Towards Fast Algorithms for Processing Type-2 Fuzzy Data: Extending Mendel's Algorithms From Interval-Valued to a More General Case

Vladik Kreinovich and Gang Xiang

***Abstract*—It is known that processing of data under general type-1 fuzzy uncertainty can be reduced to the simplest case – of interval uncertainty: namely, Zadeh's extension principle is equivalent to level-by-level interval computations applied to α-cuts of the corresponding fuzzy numbers.**

However, type-1 fuzzy numbers may not be the most adequate way of describing uncertainty, because they require that an expert can describe his or her degree of confidence in a statement by an exact value. In practice, it is more reasonable to expect that the expert estimates his or her degree by using imprecise words from natural language – which can be naturally formalized as fuzzy sets. The resulting *type-2* fuzzy numbers more adequately represent the expert's opinions, but their practical use is limited by the seeming computational complexity of their use. In his recent research, J. Mendel has shown that for the practically important case of interval-valued fuzzy sets, processing such sets can also be reduced to interval computations. In this paper, we show that Mendel's idea can be naturally extended to arbitrary type-2 fuzzy numbers.

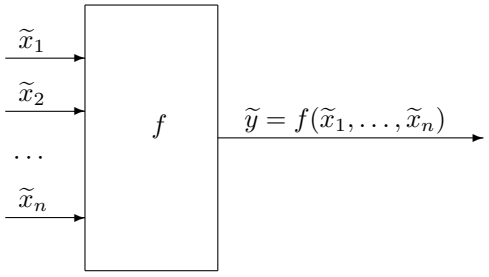
I. Why Data Processing and Knowledge Processing Are Needed in the First Place

Some quantities y we can simply directly measure. For example, when we want to know the current state of a patient in a hospital, we can measure the patient's body temperature, blood pressure, weight, and many other important characteristics. In some situations, we do not even need to measure: we can simply ask an expert, and the expert will provide us with an (approximate) value $\widetilde{y}$ of the quantity y.

However, many other quantities of interest are difficult or even important to measure or estimate directly. Examples of such quantities include the amount of oil in a given well or a distance to a star. Since we cannot directly measure the values of these quantities, the only way to learn some information about them is: to measure (or ask an expert to estimate) some other easier-to-measured quantities $x_1, \ldots, x_n$, and then to estimate y based on the measured values $\widetilde{x}_i$ of these auxiliary quantities x_i.

For example, to estimate the amount of oil in a given well, we perform *seismic* experiments: we set up small explosions at some locations and measure the resulting seismic waves at different distances from the location of the explosion. To find the distance to a faraway star, we measure the direction to the star from different location on Earth (and/or at different seasons) and the coordinates of (and the distances between) the locations of the corresponding telescopes.

To estimate the value of the desired quantity y, we must know the relation between y and the easier-to-measure (or easier-to-estimate) quantities $x_1, \ldots, x_n$. Specifically, we want to use the estimates of x_i to come up with an estimate for y. Thus, the relation between y and x_i must be given in the form of an *algorithm* $f(x_1, \ldots, x_n)$ which transforms the values of x_i into an estimate for y. Once we know this algorithm f and the measured values $\widetilde{x}_i$ of the auxiliary quantities, we can estimate y as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.



In different practical situations, we have algorithms f of different complexity. For example, to find the distance to star, we can usually have an explicit analytical formula coming from geometry. In this case, f is a simple formula. On the other hand, to find the amount of oil, we must numerically solve a complex partial differential equation. In this case, f is a complex iterative algorithm of solving this equation.

In the case when the values x_i are obtained by measurement, this two-stage process does involve measurement. To distinguish it from *direct* measurements (i.e., measurements which directly measure the values of the desired quantity), the above two-stage process is called an *indirect* measurement.

When the inputs come from measurements – i.e., constitute *data* – the computational part of the corresponding procedure is called *data processing*. When the inputs come from experts – i.e., constitute *knowledge* – the computational part of the corresponding procedure is called *knowledge processing*.

II. Need to Take Uncertainty Into Account

In the case of data processing, we start with measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$. Measurements are never exact. There is a non-zero difference $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ between the

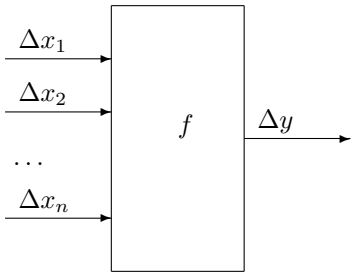
Vladik Kreinovich is with the Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968 (email vladik@utep.edu). Gang Xiang is with the Philips Healthcare Informatics (email gxiang@acm.org).

This work was supported in part by NSF grants HRD-0734825, EAR-0225670, and EIA-0080940, by Texas Department of Transportation contract No. 0-5453, by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and by the Max Planck Institut für Mathematik.

(approximate) measurement result $\widetilde{x}_i$ and the (unknown) actual value x_i of the i-th quantity x_i. This difference is called the *measurement error*. The result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of applying the algorithm f to the measurement results $\widetilde{x}_i$ is, in general, different from the result $y = f(x_1, \ldots, x_n)$ of applying this algorithm to the actual values x_i. Thus, our estimate $\widetilde{y}$ is, in general, different from the actual value y of the desired quantity: $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y \neq 0$.

In many practical applications, it is important to know not only the desired estimate for the quantity y, but also how accurate this estimate is. For example, in geophysical applications, it is not enough to known that the amount of oil in a given oil files is about 100 million tons. It is important to know how accurate is this estimate. If the amount is 100 $\pm$ 10, this means that the estimates are good enough, and we should start exploring this oil field. On the other hand, if it is 100 ± 200, this means that it is quite possible that the actual value of the desired quantity y is 0, i.e., that there is no oil at all. In this case, it may be prudent to perform additional measurements before we invest a lot of money into drilling oil wells.

It is therefore desirable to find out the uncertainty Δy caused by the uncertainties Δx_i in the inputs:



Comment. We assumed that the relation f provides the *exact* relation between the variables $x_1, \ldots, x_n$, and the desired value y. In this case, in the ideal case when we plug in the actual (unknown) values of x_i into the algorithm f, we get the exact value $y = f(x_1, \ldots, x_n)$ of y.

In many real-life situations, the relation f between x_i and y is only *approximately* known. The corresponding *model uncertainty* has to be estimated separately and added to the uncertainty caused by the measurement errors.

III. From Probabilistic to Interval Uncertainty

To estimate the uncertainty Δy caused by the measurement uncertainties Δx_i, we need to have some information about these original uncertainties Δx_i. The whole idea of uncertainty is that we do not know the exact value of x_i (hence, we do not know the exact value of Δx_i). In order words, there are several possible values of Δx_i. So, the first thing we would like to know is what is the *set* of possible values of Δx_i.

We may also know that some of these possible values are more frequent than the others. In other words, we may also have some information about the *probabilities* of different possible values Δx_i.

The manufacturers of a measuring device usually provide us with an upper bound Δ_i for the (absolute value of) possible measurement errors, i.e., with the bound Δ_i for which we are guaranteed that $|\Delta x_i| \leq \Delta_i$.

The need for such a bound comes from the very nature of a measurement process. Indeed, if no such bound is provided, this means that the actual value x_i can be as different from the "measurement result" $\widetilde{x}_i$ as possible. Such a value $\widetilde{x}_i$ is not a measurement, it is a wild guess.

Since the (absolute value of the) measurement error $\Delta x_i = \widetilde{x}_i - x_i$ is bounded by the given bound Δ_i, we can therefore guarantee that the actual (unknown) value of the desired quantity belongs to the interval

$$\mathbf{x}_i \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i].$$

For example, if the measured value of a quantity is $\widetilde{x}_i = 1.0$, and the upper bound Δ_i on the measurement error is 0.1, this means that the (unknown) actual value of the measured quantity can be anywhere between $1 - 0.1 = 0.9$ and $1 + 0.1 = 1.1$, i.e., that it can take any value from the interval $[0.9, 1.1]$.

In many practical situations, we not only know the interval $[-\Delta_i, \Delta_i]$ of possible values of the measurement error; we also know the probability of different values Δx_i within this interval [13].

In most practical applications, it is assumed that the corresponding measurement errors are normally distributed with 0 means and known standard deviation. Numerous engineering techniques are known (and widely used) for processing this uncertainty; see, e.g., [13].

In practice, we can determine the desired probabilities of different values of Δx_i by comparing

- the result $\widetilde{x}_i$ of measuring a certain quantity with this instrument and
- the result $\widetilde{x}_{i\,\text{st}}$ of measuring the same quantity by a standard (much more accurate) measuring instrument.

Since the standard measuring instrument is much more accurate than the one we use, i.e., $|\widetilde{x}_{i\,\text{st}} - x_i| \ll |\widetilde{x}_i - x_i|$, we can assume that $\widetilde{x}_{i\,\text{st}} = x_i$, and thus, that the difference $\widetilde{x}_i - \widetilde{x}_{i\,\text{st}}$ between these two measurement results is practically equal to the measurement error $\Delta x_i = \widetilde{x}_i - x_i$.

Thus, the empirical distribution of the difference $\widetilde{x}_i - \widetilde{x}_{i\,\text{st}}$ is close to the desired probability distribution for measurement error.

There are two cases, however, when this determination is not done:

- First is the case of *cutting-edge* measurements, e.g., measurements in fundamental science. When a Hubble telescope detects the light from a distant galaxy, there is no "standard" (much more accurate) telescope floating nearby that we can use to calibrate the Hubble: the Hubble telescope is the best we have.
- The second case is the case of real *industrial* applications (such as measurements on the shop floor). In this case, in principle, every sensor can be thoroughly calibrated, but sensor calibration is so costly – usually

costing several orders of magnitude more than the sensor itself – that manufacturers rarely do it (only if it is absolutely necessary).

In both cases, we have no information about the probabilities of Δx_i; the only information we have is the upper bound on the measurement error.

In this case, after performing a measurement and getting a measurement result $\widetilde{x}_i$, the only information that we have about the actual value x_i of the measured quantity is that it belongs to the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

In other words, we do know not the actual value x_i of the i-th quantity. Instead, we know the *interval* $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ that contains x_i.

In this situation, for each i, we know the interval $\mathbf{x}_i$ of possible values of x_i, and we need to find the range

$$\mathbf{y} \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}$$

of the given function $f(x_1, \ldots, x_n)$ over all possible tuples $x = (x_1, \ldots, x_n)$ with $x_i \in \mathbf{x}_i$.

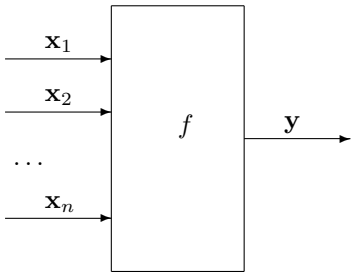
Since the function $f(x_1, \ldots, x_n)$ is usually continuous, this range is also an interval, i.e., $\mathbf{y} = [\underline{y}, \overline{y}]$ for some $\underline{y}$ and $\overline{y}$. So, to find this range, it is sufficient to find the endpoints $\underline{y}$ and $\overline{y}$ of this interval.

Let us formulate the corresponding *interval computations* problem of interval computations in precise terms. We are given:

- an integer n;
- n intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$, and
- an algorithm $f(x_1, \ldots, x_n)$ which transforms n real numbers into a real number $y = f(x_1, \ldots, x_n)$.

We need to compute the endpoints $\underline{y}$ and $\overline{y}$ of the interval

$$\mathbf{y} = [\underline{y}, \overline{y}] = \{f(x_1, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]\}.$$



There exist many efficient algorithms and software packages for solving interval computation problems, and these algorithms has led to numerous useful applications; see, e.g., [1], [2].

IV. Knowledge Processing and Fuzzy Uncertainty

A. Need to Process Fuzzy Uncertainty

In many practical situations, we only have expert estimates for the inputs x_i. Sometimes, experts provide guaranteed bounds on x_i, and even the probabilities of different values within these bounds. However, such cases are rare. Usually, the experts' opinion about the uncertainty of their estimates are described by (imprecise, "fuzzy") words from natural language. For example, an expert can say that the value x_i of the i-th quantity is approximately equal to 1.0, with an accuracy most probably about 0.1. Based on such "fuzzy" information, what can we say about $y = f(x_1, \ldots, x_n)$?

The need to process such "fuzzy" information was first emphasized in the early 1960s by L. Zadeh who designed a special technique of *fuzzy logic* for such processing; see, e.g., [3], [12]. In this technique, our imprecise knowledge about x_i is described by assigning, to each possible real value x_i, the degree $m_i(x_i) \in [0, 1]$ with which this value is a possible value of the i-th input.

In most practical situations, the membership function starts with 0, continuously increases until a certain value and then continuously decreases to 0. Such membership function describe usual expert's expressions such as "small", "medium", "reasonably high", "approximately equal to a with an error about σ", etc. Since membership functions of this type are actively used in expert estimates of number-valued quantities, they are usually called *fuzzy numbers*.

B. Zadeh's Extension Principle

Let us recall how fuzzy techniques can be used for processing fuzzy uncertainty.

We know an algorithm $y = f(x_1, \ldots, x_n)$ that relates the value of the desired difficult-to-estimate quantity y with the values of easier-to-estimate auxiliary quantities $x_1, \ldots, x_n$. We also have expert knowledge about each of the quantities x_i. For each i, this knowledge is described in terms of the corresponding membership function $m_i(x_i)$. Based on this information, we want to find the membership function $m(y)$ which describes, for each real number y, the degree of confidence that this number is a possible value of the desired quantity.

Intuitively, y is a possible value of the desired quantity if for some values $x_1, \ldots, x_n$, x_1 is a possible value of the 1st input quantity, and x_2 is a possible value of the 1st input quantity, ..., and $y = f(x_1 \ldots, x_n)$. We know that the degree of confidence that x_1 is a possible value of the 1st input quantity is equal to $m_1(x_1)$, that the degree of confidence that x_2 is a possible value of the 2nd input quantity is equal to $m_2(x_2)$, etc. The degree of confidence $d(y, x_1, \ldots, x_n)$ in an equality $y = f(x_1 \ldots, x_n)$ is, of course, equal to 1 if this equality holds, and to 0 if this equality does not hold.

The simplest way to represent "and" is to use min. Thus, for each combination of values $x_1, \ldots, x_n$, the degree of confidence in a composite statement "x_1 is a possible value of the 1st input quantity, and x_2 is a possible value of the 1st input quantity, ..., and $y = f(x_1 \ldots, x_n)$" is equal to

$$\min(m_1(x_1), m_2(x_2), \ldots, d(y, x_1, \ldots, x_n)).$$

We can simplify this expression if we consider two possible cases: when the equality $y = f(x_1 \ldots, x_n)$ holds, and when this equality does not hold.

When the equality $y = f(x_1 \ldots, x_n)$ holds, we get $d(y, x_1, \ldots, x_n) = 1$, and thus, the above degree of confidence is simply equal to

$$\min(m_1(x_1), m_2(x_2), \ldots, d(y, x_1, \ldots, x_n)).$$

When the equality $y = f(x_1 \ldots, x_n)$ does not hold, we get $d(y, x_1, \ldots, x_n) = 0$, and thus, the above degree of confidence is simply equal to 0.

We want to combine these degrees of belief into a single degree of confidence that "for some values $x_1, \ldots, x_n$, x_1 is a possible value of the 1st input quantity, and x_2 is a possible value of the 1st input quantity, ..., and $y = f(x_1 \ldots, x_n)$". The words "for some values $x_1, \ldots, x_n$" means that the following composite property hold either for one combination of real numbers $x_1, \ldots, x_n$, or from another combination – until we exhaust all (infinitely many) such combinations. The simplest way to represent "or" is to use $\max$. Thus, the desired degree of confidence $m(y)$ is equal to the maximum of the degrees corresponding to different combinations $x_1, \ldots, x_n$. Since we have infinitely many possible combinations, maximum is not necessarily attained, so we should, in general, consider supremum instead of the maximum:

$$m(y) = \sup \min(m_1(x_1), m_2(x_2), \ldots, d(y, x_1, \ldots, x_n)),$$

where the supremum is taken over all possible combinations.

Since we know that the maximized degree is non-zero only when $y = f(x_1 \ldots, x_n)$, it is sufficient to only take supremum over such combinations. For such combinations, we can omit the term $d(y, x_1, \ldots, x_n)$ in the maximized expression, so we arrive at the following formula:

$$m(y) = \sup\{\min(m_1(x_1), m_2(x_2), \ldots) :$$

$$y = f(x_1, \ldots, x_n)\}.$$

This formula describes a reasonable way to extend an arbitrary data processing algorithm $f(x_1, \ldots, x_n)$ from real-valued inputs to a more general case of fuzzy inputs. It was first proposed by L. Zadeh and is thus called *Zadeh's extension principle*. This is the main formula that describes knowledge processing under fuzzy uncertainty.

C. Reduction to Interval Computations

It is known that from the computational viewpoint, the application of this formula can be reduced to interval computations – and indeed, this is how knowledge processing under fuzzy uncertainty is usually done, by using this reduction; see, e.g., [3], [8], [12].

Specifically, for each fuzzy set with a membership function $m(x)$ and for each $\alpha \in (0, 1]$, we can define this set's *α-cut* as $\mathbf{x}(\alpha) \stackrel{\text{def}}{=} \{x : m(x) \geq \alpha\}$. Vice versa, if we know the α-cuts for all α, we, for each x, can reconstruct the value $m(x)$ as the largest value α for which $x \in \mathbf{x}(\alpha)$.

It is known that when the inputs $m_i(x_i)$ are fuzzy numbers, and the function $y = f(x_1, \ldots, x_n)$ is continuous, then for each α, the α-cut $\mathbf{y}(\alpha)$ of y is equal to the range of possible values of $f(x_1, \ldots, x_n)$ when $x_i \in \mathbf{x}_i(\alpha)$ for all i:

$$\mathbf{y}(\alpha) = f(\mathbf{x}_1(\alpha), \ldots, \mathbf{x}_n(\alpha)).$$

Thus, from the computational viewpoint, the problem of processing data under fuzzy uncertainty can be reduced to several problems of data processing under interval uncertainty – as many problems as there are α-levels.

As we have mentioned, there exist many efficient algorithms and software packages for solving interval computations problems. So, the above reduction can help to efficiently solve the problems of fuzzy data processing as well.

V. Type-2 Fuzzy Sets

A. Need for Type-2 Fuzzy Sets

The main objective of fuzzy logic is to describe uncertain ("fuzzy") knowledge, when an expert cannot describe his or her knowledge by an exact value or by a precise set of possible values. Instead, the expert describe this knowledge by using words from natural language. Fuzzy logic provides a procedure for formalizing these words into a computer-understandable form – as fuzzy sets.

In the traditional approach to fuzzy logic, the expert's degree of certainty in a statement – such as the value $m_A(x)$ describing that the value x satisfies the property A (e.g., "small") – is described by a number from the interval $[0, 1]$. However, we are considering situations in which an expert is unable to describe his or her knowledge in precise terms. It is not very reasonable to expect that in this situation, the same expert will be able to meaningfully express his or her degree of certainty by a precise number. It is much more reasonable to assume that the expert will describe these degrees also by words from natural language.

Thus, for every x, a natural representation of the degree $m(x)$ is not a number, but rather a new fuzzy set. Such situations, in which to every value x we assign a fuzzy number $m(x)$, are called *type-2* fuzzy sets.

B. Successes of Type-2 Fuzzy Sets

Type-2 fuzzy sets are actively used in practice; see, e.g., [4], [5]. Since type-2 fuzzy sets provide a more adequate representation of expert knowledge, it is not surprising that such sets lead to a higher quality control, higher quality clustering, etc., in comparison with the more traditional type-1 sets.

C. The Main Obstacle to Using Type-2 Fuzzy Sets

If type-2 fuzzy sets are more adequate, why are not they used more? The main reason why their use is limited is that the transition from type-1 to type-1 fuzzy sets leads to an increase in computation time. Indeed, to describe a traditional (type-1) membership function function, it is sufficient to describe, for each value x, a single number $m(x)$. In contrast, to describe a type-2 set, for each value x, we must describe the entire membership function – which needs several parameters to describe. Since we need more numbers just to store such information, we need more computational time to process all the numbers representing these sets.

D. Interval-Valued Fuzzy Sets

In line with this reasoning, the most widely used type-2 fuzzy sets are the ones which require the smallest number of parameters to store. We are talking about *interval-valued* fuzzy numbers, in which for each x, the degree of certainty $m(x)$ is an interval $\mathbf{m}(x) = [\underline{m}(x), \overline{m}(x)]$. To store each interval, we need exactly two numbers – the smallest possible increase over the single number needed to store the type-1 value $m(x)$.

VI. Mendel's 2007 Algorithm for Processing Interval-Valued Fuzzy Data

In his plenary talk [6], J. M. Mendel provided a new groundbreaking algorithm which drastically reduced the computational complexity of processing interval-valued fuzzy data. Specifically, he showed that processing interval-valued fuzzy data can be efficiently reduced to interval computations. Since there exist many efficient algorithms and software packages for solving interval computation problems, Mendel's reduction means that we can use these packages to also process interval-valued fuzzy data – and thus, that processing interval-valued fuzzy data is (almost) as efficient as processing the traditional (type-1) fuzzy data.

Mendel's algorithm can be explained as follows. In the case of interval-valued fuzzy data, we do not know the exact numerical values $m_i(x_i)$ of the membership functions, we only know the interval $\mathbf{m}_i(x_i) = [\underline{m}_i(x), \overline{m}_i(x)]$ of possible values of $m_i(x_i)$, By applying Zadeh's extension principle to different combinations of values $m_i(x_i) \in [\underline{m}_i(x), \overline{m}_i(x)]$, we can get, in general, different values of

$$m(y) = \sup\{\min(m_1(x_1), m_2(x_2), \ldots) : y = f(x_1, \ldots, x_n)\}.$$

The result of processing interval-valued fuzzy numbers can be thus described if for each y, we describe the set of possible values of $m(y)$.

When the values $m_i(x_i)$ continuously change, the value $m(y)$ also continuously change. So, for every y, the set $\mathbf{m}(y)$ of all possible values of $m(y)$ is an interval: $\mathbf{m}(y) = [\underline{m}(y), \overline{m}(y)]$. Thus, to describe this set, it is sufficient, for each y, to provide the lower endpoint $\underline{m}(y)$ and the upper endpoint $\overline{m}(y)$ of this interval.

This computation is a particular case of the general problem of interval computations. Indeed, in general, we start with the intervals of possible values of the input, and we want to compute the interval of possible values of the output. In our case, we start with the intervals of possible values of $m_i(x_i)$, and we want to find the set of possible values of $m(y)$.

It is worth mentioning that the corresponding interval computation problem is easier than the general problem because the expression described by Zadeh's extension principle is monotonic – to be more precise, (non-strictly) increasing. Namely, if we increase one of the values $m_i(x_i)$, then the resulting value $m(y)$ can only increase (or stay the same). For monotonic functions, the range of possible values is easy to compute:

- the function attains its smallest value when all the inputs are the smallest, and
- the function attains its largest value when all the inputs are the largest.

In our case, for each input $m_i(x_i)$, the smallest possible value of $\underline{m}_i(x_i)$, and the largest possible value is $\overline{m}_i(x_i)$. Thus, we conclude that:

$$\underline{m}(y) = \sup\{\min(\underline{m}_1(x_1), \underline{m}_2(x_2), \ldots) : y = f(x_1, \ldots, x_n)\};$$

$$\overline{m}(y) = \sup\{\min(\overline{m}_1(x_1), \overline{m}_2(x_2), \ldots) : y = f(x_1, \ldots, x_n)\}.$$

In other words,

- to compute the lower membership function $\underline{m}(y)$, it is sufficient to apply the standard Zadeh's extension principle to the lower membership functions $\underline{m}_i(x_i)$, and
- to compute the upper membership function $\overline{m}(y)$, it is sufficient to apply the standard Zadeh's extension principle to the upper membership functions $\overline{m}_i(x_i)$.

We already know that for type-1 fuzzy sets, Zadeh's extension principle can be reduced to interval computations. Thus, we conclude that for every level $\alpha \in (0, 1]$, we have

$$\underline{\mathbf{x}}(\alpha) = f(\underline{\mathbf{x}}_1(\alpha), \ldots, \underline{\mathbf{x}}_n(\alpha))$$

and

$$\overline{\mathbf{x}}(\alpha) = f(\overline{\mathbf{x}}_1(\alpha), \ldots, \overline{\mathbf{x}}_n(\alpha)),$$

where

$$\underline{\mathbf{x}}_i \stackrel{\text{def}}{=} \{x_i : \underline{m}_i(x_i) \geq \alpha\} \text{ and } \overline{\mathbf{x}}_i \stackrel{\text{def}}{=} \{x_i : \overline{m}_i(x_i) \geq \alpha\}.$$

These are, in effect, the formulas proposed by Mendel in [5].

VII. New Result: Extension of Mendel's Formulas to General Type-2 Fuzzy Numbers

Let us show that Mendel's idea can be extended beyond interval-valued fuzzy numbers, to arbitrary type-2 fuzzy numbers. Indeed, for arbitrary type-2 fuzzy numbers, for each x_i, the value $m_i(x_i)$ is also a fuzzy number. The relation between the input fuzzy numbers $m_i(x_i)$ and the desired fuzzy number $m(y)$ can be expressed by the same Zadeh's principle:

$$m(y) = \sup\{\min(m_1(x_1), m_2(x_2), \ldots) : y = f(x_1, \ldots, x_n)\},$$

but this time, all the values $m_i(x_i)$ and $m(y)$ are fuzzy numbers. How can we describe this relation between fuzzy numbers?

Let us first describe the fuzzy numbers themselves. By definition, a fuzzy number is a function that maps every possible value to a degree from the interval $[0, 1]$ describing to what extend this value is possible. Thus, e.g., for each y,

the corresponding fuzzy number $m(y)$ is a mapping which maps all possible values $t \in [0,1]$ into a degree (from the interval $[0,1]$) with which t is a possible value of $m(y)$. Let us denote this degree by $m(y,t)$.

Similarly, for each i and for each real number x_i, the fuzzy number $m_i(x_i)$ is a mapping which maps all possible values $t \in [0,1]$ into a degree (from the interval $[0,1]$) with which t is a possible value of $m_i(x_i)$. Let us denote this degree by $m_i(x_i,t)$.

As we have already mentioned, processing fuzzy numbers can be reduced to processing the corresponding α-cuts. In this case, all the values $m_i(x_i)$ and $m(y)$ are fuzzy numbers, we conclude that, for every $\alpha \in (0,1]$, the α-cut $(m(y))(\alpha)$ for the fuzzy number $m(y)$ can be obtained by processing the corresponding α-cuts $(m(y))(\alpha)$ for $m_i(x_i)$. To avoid confusion between standard α-cuts, let us denote the corresponding threshold not as α but as β. As a result, we conclude that

$$m(y)(\beta) = \sup\{\min(m_1(x_1)(\beta), m_2(x_2)(\beta), \ldots) : y = f(x_1, \ldots, x_n)\}.$$

For fuzzy numbers, the corresponding β-cuts are intervals: $m(y)(\beta) = [\underline{m(y)(\beta)}, \overline{m(y)(\beta)}]$ and $m_i(x_i)(\beta) = [\underline{m_i(x_i)(\beta)}, \overline{m_i(x_i)(\beta)}]$.

From our description of Mendel's result, we already know that in the interval case, since the expression corresponding to Zadeh's extension principle is monotonic,

- the lower endpoints of the output can be obtained form the lower endpoints of the inputs, and
- the upper endpoint of the output can be obtained from the upper endpoints of the inputs,

hence, that

$$\underline{m(y)(\beta)} = \sup\{\min(\underline{m_1(x_1)(\beta)}, \underline{m_2(x_2)(\beta)}, \ldots) : y = f(x_1, \ldots, x_n)\};$$

$$\overline{m(y)(\beta)} = \sup\{\min(\overline{m_1(x_1)(\beta)}, \overline{m_2(x_2)(\beta)}, \ldots) : y = f(x_1, \ldots, x_n)\}.$$

For the corresponding functions $\underline{m(y)(\beta)}$, $\underline{m_i(x_i)(\beta)}$, $\overline{m(y)(\beta)}$, and $\overline{m_i(x_i)(\beta)}$, we get the standard Zadeh's extension principle relation between membership functions. We already know that this relation can be described in terms of interval computations. Thus, we conclude that

$$\underline{\mathbf{y}}(\alpha,\beta) = f(\underline{\mathbf{x}}_1(\alpha,\beta), \ldots, \underline{\mathbf{x}}_n(\alpha,\beta))$$

and

$$\overline{\mathbf{y}}(\alpha,\beta) = f(\overline{\mathbf{x}}_1(\alpha,\beta), \ldots, \overline{\mathbf{x}}_n(\alpha,\beta)),$$

where

$$\underline{\mathbf{y}}(\alpha,\beta) = \{x : \underline{y(\beta)} \geq \alpha\} \text{ and } \overline{\mathbf{y}}(\alpha,\beta) = \{x : \overline{y(\beta)} \geq \alpha\}$$

are the α-cuts of the corresponding membership functions $\underline{m(y)(\beta)}$, and $\overline{m(y)(\beta)}$, and similarly,

$$\underline{\mathbf{x}}_i(\alpha,\beta) = \{x : \underline{x_i(\beta)} \geq \alpha\}$$

and

$$\overline{\mathbf{x}}_i(\alpha,\beta) = \{x : \overline{x_i(\beta)} \geq \alpha\}$$

are the α-cuts of the corresponding membership functions $\underline{m_i(x_i)(\beta)}$, and $\overline{m_i(x_i)(\beta)}$.

Thus, from the computational viewpoint, the problem of processing data under type-2 fuzzy uncertainty can be reduced to several problems of data processing under interval uncertainty – as many problems as there are (α,β)-levels.

VIII. Conclusion

Type-2 fuzzy sets more adequately describe expert's opinion than the more traditional type-1 fuzzy sets. Because of this, in many practical applications, the use of type-2 fuzzy sets has led to better quality control, better quality clustering, etc. The main reason why they are not universally used is that when we go from type-1 sets to type-2 sets, the computational time of data processing increases. In his 2007 paper, J. Mendel has shown that for the practically important case of interval-valued fuzzy numbers, processing of such such data can be reduced to processing interval data – and is, thus, (almost) as fast as processing type-1 fuzzy data. In this paper, we show that Mendel's idea can be extended to arbitrary type-2 fuzzy numbers – and thus, that processing general type-2 fuzzy numbers is also (almost) as fast as processing type-1 fuzzy data. This result will hopefully lead to more practical applications of type-2 fuzzy sets – which more adequately describe expert knowledge.

References

[1] Interval computations website http://www.cs.utep.edu/interval-comp

[2] Jaulin, L., Kieffer, M., Didrit, O, and Walter, E.: Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics, Springer-Verlag, London, 2001.

[3] Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall (1995)

[4] Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall (2001)

[5] Mendel, J.M.: Type-2 Fuzzy Sets and Systems: an Overview, IEEE Computational Intelligence Magazine **2**, 20–29 (2007)

[6] Mendel, J.M.: Novel Weighted Averages as a Computing With Words Engine, Plenary talk at IFSA World Congress IFSA'07, Cancun, Mexico, June 18–21, 2007.

[7] Nguyen, H.T.: A note on the extension principle for fuzzy sets. J. Math. Anal. and Appl. **64**, 369–380 (1978)

[8] Nguyen, H.T., Kreinovich, V.: Nested intervals and sets: concepts, relations to fuzzy sets, and applications. In: R.B. Kearfott, V. Kreinovich (eds.) Applications Of Interval Computations, pp. 245–290. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands (1996)

[9] Nguyen, H.T., Kreinovich, V.: Applications Of Continuous Mathematics To Computer Science. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands (1997)

[10] Nguyen, H.T., Kreinovich, V.: Methodology of fuzzy control: an introduction. In: H.T. Nguyen, M. Sugeno (eds.) Fuzzy Systems: Modeling and Control, pp. 19–62. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands (1998)

[11] Nguyen, H.T., Kreinovich, V., Zuo, Q.: Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS) **5**, 317–358 (1997)

[12] Nguyen, H.T., Walker, E.A.: First Course In Fuzzy Logic. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA (2006)

[13] Rabinovich, S.: Measurement Errors and Uncertainties: Theory and Practice. American Institute of Physics, New York, NY, USA (2005)