

From Processing Interval-Valued Fuzzy Data to General Type-2: Towards Fast Algorithms

Vladik Kreinovich

Department of Computer Science

University of Texas at El Paso, El Paso, TX 79968

Email: vladik@utep.edu

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. It turns out that for the practically important case of interval-valued fuzzy sets, processing such sets can also be reduced to interval computations – and that this idea can be naturally extended to arbitrary type-2 fuzzy numbers.

I. 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 $y = f(x_1, \dots, x_n)$?

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], [14]. 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 $y = f(x_1, \dots, x_n)$ that relates the value of the desired difficult-to-estimate quantity y with the values of easier-to-estimate auxiliary quantities x_1, \dots, 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, \dots, x_n , x_1 is a possible value of the 1st input quantity, and x_2 is a possible value of the 2nd input quantity, \dots , and $y = f(x_1, \dots, x_n)$. 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 $d(y, x_1, \dots, x_n)$ in an equality $y = f(x_1, \dots, x_n)$ 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, \dots, 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 2nd input quantity, \dots , and $y = f(x_1, \dots, x_n)$ ” is equal to

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

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

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

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

When the equality $y = f(x_1, \dots, x_n)$ does not hold, we get $d(y, x_1, \dots, x_n) = 0$, 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, \dots, 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, \dots , and $y = f(x_1, \dots, x_n)$ ". The words "for some values x_1, \dots, x_n " means that the following composite property hold either for one combination of real numbers x_1, \dots, 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, \dots, 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:

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

where the supremum is taken over all possible combinations.

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

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

This formula describes a reasonable way to extend an arbitrary data processing algorithm $f(x_1, \dots, x_n)$ 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], [10], [14].

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 $y = f(x_1, \dots, x_n)$ is continuous, then for each α , the α -cut $\mathbf{y}(\alpha)$ of y is equal to the range of possible values of $f(x_1, \dots, x_n)$ when $x_i \in \mathbf{x}_i(\alpha)$ for all i :

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

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.

There exist many efficient algorithms and software packages for solving interval computations problems; see, e.g., [1], [2], [8] and references therein. So, the above reduction can help to efficiently solve the problems of fuzzy data processing as well.

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., [5], [6], [7]. 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 $m(x)$.

III. TOWARDS FAST ALGORITHMS FOR PROCESSING INTERVAL-VALUED FUZZY DATA

In their papers and books, J. M. Mendel and his co-authors provided new algorithms which drastically reduced the computational complexity of processing interval-valued fuzzy data; see, e.g., [5], [6], [7] and references therein. In particular, they 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, this 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.

The corresponding reduction 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

$$m(y) = \sup\{\min(m_1(x_1), m_2(x_2), \dots)\};$$

$$y = f(x_1, \dots, x_n).$$

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:

$$\underline{m}(y) = \sup\{\min(\underline{m}_1(x_1), \underline{m}_2(x_2), \dots)\};$$

$$y = f(x_1, \dots, x_n);$$

$$\overline{m}(y) = \sup\{\min(\overline{m}_1(x_1), \overline{m}_2(x_2), \dots)\};$$

$$y = f(x_1, \dots, x_n).$$

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

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

and

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

where

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

IV. EXTENSION TO GENERAL TYPE-2 FUZZY NUMBERS

Let us show that the above techniques can be extended beyond interval-valued fuzzy numbers, to arbitrary type-2 fuzzy numbers; see, e.g., [4]. 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:

$$m(y) = \sup\{\min(m_1(x_1), m_2(x_2), \dots)\};$$

$$y = f(x_1, \dots, x_n),$$

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 extent 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

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

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 the interval-valued fuzzy case, 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 from 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

$$\begin{aligned} \underline{m(y)(\beta)} &= \sup\{\min(\underline{m_1(x_1)(\beta)}, \underline{m_2(x_2)(\beta)}, \dots) : \\ & y = f(x_1, \dots, x_n)\}; \\ \overline{m(y)(\beta)} &= \sup\{\min(\overline{m_1(x_1)(\beta)}, \overline{m_2(x_2)(\beta)}, \dots) : \\ & y = f(x_1, \dots, x_n)\}. \end{aligned}$$

For the corresponding functions

$$\underline{m(y)(\beta)}, \underline{m_i(x_i)(\beta)}, \overline{m(y)(\beta)}, \overline{m_i(x_i)(\beta)},$$

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

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

and

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

where

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

are the α -cuts of the corresponding membership functions

$$\underline{m(y)(\beta)} \text{ and } \overline{m(y)(\beta)},$$

and similarly,

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

are the α -cuts of the corresponding membership functions

$$\underline{m_i(x_i)(\beta)} \text{ and } \overline{m_i(x_i)(\beta)}.$$

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.

V. 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.

For the practically important case of interval-valued fuzzy numbers, processing of 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 argue that these reduction techniques 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.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation grants HRD-0734825 and DUE-0926721 and by Grant 1 T36 GM078000-01 from the National Institutes of Health.

REFERENCES

- [1] Interval computations website <http://www.cs.utep.edu/interval-comp>
- [2] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer-Verlag, London, 2001.
- [3] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, 1995.
- [4] V. Kreinovich and G. Xiang, "Towards Fast Algorithms for Processing Type-2 Fuzzy Data: Extending Mendel's Algorithms From Interval-Valued to a More General Case", *Proceedings of the 27th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2008*, New York, New York, May 19–22, 2008.
- [5] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, 2001.
- [6] J. M. Mendel, "Type-2 Fuzzy Sets and Systems: an Overview", *IEEE Computational Intelligence Magazine*, 2007, Vol. 2, pp. 20–29.
- [7] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*, IEEE Press and Wiley, 2010.
- [8] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM Press, Philadelphia, Pennsylvania, 2009.
- [9] H. T. Nguyen, "A note on the extension principle for fuzzy sets", *J. Math. Anal. and Appl.*, 1978, Vol. 64, pp. 369–380.
- [10] H. T. Nguyen and V. Kreinovich, "Nested intervals and sets: concepts, relations to fuzzy sets, and applications" In: R. B. Kearfott and V. Kreinovich (eds.), *Applications Of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 245–290.
- [11] H. T. Nguyen and V. Kreinovich, *Applications of Continuous Mathematics to Computer Science*, Kluwer, Dordrecht, 1997.
- [12] H. T. Nguyen and V. Kreinovich, "Methodology of fuzzy control: an introduction" In: H. T. Nguyen and M. Sugeno (eds.), *Fuzzy Systems: Modeling and Control*, Kluwer, Dordrecht, 1998, pp. 19–62.
- [13] H. T. Nguyen, V. Kreinovich, and Q. Zuo, "Interval-valued degrees of belief: applications of interval computations to expert systems and intelligent control", *Int'l J. of Uncertainty, Fuzziness, and Knowledge-Based Systems (IJUFKS)*, 1997, Vol. 5, pp. 317–358.
- [14] H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, FL, 2006.
- [15] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, American Institute of Physics, New York, NY, 2005.