Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Combining Weights into Scores: A Linear Transform Approach

Sam Yuan Sung
Department of Computer Science
South Texas College
McAllen, USA
sysung@southtexascollege.edu

Tianming Hu
Department of Computer Science
Dongguan University of Technology
Dongguan, China
tmhu@ieee.org

***Abstract*—Ranking has been widely used in many applications. A ranking scheme usually employs a scoring rule that assigns a final numerical value to every object to be ranked. A scoring rule normally involves the use of one to many scores, and it gives more weight to the scores that are more important. In this paper, we give a scheme that can combine weights into scores in a natural way and compare our scheme to the formula given by Fagin. Also given are some additional properties that are desirable for weighted scoring rules. Finally, we discuss other interesting issues on weighted scoring rules.**

***Index Terms*—linear transform, ranking, scoring rule, weighted method**

I. Introduction

Ranking has been used in many applications. Examples include multimedia databases, information retrievals, image and pictorial databases, and many others. In these applications, ranking is a pre-requisite because it is very common that users will issue similarity searches. Similarity is defined in terms of a similarity distance function. The smaller the similarity distance value, the more similar are two objects.

Typical similarity search queries are the similarity range query and the k-nearest neighbor query. A similarity range query is specified by a query object and a similarity distance range. A k-nearest neighbor query is specified by a query object and a number k, the k most similar objects to the query object are to be retrieved. There is a large literature in the area of efficient support of similarity search, e.g., [1], [2]. In all of similarity searches, ranking or scoring rule is used for the purpose of sorting qualified objects, or filtering out unqualified objects.

In [3], [4], Fagin and Wimmers have addressed the problem of "incorporating weights into scoring rules", because it is often desirable and/or necessary that extra weight can be assigned to more important attributes (or subqueries, judges, etc) in order to get a better or fair result. This problem [5] is also our main focus in this paper. Fagin and Wimmers gave a simple but elegant formula for incorporating weights into ranking rules. Their formula has several desirable properties. Most importantly, their formula can be applied to almost any underlying ranking rules straightforwardly. However, the formula also causes certain undesirable problems, some of which are noted by the formula's authors themselves. To study and overcome the problems encountered by Fagin's method is one of our motivations.

Our main contributions in this paper are

- we study Fagin's method and the reason behind its problems;
- we propose our solution for the problem of incorporating weights into scores;
- we discuss some other interesting issues related to this topic.

The rest of paper is organized as follows. The problem formulation is given in Section II. In Section III we present a brief introduction to the solution of incorporating weights into scores proposed by Fagin and Wimmers. We listed some of the drawbacks of their method. In Section IV, we propose a new method. Our method is based on linear transform, a natural way of extending the ranking rule from the unweighted case to the weighted case. We show that our method has overcome the drawbacks of the previous work. In particular, we apply our method on the minimum ranking rule, and show the advantages of our method over two other methods. In Section V, we discuss some interesting related problems. Our conclusions are given in Section VI.

II. Problem Formulation

The general ranking rule is as follows. Each object X has a number of scores associated with it. A "ranking rule" is applied to X to generate an overall score of X. The overall score is then used to rank X. A ranking rule is normally a mathematical function which uses an object's scores as arguments.

Let the number of scores associated with an object X be denoted as $(x_1, x_2, ..., x_n)$. Without loss of generality, we can assume that (1) $0 \leq x_i \leq 1$, for all $i = 1, ..., n$, where 1 represents a perfect match; (2) the higher the overall score is, the higher the rank has.

A popular ranking rule is to use geometric distance function of the following:

$$f(x_1, x_2, ..., x_n) = (x_1^\alpha + x_2^\alpha + ... + x_n^\alpha)^{\frac{1}{\alpha}} \quad (1)$$

where $\alpha = 1, 2, ..., \infty$. In particular, we have

- the sum rule: when $\alpha = 1$, $f(x_1, x_2, ..., x_n) = (x_1 + x_2 + ... + x_n)$.

- the Euclidean distance: when $\alpha = 2$, $f(x_1, x_2, ..., x_n) = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$.

Often used in multimedia database retrieval and information retrieval, the minimum and maximum rules are attractive because they are equivalent to the conjunctive and disjunctive in the standard fuzzy semantics [6].

- the minimum rule:

$$f(x_1, x_2, ..., x_n) = \min(x_1, x_2, ..., x_n) \tag{2}$$

- the maximum rule:

$$f(x_1, x_2, ..., x_n) = \max(x_1, x_2, ..., x_n) \tag{3}$$

Other forms of ranking rules can be found in many other areas, such as multicriterion decision-making [7] which has a large literature in economics [8], [9], sports competition [10] and multimodal biometric systems [11].

In practice, it is obvious that in many cases different weights need to be assigned to different attributes, different judges, etc. For example, in a multimedia database, the user might like to give more weight to color than shape in a search for similar pictures. Therefore, It is natural to ask for some meaningful and systematic way to combine weights and ranking rules. Fagin and Wimmers have proposed a formula for incorporating weights into ranking rules. Their formula has many desirable characteristics. However, the formula also has some undesirable features, and that inspires us to study the reasons and to find other alternate solutions.

III. Fagin and Wimmers' Solution

We now describe the solution of Fagin and Wimmers. More details can be found in [4]. Given an unweighted rule $f(x_1, x_2, ..., x_n)$, and a set of weights $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$, where each θ_i is the weight with respect to the score x_i. Assume that $\theta_1, \theta_2, ..., \theta_n$ are all nonnegative and sum to one, and without loss of generality, $\theta_1 \geq \theta_2 \geq ... \geq \theta_n$. A corresponding weighted rule is generated as follows, which is referred to as "FW formula" in this paper.

$$\begin{aligned} f_{(\theta_1, \theta_2, ..., \theta_n)}(x_1, x_2, ..., x_n) \quad = \quad & (\theta_1 - \theta_2) f(x_1) + \\ & 2(\theta_2 - \theta_3) f(x_1, x_2) + \\ & 3(\theta_3 - \theta_4) f(x_1, x_2, x_3) + \\ & ... + \\ & n\theta_n f(x_1, ..., x_n) \end{aligned} \tag{4}$$

An intuitive explanation of the FW formula is that x_1 has an extra weight of $\theta_1 - \theta_2$ than all the other x_is, so we first compute this excessive score for x_1. This gives the first term $(\theta_1 - \theta_2) f(x_1)$ in the formula. Similarly, both x_1 and x_2 have an extra weight of $\theta_2 - \theta_3$ than the other x_is, so we then compute this excessive score for x_1 and x_2, and get the second term in the formula. Continuing in this way, we finally get the last term of the formula.

A. Desirable Properties

The FW formula is simple in format, and elegant in intuition. It has the beauty that it works with any underlying unweighted rule in a straightforward manner. The weighted rule generated from the FW formula has the following desirable properties:

- D1: $f_{(\theta_1, ..., \theta_n)}(x_1, ..., x_n) = f(x_1, ..., x_n)$, if $\theta_1 = ... = \theta_n$. That is, if all weights in θ are equal, then the weighted rule f_θ coincides with the unweighted rule f.
- D2: $f_{(\theta_1, ..., 0)}(x_1, ..., x_n) = f_{(\theta_1, ..., \theta_{n-1})}(x_1, ..., x_{n-1})$. That is, if a particular argument has zero weight, then that argument can be dropped without affecting the value of the result.
- D3: $f_{(\theta_1, ..., \theta_n)}(x_1, ..., x_n)$ is a continuous function of $\theta_1, ..., \theta_n$.

The weighted rule f_θ obtained from the FW formula inherits many properties of the underlying unweighted rule f. For example, one important such property is monotonic. We write $X \geq X'$ for two objects X and X' if $x_i \geq x_i'$ for each i, and $X > X'$ if $x_i > x_i'$ for each i. We say an unweighted rule f is monotonic if $f(X) \geq f(X')$ whenever $X \geq X'$. Also, f is strictly monotonic if $f(X) > f(X')$ whenever $X > X'$. Analogously, a weighted rule is monotonic (strictly monotonic) if each f_θ is monotonic (strictly monotonic). The monotonic property means if each individual score of X is at least as big as the corresponding score of X', then the overall score of X should be at least as big as the overall score of X'. A rule is monotonic but not strictly monotonic means there is a portion of the domain where the ranking rule is insensitive; and such a rule might be considered as undesirable. In the next section, we will mention a few more properties.

B. Drawbacks

Unfortunately, there are some serious drawbacks with the FW formula.

1) Equivalence property is not preserved: We call two ranking rules f and g equivalent if $f(x) < f(y)$ whenever $g(x) < g(y)$ and vice versa. For example, the function sum $f(x_1, ..., x_n) = x_1 + ... + x_n$ and the function $average$ $f(x_1, ..., x_n) = (x_1 + ... + x_n)/n$ are equivalent. The functions $f(x_1, ..., x_n) = \sqrt{x_1^2 + ... + x_n^2}$ and $f(x_1, ..., x_n) = x_1^2 + ... + x_n^2$ are also equivalent.

In ranking, it is the relative, rather than the absolute, score that matters. For example, to find the winner of two arbitrary objects, using the function sum or average makes no difference – the winner is always the same. Thus, the equivalence property is very useful in ranking since it means two equivalent rules will give the same results as far as ranking is concerned. In practice, which rule to use (among several equivalent rules) is normally a consideration of efficiency, or simply a matter of convenience.

Under the FW formula, equivalent unweighted rules are often not generating equivalent weighted rules. For example,

TABLE I
AN EXAMPLE OF AGGREGATION ANOMALY

X	$w_1 = 0.6$	$w_2 = 0.4$	X'	$w_1 = 0.6$	$w_2 = 0.4$
$\theta_1 = 0.6$	0.8	0.6	$\theta_1 = 0.6$	0.6	0.8
$\theta_2 = 0.4$	0.6	0.8	$\theta_2 = 0.4$	0.8	0.6

assume the weighted sum is

$$f_{(\theta_1,\theta_2)}(x_1, x_2) = (\theta_1 - \theta_2)x_1 + 2\theta_2(x_1 + x_2) = x_1 + 2\theta_2 x_2 \tag{5}$$

and the weighted average is

$$f_{(\theta_1,\theta_2)}(x_1, x_2) = (\theta_1 - \theta_2)x_1 + 2\theta_2 \frac{x_1 + x_2}{2} = \theta_1 x_1 + \theta_2 x_2 \tag{6}$$

Let $\theta_1 = 2/3$ and $\theta_2 = 1/3$, we get rank$(0.4, 0.8) >$ rank$(0.8, 0.1)$ for the weighted sum, while rank$(0.4, 0.8) <$ rank$(0.8, 0.1)$ for the weighted average.

2) Lack of natural generalization and variation: For a given unweighted rule, the FW formula generates only one corresponding weighted rule. Also, this generalization is not a natural extension of the underlying unweighted rule. For example, the FW formula generates the following weighted rule for the Euclidean distance function $f_\theta(x) = (\theta_1 - \theta_2)x_1 + 2\theta_2\sqrt{x_1^2 + x_2^2}$. The above rule is not a natural generalization of the Euclidean distance. It also does not give the rich variation that a weighted Euclidean distance can be. For example, both $f_\theta(x) = \sqrt{\theta_1 x_1^2 + \theta_2 x_2^2}$ and $f_\theta(x) = \sqrt{\theta_1^2 x_1^2 + \theta_2^2 x_2^2}$ are two different natural extension of the unweighted Euclidean distance, but none of them is equivalent to the above function.

3) Aggregation anomaly: We show an illustrative 2-dimensional example in Table I, and give more details of this part in Section V.

Assume $w_1 = 0.6$, $w_2 = 0.4$, $\theta_1 = 0.6$ and $\theta_2 = 0.4$. Suppose the underlying rule used is sum. With the joint weighted rule defined in Eq. (13), the overall score for object X is 2.368. However, if we compute by row first and then followed by column, with Eq. (14), we get 1.28 for the 1st row, 1.24 for the 2nd row, and then by column we get the overall score 2.272. Similarly, for object X', we get the overall scores 2.392 and 2.264, respectively.

Some observations are in order: (1) if we apply the FW formula to individual data directly, the result might be different from if we apply the formula aggregately (row by column in this example); and (2) the two weighted rules (direct versus aggregate) are not equivalent.

4) Computational efficiency: Since the FW formula needs to compute the underlying function n times, it is in general less efficient than a weighted rule that computes the underlying function only once or less than n times.

5) Embedding into search method: A ranking rule is often used together with a search method, especially in database applications. For example, in finding the nearest k objects in similarity search [1], or the top k answers in multimedia search [2], or finding the relevant documents in information retrieval [12], etc. Fagin has shown that the weighted rule generated according to the FW formula can be straightaway used on several search algorithms [2], [13] with no or little modification. However, it is important to note that there are many other search algorithms where we do not know whether we can convert an algorithm for the unweighted case into an algorithm for the weighted case, without drastically changing the structure or affecting the performance. However, further discussion on this particular issue is beyond our scope here.

We find two reasons that might explain the above problems caused by the FW formula. The first one is the weights in the FW formula do not tightly embed into the underlying function. That is, the weights are not mixed with the scores directly, they are simply outside the functions. This makes the weighted rules unnatural and lack of versatility.

The second reason is the partial set evaluation. Note that the FW formula uses only subsets in its first $n-1$ terms, and only the last term uses the full set of $\{x_1, x_2, ..., x_n\}$. This partial set evaluation is the direct cause for most of the above problems, e.g., the loss of equivalence. It also accounts for the problem of "counterfactual" [14], where one is forced to consider worlds that are commonly known as impossible. One such example described in [4] is in the diving competition where the overall score is obtained by eliminating the top and bottom scores, then summing the remaining scores and multiplying by the degree of difficulty of the dive. Suppose there are seven judges, and each judge i gives a score x_i. The unweighted rule is as follows: $f(X) = d \times g(X)$, where d is the degree of difficulty of the dive, and

$$g(X) = \sum_{i=1}^{7} x_i - \max\{x_1, ..., x_7\} - \min\{x_1, ..., x_7\} \tag{7}$$

The weighted rule according to the FW formula not only has to evaluate $f(X)$ when X is over the full set $I = \{x_1, x_2, ..., x_7\}$, but it also has to evaluate $f(X)$ when X is over a proper subset of I. It is strange that even if we know there are always exactly seven judges, we are forced to consider the situation where there are fewer judges. In addition, it is not clear how to properly handle the case where there are less than three judges.

IV. OUR APPROACH

The essential principle of a ranking rule is that the highest ranked object should have the shortest distance from the perfect point. The farther an object is from the perfect point the lower the rank it will have. The perfect point in our case is $A^+ = (1, ..., 1)$. Analogously, we can select $A^- = (0, ..., 0)$ as the negative perfect point. In general the farther an object is from the negative perfect point the higher the rank will be.

The rationale for our approach is that the strength of a score is in general proportional to its weight. In addition, we intend to use a full set evaluation, and directly combine weights with scores. The idea of our method is based on transformation. The simplest form of our method is linear transform.

A. Linear Transform

Suppose the underlying unweighted rule is $f(x_1, ..., x_n)$. Let the weights be $\theta = (\theta_1, ..., \theta_n)$, where $\theta_1, ..., \theta_n$ are all nonnegative and sum to one. For an object $X = (x_1, ..., x_n)$, the overall score of X under our weighted rule is

$$f_\theta(X) = f(\theta_1 x_1, ..., \theta_n x_n) \tag{8}$$

For example, if $f(x_1, ..., x_n) = (x_1^\alpha + ... + x_n^\alpha)^{\frac{1}{\alpha}}$, where $\alpha = 1, 2, ...\infty$, the linear transform method will generate the corresponding weighted rule $f_\theta(x_1, ..., x_n) = (\theta_1^\alpha x_1^\alpha + ... + \theta_n^\alpha x_n^\alpha)^{\frac{1}{\alpha}}$.

Our method preserves the equivalence property.

Theorem 1. *If f and g are equivalent in the unweighted case, then they are also equivalent in the weighted case.*

Proof: Let $f_\theta(X) < f_\theta(Y)$. Since f and g are equivalent, from Eq. (8), we have also $g_\theta(X) < g_\theta(Y)$. ■

It is obvious that our method also preserves the monotonic and strictly monotonic property.

Since in ranking, it is the relative overall score that counts, we can slightly modify the first desirable property D1 as follows:

- D1': If all weights in θ are equal, then the weighted rule f_θ is equivalent to the unweighted rule f.

The above change does not compromise the significance of the property.

Since our method is more tightly coupled with the underlying function, when we check whether a desirable property is satisfied by the weighted rule generated using our method, it is natural that we need to examine the underlying function themselves.

The property D1' is satisfied if the underlying function $f(X)$ is equivalent to the function $g(X) = f(cX)$ where c is a constant. In general we expect the function $g(X)$ to be equivalent to $f(X)$ if f is a reasonable ranking function. Because intuitively, $f(X) < f(Y)$ means X is closer to the negative perfect point $A^- = (0, 0, ..., 0)$ than Y, so we expect that cX is also closer to A^- than cY for any constant c (technically we can restrict $c < 1$).

For all the functions we mentioned so far, namely Eqs.(2),(3),(7), they all have the character of $f(cX) = cf(X)$ when c is some constant. Thus, they all satisfy property D1'.

Property D2, in our case, means the underlying function f has the character that a zero score will contribute nothing to the overall score. All functions we mentioned, except min, has this character.

Property D3 is true for any underlying function f that is continuous.

The overall score obtained by Eq. (8) is normally in the range $[0, \infty)$. It is sometimes desirable to have the overall score within a range $[0, 1]$ with maximum and minimum achieved by A^+ and A^-, respectively.

Definition 1. A ranking rule f is in standard format if $0 \le f(X) \le 1$, $f(A^-) = 0$ and $f(A^+) = 1$.

The following lemma shows a weighted rule obtained by our method can always be converted into a standard format.

Lemma 1. *Let $g_\theta(X) = \frac{f_\theta(X) - f_\theta(A^-)}{f_\theta(A^+) - f_\theta(A^-)}$. Then g_θ is in standard format and equivalent to f_θ.*

B. Other Transforms

Besides linear transform, other types of transforms might be useful and meaningful. For example, in Euclidean distance, a linear transform $x_i \to \theta_i x_i$ gives the weighted rule $\sqrt{\sum_{i=1}^n \theta_i^2 x_i^2}$. A square-root transform $x_i \to \sqrt{\theta_i} x_i$ would give another weighted rule $\sqrt{\sum_{i=1}^n \theta_i x_i^2}$. Both have been used in practice[15], [16], and both seem to be natural generalization of the unweighted Euclidean rule, but they are not equivalent.

Examples of other types of transforms include exponential transform $x_i \to x_i^{\theta_i}$. If the underlying function is a form of product, e.g. $f(X) = x_1 \cdot x_2 \cdots x_n$, then exponential transformation is more meaningful to use.

On the other hand, there are situations where there is no natural generation in the weighted case; min is one good example[4]. However, it is interesting to find that our method indeed generates a very natural weighted min rule.

C. Weighted Min Rule

At the first glance, it seems that a weighted minimum rule can not be generalized using our method. A direct application of the linear transform to minimum function gives the weighted rule $\min\{\theta_1 x_1, ..., \theta_n x_n\}$, which is obviously not valid. The reason is that the weighted min in this straightaway extension is counter-intuitive to the concept of min.

The reason why directly applying linear transform to min does not work is (it is supposed) that the scores with higher weight will get more emphasized, and that contradicts with the concept of min (however, it is in line with max). For example, suppose two competitors X with scores $(3, 4)$ and Y with $(5, 2)$. Under min, X gets 3, Y gets 2, and X is better than Y (the higher score the better or more similar). If weights are 0.3 and 0.7, then by directly applying linear transform, we have $(0.9, 2.8)$ for X and $(1.5, 1.4)$ for Y. So X has 0.9 and Y has 1.4 as result of min. In the weighted case, Y is better than X. This is wrong because more weight (0.7) has been assigned to the attribute-2, so Y should be even worse than X.

As explained at the beginning of this Section, we can take the minimum from the point of view of either A^- or A^+. It is true that

$$\min\{x_1, ..., x_n\} = 1 - \max\{1 - x_1, ..., 1 - x_n\} \tag{9}$$

This shows that if we view minimum from the point of A^+, we can use max instead of min. Since the concept of max has no conflict with weighting, we can apply our method to it and expect to generate a reasonable weighted rule.

Applying the linear transform, the point A^+ is transformed to $(\theta_1, ..., \theta_n)$. Thus, under our linear transform method, the weighted min rule is formulated as

$$f_\theta(X) = 1 - \max\{\theta_1 - \theta_1 x_1, ..., \theta_n - \theta_n x_n\} \tag{10}$$

TABLE II

AN EXAMPLE OF 2-DIMENSIONAL WEIGHTS ASSIGNMENT

X	w_1	w_2	...	w_m	
θ_1	x_{11}	x_{12}	...	x_{1m}	y_1
θ_2	x_{21}	x_{21}	...	x_{11}	y_2
...	...	...	...	...	...
θ_n	x_{n1}	x_{n2}	...	x_{nm}	y_n

Using Lemma 1, we convert the above into a standard format of the *weighted min rule* as follows:

$$f_\theta(X) = 1 - \frac{\max\{\theta_1 - \theta_1 x_1, ..., \theta_n - \theta_n x_n\}}{\max\{\theta_1, ..., \theta_n\}} \qquad (11)$$

D. Comparison of Weighted Min Rules

We now compare our weighted min rule with the one obtained by the FW formula. It is instructive to also include some other weighted min function in the comparison, like the one given by Dubois and Prade [17] which has been used in fuzzy set theory:

$$f_\theta(X) = \min_{i\in\{1,...,n\}}\{\max\{1-(\frac{\theta_i}{M}), x_i\}\} \qquad (12)$$

where $M = \max\{\theta_1, ..., \theta_n\}$. All three rules satisfy properties D1-D3, and monotonic. But Eq. (12) is not strictly monotonic while the other two are.

Fig. 1 is a 2-dimensional visualization of the three rules, with weights $\theta_1 = 2/3, \theta_2 = 1/3$. All three rules have $min = x_1$ when $x_1 \leq x_2$. Dubois and Prades formula gives $min = 1-\frac{\theta_2}{\theta_1}$ when $x_1 \geq \frac{\theta_2}{\theta_1} \geq x_2$, which is rather undesirable. When $x_2 > x_1$, under FW, both x_1 and x_2 contribute to the overall score. Under linear transform, there is exactly one, either x_1 or x_2, contributing to the overall score.

V. OTHER RELATED PROBLEMS

A. Multi-dimensional Weights

So far, the weights we discussed are from 1-dimension only. In some applications, the weights can be formed in 2-dimension or higher. For example, in a search for similar events, one set of weights is given for the attributes of the event, another set of weights is given for the time of that event occurred (the more recent, the higher weight). Given a 2-dimensional type of weights, as shown in above Table II, we can evaluate the overall score for object X on individual score as follows:

$$f_{\theta w}(X) = f_{(\theta_1 w_1, ..., \theta_n w_m)}(x_{11}, ..., x_{nm}) \qquad (13)$$

We call the above *joint weighted rule*.

Another way is to have an aggregated score first (e.g., by aggregating over row), then evaluate on the aggregated score to get the overall score. That is,

$$f_{\theta w}(X) = f_{(\theta_1, ..., \theta_n)}(y_1, ..., y_n) \qquad (14)$$

where

$$y_i = f_{(w_1, ..., w_m)}(x_{i1}, ..., x_{im}) \qquad (15)$$

We name the above *aggregate weighted rule*.

TABLE III

AN EXAMPLE FOR THE MIN WEIGHTED RULE

X	w_1	w_2	y	X'	w_1	w_2	y
θ_1	0.8	0.6	0.64	θ_1	0.7	0.6	0.62
θ_2	0.6	0.8	0.6	θ_2	0.7	0.7	0.7

TABLE IV

AN EXAMPLE OF PARTIAL SET RANKING

	E1(5)	E2(5)	E3(4)	E4(1)	Average	Weighted average
X	10	6			8	8
Y	10		6		8	8.2
Z		9		5	7	8.3

There are three reasons that we want the joint method and the aggregate method to be equivalent: (1) for security and/or confidential purpose, sometimes individual data is not disclosed, only summary data (i.e. aggregated data) is shown; (2) in some cases of data warehousing [18], only summary data can be obtained quickly, to access individual data can be slow and expensive; and (3) even when both individual and summary data are available, it is usually more efficient to compute from the summary data compared to computing directly from individual data [19].

The two rules in fact, under linear transform method, are the same for many cases, including Eqs. (1),(3) and (9). We show the case of Eq. (9) in the following and omit the other two which are straightforward. For min rule Eq. (9), we have

$$\begin{aligned} y_i &= f_{(w_1,...,w_m)}(x_{i1}, ..., x_{im}) \\ &= 1 - \frac{\max\{w_1(1-x_{i1}), ..., w_m(1-x_{im})\}}{\max\{w_1, ..., w_m\}} \end{aligned}$$

Thus, the aggregate min rule is

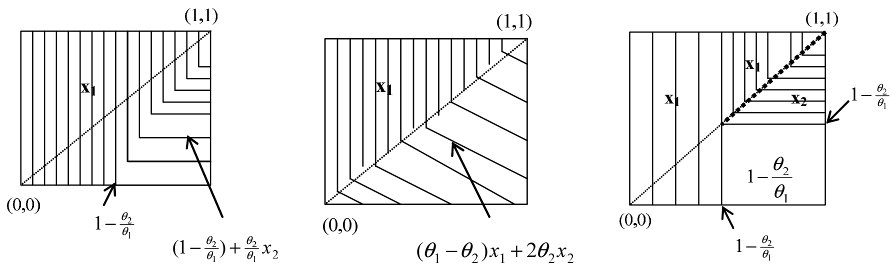
$$\begin{aligned} f_{\theta w}(X) &= f_{(\theta_1,...,\theta_n)}(y_1, ..., y_n) \\ &= 1 - \frac{\max\{\theta_1(1-y_1), ..., \theta_n(1-y_n)\}}{\max\{\theta_1, ..., \theta_n\}} \end{aligned}$$

It is not hard to verify that the above is equal to the joint min rule.

However, the two rules are not equivalent under the FW formula. One example of the weighted sum rule is shown in Section III. Table III gives another example for the case of the weighted min rule. Assume $w_1 = 0.6$, $w_2 = 0.4$, $\theta_1 = 0.6$, and $\theta_2 = 0.4$. Under FW, the aggregate min has overall score 0.608 for X, and 0.62 for X'. The joint min has overall scores 0.624 and 0.612 for X and X', respectively. This shows that the two rules, aggregate min and joint min, are not equivalent under the FW formula.

B. Partial Set Ranking

Another interesting ranking related problem is what we call partial set ranking problem. In real life, it often happens that a ranking is required to be completed within a short time period, but the number of evaluators available is much smaller than the number of objects that need to be evaluated. Examples are proposals submitted for a project, papers submitted to a conference, etc. Under the circumstance, it is usually very



(a) Linear transform (b) FW formula (c) Dubois and Prades formula

Fig. 1. 2D contours of three weighted min ranking rules

difficult, or even impossible, to ask every evaluator to evaluate every proposal. As a result, each object (proposal etc.) is normally evaluated by a small number of evaluators. In practice, we also like to assign weights to evaluators according to some measurement such as their experience or professional knowledge.

Without further elaboration, we can illustrate the problem caused by the partial set ranking by an example in Table IV, where there are four evaluators E1-E4, with weights 5, 5, 4 and 1, respectively. Intuitively, object X should be ranked at least as high as Y, because X's score of 6 is from an evaluator with higher weight. Both X and Y should be ranked higher than Z, as reflected by their average overall scores. However, the weighted average tells a different story that is contradicting to our intuition. We call this problem the partial ranking anomaly, where a missing score is not equal to a score of zero [20]. How to solve the partial ranking anomaly problem, or to a larger extent, how to conduct the partial ranking in a fair manner, remains to be seen.

VI. Conclusions

There is a large body of algorithms using ranking for filtering and sorting purpose. However, there is a lack of systematic way for combining weights into account in the ranking process. Fagin and Wimmers developed a formula which can take assigned weights into any ranking rules in a straightforward way. However, there are certain shortcomings we discovered in their method.

We proposed a new method that has a few desirable properties. For example, (1) it preserves equivalence; (2) it can be used for multi-dimensional weights; and (3) it is a natural extension to the underlying unweighted case. Our method is not a mechanical formula, it is a "conceptual" method in the sense that sometimes the underlying function needs to be taken into consideration as well. A good example is the construction of a weighted min function using our method. We also described an interesting but important related issue, the partial set ranking problem. Its solution remains to be found in the future.

References

[1] T. Seidl and H. Kriegel, "Optimal multi-step k-nearest neighbor search," in *Proceedings of 1998 ACM SIGMOD International Conference on Management of Data*, pp. 154–165, 1998.
[2] R. Fagin, "Fuzzy queries in multimedia database systems," in *Proceedings of 1998 ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1998.
[3] R. Fagin and E. Wimmers, "Incorporating user preferences in multimedia queries," in *Proceedings of 1997 International Conference on Database Theory*, pp. 247–261, 1997.
[4] R. Fagin and E. L. Wimmers, "A formula for incorporating weights into scoring rules," *Theoretical Computer Science*, vol. 239, no. 2, pp. 309–338, 2000.
[5] N. Schulz and I. Schmitt, "A survey of weighted scoring rules in multimedia database systems," Preprint 7, Fakultat fur Informatik, Universitat Magdeburg, 2002.
[6] H. Zimmermann, *Fuzzy Set Theory*. Kluwer Academic Publishers, 1996.
[7] E. Tsiporkova and V. Boeva, "Multi-step ranking of alternatives in a multi-criteria and multi-expert decision making environment," *Information Sciences*, vol. 176, no. 18, pp. 2673–2697, 2006.
[8] K. Arrow and H. Raynaud, *Social Choice and Multicriterion Decision-Making*. MIT Press, 1986.
[9] A. Gibbard, "Manipulation of voting schemes: A general result," *Econometrica*, vol. 41, no. 4, pp. 587–601, 1973.
[10] G. Bassett and J. Persky, "Rating skating," *Jounal of the American Statistical Association*, vol. 89, pp. 1075–1079, 1994.
[11] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, pp. 2270–2285, 2005.
[12] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.
[13] R. Fagin and Y. Maarek, "Allowing users to weight search terms in information retrieval," Tech. Rep. RJ 10108, IBM Research Lab, 1998.
[14] D. Lewis, *Counterfactuals*. Harvard University Press, 1973.
[15] A. Bimbo, M. Mugnaini, P. Pala, and F. Turco, "PICASSO: Visula querying by color perspective regions," in *Proceedings of the 2nd International Conference on Visual Information Systems*, pp. 125–131, 1997.
[16] M. Carey and L. Haas, "Towards heterogeneous multimedia information systems: The garlic approach," in *Proceedings of the 5th International Workshop on Research Issues in Data Engineering*, pp. 124–131, 1995.
[17] D. Dubois and H. Prade, "Weighted minimum and maximum operations in fuzzy set theory," *Information Sciences*, vol. 39, pp. 205–210, 1986.
[18] J. Widom, "Research problems in data warehousing," in *Proceedings of ACM Conference on Information and Knowledge Management*, 1995.
[19] I. Mumick, D. Quass, and B. Mumick, "Maintenance of data cubes and summary tables in a warehouse," in *Proceedings ACM SIGMOD International Conference on Management of Data*, pp. 100–111, 1997.
[20] J. W. Grzymala-Busse and M. Hu, "A comparison of several approaches to missing attribute values in data mining," in *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing*, 2001.