedia data become more prevalent, it is important to permit various shades of gray. For example, in searching for a red picture, the user is unlikely to want a Boolean value that says whether the picture is red or not. More likely, the user would prefer a score giving the redness of a particular picture.

In general, a user might want to query not only over a single multimedia property, but might wish to take into account several properties. For example, the user might be interested in a movie clip that has a predominantly red scene with a loud noise in the sound track. In this case, there is likely to be a score giving the redness of the scene and a different score giving the loudness of the sound. These two scores must be combined into a single score. Such an approach is taken by the Garlic system, which is being developed at the IBM Almaden Research Center, and which provides access to a variety of data sources, including multimedia. See [6, 10] for a discussion of the Garlic system, and [14] and [7] (along with Section 11 of this paper) for a discussion of algorithms with a low middleware cost for computing distributed scores (where different "black boxes" produce the various scores that must be combined).

[2] We shall consider various Olympic sports, such as diving, as examples in this paper.

[3] The rules of scoring in (artistic) gymnastics are similar, in that the top and bottom scores are eliminated, and the remaining scores are averaged. Interestingly, there are differences in the weighted case between considering the sum of scores (as in diving) and the average of scores (as in gymnastics): see Section 6.

However, there is an additional problem that must be addressed. It is unlikely that the user equally values the attributes being queried. For example, the user might like to inform the multimedia system to give extra weight to the picture and less weight to the sound. In the user interface, *sliders* are often used to convey this information to the system. Sliders are bars on the screen that indicate the importance of each attribute. The user moves his mouse to slide an indicator along the bar in order to increase or decrease the weighting of a given attribute.

The contribution of this paper is to give an explicit formula for incorporating weights that can be applied no matter what the underlying scoring rule is. The formula we give is surprisingly simple, in that it involves far fewer terms than we might have guessed. It has three further desirable properties. The first desirable property is that when all of the weights are equal, then the result obtained is simply the underlying scoring rule. Intuitively, this says that when all of the weights are equal, then this is the same as considering the unweighted case. In the case of a multimedia database system where the weights are determined by sliders, another way to describe this desirable property is to say that if the user does not see any sliders, then this is the same as if the sliders exist and are all set to a default value where the weights are equal. The second desirable property is that if a particular argument has zero weight, then that argument can be dropped without affecting the value of the result. The third desirable property is that the value of the result is a continuous function of the weights. It turns out that if these desirable properties hold, then under one additional assumption (a type of local linearity), our formula gives the unique possible answer. We note that our formula was developed in the context of the Garlic system.

In Section 2 we discuss min with two arguments as an example. In Section 3 we give some basic definitions. In Section 4 we discuss desiderata in the weighted case. Section 5 contains our main result, which is an explicit formula for the scores in the weighted case. In Section 6, we consider a number of examples of incorporating weights into scoring rules. In particular, we discuss the effects of

- weighting the importance of the conjuncts in fuzzy logic (that is, considering "weighted min");
- weighting search terms in information retrieval (this example is from [15]);
- weighting the cost of pages in page replacement algorithms, such as considering "weighted LRU" (this example is joint with Alain Azagury);
- weighting the importance of judges in competitions such as diving; and
- weighting the importance of criteria in multicriterion decision-making (where it turns out to be convenient to take the range of a scoring rule to be a vector space).

In Section 7, we take a geometric viewpoint of our formula. This viewpoint explains why our formula has so few terms, and shows the uniqueness of our formula under certain assumptions. In Section 8 we show that a certain strong form of linearity cannot hold except in very special cases. In Section 9 we show that our system of incorporating weights preserves many of the properties (such as monotonicity) of the scoring rules. In Section 10 we discuss some other methods of incorporating weights that have been considered in the literature, including the Choquet integral. In Section 11 we reconsider the example of queries in a multimedia database system, and discuss how the results from [14] on minimizing the middleware cost carry over in the weighted case. In Section 12 we give our conclusions.

2. Min with two arguments

Assume that there are two scores, namely x_1 and x_2. In the case of queries in a multimedia database system, these scores are numbers (typically between 0 and 1, where 1 represents a perfect match) that represent how well an object rates on a particular attribute. For example, x_1 might be a score indicating how red a picture is, and x_2 might be a score indicating how loud a sound is. How should these scores be combined to reflect an "overall score" that reflects both the redness and the loudness? Should we take the average of the scores? Or what should we do? Not surprisingly, there are many possible answers, depending on the issues at hand. Garlic allows an arbitrary scoring rule to be "plugged in".

Our paper deals with the following issue. Assume that some scoring rule, such as the average or the min, is given for combining scores. How do we modify this scoring rule if we decide now that we do not want to assign equal weight to the scores? In the multimedia database example, assume that the user cares twice as much about the color of the picture as he does about the loudness of the sound. How should we combine the color score and the loudness score to obtain an overall score? If the scoring rule is simply to take the average, then the answer is fairly clear. We would assign a weight $\theta_1 = \frac{2}{3}$ to the color, and a weight $\theta_2 = \frac{1}{3}$ to the loudness. (The weights must sum to one, and the weight for color, namely θ_1, should be twice the weight θ_2 for loudness.) We then take the weighted sum $\theta_1 x_1 + \theta_2 x_2$. But what if we are using a different underlying scoring rule than the average for combining scores?

For the rest of this section, we assume that as in standard fuzzy logic, the scoring rule is to take the min. Assume again that we wish to weight the scores, where θ_1 is the weight for color, and θ_2 is the weight for loudness. Then we cannot simply take the result to be $\theta_1 x_1 + \theta_2 x_2$. For example, if we are indifferent to color versus loudness, so that we weight them equally with $\theta_1 = \theta_2 = \frac{1}{2}$, then we would get the wrong answer by using $\theta_1 x_1 + \theta_2 x_2$, since this does not give us the min of x_1 and x_2. (We are assuming here that we use the underlying, or "unweighted", scoring rule for combining scores when the θ_i's are equal. Later, we shall make such assumptions explicit.) What should the answer be, as a function of x_1, x_2, and θ_1? (Here we do not need to include θ_2 as a parameter, since $\theta_2 = 1 - \theta_1$.)

Assume without loss of generality that $x_1 \leqslant x_2$. If $\theta_1 = \frac{1}{2}$, then the answer should be x_1, since as we noted, when the weights are equal, we should use the unweighted rule for combining, which in this case is the min. If $\theta_1 = 0$, then the answer should be x_2. This is under the assumption that when an argument has 0 weight, then it can be "dropped"; this is another assumption that will be made explicit later. Similarly, if $\theta_1 = 1$, so that $\theta_2 = 0$, then the answer should be x_1.

What about values of θ_1 other than 0, 1, or $\frac{1}{2}$? Since the value is x_1 when $\theta_1 = \frac{1}{2}$, it is reasonable to argue that the value should be x_1 whenever $\theta_1 \geqslant \frac{1}{2}$; after all, if the value of x_2 becomes irrelevant (as long as it is bigger than x_1) for $\theta_1 = \frac{1}{2}$, then surely it should be irrelevant for any larger value of θ_1, where we are weighting the first value (the x_1 value) even more highly. Another argument that the value should be x_1 whenever $\theta_1 \geqslant \frac{1}{2}$ is that the value is x_1 for both $\theta_1 = \frac{1}{2}$ and $\theta_1 = 1$, and so it should be the same for intermediate values. Later, we shall give a local linearity argument that says that the value should be x_1 whenever $\theta_1 \geqslant \frac{1}{2}$. Furthermore, this local linearity argument says that the value when $\theta_1 < \frac{1}{2}$ should be the linearly interpolated value between the value x_2 when $\theta_1 = 0$, and the value x_1 when $\theta_1 = \frac{1}{2}$: this value is $2(x_1 - x_2)\theta_1 + x_2$.

What would we do when there are three arguments x_1, x_2, and x_3, and three weights θ_1, θ_2, and θ_3? Here the answer is not at all clear *a priori*. Our results enable us to answer this question. Our methods in this paper work for arbitrary scoring rules, not just average and min.

3. Definitions

We assume that we are given a finite index set $\mathscr{I}$. In the case where scores are assigned to different attributes, such as in the multimedia database example, we think of $\mathscr{I}$ as the set of all attributes. In the multimedia database example, these attributes would include color and loudness. In the case where scores are assigned by different scorers (or judges) on the same attribute, such as in scoring in competitive diving, we think of $\mathscr{I}$ as the set of scorers. We typically use I to denote some non-empty subset of $\mathscr{I}$. Let D be the set of possible entries of the tuples in the domain of a scoring rule. It is common in fuzzy logic to take D to be the closed interval $[0,1]$. We shall take a *tuple* X (*over* I) to be a function with domain I and range D. We shall usually write x_i for $X(i)$. If $I = \{1,\ldots,m\}$, then we may write $(x_1,\ldots,x_m)$ for X. If $I' \subseteq I$, then by $X \upharpoonright I'$, we mean the tuple that is the restriction of X to the domain I'.

A *scoring rule* is a function whose domain is the set of all tuples over nonempty subsets of $\mathscr{I}$. Henceforth we shall usually refer to a scoring rule as simply a *rule*, or an *unweighted rule* (to contrast it with a "weighted rule", which we shall define shortly).

Let S be a set, that we shall take to be the range of the rule. It is common in fuzzy logic to take S (like D) to be the closed interval $[0,1]$. The only requirement we impose on S is that it be a convex set, so that if $\alpha_1,\ldots,\alpha_m$ are nonnegative real numbers[4] that sum to 1, and $s_1,\ldots,s_m$ are members of S, then $\sum_{i=1}^{m} \alpha_i s_i$ is also a member of S. Later, we shall consider certain situations where it is convenient to take S to be a vector space. When $D = S$, as is common in fuzzy logic, the rule combines a collection of scores to obtain an overall score. In this case, it would certainly be natural to assume that $f(x_i) = x_i$. There are other situations where we do not have $f(x_i) = x_i$. For example, in the case of information retrieval, where the entries of the tuples are

[4] In some situations, we might want to restrict our attention to rational weights; in this case, we would assume also that the α_i's are rational.

search terms and the range is a set of numerical scores, such an assumption would not make sense. In any case, we do not assume that $f(x_i)=x_i$, even when $D=S$.

A *weighting* (*over* I) is a function Θ with domain a nonempty index set $I \subseteq \mathscr{I}$ and range the closed interval $[0,1]$, whose values $\Theta(i)$ sum to 1. Addition and scalar multiplication are defined in the usual way: $(\alpha \cdot \Theta)(i)=\alpha \cdot \Theta(i)$ for real numbers α, and $(\Theta+\Theta')(i)=\Theta(i)+\Theta'(i)$. We shall write θ_i for $\Theta(i)$. If $I=\{1,\dots,m\}$, then we may write $(\theta_1,\dots,\theta_m)$ for Θ.

A *weighted rule* is a function whose domain is the set of all pairs (Θ,X), where Θ is a weighting and X a tuple over the same nonempty subset of $\mathscr{I}$. For simplicity in notation, we shall usually write $f_\Theta(X)$ for the result of evaluating the weighted rule with argument (Θ,X). It is convenient then to consider f_Θ as a function whose domain is the set of all tuples X over I, where I is that subset of $\mathscr{I}$ such that Θ is over I.

If Θ is over I, we define the *support* of Θ to be the subset of I consisting of all $i \in I$ such that $\theta_i > 0$.

4. Desiderata

Assume that we are given an unweighted rule. Thus, we are given a rule for assigning values to tuples. We wish to define a weighted rule from the unweighted rule. Thus, intuitively, we want to determine how to modify the rule when we weight the importance of the arguments. In this section, we consider some desirable properties for the relationship between the weighted and unweighted rule. Later (Theorem 5.1), we shall show that under the additional natural assumption of a type of local linearity (that we also define in this section), there is a unique choice of the weighted rule that satisfies these properties.

Our first desirable property says intuitively that when all of the weights are equal, then the weighted rule gives the same answer as the unweighted rule. This corresponds to the intuition that the unweighted rule tells how to assign values to tuples in the case where no argument has higher weight than any other argument. Formally, denote the evenly balanced weighting over I by E_I; thus, $(E_I)_i = 1/card(I)$ for each $i \in I$, where $card(I)$ denotes the cardinality of I. We say that the weighted rule is *based on* the unweighted rule f if whenever I is a nonempty subset of $\mathscr{I}$, and X is a tuple over I, then $f_{E_I}(X)=f(X)$. Thus, our first desirable property is:

- The weighted rule is based on the unweighted rule. This says that $f_{(1/m,\dots,1/m)}(x_1,\dots,x_m)=f(x_1,\dots,x_m)$.

Our second desirable property says intuitively that if a particular argument has zero weight, then that argument can be dropped without affecting the value of the result. Formally, a weighted rule is *compatible* if whenever Θ and X are over the same index set I, and J is the support of Θ, then $f_\Theta(X)=f_{\Theta\restriction J}(X\restriction J)$. Thus, our second desirable property is

- The weighted rule is compatible. This says that $f_{(\theta_1,\dots,\theta_{m-1},0)}(x_1,\dots,x_m)=f_{(\theta_1,\dots,\theta_{m-1})}(x_1,\dots,x_{m-1})$.

Our third desirable property is that the value of the result is a continuous function of the weights. That is, if Θ and X are over the same index set, and if X is held fixed, then $f_\Theta(X)$ is a continuous function of Θ. Thus, our third desirable property is:

- $f_{(\theta_1,\ldots,\theta_m)}(x_1,\ldots,x_m)$ is a continuous function of $\theta_1,\ldots,\theta_m$.

These three desirable properties are really essential — any method of going from an unweighted to a weighted rule that does not satisfy these three properties is seriously flawed. The weighted case must bear some relation to the unweighted case, and our notion of "based on" is the only natural choice. The notion of "compatibility" is also the only natural choice for handling zero weights. And surely we would expect continuity, since a small change in the weights should lead to at most a small change in the value of the result.

These three properties are not sufficient to determine a unique weighted rule from an unweighted rule; another property is needed. Perhaps the most natural additional property would be *linearity*: this would say that

$$f_{\alpha\cdot\Theta+(1-\alpha)\cdot\Theta'}(X)=\alpha\cdot f_\Theta(X)+(1-\alpha)\cdot f_{\Theta'}(X), \tag{1}$$

whenever $\alpha\in[0,1]$ and Θ,Θ',X are over the same index set. Unfortunately, as we shall discuss in Section 8, linearity is incompatible with the other desirable properties, except in the case where the unweighted rule is essentially simply the average.

Since we are not able to have (total) linearity, we shall settle for a weaker variation, which we call "local linearity". Local linearity turns out to imply continuity. Together with the other desirable properties, local linearity does uniquely determine a weighted rule. Furthermore, this new property leads to a simple formula for the weighted rule. To define this property, we need some more definitions.

Two weightings are called *comonotonic* if they agree on the order of importance of the arguments. [5] Formally, assume that Θ,Θ' are weightings over I. Then Θ,Θ' are *comonotonic* if there do *not* exist $i,j\in I$ with $\theta_i<\theta_j$ and $\theta'_j<\theta'_i$ both holding. For example, (.2,.7,.1) and (.3,.5,.2) are comonotonic because in both cases, the second entry is biggest, the first entry is next-biggest, and the third entry is smallest. It is clear that comonotonicity is reflexive and symmetric. Comonotonicity is *not* transitive, since for example (0,1) and (1,0) are not comonotonic, while (0.5,0.5) is comonotonic to both (0,1) and (1,0).

We now define local linearity, and argue that it is fairly natural. Intuitively, local linearity says that the scoring rule acts like a balance. If two weightings are comonotonic, then local linearity demands that the weighting that is the midpoint of two comonotonic weightings should produce a value that is the midpoint of the two values produced by the given weightings. In fact, local linearity extends beyond the midpoint to any weighting that is a convex combination of two comonotonic weightings: if a

[5] In earlier versions of this paper, we referred to comonotonicity as *order-equivalence*. We changed the name to coincide with Schmeidler's [27] term he uses in the context of Choquet integrals (see Section 10.2).

weighting is a convex combination of two weightings that are comonotonic, then local linearity demands that the associated value should be the same convex combination of the values associated with the given weightings. Formally, we say that a weighted rule is *locally linear* if Eq. (1) holds whenever (a) $\alpha \in [0,1]$, (b) Θ, Θ', X are over the same index set I, and (c) Θ and Θ' are comonotonic.

The next proposition says that local linearity implies continuity, as a function of the weights. (To define continuity, we need to define the distance between two weightings Θ and Θ'. We simply take the distance to be the Euclidean distance.)

Proposition 4.1. *Assume that the weighted rule is locally linear. Then $f_\Theta(X)$ is a continuous function of Θ, for each fixed X.*

Proof. The proof is given in Section 7. □

Since we allow the possibility that the unweighted rule not be a continuous function of X, this can certainly happen also in the weighted case. As we shall see in Section 9, under our method of obtaining the weighted rule from the unweighted rule, the weighted rule is continuous as a function of X if this is true of the unweighted rule.

Our main theorem (Theorem 5.1) gives an explicit, simple formula for obtaining a weighted rule from an unweighted rule. The weighted rule is based on the unweighted rule, compatible, and locally linear. Furthermore, the theorem says that our formula gives the unique such weighted rule.

A weighted rule is *totally linear* if Eq. (1) holds even when Θ and Θ' are not necessarily comonotonic. As we mentioned above, we shall see from Theorem 8.1 that extending linearity to hold for all weightings and not merely the comonotonic ones severely restricts the possible choices for the unweighted case. That is, total linearity of the weighted rule can be obtained only for certain very restricted choices of the unweighted rule, which essentially correspond to taking the average. Although we argued above that local linearity is a reasonable assumption, we might argue that total linearity is perhaps too strong. When the order of importance of two components changes, a dramatic shift might occur, and there is no reason to assume that the value associated with the midpoint has any relation to the value associated with the endpoint weightings. It might even occur that the midpoint of two weightings is not comonotonic to either weighting. For example, $(0.3, 0.4, 0.3)$ is the midpoint of $(0.1, 0.4, 0.5)$ and $(0.5, 0.4, 0.1)$ but is not comonotonic to either one.

It is helpful to have a notation for selecting the most important (i.e. the largest) component of a weighting, down to the least important (i.e. the smallest) component of a weighting. A bijection σ that provides such a service is said to "order" the weighting. If σ orders a given weighting, then $\sigma(1)$ represents the most important component and $\sigma(m)$ represents the least important component (where m is the number of components). This is formalized in the next definition.

Assume that $m = card(I)$. A bijection σ from $\{1,\dots,m\}$ onto I is said to *order* a weighting Θ over I if $\theta_{\sigma(1)} \geqslant \theta_{\sigma(2)} \geqslant \cdots \geqslant \theta_{\sigma(m)}$. It is easy to see that every weighting is ordered by some bijection σ.

5. Main theorem

Our original goal in doing the research for this paper was to obtain a procedure that, given an unweighted rule, would give a corresponding weighted rule, where three essential properties hold:

1. The weighted rule is based on the unweighted rule.
2. The weighted rule is compatible.
3. The value $f_\Theta(X)$ of the result is a continuous function of Θ.

As we shall discuss, we obtain a procedure where not only do (1) and (2) above hold, but where (3) is replaced by the stronger (3′) below ((3′) is stronger than (3) by Proposition 4.1):

3.′ The weighted rule is locally linear.

Not only do we obtain a procedure, but we in fact obtain a closed-form formula for the weighted rule, that we give in the following theorem. As we shall explain, this formula is surprisingly simple, in that it involves far fewer terms than we might have guessed. In fact, the simplicity of the formula is an argument in favor of local linearity. Furthermore, the theorem says that this weighted rule is the unique one that satisfies (1), (2), and (3′).

Theorem 5.1. *For every unweighted rule f there exists a unique weighted rule that is based on f, compatible, and locally linear. If $I = \{1,\dots,m\}$, and X and Θ are over I with $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m$, then*

$$f_\Theta(X) = (\theta_1 - \theta_2)\cdot f(x_1) + 2\cdot(\theta_2-\theta_3)\cdot f(x_1,x_2) + 3\cdot(\theta_3-\theta_4)\cdot f(x_1,x_2,x_3) + \cdots + m\cdot\theta_m\cdot f(x_1,\dots,x_m). \tag{2}$$

We note for later use that if we no longer assume that $I = \{1,\dots,m\}$ and that $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m$, but simply that $m = card(I)$ and that Θ is ordered by σ, then (2) becomes

$$f_\Theta(X) = \left(\sum_{i=1}^{m-1} i\cdot(\theta_{\sigma(i)} - \theta_{\sigma(i+1)})\cdot f(X\restriction\{\sigma(1),\dots,\sigma(i)\})\right) + m\cdot\theta_{\sigma(m)}\cdot f(X). \tag{3}$$

We shall often refer to the formula in (2) as "the weighting formula". If we were to adopt the natural convention that $\theta_{m+1} = 0$, then the last term $m\cdot\theta_m\cdot f(x_1,\dots,x_m)$ in the weighting formula could be written as $m\cdot(\theta_m - \theta_{m+1})\cdot f(x_1,\dots,x_m)$, which makes it have the same form as the other terms.

The proof of Theorem 5.1 is given at the end of this section.

The simplicity of the weighting formula is rather surprising, since it is not clear *a priori* that $f_\Theta(X)$ should depend only on $f(x_1), f(x_1,x_2),\dots,f(x_1,\dots,x_m)$, and not on other values of f. For example, when $m=3$, f is min, and $\theta_1 \geqslant \theta_2 \geqslant \theta_3$, then the formula for $f_\Theta(X)$ is a convex combination of the three terms $x_1, \min(x_1,x_2)$, and $\min(x_1,x_2,x_3)$ only, and not of any of the terms $x_2, x_3, \min(x_1,x_3)$, or $\min(x_2,x_3)$. In general, $f_\Theta(X)$ depends on the values of $f(X\restriction Z)$ for only m of the 2^m-1 possible choices of Z. Thus, the formula is not only simple, but it is surprisingly simple. In Section 7, we show how a geometric view of our formula explains why there are only m terms.

As we shall show when we prove Theorem 5.1, the weighting formula is well defined, even when some of the θ_i's are equal. For example, if $\theta_2 = \theta_3$, then should the second term of the weighting formula involve $f(x_1, x_2)$ or $f(x_1, x_3)$? The point is that it does not matter, since in either case the result is multiplied by $(\theta_2 - \theta_3)$, which is 0.

Note that if $f(X)$ is rational for each X, and if each θ_i is rational, then $f_\Theta(X)$ is also rational (cf. footnote 4 of Section 3).

The following corollary is useful.

Corollary 5.2. *Let α_i be the coefficient of $f(x_1, \ldots, x_i)$ in the weighting formula, for $1 \leqslant i \leqslant m$, so that $f_\Theta(X) = \sum_{i=1}^m \alpha_i \cdot f(x_1, \ldots, x_i)$. Then $\alpha_i \geqslant 0$ for each i, and $\sum_{i=1}^m \alpha_i = 1$.*

Proof. The fact that $\alpha_i \geqslant 0$ for each i follows immediately from the assumption in Theorem 5.1 that $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m \geqslant 0$. We now show that $\sum_{i=1}^m \alpha_i = 1$. We have $\sum_{i=1}^m \alpha_i = \sum_{i=1}^{m-1} i \cdot (\theta_i - \theta_{i+1}) + m \cdot \theta_m = \sum_{i=1}^m i \cdot \theta_i - \sum_{i=1}^{m-1} i \cdot \theta_{i+1} = \sum_{i=1}^m i \cdot \theta_i - \sum_{i=2}^m (i-1) \cdot \theta_i = \theta_1 + \sum_{i=2}^m \theta_i = \sum_{i=1}^m \theta_i = 1$. □

Note from Corollary 5.2 that $f_\Theta(X)$ is a convex combination of members of the range S (since $f(X)$ is in S for each X). This is why we took the range S to be convex: the weighted rule has the same range S as the unweighted rule.

We close this section with a proof of Theorem 5.1.

Proof of Theorem 5.1. We begin by showing that the weighting formula is well defined, even when some of the θ_i's are equal. Since we shall need to use different permutations of the index set I, we shall need to make use of (3). Thus, assume that Θ is ordered by both σ and σ'; we must show that

$$\left(\sum_{i=1}^{m-1} i \cdot (\theta_{\sigma(i)} - \theta_{\sigma(i+1)}) \cdot f(X \upharpoonright \{\sigma(1), \ldots, \sigma(i)\}) \right) + m \cdot \theta_{\sigma(m)} \cdot f(X) \tag{4}$$

equals

$$\left(\sum_{i=1}^{m-1} i \cdot (\theta_{\sigma'(i)} - \theta_{\sigma'(i+1)}) \cdot f(X \upharpoonright \{\sigma'(1), \ldots, \sigma'(i)\}) \right) + m \cdot \theta_{\sigma'(m)} \cdot f(X). \tag{5}$$

Let J be the subset of $\{1, \ldots, m-1\}$ consisting of all i such that $\theta_{\sigma(i)}$ is strictly greater than $\theta_{\sigma(i+1)}$ (possibly J is empty). Define J^+ to be $J \cup \{m\}$. Write the members of J^+ as $i_1, \ldots, i_s$, where $i_1 < \cdots < i_s = m$. Thus, there are exactly s distinct values of the θ_i's, namely $\theta_{\sigma(i_1)}, \ldots, \theta_{\sigma(i_s)}$, with $\theta_{\sigma(i_1)} > \cdots > \theta_{\sigma(i_s)}$. Among the θ_i's, there are exactly i_1 with the highest value, exactly $i_2 - i_1$ with the next highest value, and so on. Since the previous sentence is independent of the permutation (σ or σ'), it follows that the values $i_1,\ldots,i_s$ are independent of the permutation, and so J^+ is the same whether it is defined using σ (as we did) or using σ'. Hence, $J = J^+ \cap \{1,\ldots,m-1\}$ is also independent of the permutation. Therefore, J is equal to the subset of $\{1,\ldots,m-1\}$ consisting of all i such that $\theta_{\sigma'(i)}$ is strictly greater than $\theta_{\sigma'(i+1)}$.

Now $\theta_{\sigma(1)},\theta_{\sigma(2)},\ldots,\theta_{\sigma(i_1)}$ are precisely those θ_i's with the highest value; $\theta_{\sigma(i_1+1)},\ldots,\theta_{\sigma(i_2)}$ are precisely those θ_i's with the second highest value; and so on. Identically, we have that $\theta_{\sigma'(1)},\theta_{\sigma'(2)},\ldots,\theta_{\sigma'(i_1)}$ are precisely those θ_i's with the highest value; $\theta_{\sigma'(i_1+1)},\ldots,\theta_{\sigma'(i_2)}$ are precisely those θ_i's with the second highest value; and so on. So $\{\sigma(1),\ldots,\sigma(i_j)\}$ and $\{\sigma'(1),\ldots,\sigma'(i_j)\}$ each contain exactly those σ_i's that have one of the j highest distinct values. Therefore,

$$\{\sigma(1),\ldots,\sigma(i)\} = \{\sigma'(1),\ldots,\sigma'(i)\} \text{ for every } i \text{ in } J. \tag{6}$$

We now show that

$$\theta_{\sigma(i)} = \theta_{\sigma'(i)} \text{ for every } i. \tag{7}$$

If $1 \leqslant i \leqslant i_1$, then $\theta_{\sigma(i)}$ and $\theta_{\sigma'(i)}$ are both equal to the highest distinct value of the θ_i's, and hence are equal. If $i_1+1 \leqslant i \leqslant i_2$, then $\theta_{\sigma(i)}$ and $\theta_{\sigma'(i)}$ are both equal to the second highest distinct value of the θ_i's, and hence are equal. Continuing this argument, we see that $\theta_{\sigma(i)} = \theta_{\sigma'(i)}$ for every i, as desired.

Since $\theta_{\sigma(i)} = \theta_{\sigma(i+1)}$ for all i with $1 \leqslant i \leqslant m-1$ and $i \notin J$, it follows that we can change the indices of summation in (4) to $i \in J$; that is, (4) equals

$$\left(\sum_{i\in J} i\cdot(\theta_{\sigma(i)}-\theta_{\sigma(i+1)})\cdot f(X\restriction\{\sigma(1),\ldots,\sigma(i)\})\right) + m\cdot\theta_{\sigma(m)}\cdot f(X). \tag{8}$$

Identically, (5) equals

$$\left(\sum_{i\in J} i\cdot(\theta_{\sigma'(i)}-\theta_{\sigma'(i+1)})\cdot f(X\restriction\{\sigma'(1),\ldots,\sigma'(i)\})\right) + m\cdot\theta_{\sigma'(m)}\cdot f(X). \tag{9}$$

But it follows from (6) and (7) that (8) and (9) are equal. This concludes the proof that f_Θ is well defined.

If $\Theta = E_I$, then $\theta_1 = \cdots = \theta_m = 1/m$. Hence, $f_\Theta(X) = \sum_{i=1}^{m-1} i\cdot(1/m - 1/m)\cdot f(x_1,\ldots,x_i) + m\cdot(1/m)\cdot f(x_1,\ldots,x_m) = f(x_1,\ldots,x_m)$. Thus, the weighted rule is based on f.

The fact that the weighted rule is locally linear follows easily from the fact that the weighting formula is explicitly a linear function of the θ_i's (within each simplex consisting of a set of Θ's that are comonotonic).

We now show that the weighted rule is compatible. Assume as before that $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m$. Let k be maximal such that $\theta_k > 0$. Thus, $\{1,\ldots,k\}$ is the support of Θ. We must show that $f_\Theta(X)$ as given by (2) is equal to the result of replacing m by k in the formula given in (2). But it is simple to verify that both are equal to the sum of the first k summands in the weighting formula (2).

Finally, uniqueness is shown in Section 7. □

6. Examples

To illustrate the simplicity and usefulness of the weighting formula, it is helpful to see some examples. In particular, in this section we illustrate its wide range of applicability by including examples from various domains.

Example 1. There is one unweighted scoring rule where we know what the corresponding weighted scoring rule "should be". Namely, if the unweighted scoring rule is to take the average $(x_1+\cdots+x_m)/m$, then we expect the weighted scoring rule to be the weighted average $\theta_1 x_1+\cdots+\theta_m x_m$. This is fairly straightforward to verify; let us do so for the case $m=3$. In this case, the weighting formula is

$$(\theta_1-\theta_2)x_1+2(\theta_2-\theta_3)\frac{x_1+x_2}{2}+3\theta_3\frac{x_1+x_2+x_3}{3}=\theta_1 x_1+\theta_2 x_2+\theta_3 x_3,$$

as desired.

Example 2. We now show that the weighting formula gives the "right" values, as discussed in Section 2, in the case of min with two arguments. This follows from the fact that the weighted rule described in Section 2 satisfies the conditions of Theorem 5.1, and so coincides with the weighting formula by uniqueness. Again, it is instructive to give a direct derivation of this equality. Assume as in Section 2 that $x_1\leqslant x_2$. If $\theta_1\geqslant 1/2$, then the weighting formula is

$$\begin{aligned}(\theta_1-\theta_2)x_1+2\theta_2 x_1 &= (\theta_1+\theta_2)x_1\\ &= x_1,\end{aligned}$$

as in Section 2. If $\theta_1\leqslant 1/2$ (so that $\theta_2\geqslant\theta_1$), then the weighting formula is

$$\begin{aligned}(\theta_2-\theta_1)x_2+2\theta_1 x_1 &= (1-2\theta_1)x_2+2\theta_1 x_1\\ &= 2(x_1-x_2)\theta_1+x_2,\end{aligned}\tag{10}$$

as in Section 2. (The first equality in (10) holds since $\theta_1+\theta_2=1$.)

Example 3. What does the weighting formula give us in the case of min with three arguments? Assume $\theta_1\geqslant\theta_2\geqslant\theta_3$. The formula is

$$(\theta_1-\theta_2)x_1+2(\theta_2-\theta_3)\min(x_1,x_2)+3\theta_3\min(x_1,x_2,x_3).$$

Note how simple this formula is. There is probably no way we would have guessed this formula a priori.

Example 4. We now consider the case of information retrieval, which we mentioned in the introduction. Here the user issues a query against a given repository by presenting a set of search terms to an information retrieval system, and the system assigns a relevance score to each document, based on the search terms. The system then presents the user with a list of documents, ranked by their relevance scores.

To apply our methodology, let us hold fixed some document d, and let X be a tuple whose entries consist of search terms (we are assuming in this example that the order of the search terms does not affect the relevance score). We define $f(X)$ to be the relevance score of document d using the search terms in X. Our methodology then gives us a way to assign a relevance score when a weight is assigned to each search term.

We study this example in detail in [15].

Example 5. Our next example is concerned with obtaining versions of page replacement algorithms, where the pages have possibly different weights. [6] A computer system typically has a storage hierarchy. When a page of data is needed, it is brought into the top level of the storage hierarchy (we shall refer to this top level as the *cache*). Typically, to make room for this page, another page is removed from the cache according to some *page replacement algorithm*. For example, the LRU ("least recently used") page replacement algorithm [21] removes the page that has been least recently used.

There are some situations where certain pages are more expensive to retrieve than others. For example, if a page is retrieved from a remote site, we might consider the time to retrieve the page to be its cost. It would be desirable to use a page replacement algorithm that takes this cost into consideration. Thus, if there is a choice between removing a more expensive page and a less expensive page from the cache, the page replacement algorithm should somehow favor keeping the more expensive page in the cache, and thereby tend to remove the less expensive page.

Assume that each page in the cache has a weight: pages with a higher cost should have a bigger weight. For example, we could assign weights that are proportional to the costs. Our methodology enables us to obtain a weighted version of page replacement algorithms (such as "weighted LRU"). The details, which are somewhat messy, are omitted. [7]

Example 6. It is instructive to reconsider the case of the scoring rule used in judging diving that we mentioned in the introduction, where the overall score is obtained by eliminating the top and bottom scores, summing the remaining scores and multiplying by the degree of difficulty of the dive. Let X be a tuple (of scores). Define $g(X)$ to be the result of summing all of the entries of X except the biggest and smallest. Thus, if the index set I of X has at least two members, then

$$g(X) = \left(\sum_{i \in I} x_i \right) - \max_{i \in I} x_i - \min_{i \in I} x_i. \tag{11}$$

If f is a tuple over the set $\mathscr{I}$ of all judges, then the unweighted rule is as follows:

$$f(X) = d \cdot g(X), \tag{12}$$

where d is the degree of difficulty of the dive.

[6] This example is joint with Alain Azagury.

[7] We note that there are several papers on weighted versions of page replacement algorithms [9, 20, 25, 29, 33, 34].

Let us say that the set $\mathscr{I}$ of judges is of size seven (this is the standard number of judges in international competition). In order to apply the weighting formula when we assign weights to the scores of the seven judges, we need not only the definition of $f(X)$ when X is a tuple over the set $\mathscr{I}$ of all judges, as given by (12), but also the definition of $f(X)$ when X is over a proper subset of $\mathscr{I}$. How should $f(X)$ be defined in this case? There are (at least) two possibilities.

Possibility 1. We strictly mimic (12). In particular, when the index set of X has either one or two members (so that there are either one or two judges), then $f(X)$ is identically zero: this is because when we eliminate the top and bottom scores, then no scores are left.

Possibility 2. What is probably more reasonable is to take

$$f(X) = d \cdot g(X), \tag{13}$$

when the index set I of X has at least three members, and otherwise to take

$$f(X) = d \cdot \sum_{i \in I} x_i. \tag{14}$$

Thus, in this case, if there are only one or two judges, then no scores are eliminated, and so the overall score is simply the sum of the scores times the degree of difficulty of the dive. [8]

In retrospect, it is not surprising that we must consider the situation when there are less than seven judges to determine what to do in the weighted case, since, for example, if we assign weights of zero to, say, three judges, then by compatibility, we are reduced to considering the scoring rule when there are four judges. However, it is somewhat intriguing from an epistemological point of view that even if we know that there are always exactly seven judges, we are forced to consider "possible worlds" where there are fewer judges. Such reasoning, where we are forced to consider worlds that are commonly known to be impossible, is called *counterfactual* [19].

Let us now look closer at the situation where we weight the importance of the various judges' scores: for example, some senior judge's score might be more important than the scores of any of the other judges. Assume for definiteness that $\theta_1 > \theta_2 \geqslant \cdots \geqslant \theta_m$, so that in particular the weight θ_1 assigned to the score of judge 1 is strictly larger than any of the other weights. What does the weighting formula then say in the weighted case? Under possibility 2 above, we see that the score of judge 1 is then never completely ignored, even if his score is the highest or the lowest. Instead, some multiple of judge 1's score is always averaged in, because the multiplicative factor $(\theta_1 - \theta_2)$ is strictly positive. Under possibility 1 above, however, it is straightforward to verify that in the weighted case, just as in the unweighted case, judge 1's score is ignored if it is the highest or the lowest.

[8] Note that, as is traditional in judging diving, we do not normalize by dividing by the number of judges.

Example 7. Let us now consider a simplification of the scoring rule for diving, where we simply take as the overall score the sum of the scores of the judges (thus, no scores are eliminated, and the degree of difficulty is not considered). Similarly, we can consider a simplification of the scoring rule for gymnastics (cf. footnote 3 of Section 1) where we simply take as the overall score the average of the scores of the judges (thus, no scores are eliminated). At first glance, these two simplifications (taking the sum versus taking the average) seem to be essentially identical. After all, the results (as to who finished first, who finished second, and so on) are unchanged when we simply multiply every score by the same positive constant (namely, the number of judges). But interestingly enough, in the weighted case it makes a difference as to whether the score is taken to be the sum or the average, as we now show.

Take the simple case where there are two contestants (A and B) and two judges (1 and 2). Assume that judge i gives A the score a_i, and gives B the score b_i, for $i=1,2$. In the unweighted case, contestant A wins if $a_1+a_2>b_1+b_2$, ties if $a_1+a_2=b_1+b_2$, and loses if $a_1+a_2<b_1+b_2$. This is true whether the score is taken to be the sum or the average. So in the unweighted case, the winner is the same, whether the score is taken to be the sum or the average.

Assume that the weight for judge i is θ_i, for $i=1,2$ (so of course $\theta_1+\theta_2=1$). Now consider the weighted case where $\theta_1>\frac{1}{2}$, so that the first judge has more weight than the second judge. Assume that $a_1>b_1$, so that the first judge gives a higher score to contestant A than to contestant B. Assume also that $a_2<b_2$, so that the second judge gives a higher score to contestant B than to contestant A. When the scoring rule is the sum, then by the weighting formula, the overall score for contestant A is $(\theta_1-\theta_2)a_1+2\theta_2(a_1+a_2)$, which equals $a_1+2\theta_2 a_2$ (since $\theta_1+\theta_2=1$). Similarly, the overall score for contestant B is $b_1+2\theta_2 b_2$. So contestant A wins precisely if

$$a_1+2\theta_2 a_2>b_1+2\theta_2 b_2. \tag{15}$$

On the other hand, when the scoring rule is the average, then contestant A wins precisely if

$$\theta_1 a_1+\theta_2 a_2>\theta_1 b_1+\theta_2 b_2. \tag{16}$$

It is straightforward to assign the parameters so that (15) fails but (16) holds.[9] This means that contestant A loses when the unweighted score is taken to be the sum, but wins when the unweighted score is taken to be the average. Perhaps there is some interesting theory underlying this example.

Example 8. We now consider scoring in figure skating (which, as we shall see, can be viewed as a special case of *voting schemes*, or of *multicriterion decision-making*). In figure skating, it turns out that the numerical values of the scores given by the judges are unimportant: all that matters is the rank order of the contestants for each judge. So in the case of figure skating, the domain D (the entries of the tuples) is the set

[9] For example, take $\theta_1, a_1, a_2, b_1, b_2$, respectively, to be .93, .1, 0, 0, 1.

of all permutations [10] of the contestants, one permutation for each judge. There is a complicated rule that determines the overall order of finish of the contestants. [11] Let us assume that there are n contestants. For the sake of this example, let us take the range S of the scoring rule to be the Euclidean space $\Re^n$, where $\Re$ is the set of real numbers. Let X be a tuple of permutations of the contestants (one permutation for each judge), and let I be the set of judges. If the rules of figure skating say that in this situation the jth contestant finishes in i_jth place, for $1 \leqslant j \leqslant n$, then we take $f(X) = (i_1, \ldots, i_n)$. We can think of the jth entry of the vector $f(X)$ as giving the score of contestant j, where the contestant that finishes in first place gets a score of 1, the contestant that finishes in second place gets a score of 2, and so on. In the weighted case, where we assign weights to the judges, $f_\Theta(X)$ as defined in (2) then gives us a vector of scores for the contestants, where again the jth entry of the vector $f_\Theta(X)$ is the score of contestant j. The contestant with the lowest score wins, the contestant with the second-lowest score is in second place, etc.

In this example, the range is taken to be a vector space, rather than, as in the previous examples, simply a set of real numbers. We note that we did not really need to take the range to be a vector space. Instead, we could have taken a separate unweighted rule f^j for each contestant j, such that f^j is the projection onto the jth component of the rule f that we described above.

Scoring in figure skating fits the paradigm of *multicriterion decision-making*, which has a large literature in economics (see, for example [3]). Here, there are a set of *alternatives*, and a set of relevant *criteria*. For each criterion, there is a permutation of the set of alternatives, which tells the best alternative under that criterion, the second-best, and so on. There is then an *aggregation method* that decides on an overall ranking of the alternatives, based on the criteria. A *voting scheme* [2, 16], such as that for judging figure skating, fits this paradigm, where we think of the contestants as the alternatives, the judges (or voters) as the criteria, and the rule that tells the order of finish of the contestants as the aggregation method. The method we have described in this example for applying our methodology can be easily adapted to this general framework of multicriterion decision-making. We are thereby able to take an arbitrary aggregation method and modify it to allow us to assign a weight to each criterion.

In a traditional voting scheme, the voters are treated homogeneously, in a symmetric manner (at least in the unweighted case). This would not necessarily be the case in multicriterion decision-making, where different criteria might be treated differently, even in the unweighted case. There is already another natural way (other than our method) to find a weighted version of a voting scheme (at least in the case where the weights are rational). We simply allow multiple copies of voters, where the number of multiple copies is proportional to the weight, and treat these multiple copies as independent voters. It is an interesting question, which will require future research, to

[10] For simplicity, we are ignoring ties throughout this example.

[11] See [4] for a discussion of the rules of figure skating.

understand the differences between the results of our approach and of this “multiple copy” approach.

7. Viewing the weighting formula geometrically

In this section, we take a geometric viewpoint. Thereby, we see why the weighting formula is unique, and why it has only a linear number of terms. Furthermore, we use our geometric machinery to prove Proposition 4.1, which says that local linearity implies continuity, as a function of the weights.

For simplicity in notation, let us assume throughout this section that X is fixed. Assume that we are given a weighted rule (a function of (Θ, X)), where again we denote the value at (Θ, X) by $f_\Theta(X)$. As in the proof of Proposition 4.1, define a function h with domain the set of weightings, such that $h(\Theta) = f_\Theta(X)$.

Local linearity of the weighted rule says that if Θ and Θ' are comonotonic and $\alpha \in [0,1]$, then

$$h(\alpha \cdot \Theta + (1-\alpha) \cdot \Theta') = \alpha \cdot h(\Theta) + (1-\alpha) \cdot h(\Theta'). \tag{17}$$

The next simple lemma says that (17) generalizes to convex combinations of an arbitrary number of weightings, not just two.

Lemma 7.1. *Assume that h satisfies* (17) *whenever* Θ *and* Θ' *are comonotonic and* $\alpha \in [0,1]$. *Assume that* $\Theta^1, \ldots, \Theta^m$ *are weightings, each pair of which is comonotonic. Assume that* $\alpha_1, \ldots, \alpha_m$ *are nonnegative, and* $\sum_{i=1}^m \alpha_i = 1$. *Then*

$$h\left(\sum_{i=1}^{m} \alpha_i \cdot \Theta^i\right) = \sum_{i=1}^{m} \alpha_i \cdot h(\Theta^i). \tag{18}$$

Proof. For simplicity, we prove this result only when $m = 3$; the general result can be proved similarly, by induction.

The result is clearly true if $\alpha_1 = 1$, so assume that $\alpha_1 \neq 1$. Then

$$\begin{aligned}
& h(\alpha_1 \cdot \Theta^1 + \alpha_2 \cdot \Theta^2 + \alpha_3 \cdot \Theta^3) \\
&= h(\alpha_1 \cdot \Theta^1 + (1-\alpha_1)(\tfrac{\alpha_2}{1-\alpha_1} \cdot \Theta^2 + \tfrac{\alpha_3}{1-\alpha_1} \cdot \Theta^3)) \\
&= \alpha_1 \cdot h(\Theta^1) + (1-\alpha_1) h(\tfrac{\alpha_2}{1-\alpha_1} \cdot \Theta^2 + \tfrac{\alpha_3}{1-\alpha_1} \cdot \Theta^3) \quad \text{by (17)} \\
&= \alpha_1 \cdot h(\Theta^1) + (1-\alpha_1)(\tfrac{\alpha_2}{1-\alpha_1} \cdot h(\Theta^2) + \tfrac{\alpha_3}{1-\alpha_1} \cdot h(\Theta^3)) \quad \text{by (17)} \\
&= \alpha_1 \cdot h(\Theta^1) + \alpha_2 \cdot h(\Theta^2) + \alpha_3 \cdot h(\Theta^3).
\end{aligned}$$

This was to be shown. □

Let U be the simplex defined by $\theta_1 + \cdots + \theta_m = 1$ and $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m \geqslant 0$. (A similar argument applies when instead of assuming that $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m \geqslant 0$, we assume more generally that $\theta_{\sigma(1)} \geqslant \theta_{\sigma(2)} \geqslant \cdots \geqslant \theta_{\sigma(m)} \geqslant 0$. We are assuming that $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m \geqslant 0$ simply for ease in notation.) We now show that the extreme points [12] of U are precisely the points $e_1, \ldots, e_m$, defined as follows:

$$\begin{aligned} e_1 &= (1, 0, 0, 0, \ldots, 0), \\ e_2 &= \left(\tfrac{1}{2}, \tfrac{1}{2}, 0, 0, \ldots, 0\right), \\ e_3 &= \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}, 0, \ldots, 0\right), \\ &\ldots \\ e_m &= \left(\tfrac{1}{m}, \tfrac{1}{m}, \tfrac{1}{m}, \ldots, \tfrac{1}{m}\right). \end{aligned}$$

First, every point $(\theta_1, \ldots, \theta_m)$ in U is a convex combination of $e_1, \ldots, e_m$, because of the following equality, which is straightforward to verify:

$$\begin{aligned} (\theta_1, \ldots, \theta_m) = (\theta_1 - \theta_2) \cdot e_1 + 2 \cdot (\theta_2 - \theta_3) \cdot e_2 + 3 \cdot (\theta_3 - \theta_4) \cdot e_3 \\ + \cdots + m \cdot \theta_m \cdot e_m. \end{aligned} \tag{19}$$

Thus, no point other than $e_1, \ldots, e_m$ can be an extreme point of U. Now the points $e_1, \ldots, e_m$ are linearly independent, since the matrix whose ith row is e_i is a triangular matrix with only positive elements on the diagonal, and so the determinant is positive. It is not hard to show that since every point in U is a convex combination of $e_1, \ldots, e_m$, and since the points $e_1, \ldots, e_m$ are linearly independent, it follows that the extreme points of U are precisely $e_1, \ldots, e_m$.

Let f be an unweighted rule, and assume we are given a weighted rule that is based on f, compatible, and locally linear. Again, we denote the value at (Θ, X) by $f_\Theta(X)$, and write $f_\Theta(X)$ as $h(\Theta)$. By Eq. (19) and Lemma 7.1, we have

$$\begin{aligned} h(\theta_1, \ldots, \theta_m) = (\theta_1 - \theta_2) \cdot h(e_1) + 2 \cdot (\theta_2 - \theta_3) \cdot h(e_2) + 3 \cdot (\theta_3 - \theta_4) \cdot h(e_3) \\ + \cdots + m \cdot \theta_m \cdot h(e_m). \end{aligned} \tag{20}$$

But what is $h(e_i)$ in this equation? Let $I = \{1, \ldots, i\}$. Note that the restriction of e_i to $\{1, \ldots, i\}$ is the evenly balanced weighting E_I, where $(E_I)_j = 1/i$ for each $j \in I$. Then

$$\begin{aligned} h(e_i) &= f_{e_i}(X) \\ &= f_{E_i}(x_1, \ldots, x_i) \quad \text{by compatibility} \\ &= f(x_1, \ldots, x_i) \quad \text{since the weighted rule is based on } f. \end{aligned}$$

When we substitute this value for $h(e_i)$ into (20), we find that $h(\Theta)$, that is, $f_\Theta(X)$, equals the weighting formula in (2). This proves the uniqueness part of Theorem 5.1, as promised.

This analysis shows a reason why there are only m terms in the weighting formula: it is because the simplex U has only m extreme points. The key to the uniqueness proof is that the formula is determined uniquely by its values at the m extreme points.

[12] An *extreme point* of a simplex is a point in the simplex that is not a convex combination of other points in the simplex.

We now prove Proposition 4.1, which says that local linearity implies continuity, as a function of the weights. Assume that the weighted rule is locally linear. As before, if we denote $f_\Theta(X)$ by $h(\Theta)$, then we see by Eq. (19) and Lemma 7.1 that (20) holds. This shows that h is a linear (and hence continuous) function in the simplex defined by $\theta_1 + \cdots + \theta_m = 1$ and $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_m \geqslant 0$ (and similarly in each simplex defined by $\theta_1 + \cdots + \theta_m = 1$ and $\theta_{\sigma(1)} \geqslant \theta_{\sigma(2)} \geqslant \cdots \geqslant \theta_{\sigma(m)} \geqslant 0$). Since h is continuous in each such simplex, and since there are a finite number of such simplexes, each of which is a closed set, it follows that h is continuous everywhere. Proposition 4.1 follows.

We note that historically, we derived the weighting formula by a different geometric view than that which we just presented. We now discuss this other view, and how it enabled us to determine a formula for f_Θ by induction on the number of nonzero entries of Θ. To start off the induction, note that if Θ has only one nonzero entry, so that $\theta_i = 1$ for some i, then f_Θ is uniquely determined, since then $f_\Theta(X) = f_{E_{\{i\}}}(x_i)$ by compatibility, and in turn $f_{E_{\{i\}}}(x_i) = f(x_i)$ since the weighted rule is based on the unweighted rule. Assume now that Θ has m nonzero entries. By compatibility, we can assume that Θ is over an index set I of size m. Let V be the $(m-1)$-dimensional hyperplane in m-dimensional Euclidean space (indexed by I), where V is defined by $\sum_{i \in I} \theta'_i = 1$. Let R be the (bounded) subregion of V where $\theta'_i \geqslant 0$ for each $i \in I$. For each $i \in I$, let B_i be the $(m-2)$-dimensional hyperplane that is the intersection of V with the $(m-1)$-dimensional hyperplane defined by $\theta'_i = 0$. Then the boundary B of R is the union of the B_i's. Each Θ' in B has at least one 0 entry. Therefore, by induction hypothesis (and by compatibility), we can assume that we already have determined a formula for $f_{\Theta'}$ for each Θ' in B. Now Θ is a linear combination of E_I and of some Θ' in B; say $\Theta = \alpha \cdot E_I + (1-\alpha) \cdot \Theta'$, where $\alpha \in [0,1]$. (In fact, $\alpha = m \cdot \theta_{\sigma(m)}$ when Θ is ordered by σ.) Assume that X is over I. We know that $f_{E_I}(X) = f(X)$, since the weighted rule is based on f, and by induction hypothesis we know a formula for $f_{\Theta'}$. By local linearity, we know that we can then take $f_\Theta(X)$ to be $\alpha \cdot f(X) + (1-\alpha) \cdot f_{\Theta'}(X)$. This turns out to give us the weighting formula.

We remark that unlike the geometric viewpoint given at the start of this section, under this latter geometric viewpoint it is not at all clear that the weighting formula should have a linear number of terms. Indeed, we were surprised when our computations yielded only a linear number of terms.

8. Totally linear rules

When is the weighted rule not only locally linear, but even totally linear? Theorem 8.1 below tells us that this happens only for very special unweighted rules f, namely, those where $f(x_1, \ldots, x_m) = (f(x_1) + \cdots + f(x_m))/m$.

Theorem 8.1. *Assume* $I = \{1, \ldots, m\}$. *The following are equivalent*:

1. *The weighted rule given by the weighting formula is totally linear.*

2. $f(x_1,\ldots,x_m)=(f(x_1)+\cdots+f(x_m))/m$ *for each* m.
3. $f_{(\theta_1,\ldots,\theta_m)}(x_1,\ldots,x_m)=\theta_1\cdot f(x_1)+\cdots+\theta_m\cdot f(x_m)$, *for each* m.

Proof. ($1\Rightarrow 2$). For each i in I, let Θ^i be the weighting over I that is entirely concentrated on component i. Thus, $(\Theta^i)(i)=1$ and $(\Theta^i)(j)=0$ if $j\neq i$. Therefore, $E_I=(\Theta^1+\cdots+\Theta^m)/m$. Then

$$\begin{aligned} f(X) &= f_{E_I}(X) \quad \text{since the weighted rule is based on } f \\ &= \left(\sum_{i=1}^{m} f_{\Theta^i}(X)\right)\Big/m \quad \text{since the weighted rule is totally linear} \\ &= \left(\sum_{i=1}^{m} f(x_i)\right)\Big/m \quad \text{since the weighted rule is compatible and based on } f. \end{aligned}$$

($2\Rightarrow 3$). Here the proof is as in Example 1.

($3\Rightarrow 1$). It is easy to see from a straightforward calculation that part (3) implies that the weighted rule is totally linear. □

Under the natural assumption (discussed near the beginning of Section 3) that $f(x_i)=x_i$, part (2) says that $f(x_1,\ldots,x_m)=(x_1+\cdots+x_m)/m$, so that the unweighted rule is precisely the average. Part (3) then says that $f_{(\theta_1,\ldots,\theta_m)}(x_1,\ldots,x_m)=\theta_1\cdot x_1+\cdots+\theta_m\cdot x_m$, the weighted average.

9. Inherited properties

In this section, we restrict our attention (except at the end of the section) to situations where the domain D (which represents the values of the arguments of the scoring rule) and the range S (the values the scoring rule takes) are each sets of numbers. For example, these might correspond to situations where we are combining a collection of scores to obtain an overall score. So far we have not considered any other restrictions on a scoring rule. In practice, a scoring rule usually enjoys many properties such as continuity, monotonicity, etc., in its argument X. As we discuss in this section, these properties are inherited by the weighted rule.

We say that an unweighted rule is *continuous* if for each choice of index set I, it is continuous when restricted to tuples X over I. Similarly, a weighted rule is continuous if each f_Θ is continuous in the argument X. Note that this is in contrast to the discussion in Section 4, where we were concerned with continuity in the weighting Θ. In the case where the scoring rule is combining individual scores to obtain an overall score, we would expect the scoring rule to be continuous: slight changes in individual scores should lead to only slight changes in the overall score.

If X and X' are each tuples over the same index set I, let us write $X\geqslant X'$ if $x_i\geqslant x_i'$ for each $i\in I$, and $X>X'$ if $x_i>x_i'$ for each $i\in I$. We say that an unweighted rule is *monotonic* if for each choice of index set I, it is monotonic when restricted to tuples over I. That is, an unweighted rule f is monotonic if $f(X)\geqslant f(X')$ whenever $X\geqslant X'$.

An unweighted rule is *strictly monotonic* if $f(X) > f(X')$ whenever $X > X'$. Similarly, a weighted rule is monotonic (resp. strictly monotonic) if each f_Θ is monotonic (resp. strictly monotonic). In the case where the scoring rule is combining individual scores to obtain an overall score, we would certainly expect the scoring rule to be monotonic: intuitively, if the individual scores according to X are each at least as big as the corresponding scores according to X', then the overall score of X should be at least as big as the overall score of X'. Similarly, we expect a scoring rule to be strictly monotonic; if it is monotonic but not strictly monotonic, then there is a portion of the domain where the scoring rule is insensitive. In fact, in Section 10 we shall mention an example of a weighted rule that is monotonic but not strictly monotonic, that arises under a certain method for obtaining the weighted rule; such a rule might be considered undesirable.

We now define a notion of an unweighted rule being *strict*. This notion will be important in Section 11. For this notion, we assume that, as is common in fuzzy logic, the domain D and the range S are both the closed interval $[0,1]$. Intuitively, an unweighted rule is strict if it takes on the value 1 precisely when it is over a tuple of all 1's. Formally, we say that an unweighted rule f is *strict* if whenever X is over I, then $f(X)=1$ iff $x_i=1$ for every $i \in I$. Strictness is certainly a property we would expect of any scoring rule that is used to evaluate the conjunction. We now define strictness in the weighted case. Assume that Θ is over I. We say that Θ *has full support* if the support of Θ is I, that is, if θ_i is nonzero for every $i \in I$. We say that a weighted rule is strict if whenever Θ and X are over I, and Θ has full support, then $f_\Theta(X)=1$ iff $x_i=1$ for every $i \in I$.

An unweighted rule f is called *translation-preserving* if $f(X')=f(X)+a$, provided X and X' are over the same index set I, and $x_i'=x_i+a$ for every $i \in I$. Similarly, a weighted rule is translation-preserving if each f_Θ is translation-preserving in this sense. The idea behind a translation-preserving scoring rule is that if all the input scores are increased by the same amount, then the output score is increased by that same amount. Unlike the situation with continuity and monotonicity, we do not usually expect a scoring rule to be translation-preserving, even in the case where the scoring rule is combining scores to obtain an overall score. Of course, the min function is translation-preserving. In fact, as we shall discuss later (Proposition 10.1), min is the unique monotonic, translation-preserving binary function (up to boundary conditions).

An unweighted rule f *satisfies betweenness* if $\min X \leqslant f(X) \leqslant \max X$ for every X. This says that the resulting score lies between the smallest and largest of its arguments. This is certainly a natural property that we might expect of a scoring rule that combines scores to obtain an overall score. An unweighted rule f *is idempotent* if $f(x,\dots,x)=x$ for every x in the domain D. This says that if all of the "input scores" are equal, then the resulting output score has this same value. It is clear that if a scoring rule satisfies betweenness, then it is also idempotent. As before, we say that a weighted rule satisfies betweenness (resp. is idempotent) if each f_Θ satisfies betweenness (resp. is idempotent) in this sense.

The next theorem says that the properties we have discussed in this section are inherited by the weighted rule.

Theorem 9.1. *If an unweighted rule is continuous* (*resp. is monotonic, is strictly monotonic, is strict, is translation-preserving, satisfies betweenness, is idempotent*), *then the corresponding weighted rule given by the weighting formula is continuous* (*resp. is monotonic, is strictly monotonic, is strict, is translation-preserving, satisfies betweenness, is idempotent*) *as well.*

Proof. All of the results follow from Corollary 5.2. The only result that does not follow easily is for strictness, which we now consider.

Assume that f is strict, and that $(\theta_1,\ldots,\theta_m)$ has full support (which means that each θ_i is nonzero). By Corollary 5.2, we know that $f_\Theta(X)$ is a convex combination $\sum_{i=1}^m \alpha_i \cdot f(x_1,\ldots,x_i)$ of the terms $f(x_1,\ldots,x_i)$, for $1 \leqslant i \leqslant m$. From Theorem 5.1, we see that $\alpha_m = m \cdot \theta_m$, and so $\alpha_m > 0$.

Assume that $f_\Theta(X) = 1$. Since by assumption $0 \leqslant f(x_1,\ldots,x_i) \leqslant 1$ for each i, it follows that for each i where $\alpha_i > 0$, necessarily $f(x_1,\ldots,x_i) = 1$ (otherwise the convex combination could not take on the maximal value of 1). In particular, $f(x_1,\ldots,x_m) = 1$. Since by assumption f is strict, it follows that $x_i = 1$, for $1 \leqslant i \leqslant m$. This was to be shown. □

In the remainder of this section, we no longer assume that the domain D and the range S are each sets of numbers. We would like to show that some sort of symmetry is inherited by the weighted rule from the unweighted rule. Normally, a function is called symmetric if it is unchanged by any permutation of its arguments. In our setting, this translates to saying that we can take any permutation of the indices without changing the result. We now formally define the notion of symmetry. In this definition, $\circ$ represents functional composition, and $\delta(I)$ represents the image of the set I under the function δ when I is a subset of the domain of δ.

An unweighted rule f is called *symmetric* if $f(X \restriction \delta(I)) = f((X \circ \delta) \restriction I)$ for each permutation δ of $\mathscr{I}$, each nonempty $I \subseteq \mathscr{I}$, and each X over $\delta(I)$. A weighted rule is called *symmetric* if $f_\Theta(X) = f_{\Theta \circ \delta}(X \circ \delta)$ for each permutation δ of $\mathscr{I}$, each Θ over $\delta(I)$, and each X over $\delta(I)$. (Note that f_Θ and X are over $\delta(I)$, and that $f_{\Theta\circ\delta}$ and $X \circ \delta$ are over I.) Being symmetric means intuitively that we do not distinguish among the arguments.

Theorem 9.2. *If an unweighted rule is symmetric, then the corresponding weighted rule given by the weighting formula is symmetric as well.*

Proof. Since we shall need to use different permutations of the index set I, we shall again need to make use of (3).

Assume that f is a symmetric, unweighted rule. Assume that the weighted rule is based on f, compatible, and locally linear, that δ is a permutation of $\mathscr{I}$, and that Θ

and X are over $\delta(I)$. Let $\Theta' = \Theta \circ \delta$ and let $X' = X \circ \delta$. Note that Θ' and X' are over I. It suffices to show that $f_\Theta(X) = f_{\Theta'}(X')$.

Let $m = card(I)$. Let σ be a bijection from $\{1,\ldots,m\}$ onto $\delta(I)$ such that Θ is ordered by σ. Let $\sigma' = \delta^{-1} \circ \sigma$ (here δ^{-1} represents the inverse function, which is defined since δ is injective). Note that σ' is a bijection from $\{1,\ldots,m\}$ onto I.

Assume that $i \leqslant m$. Then $\theta'_{\sigma'(i)} = \Theta'(\sigma'(i)) = \Theta \circ \delta(\delta^{-1} \circ \sigma(i)) = \Theta(\delta(\delta^{-1}(\sigma(i)))) = \Theta(\sigma(i)) = \theta_{\sigma(i)}$. Hence, $\theta'_{\sigma'(i)} = \theta_{\sigma(i)}$ for $i \leqslant m$. Since Θ is ordered by σ, it follows that Θ' is ordered by σ'. Now, $f(X \restriction \{\sigma(1),\ldots,\sigma(i)\}) = f(X \restriction \delta(\{\sigma'(1),\ldots,\sigma'(i)\})) = f(X \circ \delta \restriction \{\sigma'(1),\ldots,\sigma'(i)\}) = f(X' \restriction \{\sigma'(1),\ldots,\sigma'(i)\})$. Here the second equality holds since f is symmetric. Thus, $f(X \restriction \{\sigma(1),\ldots,\sigma(i)\}) = f(X' \restriction \{\sigma'(1),\ldots,\sigma'(i)\})$ for $i \leqslant m$.

From (3), it follows that

$$\begin{aligned} f_\Theta(X) &= \sum_{i=1}^{m-1} i \cdot (\theta_{\sigma(i)} - \theta_{\sigma(i+1)}) \cdot f(X \restriction \{\sigma(1),\ldots,\sigma(i)\}) + m \cdot \theta_{\sigma(m)} \cdot f(X) \\ &= \sum_{i=1}^{m-1} i \cdot (\theta'_{\sigma'(i)} - \theta'_{\sigma'(i+1)}) \cdot f(X' \restriction \{\sigma'(1),\ldots,\sigma'(i)\}) + m \cdot \theta'_{\sigma'(m)} \cdot f(X') \\ &= f_{\Theta'}(X') \end{aligned}$$

which is the desired result. □

It is quite common for naturally occurring scoring rules to be symmetric. For example, average, max, and min are symmetric scoring rules. Of course, some circumstances might provide a reason to treat the arguments differently and occasion the use of a non-symmetric scoring rule. One such scenario could arise if, say, we are considering scores assigned to different attributes, and all of the scores about one particular attribute are guaranteed to be in the interval $[0, \frac{1}{2}]$, but for the other attributes the scores can range throughout the interval $[0, 1]$. Assume that in this situation we are "designing" a scoring rule. Consider the case where the tuple X has two entries x_1 and x_2, and where x_1 is guaranteed to be in the interval $[0, \frac{1}{2}]$. Instead of, say, taking $f(X)$ in this case to be the average $(x_1 + x_2)/2$, it would probably be more reasonable to "normalize" and take it to be $(2x_1 + x_2)/2$ instead. This leads to an unweighted rule that is not symmetric.

10. Related work

There is much work in the economics literature about indifference curves. This includes work on computing optimal indifference curves (which depend on user-supplied weightings). That work is only tangentially related to ours, since the focus there is on computing optimality. See [18] for a more complete discussion. In Example 8 of Section 6, we discussed voting schemes as an example where there is a "competing" approach to ours for incorporating weights. As we noted, future research is required to understand the issues involved. In Section 10.1, we discuss other methods for obtaining a weighted rule from an unweighted rule. In Section 10.2, we consider the Choquet integral, which is also intended to deal with weightings.

10.1. Other approaches

We now discuss three methods from the literature for obtaining a weighted rule from an unweighted rule, and compare them with our approach. Each of these methods deals only with one or a few particular unweighted rules, rather than, as in our approach, with arbitrary unweighted rules.

Method 1. The first method is inspired by a paper by Dubois and Prade [13]. It deals with the case where the min function is used in the unweighted case (in fact, the title of Dubois and Prade's paper is "Weighted Minimum and Maximum Operations in Fuzzy Set Theory"). Their underlying scenario and goals are actually quite different from ours (for example, instead of dealing with probabilities θ_i, they deal with possibility distributions [36]). Nonetheless, it is instructive to compare the explicit formula that they obtain for the "weighted min", and see how it fares under our criteria.

Let X be a tuple over I with range $[0,1]$, let $f(X)=\min_{i\in I}\{x_i\}$ and let

$$f_\Theta(X)=\min_{i\in I}\{\max\{1-(\theta_i/M),x_i\}\}, \tag{21}$$

where $M=\max_{i\in I}\{\theta_i\}$. It is easy to check that the weighted rule is compatible, and based on f. Note that the weighted rule is not locally linear.

An attractive feature of this weighted rule is that it is simple, continuous (both in Θ and X), monotonic, strict, symmetric, and idempotent, and it satisfies betweenness. However, it is not strictly monotonic. For example, let $\theta_1=\frac{2}{3}$ and $\theta_2=\frac{1}{3}$. The reader can easily verify that $f_\Theta(.7,.3)$ and $f_\Theta(.8,.4)$ are each equal to .5. In fact, it is easy to verify that if $x_2\leqslant .5\leqslant x_1$, then $f_\Theta(X)=.5$. We consider this undesirable, since it says intuitively that f_Θ is insensitive to its arguments in this region.

The fact that the weighted rule (21) is compatible and based on f depends on the assumption that we are considering min only over tuples with range $[0,1]$. If we were to allow tuples with a different range, and in particular the range $(-\infty,+\infty)$, then it is not clear how to modify the weighted rule (21). In contrast, our weighting formula for min is the same in all cases.

Also, this weighted rule is not translation-preserving, since $f_\Theta(.8,.4)\neq f_\Theta(.7,.3)+.1$. We consider this undesirable because, as the following proposition shows, the key feature of min (the underlying scoring rule) is that it is monotonic and translation-preserving. Thus, up to boundary conditions, it is uniquely determined by being monotonic and translation-preserving.

Proposition 10.1. min *is the unique monotonic*, *translation-preserving binary function* f *on* $[0,1]$ *for which* $f(0,1)=0=f(1,0)$.

Proof. It is easy to check that min has all the desired properties. So assume that f has the properties with the goal of proving that $f(x_1,x_2) = \min(x_1,x_2)$ for all $x_1,x_2 \in [0,1]$. By monotonicity it follows that $f(0,t) = 0 = f(t,0)$ for all $t \in [0,1]$. The rest of proof breaks into two cases depending on whether x_1 or x_2 is larger.

1. If $x_1 \leqslant x_2$, then $f(x_1,x_2) = f(0 + x_1,(x_2 - x_1) + x_1) = f(0,x_2 - x_1) + x_1 = 0 + x_1 = \min(x_1,x_2)$.
2. If $x_1 \geqslant x_2$, then $f(x_1,x_2) = f((x_1 - x_2) + x_2, 0 + x_2) = f(x_1 - x_2, 0) + x_2 = 0 + x_2 = \min(x_1,x_2)$.

In either case, $f(x_1,x_2) = \min(x_1,x_2)$. □

Similar characterizations of min appear in the literature [1, 24]. The formula in (21) is computationally a little simpler than the weighting formula. Therefore, there might be situations, where, say, computational simplicity is more important than strict monotonicity and translation invariance, when (21) would be preferable to use rather than the weighting formula.

Method 2. Sung [28] recently wrote a follow-on to our paper, in which he considers other possible desiderata. In doing so, he develops techniques for obtaining a weighted rule from an unweighted rule that applies in certain special cases. In particular, he gives another method of obtaining a "weighted min", which we now describe.

Let X be a tuple over I with range $[0,1]$, let $f(X) = \min_{i \in I} \{x_i\}$ and let

$$f_\Theta(X) = 1 - \frac{1}{M} \max_{i \in I} \{\theta_i - \theta_i x_i\}, \tag{22}$$

where $M = \max_{i \in I}\{\theta_i\}$. It is straightforward to check that like (21), this weighted rule is compatible and based on f, but not locally linear. As with (21), an attractive feature of this weighted rule is that it is simple, continuous (both in Θ and X), monotonic, strict, symmetric, and idempotent, and it satisfies betweenness. Unlike (21), this formula has the additional desirable feature that it is strictly monotonic. However, like (21), it is not translation-preserving. As before, we consider this undesirable because, as Proposition 10.1 shows, the key feature of min (the underlying scoring rule) is that it is monotonic and translation-preserving.

Finally, as with (21), it is not clear how to modify the weighted rule (22) for min over tuples with a different range than $[0,1]$, and in particular with range $(-\infty, +\infty)$.

Method 3. The third method is from a paper by Salton et al. on information retrieval [26], and deals with the case where a version of the Euclidean distance is used in the unweighted case.

Let X be a tuple over I, let

$$f(X) = \sqrt{\frac{\sum_{i \in I} x_i^2}{card(I)}} \tag{23}$$

and let

$$f_\Theta(X)=\sqrt{\frac{\sum_{i\in I}\theta_i^2 x_i^2}{\sum_{i\in I}\theta_i^2}}, \tag{24}$$

when Θ is over I. It is easy to check that the weighted rule is compatible and is based on f. Note that the weighted rule is not locally linear. Unlike other scoring rules we have discussed, these take us out of the rationals.

The weighted rule given by the formula (24) is quite reasonable: it gives a natural generalization of the unweighted formula (23); it is continuous (both in Θ and X), strictly monotonic, strict, symmetric, and idempotent, and it satisfies betweenness. It is not translation-preserving, but we would not expect it to be, since the unweighted rule is not translation-preserving.

As was the case with Methods 1 and 2 in this section, the formula for f_Θ in (24) is computationally easier than the weighting formula in this case, since only one square root is involved in (24), whereas m square roots are involved in the weighting formula.

One possible objection to (24) is that it is not clear why θ_i^2, rather than θ_i, is being used in the formula; either seems like a reasonable alternative. In fact, the QBIC [13] system [23] also uses a variation of Euclidean distance, and in the weighted case uses θ_i rather than θ_i^2. Because of the specific form of the unweighted rule (23), there is a natural extension to the weighted rule, as given by (24). There are other examples where there is a natural way (other than ours) to modify an unweighted rule to obtain a weighted rule. For example, if $f(X)$ is the geometric mean $(x_1\cdots x_m)^{1/m}$, then a natural generalization in the weighted case would be $x_1^{\theta_1}\cdots x_m^{\theta_m}$. On the other hand, there are often situations where there is no natural generalization in the weighted case; min is a good example.

The point of this example is that for certain special unweighted rules, there may be a natural way to obtain a weighted rule that is not the weighted rule given by the weighting formula that our methodology gives us. In fact, the extension (24) is more in the spirit of the unweighted case (23) than our extension, the weighting formula (2). But our methodology has the advantage that it *always* gives us a (simple) way to obtain a weighted rule from an unweighted rule, no matter what the unweighted rule is.

10.2. Choquet integral

The Choquet integral [8, 27] of a function generates a weighted version of the function. A nice overview appears in [17].

Let g be a function with finite domain X and range $[0,1]$. Let μ be a set function (that is, a function with domain the power set of X) and range $[0,1]$. It is common to make various assumptions about μ, such as that $\mu(\emptyset)=0$, that $\mu(X)=1$, and that μ is monotone (i.e., $\mu(A)\leqslant\mu(B)$ if $A\subseteq B$). We think of μ as a generalized measure.

[13] QBIC, which stands for Query By Image Content, is a trademark of IBM Corporation.

Assume that X consists of the m elements $x_1,\dots,x_m$, and that the indices are selected so that $g(x_1)\geqslant\cdots\geqslant g(x_m)$. The *Choquet integral of g with respect to μ* is defined to be

$$(g(x_1)-g(x_2))\cdot\mu(\{x_1\})+(g(x_2)-g(x_3))\cdot\mu(\{x_1,x_2\}) \\ +\cdots+(g(x_{m-1})-g(x_m))\cdot\mu(\{x_1,\dots,x_{m-1}\})+(g(x_m))\cdot\mu(\{x_1,\dots,x_m\}). \quad (25)$$

Note that if μ is additive, so that $\mu(\{x_1,\dots,x_k\})=\mu(x_1)+\cdots+\mu(x_k)$, then the Choquet integral (25) reduces to $\mu(x_1)\cdot g(x_1)+\cdots+\mu(x_m)\cdot g(x_m)$, which is a weighted average of $g(x_1),\dots,g(x_m)$. Thus, the intuition behind the Choquet integral is that it is some sort of weighted average of $g(x_1),\dots,g(x_m)$, even in the case when μ is not necessarily additive.

We now show a sense in which our weighting formula in (2) is a Choquet integral.[14] Let us define a set function μ by taking $\mu(\{x_{i_1},\dots,x_{i_s}\})$ to be $s\cdot f(x_{i_1},\dots,x_{i_s})$ for each subset $\{x_{i_1},\dots,x_{i_s}\}$ of X of size s. If Θ is a weighting, then define g (over X) by taking $g(x_i)=\theta_i$. It is straightforward to see that the Choquet integral of g with respect to μ gives our weighting formula in (2).

Because of the intuition of a Choquet integral of g with respect to μ as being some sort of weighted average of $g(x_1),\dots,g(x_m)$, we might have expected our weighting formula to be a Choquet integral of some variation of f with respect to some variation of Θ. Instead, somewhat mysteriously, our weighting formula turns out to a Choquet integral of a variation of Θ with respect to a variation of f. It is also interesting to note that in order to define a Choquet integral that gives our weighting formula, it is necessary to define $\mu(\{x_{i_1},\dots,x_{i_s}\})$ to be $s\cdot f(x_{i_1},\dots,x_{i_s})$ rather than the more natural choice of simply $f(x_{i_1},\dots,x_{i_s})$.

Schmeidler [27] refers to two functions g and g' on the same domain as being *comonotonic* if there do not exist x,y with $g(x)<g(y)$ and $g'(x)>g'(y)$. Since in our case we have $g(x_i)=\theta_i$, it is clear that Schmeidler's notion of comonotonicity gives exactly our notion of comonotonicity (this is why we use the term). Dellacherie [11] shows that the Choquet integral is additive for comonotonic functions. It follows easily that the Choquet integral of Θ (with respect to an arbitrary μ) is locally linear. Marinacci (personal communication, 1998) proves the converse, namely, that each locally linear function of Θ is a Choquet integral of Θ. Marinacci uses this to give an alternative proof of the uniqueness part of Theorem 5.1. It might be interesting to explore other consequences of viewing our weighting formula as a Choquet integral.

11. Low middleware cost in a multimedia database system

In this section we focus on the case, mentioned in the introduction, of queries in a multimedia database system. Garlic [6, 10] is a multimedia database system being developed at the IBM Almaden Research Center. It is designed to be capable of integrating data that resides in different database systems as well as a variety of nondatabase data

[14] This was pointed out to us by David Schmeidler.

servers. A single Garlic query can access data in a number of different subsystems. An example of a nontraditional subsystem that Garlic accesses is QBIC [23] ("Query By Image Content"). QBIC can search for images by various visual characteristics such as color and texture. In [14], the first author developed an efficient algorithm for evaluating conjunctions in such a system, when the conjuncts are independent. In this section, we show that this algorithm can be carried over to the weighted case.

Let us begin with an example, where we deal first with the unweighted case. Consider again the fuzzy conjunction $(Color = \text{'}red\text{'}) \wedge (Sound = \text{'}loud\text{'})$. We denote this query by Q. Assume that two different subsystems deal with color and sound (for example, QBIC might deal with color). Garlic is a middleware system that has to piece together information from both subsystems in order to answer the query Q. Let I be the index set {Color,Sound}, and let o be an object. Assume that the redness score of object o, as determined by the subsystem dealing with color, is x_1, and the loudness score of object o, as determined by the subsystem dealing with sound, is x_2. Let X be a tuple over I whose Color value is x_1 and whose Sound value is x_2. Then, in the setup of this paper, we would take the overall score of object o under query Q to be $f(X)$, where f is the scoring rule.

Let us say that we are interested in finding the top 10 answers to query Q (that is, the 10 objects with the highest overall scores, along with their scores). One way to do this would be to evaluate the query on every single object in the database, and take the 10 objects with the highest overall scores (ties would be broken arbitrarily). The problem with this naive algorithm is that there is a very high middleware cost: [15] every single object in the database must be accessed. The first author [14] gives an algorithm that is much more efficient, provided that the conjuncts are independent. He shows that if the scoring rule (in this case, f) is monotonic, then the middleware cost is of the order of the square root of the number of objects in the database. (More precisely, it is shown that if there are m conjuncts, and N objects in the database, then the middleware cost for finding the top k objects in the database is $O(N^{(m-1)/m}k^{1/m})$, with arbitrarily high probability. For details about the probabilistic assumptions, see [14].) Furthermore, it is shown that if the scoring rule is strict, then this is optimal.

What about the weighted case, where, say, we care twice as much about the color as about the sound? It follows from Theorem 9.1 that if the unweighted rule is monotonic and strict, then so is the weighted rule. Now the upper bound in [14] depends only on the scoring rule being monotonic, and the matching lower bound depends only on the scoring rule being strict. Therefore, we have the following theorem.

Theorem 11.1. *Assume that the unweighted rule is monotonic and strict. Then for the corresponding weighted rule given by the weighting formula, there is an algorithm* $\mathscr{A}$ *for finding the top* k *answers to the query determined by* f_Θ. *If the support of*

[15] The cost model, including the definition of "middleware cost", is defined formally in [14]. Intuitively, the middleware cost corresponds to the number of elements accessed in the database.

Θ consists of m independent attributes, then the middleware cost for algorithm $\mathscr{A}$ is $O(N^{(m-1)/m}k^{1/m})$ *with arbitrarily high probability, and this is optimal.*

12. Summary

There are numerous situations where there is a rule for assigning values to tuples. It is often the case that we do not wish to give equal weight to all of the components. This paper presents, by means of a surprisingly simple formula, a general method that extends any rule to a weighted version of the rule. Our method is the unique one that satisfies certain natural properties.

Acknowledgements

We thank Joe Halpern for many useful discussions and for the suggestion that we extend our results to include the non-symmetric case. We also thank Shmuel Gal, Gil Kalai, Laura Haas, Ariel Landau, Yoëlle Maarek, Massimo Marinacci, Nimrod Megiddo, David Schmeidler, Madhu Sudan, and Avi Wigderson for helpful comments. We are also grateful to Moshe Vardi for making us aware of Sung's paper [28].

References

[1] C. Alsina, E. Trillas, Additive homogeneity of logical connectives for membership functions, in: J.C. Bezdek (Ed.), Analysis of Fuzzy Information, vol. 1, Mathematics and Logic, CRC Press, Boca Raton, 1987.
[2] K.J. Arrow, Social Choice and Individual Values, 2nd Edition, Yale University Press, New Haven, 1963.
[3] K.J. Arrow, H. Raynaud, Social Choice and Multicriterion Decision-Making, MIT Press, Cambridge, 1986.
[4] G.W. Bassett, Jr., J. Persky, Rating skating, J. Amer. Statist. Assoc. 89 (427) (1994) 1075–1079.
[5] R. Bellman, M. Giertz, On the analytic formalism of the theory of fuzzy sets, Inform. Sci. 5 (1973) 149–156.
[6] M.J. Carey, L.M. Haas, P.M. Schwarz, M. Arya, W.F. Cody, R. Fagin, M. Flickner, A.W. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J.H. Williams, E.L. Wimmers, Towards heterogeneous multimedia information systems: the Garlic approach, RIDE-DOM '95, 5th Internat. Workshop on Research Issues in Data Engineering: Distributed Object Management, 1995, pp. 124–131.
[7] S. Chaudhuri, L. Gravano, Optimizing queries over multimedia repositories, Proc. ACM SIGMOD Conf., 1996, pp. 91–102.
[8] G. Choquet, Theory of capacities, Ann. Inst. Fourier 5 (1955) 131–295.
[9] M. Chrobak, H. Karloff, T. Payne, S. Vishwanathan, New results on server problems, Proc. ACM-SIAM Symp. On Discrete Algorithms, 1990, pp. 290–300.
[10] W.F. Cody, L.M. Haas, W. Niblack, M. Arya, M.J. Carey, R. Fagin, M. Flickner, D.S. Lee, D. Petkovic, P.M. Schwarz, J. Thomas, M. Tork Roth, J.H. Williams, E.L. Wimmers, Querying multimedia data from multiple repositories by content: the Garlic Project, IFIP 2.6 3rd Working Conf. on Visual Database Systems (VDB-3), 1995.
[11] C. Dellacherie, Quelques commentaires sur les prolongements de capacités, Séminaire Probabilités V, Strasbourg, Lecture Notes in Mathematics, Vol. 191, Springer, Berlin.

[12] D. Dubois, H. Prade, Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory, in: H.-J. Zimmermann, L.A. Zadeh, B. Gaines (Eds.), Fuzzy Sets and Decision Analysis, TIMS Studies in Management Sciences, Vol. 20, 1984, pp. 209–240.
[13] D. Dubois, H. Prade, Weighted minimum and maximum operations in fuzzy set theory, Inform. Sci. 39 (1986) 205–210.
[14] R. Fagin, Combining fuzzy information from multiple systems, J. Comput. and System Sci. 58 (1999) 83–99.
[15] R. Fagin, Y.S. Maarek, Allowing users to weight search terms in information retrieval, IBM Research Report RJ 10108, March 1998.
[16] A. Gibbard, Manipulation of voting schemes: a general result, Econometrica 41 (4) (1973) 587–601.
[17] M. Grabisch, The application of fuzzy integrals in multicriteria decision making, European J. Oper. Res. 89 (1996) 445–456.
[18] R.L. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley, New York, 1976.
[19] D.K. Lewis, Counterfactuals, Harvard University Press, Cambridge, MA, 1973.
[20] M.S. Manasse, L.A. McGeoch, D.D. Sleator, Competitive algorithms for on-line problems, Proc. ACM Symp. on Theory of Computing, 1988, pp. 322–333.
[21] R.L. Mattson, J. Gecsei, D.R. Slutz, I.L. Traiger, Evaluation techniques for storage hierarchies, IBM Systems J. 9 (2) (1970) 78–117.
[22] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, The QBIC project: querying images by content using color, texture and shape, Proc. SPIE, San Jose, CA, 1993, pp. 173–187.
[23] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, The QBIC project: querying images by content using color, texture and shape, SPIE Conf. on Storage and Retrieval for Image and Video Databases vol. 1908, 1993, pp. 173–187. QBIC Web server is http://wwwqbic.almaden.ibm.com/
[24] M. Pirlot, A characterization of 'min' as a procedure for exploiting valued preference relations and related results, J. Multi-Criteria Decision Anal. 4 (1995) 37–56.
[25] P. Raghavan, M. Snir, Memory versus randomization in online algorithms, IBM J. Res. Dev. 38 (1994) 683–707.
[26] G. Salton, E.A. Fox, H. Wu, Extended information retrieval, Commun. ACM 26 (12) (1983) 1022–1036.
[27] D. Schmeidler, Integral representation without additivity, Proc. Amer. Math. Soc. 97 (2) (1986) 255–261.
[28] S.Y. Sung, A linear transform scheme for combining weights into scores, Technical Report TR98-327, Rice University, 1998.
[29] A. Tomkins, Practical and theoretical issues in prefetching and caching, Ph.D. Dissertation, Carnegie Mellon University, 1997.
[30] W. Voxman, R. Goetschel, A note on the characterization of the max and min operators, Inform. Sci. 30 (1983) 5–10.
[31] E.L. Wimmers, Minimal Bellman-Giertz Theorems, to appear.
[32] R.R. Yager, Some procedures for selecting fuzzy set-theoretic operations, Internat. J. Gen. Systems 8 (1982) 115–124.
[33] N.E. Young, On-line caching as cache size varies, Proc. ACM-SIAM Symp. on Discrete Algorithms, 1991, pp. 241–250.
[34] N.E. Young, On-line file caching, Proc. ACM-SIAM Symp. on Discrete Algorithms, 1998, pp. 82–86.
[35] L.A. Zadeh, Fuzzy sets, Inform. and Control 8 (1965) 338–353.
[36] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.
[37] H.-J. Zimmermann, Fuzzy Set Theory, 3rd ed., Kluwer Academic Publishers, Boston, 1996.