Journal of Intelligent Information Systems, 2, 421–441 (1993)

Retrieving Information by Fuzzification of Queries

RONALD R. YAGER RRY1@Iona.bitnet
Machine Intelligence Institute, Iona College, New Rochelle, NY 10801

HENRIK L. LARSEN HLL@RUC.DK
Department of Computer Science, Roskilde University, Postbox 260, DK-4000 Roskilde, Denmark

Abstract. The basic structure of an intelligent inquiring system is described. We discuss the process of generalization of requirements based on the use of fuzzy subsets. The concept of importance modification is introduced. A description of the construction of the envelope of potentially relevant items is presented. The process of criteria aggregation based on MOM and MAM operators is investigated. We discuss the use of a MAM operator to provide a ranking of the relevant items in the information base.

Keywords: information retrieval, fuzzy subsets, importance modification, aggregation, MOM and MAM operators

1. Introduction

An intelligent inquiring system is a kind of information retrieval (IR) system used to retrieve relevant information objects from an *information base*. The information base stores a collection of objects some of which are of interest to the current user. Each object in the information base can be seen to be made up of two components. The first component is the index and the second component is the body. The index usually consists of highly organized pieces of information that can be used to help identify and select the objects that may be relevant to a user. The body consists of information which may not be organized, but it contains the material that is of interest to the user. The fundamental problem in information retrieval is to find the subset of objects in the information base that is relevant to a given user. In a fuzzy information retrieval system, one can supply the list of relevant items with an ordering as to their potential interest to the user. Figure 1 shows a top-level view of the information retrieval system processes.

In the first step the user enters a request in terms of features of interest employing the keywords in the indexing system used to describe the objects. The information in this query is then used by the information retrieval system to select items that may be potentially relevant to the user. The final step is a process where the user looks at the items suggested by the system and decides the ultimate relevance of the items. This final step greatly reduces the burden of the information retrieval process, for it allows the *user* to look at the items selected

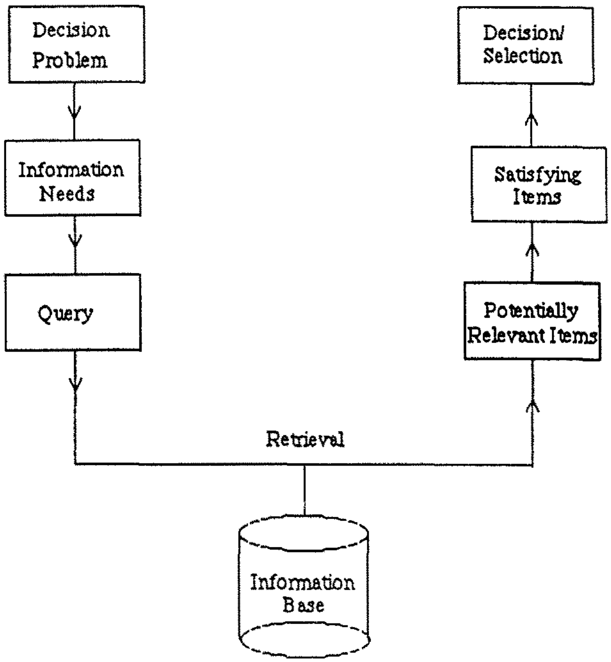

Figure 1. Information retrieval process.

and decide the ultimate relevance. This means that not all the knowledge about the decision has to be formalized in a manner that can be manipulated by the computer. The user must only supply the information that is used to search through the index.

As an example, we will consider the problem of selecting a house for purchase and assume that the user has access to an information base consisting of a collection of houses for sale. Here the user would express properties about the kind of house desired (price, size, location, etc.) in the query. The system would then search the information base and produce a listing of houses which closely match the user's request. This information contained in this listing could include text, more detailed information about the house as well as perhaps a picture of the house. The user then looks at this information and then decides which houses he wants to visit. In making this decision the user may use all kinds of subjective criteria which may be hard to quantify and not necessarily specified in his original query.

In this paper we shall describe an information retrieval system which uses fuzzy sets to help in the selection process, this kind of system can be viewed as an intelligent inquiry system. Figure 2, which is an expansion of the retrieval (filtering) process of Figure 1, illustrates the steps involved in the information retrieval process.

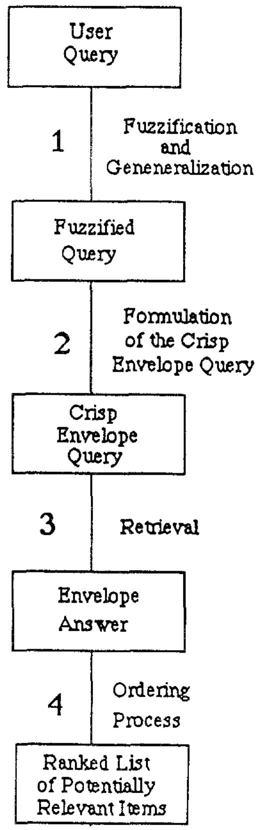

Figure 2. Information retrieval process.

In the first step the crisp information provided by the user is generalized with the aid of fuzzy sets. Using the index and a modified version of the requirements ("crisp envelope", step 2), we search through the information base (step 3), to find a subset of objects in the information base that can be considered as potentially relevant to the user. Step 3 can be based on an ordinary crisp querying language. The set of objects found in this step is called the "crisp envelope" answer. The final step in the process is a ranking of the elements in this crisp envelope which is then presented to the user.

2. Fuzzification of requirements

An important characteristic of many of the criteria supplied in a user query is that the needs they intend to represent are not crisp. If a person looking for a house indicates their desire to spend between \$100,000 and \$140,000 for the house, it is not the case that they will be totally uninterested in a house costing \$145,000. They may be less satisfied but not completely unsatisfied. The central observation here is that the boundary between a criteria being completely satisfied and not being satisfied is fuzzy rather than crisp. In building intelligent information retrieval systems we must take advantage of this fuzziness in the criteria. As we shall subsequently see, we use this fuzziness in two ways. First, we use it to help to extend the search space used for looking for potentially relevant items, in the example case, houses. In particular, we shall use it to help provide a query envelope which will crisply extend our search space. The second way we shall use this fuzzy characteristic is to provide an ordering (ranking) of the items in the information base indicating the degree to which the user query is satisfied.

The first step in the process of taking advantage of the lack of crispness in the user's requirements is to provide an appropriate representation of the requirement taking into account the noncrisp boundaries of the requirement. The most appropriate tool for representing this imprecise information is the fuzzy subset structure introduced by Zadeh (1965). As originally discussed by Zadeh, one application of the fuzzy subset is to represent concepts which have noncrisp boundaries. We recall that if X is a set, then a fuzzy subset A of X is characterized by a membership function $A(x) \in [0,1]$ such that for any element x in $X, A(x)$ indicates the degree to which the concept represented by A is satisfied by the element x. Considerable experience with fuzzy logic controllers (Lee, 1990) has indicated that trapezoidal type functions are a very effective class of functions for the representation of fuzzy concepts which have typical values.

Let $[a, b]$ be the range of values specified as being acceptable by a user for some attribute, such as price of a house. One can then *fuzzify* the range $[a, b]$ to be considered as "approximately $[a, b]$." Figure 3 shows the fuzzy subset A representing this generalization.

Formally, we can represent the fuzzy subset A by the following membership function:

$$\begin{aligned}
A(x) &= 0, && x \leq c,\\
A(x) &= \tfrac{x-c}{a-c}, && c < x < a,\\
A(x) &= 1, && a \leq x \leq b,\\
A(x) &= \tfrac{d-x}{d-b}, && b < x < d,\\
A(x) &= 0, && x \geq d.
\end{aligned}$$

In some situations, the extension of the original interval does not necessarily imply a decrease in satisfaction. For example if $[a, b]$ is the range for the price one is willing to pay for a house then paying anything less than a is completely satisfying. In this case, our fuzzy subset becomes as shown in figure 4.

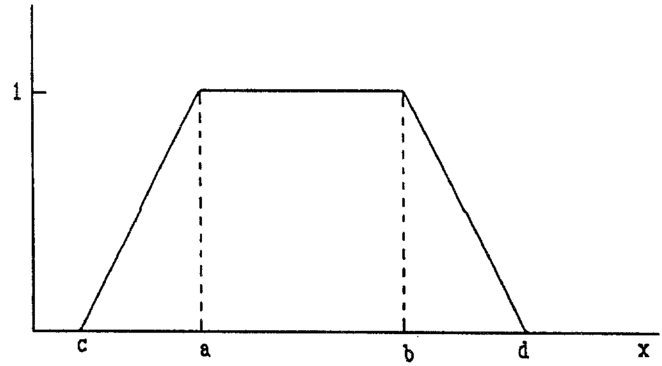

Figure 3. Representation of "approximately [*a*, *b*]."

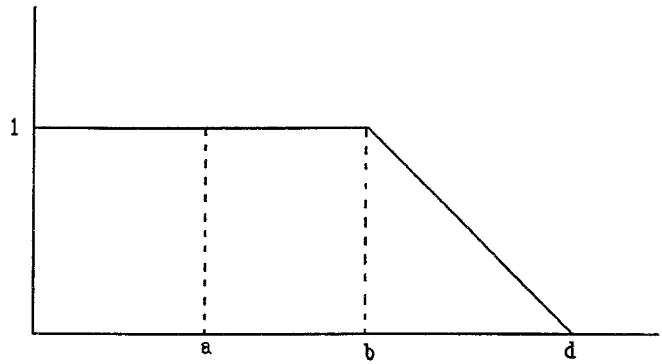

Figure 4. Unbalanced extension of [*a*, *b*].

In Figure 4 we see that in one direction, costs less then those in the range [*a*, *b*], the extension of the specifications provided by the user results in no loss of satisfaction while in the other direction there exists a decay of satisfaction, a fuzzy boundary manifested by a sloping line from *b* to *d*.

While many of the requirements specified by a user can be generalized (fuzzified) with the aid of fuzzy subsets some requirements are not amenable to generalization. For example, the desire to have a fireplace or two bathrooms are not easily fuzzified.

The above technique is most useful for variables which have numeric domains, however it can be extended to nonnumeric domains by introducing a similarity relation (Zadeh, 1971).

In addition to providing information about the values of the variables describing the objects, users can provide information about the importance of the various characteristics. Yager (1981; 1985; 1987) investigated the effect of importances of

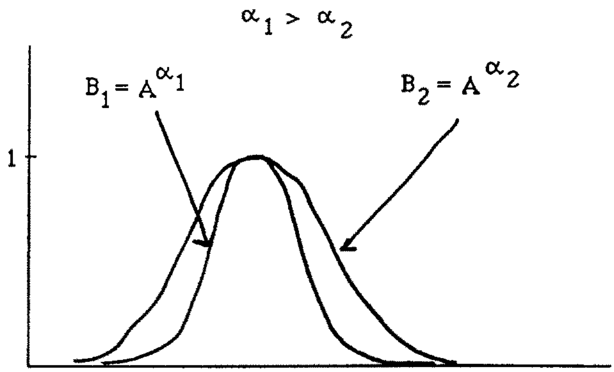

Figure 5. Effect of different importance weights.

criteria (concepts describing desired items) used in decision processes. In Yager (1978) it is suggested that if $\alpha \in [0, 1]$ is a measure of importance associated with a criterion represented by a fuzzy subset A on X then we can transform this into another fuzzy subset B such that

$$B(x) = (A(x))^{\alpha}.$$

Yager (1977) describes the effect of the inclusion of importance as being closely related to modification of the original concept by a linguistic hedge, such as "sort of." Figure 5 illustrates, for an exponential type fuzzy subset, the effect of the modification of a concept by importance. The principle effect of this operation is to cause, in the transformed fuzzy subset B, an increase in membership grade, a widening of bandwidth, as we decrease the importance of the original set A. Thus, if $\alpha_1 > \alpha_2$ then for all x

$$B_1(x) \leq B_2(x).$$

Recently Yager (1993a) has provided a general formal characterization of the association of importance with criteria. Assume A is a fuzzy subset and $\alpha \in [0, 1]$ is a measure of importance; then we can transform A into a fuzzy subset B such that

$$B(x) = g(\alpha, A(x))$$

where $g(\alpha, A(x))$ is a function having the following properties:

1. If $A_1(x) > A_2(x)$ then $g(\alpha, A_1(x)) \geq g(\alpha, A_2(x))$.
2. If $\alpha_1 > \alpha_2$ then $g(\alpha_1, A(x)) \leq g(\alpha_2, A(x))$.
3. $g(1, A(x)) = A(x)$.
4. $g(0, A(x)) = 1$.

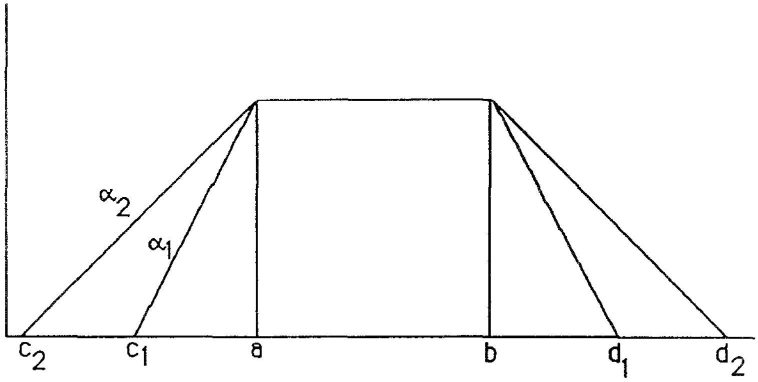

Figure 6. Including importance weights with trapezoidal membership function.

We see from condition two in the above that the effect of decreasing importance is to increase the width of the transformed fuzzy subset B, it does this by increasing the membership grade. We see that the less important a characteristic, the less restrictive the effective requirement expressed by the criterion. At the extreme, see condition 4, if something has zero importance, any object in the information base satisfies it.

Because of their inherent piecewise linearity we earlier suggested the use of trapezoidal type membership functions, as shown in Figure 3, to generalize (fuzzify) the crisp ranges, $[a, b]$, given by a user. Figure 6 illustrates a new proposed method for including importances associated with trapezoidal type membership functions.

In Figure 6, it is assumed that $\alpha_1 > \alpha_2, \alpha_2$ is less important. We see that if we denote $S_i = [c_i, d_i]$ as the support of the fuzzy subset, the less important a criterion, the wider the support. Using this approach for the inclusion of importance we may obtain an alternative representation of the fuzzy subset membership grade.

We recall that for the left fuzzy part $c_i < x < a$, we get

$$A_i(x) = \frac{x - c_i}{a - c_i}$$

Let

$$u_i = \frac{a - c_i}{a}$$

Thus u_i is the proportion of a the user is willing to give up, consider as dispensable, if the importance is α_i. u_i can be reviewed as a measure of the *flexibility*, how much it can be relaxed, as a function of the importance. We see that

$$c_i = a - u_i a$$

and thus for $x \in [a - u_i a, a]$

$$A_i(x) = \frac{x - a + u_i a}{u_i a} = 1 - \frac{1}{u_i}\left(1 - \frac{x}{a}\right).$$

Since $a - c_i$ decreases as the importance increases we see that the larger the importance the smaller the u_i. In particular, we need some function f,

$$f(\alpha) = u$$

such that as α increases, u decreases to help us get u directly from the importance value. Subsequently we shall say more about the construction of f.

In an analogous manner we get for the right fuzzy part $b < x < d_i$, that

$$A_i(x) = 1 - \frac{1}{v_i}\left(\frac{x}{b} - 1\right).$$

We note that the two fuzzy parts do not have to be symmetric. Again, we must obtain some function h which transforms importance values α_i into corresponding values of v_i. Both functions, g and h, must depend on the context.

While we have implicitly assumed the importances to be provided in terms of numeric values in the unit interval, this is neither necessary nor desirable. A preferred means of getting importance information from the user is in terms of a linguistic scale. A typical example of such a scale is the following:

I_1: very high
I_2: high
I_3: moderate
I_4: low
I_5: very low

Using such a scale, we again need functions to transform the importance value into a u_i, v_i pair. This transformation, while dependent on the domain, must be such that as the importance decreases the width of the support should increase.

An issue that must be considered in the construction of the approximate fuzzy ranges from the crisp ranges is how we get the c_i and d_i that are used for the extensions. A number of observations must be made regarding this issue.

Experience with man-machine systems tells us that we do not want too overburden the user with requests for to much information. A second observation that bears on the process of selection of the extended ranges results from the way in which the fuzzy information retrieval system is used. In using such a system, the user inputs his requirements and importances. This information then is transformed to a fuzzy subset which is used to *extend* the boundaries of the request by providing a *crisp envelope* for possible objects that satisfy the user (see Figure 7). This step allows us to use an ordinary (crisp) database system

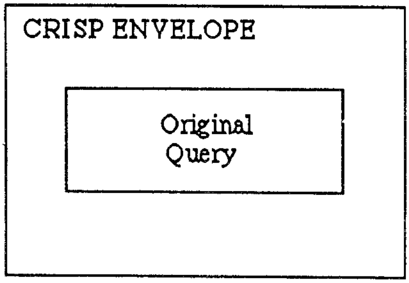

Figure 7. Crisp envelope of query.

to do the searching. We call this the extended crisp condition (see Figure 7). The objects in the database falling in the crisp envelope are ranked, where the ranking is done with the aid of the fuzzy subset. A given number of the highest ranked objects are presented to the user for their taking the appropriate action.

From this discussion we see that all that the fuzzification process does is help in providing the crisp extension of the search space and the final ranking of the objects. The issue of determining the values of c_i and d_i then becomes one of appropriately relaxing the original requirements and to obtain an appropriate ordering for the objects presented to the user.

Based upon the above observations, we feel that the values of c_i and d_i can be obtained via *default values*. A default value of course depends on the application domain and attribute range. These default values are obtained from the importance values through the function $f(\alpha)$ previously introduced.

We should emphasize a very important distinction between the problem of multicriteria based information retrieval (MCIR) and multicriteria decision making (MCDM). The important thing to highlight is that in multicriteria based information retrieval it is the user that makes the final decision as to what action to take (which houses to visit or buy) whereas in multicriteria decision making it is the computer that makes the final decision. The implication of this distinction is that the representational requirements in MCIR are less demanding then in MCDM. Since the final decision in MCIR is made by the user, this final decision can based upon additional knowledge and criteria which the user has but which he need not represent in the query presented to the system (see Figure 1). This characteristic type of man-machine interaction inherent in information retrieval systems greatly simplifies the burden on the system and the requirements that the user must formally express. Thus, once having the highest ranked objects he can bring to bear some very sophisticated reasoning and preference information in actually making the final decision. On the other hand, in MCDM systems the user must present to the computer all the criteria he desires to be considered which may be beyond the representational ability of the system.

3. Computation of Envelopes

A fuzzy query to an information/data base may be seen as a multicriteria, or multidimensional, characterization of the user needs with respect to the information base domain, the objects in the information base. From this point of view the response to a query is the set of objects in the search space that are the best instances of the concept. From another point of view, fuzzy querying is a decision problem. In this view, the query describes a multicriteria decision problem and the answer to the query is the set of objects in the information base that provides the best overall satisfaction of the decision problem.

In general, a query criterion induces a linear ordering of the objects in the information base. Thus, a multicriteria query determines a partial ordering of the objects. The global (linear) ordering needed for the answer is obtained by an aggregation of the satisfactions of the criteria. In this aggregation, each criterion is weighted with the importance of satisfying the criterion in recognizing the concept or in the decision problem. Thus, to answer a query, we must, in principle, compute the aggregated value for each object in the database. To do this in an efficient way, we should provide fuzzy indexes organized as inverted files. However, in fuzzy querying in existing database systems, only access through the crisp querying language of the systems is possible. There are two de facto standards to consider in crisp querying languages, namely SQL (structured query language) (Date, 1986) for the relational database model, and CCL (common command language) (Salton, 1989) for document bases ("text bases"). Both languages support a crisp Boolean formulation of query criteria. We will address the problem of fuzzy querying through such a crisp language as SQL.

The basic idea of our approach is to compute a so-called envelope for the fuzzy query. An envelope is a *crisp query* such that the answer to the fuzzy query is a subset of the answer to the envelope; the envelope defines the query used for accessing the database.

Assume our information base consists of a set of objects X. Let $A_1, \ldots, A_P$ be a set of criteria, fuzzy subsets, required by the user. Our procedure is to transform these fuzzy criteria into crisp criteria, $D_1, \ldots, D_p$ and then use a SQL language to search the information base to find the subset of objects Y of X that satisfy this crisp query, we call the subset Y *the envelope* answer. Once having the subset Y we can use the original fuzzy criteria to rank the objects. The advantage of using this envelope is to reduce the search burden as well as allow the use of available SQL search techniques. In this section we shall concentrate on the issue of formulating the envelope.

Consider a fuzzy criteria A represented by a trapezoidal fuzzy subset of the type shown in Figure 8.

We shall obtain from the fuzzy criteria A a related crisp criteria D as shown in Figure 9.

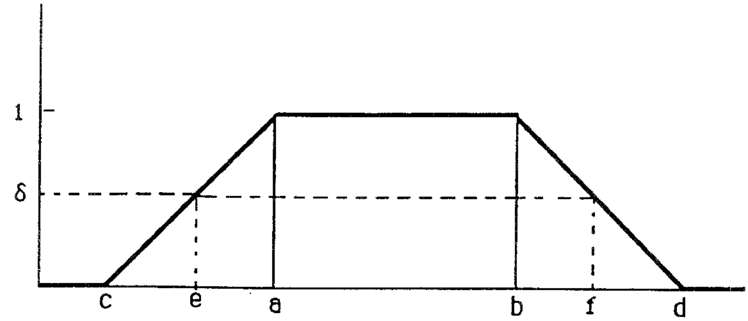

Figure 8. Fuzzy criteria.

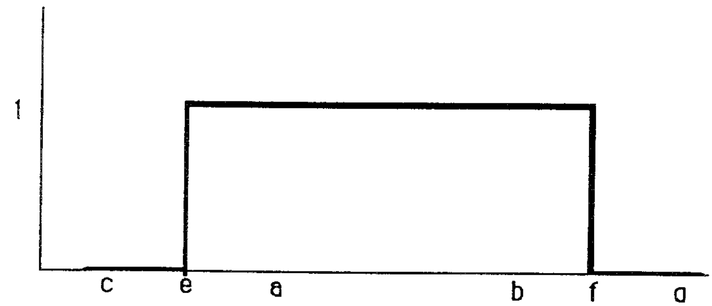

Figure 9. Crisp criteria D.

The membership grade of this crisp criteria is

$$D(x) = 1 \qquad \text{for } e \leq x \leq f,$$
$$D(x) = 0 \qquad \text{otherwise.}$$

We obtain the bounds, e and f, of the criteria D from A in the following manner. Let δ, called the threshold for satisfaction of a query, be a number in the unit interval, for example we can select $\delta = .5$. From the membership function of A we obtain e and f as the δ level points, thus, D is the δ-level set of A. In particular, in Figure 8

$$A(e) = \delta = \frac{e - c}{a - c}.$$

Therefore

$$e = \delta(a - c) + c$$

and

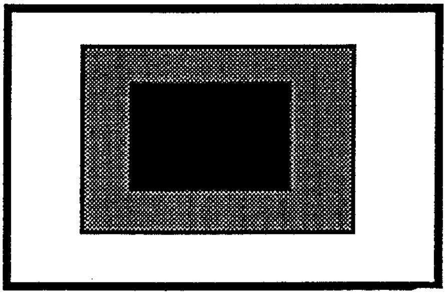

Figure 10. Construction of the envelope.

$$A(f) = \delta = \frac{d - f}{d - b}$$

hence

$$f = d - \delta(d - b)$$

In a similar manner we transform all the fuzzy criteria, A_i, into crisp criteria D_i. We then calculate the envelope Y of X as the subset of objects y such that

$$D_i(y) = 1$$

for all i. That is, the envelope is the subset of elements that satisfy all the transformed criteria.

Figure 10 illustrates the construction of the envelope for a simple two-dimensional query. The area contained within the thick black lines constitutes our database of objects. The black rectangle is meant to indicate the subset of elements in the database that satisfy the *original* crisp query specified the user. The black rectangle plus the gray area indicates the *envelope* obtained by the fuzzification of the original query followed by the crispization to D. It is on the elements in this envelope that we shall subsequently provide a ranking as to their potential usefulness to the user. Thus, via the process of fuzzification followed by crispization we have essentially extended the domain of objects we shall consider.

It should be pointed out that using our approach, the importances of the criteria have been factored into the process of obtaining the envelope.

Figure 11 illustrates the effect of importance inclusion on the construction of the envelope. Retrieved set II shows that if we *decrease* the importance of the criteria representated along the abscissa coordinate we effectively *increase* the size of the envelope in that direction. This effect results because in decreasing the importance we have increased the values of the membership grade (see figure 6) and essentially relaxed the criterion and thus allowed more objects in the information base to pass through our filter.

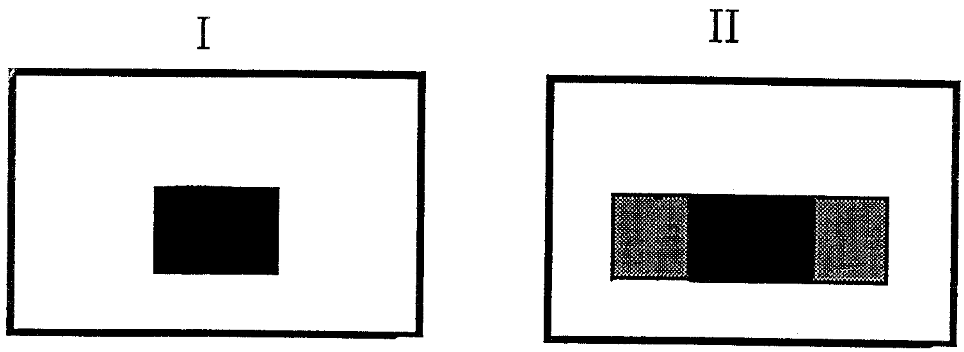

Figure 11. Effect of change of importance.

As a result of this step we have a collection of objects, the envelope answer, which contains the subset of potentially relevant objects to the user. The next step is to rank these objects, to do this we use an aggregation type process and use the original fuzzy criteria, the A_i, to provide the scores used in this aggregation process.

4. On the aggregation of criteria using MOM and MAM operators

In fuzzy logic systems, the basic aggregation operations are performed by the logical connectives *and* and *or* which provide pointwise implementations of the intersection and union operations. It has been well-established in the literature (Dubois and Prade 1985) that the appropriate characterization of these operators in the multivalued logic environment are the triangular norm operators. The t-norm operator provides the characterization of the *and* operator. It is a mapping $T : [0,1] \times [0,1] \rightarrow [0,1]$ having the following properties:

$\mathbf{T_1}$: $T(a,b) = T(b,a)$ **(commutativity)**
$\mathbf{T_2}$: $T(a,b) \geq T(c,d)$ for $a \geq c$ and $b \geq d$ **(monotonicity)**
$\mathbf{T_3}$: $T(a,T(b,c)) = T(T(a,b),c)$ **(associativity)**
$\mathbf{T_4}$: $T(a,1) =$ **(*Andness* boundary condition)**

Its dual, the t-conorm, characterizes, the *or* operator. It is a mapping S: $[0,1] \times [0,1] \rightarrow [0,1]$ having properties

$\mathbf{S_1}$: $S(a.b) = S(b,a)$ **(commutativity)**
$\mathbf{S_2}$: $S(a,b) \geq S(c,d)$ for $a \geq c$ and $b \geq d$ **(monotonicity)**
$\mathbf{S_3}$: $S(a,S(b,c)) = S(S(a,b),c)$ **(associativity)**

S_4 : $S(a, 0) = a$ *(Orness* boundary condition)

Examples of t-norm are

$$T(a, b) = \mathrm{Min}(a, b),$$
$$T(a, b) = 1 - \mathrm{Min} + [1, ((1 - a)P + (1 - b)P))^{1/p}], \quad p > 0,$$
$$T(a, b) = ab.$$

The corresponding t-conorms are

$$S(a, b) = \mathrm{Min}(a, b),$$
$$S(a, b) = 1 - \mathrm{Min} + [1, (1P - bP)^{1/p}], \quad p > 0,$$
$$S(a, b) = a + b - ab.$$

A fundamental property associated with t-norms is that

$$T(a_1, \ldots a_n) \geq T(a_1, \ldots a_n, a_{n+1}).$$

Yager (1993b, c, d) called this property the ***anti-monotonicity in cardinality*** property. This property says that as more conditions are *required* to be satisfied by an *and* aggregation, the overall satisfaction can't increase. The related property associated with the t-conorms is

$$S(a_1, a_2, \ldots, a_n) \leq S(a_1, \ldots, a_n, a_{n+1}).$$

Yager (1993b) called this the ***monotonicity in cardinality*** property. As more conditions are *allowed* to contribute to an *or* aggregation the overall satisfaction cannot decrease.

It can be shown that 1 acts as an identity element in the t-norm aggregation while 0 acts as an identity element in the t-conorm aggregation.

We shall now introduce a more generalized class of operators which provide for generalized formulations of the *and* and *or* aggregations. A **bag** (or multiset) drawn from a set X is any collection of elements, each of which is contained in X. A bag allows multiple copies of the same element. In the following, we shall restrict ourselves by constraining X to be the unit interval, I. We shall let U^I be the subset of all bags drawn from I. Assume $B =< b_1, \ldots, b_n >$ is a bag cardinality n. We say that B is in *fundamental form* if the elements are indexed such that $b_i \geq b_j$ if $i > j$. If A and B are two bags of the same cardinality and when expressed in fundamental form we have the property that $a_i \geq b_i$ for all i, then we shall denote this as $A \geq B$. If A and B are two bags we shall denote the **sum** of the bags by $D = A \oplus B$ where D is the bag consisting of the members of both A and B.

Example. Assume $A = \langle .2, .4, .8, 1, 1\rangle$ abd $B = \langle 0, .4, .6, 1\rangle$. Then

$$D = A \oplus B = \langle 0, .2, .4, .4, .6, .8, 1, 1, 1\rangle$$

DEFINITION. A bag mapping H: $U^I \rightarrow I$ is called a **MAM** (monotonic anti-monotonic) (Yager, 1993 b) operator if it has the following properties:

MA.1 If $A \geq B$ then $H(A) \geq H(B)$ (monotonicity in values).
MA.2 If $D = A + \langle 1 \rangle$ then $H(A) = H(D)$ (one identity element).

As in the case of all bag mappings, the MAM operator is commutative with respect to its arguments. We call this a generalized symmetry condition. It can easily be seen that if the following condition is satisfied

PA.1 If $D = A \oplus B$ then $H(A) \geq H(D)$ (antimonotonicity in cardinality)

We see that the MAM operator is a generalization of the t-norm operator and can be viewed as a generalized *and* aggregation. In particular, the boundary condition of the t-norm, T_4, has been weakened and included as MA.2. The associativity property of the t-norm, T_3, has been eliminated. The condition MA.1 is essentially the monotonicity condition, T_2, and the commutativity condition, T_1, is implicitly satisfied by all bag mappings.

Yager (1993 b) has shown that the t-norm operator is a special class of MAM operators.

For any bag A, since $A = \Phi \oplus A$ (Φ being the empty bag, $\langle\rangle$), from condition PA.1 it follows that $H(\Phi) \geq H(A)$. Based upon this condition we shall say that H is *normal* if $H(\Phi) = 1$. We shall say that the MAM operator is *regular* if $H(\langle 0 \rangle) = 0$. It can be seen that if H is regular, then for any bag E having zero as one of its arguments, it is the case that $H(E) = 0$. We can express $E = \langle 0 \rangle \oplus B$ where B is the bag E less the element 0. From PA.1 $H(E) \leq H(\langle 0 \rangle) \leq 0$; hence $H(E) = 0$.

Yager (1993 b) introduced a family of operators which generalized the t-conorm operator and thus induce generalized *orlike* aggregations. This class of bag mappings are called MOM (monotonic on monotonic) operators.

DEFINITION. A bag mapping $G: U^I \rightarrow I$ is called a MOM (monotonic on monotonic) operator (Yager [12]) if it has the following properties:

MO.1 If $A \geq B$, then $G(A) \geq G(B)$ (monotonicity in values).
MO.2 If $D = A + \langle 0 \rangle$ then $G(A) = G(D)$ (zero identity element).

It can be shown that the following property holds for MOM operators

PO.1 If $D = A \oplus B$ then $G(D) \geq G(A)$ (monotonicity in cardinality)

It can be shown that the MOM operator is a generalization of the t-conorm operators [13]. We say that a MOM operator G is *normal* if $G(\Phi) = 0$. We shall call G *regular* if $G(\langle 1\rangle) = 1$. It can be shown that if G is regular and F is any bag containing 1 as an element, then $G(E) = 1$.

There exists a De Morgan–like duality between the MOM and MAM operators.

DEFINITION. Assume $F : U^I \to I$ is any bag mapping which takes bags from the unit interval into the unit interval. We define the *dual of* F, $\hat{F}$, as $\hat{F}(A) = 1 - F(A^c)$. In this definition A^c is the complement of A defined to be the bag consisting of the elements b_i, where $b_i = \bar{a}_i = 1 - a_i$.

The following theorems proven in Yager [12] show that MAM and MOM operators are duals of each other.

THEOREM. Assume G is a MOM operator. Then $\hat{G}$ is a MAM operator.

THEOREM. Assume H is a MOM operator. Then $\hat{H}$ is a MAM operator.

As we have already indicated the t-norm and t-conorm are MAM and MOM operators. In the following we shall introduce another important class of MAM and MOM operators. Assume g is a mapping from the real line into the unit interval g: $R \to I$, having the property

$$g(x) \geq g(y) \quad \text{if } x > y \quad \text{(monotonic nondecreasing)}.$$

Let A be a bag drawn from the unit interval and let us define

$$\mathrm{Sum}(A) = \sum_{i=1}^{n} a_i,$$

where n is the cardinality of A. We shall call this the bag sum. It can be shown that the bag mapping

$$G(A) = g(\mathrm{Sum}(A))$$

is a MOM operator.

Using our duality theorem we can show that

$$H(A) = h(\mathrm{Sum}(A^c))$$

where h is a mapping h: $R \to I$ which is monotonically nonincreasing. If $x > y$ then $h(x) \leq h(y)$ antimonotone) is a MAM operator.

We shall now introduce the concept of weighted bags and the related ideas of weighted MOM and MAM operators. These operators will generalize the concept of weighted aggregations which can be used to associate importances with the aggregates.

DEFINITION. A *weighted bag* is a bag A whose elements are tuples (w_i, a_i) where both w_i and a_i are drawn from the unit interval. For each tuple in A we shall call a_i *the value of the tuple* and w_i *the weight of the tuple*.

Let A and B be two weighted bags of the same cardinality. Assume to each tuple (w_i, a_i) in A there exists a corresponding tuple (w_i, b_i) in B where $w_i' = w_i$. If it is the case that $a_i \geq b_i$ for all i we shall say that $A \geq_V B$. Assume that for each tuple (w_i, a_i) in A, there exists a corresponding tuple (w_i', b_i) in B where $a_i = b_i$. If it is the case that $w_i \geq w_i'$ we shall say that $A \geq_w B$. We now extend the MOM operator to act on weighted bags.

DEFINITION. A bag mapping G_w: $U^{I \times I} \to I$ is called a *weighted MOM operator* if it has the properties:

WO. 1 If $A \geq_v B$, then $G_w(A) \geq G_w(B)$.
WO. 2 If $D = A \oplus B$, then $G_w(D) \geq G_w(A)$.
WO. 3 $G_w(A \oplus \langle(1,0)\rangle) = G_w(A)$.
WO. 4 If $A \geq_w B$, then $G_w(A) \geq G_w(B)$.

We now turn to the corresponding definition for weighted MAM operators.

DEFINITION. A bag mapping H_w: $U^{I \times I} \to I$ is called a *weighted MAM operator* if it has the properties.

WA. 1 If $A \geq_v B$, then $H_w(A) \geq H_w(B)$.
WA. 2 If $D = A \oplus B$, then $H_w(D) \geq H_w(A)$.
WA. 3 $H_w(A \oplus \langle(1,1)\rangle) = H_w(A)$.
WA. 4 If $A \geq_w B$, then $H_w(A) \leq H_w(B)$.

The introduction of these weighted bags along with the weighted mappings will provide a framework for generalizing the idea of weighted aggregations. The weighted operators allow us to consider aggregations where the elements differ in importance.

Assume A is a weighted bag. We define the *right-pointed complement* of this bag as ARC where every element (w_i, a_i) in A is replaced by $(w_i, \overline{a}_i)$, with $\overline{a}_i = 1 - a_i$. We now define the concept of the dual of a weighted bag mapping in a similar way to which we defined it in the earlier section.

DEFINITION. Assume F_w: $U^{I \times I} \to I$ is a weighted bag mapping. We define the dual of $F_w, \hat{F}_w$, as

$$\hat{F}_w(A) = 1 - F(ARC).$$

We now indicate that weighted MOM and MAM operators are duals of each other.

THEOREM. Assume G_w is a weighted MOM operator then $\hat{G}_W$ is a weighted MAM operator.

THEOREM. Assume H_w is a weighted MAM operator then $\hat{H}_w$ is a weighted MOM operator.

We shall now look at some classes of weighted bag operators, the proofs of all the results in this section are found in Yager (1993b). We first define a class of weighted MOM operators based on the logical connectives.

THEOREM. If S is a t-conorm and T a t-norm then $G_w(A) = S_i(T(w_i, a_i))$ is a weighted MOM operator.

The following are examples of this class of operators:

1. $G_w(A) = \text{Max}_i[W_i \wedge a_i]$ $(\wedge = \text{Min})$.
2. $G_w(A) = \text{Max}_i[w_i a_i]$.
3. $G_w(A) = \text{Min}_i[1, \sum_{i=1}^{n} w_i a_i]$.
4. $G_w(A) = \text{Min}_i[1, \sum_{i=1}^{n} w_i \wedge a_i]$.
5. $G_w(A) = 1 - \prod_{i=1}^{n} \overline{w}_i \overline{a}_i$.

Using the duality relationship we obtain the following theorem.

THEOREM. If S is a t-conorm and T a t-norm then $H_w(A) = T_i(S(\overline{w}_i, a_i))$ is a weighted MAM operator.

Among this class of operators are

$$H_w(A) = \text{Min}_i[(\overline{w}_i \vee a_i)](\vee = \text{Max}),$$

$$H_w(A) = \prod_{i=1}^{n} (\overline{w}_i + w_i\ a_i).$$

The following theorem introduces another class of weighted MOM operators.

THEOREM. Assume g is a mapping from the real line into the unit interval g: $R \to I$ such that $g(x) \geq g(y)$ if $x > y$ (monotonic nondecreasing) then $G_w(A) = g(\sum_{i=1}^{n} w_i a_i)$ is a weighted MOM operator. We shall call these **additive weighted MOM operators.**

Using our duality theorem we can obtain a class of dual additive weighted MAM operators, generalized weighted *and* aggregators.

THEOREM. Assume h is a mapping from the real line into the unit interval $h\colon\ R \to I$ such that $h(x) \leq h(y)$ if $x > y$ (monotonic nonincreasing) then $H_w(A) = h\left(\sum_{i=1}^{n} w_i\overline{a}_i\right)$ is a weighted MAM operator.

We see that G_w essentially increases as $\sum w_i a_i$ increases, that is as more important elements are satisfied we increase our satisfaction. On the other hand, H_w works by considering the unsatisfaction, $\overline{a}_i$, and as more important aggregates are not satisfied we decrease the total aggregate score.

5. Ranking objects in the retrieval system

In the preceding, we discussed the issue of criteria aggregation. We shall now specialize this to the ranking of objects for an information retrieval system. We discussed two classes of aggregation, MOM and MAM operators. We recall that the MOM operator is a generalized *andlike* operator. In information retrieval systems we see the criteria specified by the user as being connected by an *andlike* operator, assuming the user generally wants all the criteria satisfied. That is, a person desires to obtain further information about houses in a certain price range *and* in a particular location *and* having certain amenities. Thus, the appropriate family of operators are the MAM operators. We have described two classes of MAM operators, the t-norm and the additive. We shall consider here the additive class. We recall that the weighted additive formulation is

$$\text{Score}(y) = h\left(\sum_{i=1}^{n} w_i(1 - A_i(y))\right).$$

However, in constructing the criteria function, as discussed in Section 2, we can implicitly include the importance weights, thus we can use an alternative version

$$\text{Score}(y) = h\left(\sum_{i=1}^{n}(1 - B_i(y))\right),$$

where B_i are the importance modified criteria.

In addition, since all we need is a ranking of the objects we can use any monotonically nonincreasing function for h. We shall use the exponential function, thus

$$\text{Score}(y) = \text{Exp}\left(-\sum_{i=1}^{n}(1 - B_i(y))\right) = \text{Exp}\left(-\sum_{i=1}^{n}1\right)\text{Exp}\left(\sum_{i=1}^{n}B_i(y)\right).$$

Since the term $\text{Exp}(-\sum_{i=1}^{n}1) = \text{Exp}(-n) > 0$ plays no role in determining the ranking we can use

$$\text{Score}(y) = \text{Exp}\left(\sum_{i=1}^{n} B_i(y)\right).$$

Furthermore, since taking the log of score does not effect the ordering we can use

$$\text{Score}(y) = \sum_{i=1}^{n} B_i(y)$$

Thus, the ranking of the objects can be obtained by simply ordering them according to the total of their scores of the importance modified criteria.

6. Conclusion

We have discussed a number of issues related to the construction of a weighted multicriteria information retrieval system that uses fuzzy subsets as mechanism to allow for the flexible evaluation of user requirements. We have discussed the potential use of MAM and MOM operators as a tool for the aggregation of user requirements.

7. Notes

1. In this paper we use the term information retrieval in a more colloquial sense than the narrow definition used in the technical literature where it usually refers to textual retrieval involving keywords. We refer the reader to Miyamoto and Miyake [4] for an interesting discussion on the distinction between information retrieval systems and other closely related technologies such as database management systems and query response systems.
2. The formulation of raising the fuzzy subset to a power to include importances has the drawback of not enabling us to change membership that are zero in the original fuzzy subset. This drawback can be circumvented if we restrict the use of this formulation to fuzzy subsets which are of the exponential type.
3. I would like to thank M. Zemankova for suggesting this interpretation.

References

Date, C.J. (1986). *An Introduction to Database Systems*, Reading, MA: Addison Wesley.

Dubois, D. and Prade, H. (1985). A review of fuzzy sets aggregation connectives. *Information Sciences*, 36, 85–121.

Lee, C.C. Fuzzy logic in control systems: fuzzy logic controller, Part 1. *IEEE Trans. on Systems, Man and Cybernetics, 20*, 404–418.

Miyamoto, S. and Miyake, T. On fuzzy information retrieval. *Japanese Journal of Fuzzy Theory and Systems, 3*, 93–105.

Salton, G. (1989). *Automatic Text Processing*, Reading, MA: Addison-Wesley.
Yager, R.R. (1977). Multiple objective decision making using fuzzy sets. *International Journal of Man-Machine Studies, 9*, 375–382.
Yager, R.R. (1978). Fuzzy decision making using unequal objectives. *Fuzzy Sets and Systems, 1* 87–95.
Yager, R.R. (1981). A new methodology for ordinal multiple aspect decisions based on fuzzy sets. *Decision Sciences, 12*, 589–600.
Yager, R.R. (1985). Forms of multi-criteria decision functions and preference information types. In *Approximate Reasoning in Expert Systems*. Gupta, M.M., Kandel, A., Bandler, W., and Kiska, J. B. eds. Amsterdam: North-Holland, 167–177.
Yager, R.R. (1987). A note on weighted queries in information retrieval systems, *Journal of American Society of Information Sciences*, 38, 23–24.
Yager, R.R. (1993a). On weighted median aggregation, Technical Report # MII-1317, Machine Intelligence Institute, Iona College, New Rochelle, NY.
Yager, R.R. (1993b). A unified approach to aggregation based upon MOM and MAM operators. Technical Report # MII-1316, Machine Intelligence Institute, Iona College.
Yager, R.R. (1993c). MAM and MOM operators for aggregation, *Information Sciences, 69*, 259–273.
Yager, R.R. (1993d). Toward a unified approach to aggregation in fuzzy and neural systems. Proceedings World Conference on Neural Networks, Portland, OR, Vol. II, 619–622.
Zadeh, L.A. (1965). Fuzzy sets, *Information and Control, 8*, 338–353.
Zadeh, L.A. (1971). Similarity relations and fuzzy orderings. *Information Sciences, 3*, 177–200.