How to gauge the joint effect of probabilistic, interval, and fuzzy uncertainty on the result of data processing

Vladik Kreinovich[1], Olga Kosheleva[2], and Nguyen Hoang Phuong[3]
[1,2]Departments of [1]Computer Science and [2]Teacher Education
University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, {olgak,vladik@utep.edu}
[3]Thang Long University, Nghiem Xuan Yem Road
Hoang Mai District, Hanoi, Vietnam, nhphuong2008@gmail.com

1. Formulation of the practical problem

- One of the main objectives of science and engineering is:
  - to predict the future state of the world (science) and
  - to make sure that this future state is as beneficial for us as possible (engineering).
- The more information we use, the more accurate and more reliable our predictions.
- So, it makes sense to use both measurement results and expert estimates.
- This is what many big data applications do.
- This is what all Large Language Models (LLMs) do.
- And the enormous amount of processed data explains the LLMs' spectacular successes.

2. Formulation of the problem: uncertainty

- The inputs to this data processing are not known exactly.
- So, the results of processing this data also come with uncertainty.
- For LLMs, this is well known.
- To better use these results, we need to gauge this uncertainty.
- The fact that we combine data with different types of uncertainty makes it a challenge.
- For some measurement results, we know the probability distribution of the measurement errors.
- For some – e.g., for state-of-the-art measurements – we only know upper bounds on the measurement error.
- This corresponds to interval uncertainty.
- Fuzzy techniques describe expert estimates.

3. Formulation of the problem: what is known and what are the remaining issues

- There are effective well-founded techniques for dealing with each type of uncertainty by itself.
- However, processing of heterogeneous data, with different types of uncertainty, remains largely a challenge.
- To be more precise, there are many heuristic ideas about it – many can be traced to Zadeh himself.
- However, they produce different results, and it is not clear which of them we should select.
- It is therefore desirable to come up with well-founded methods for processing such heterogeneous data.
- In this talk, we provide an overview of the corresponding ideas and results.

4. What we propose: main idea

- Our main idea:
  - instead of reducing all types of uncertainty to a single one,
  - process each type of uncertainty separately,
  - and only then combine the results.
- Let us describe all this in some detail.

5. Why data processing

- What do we humans want?
- We want to know the current state of the world.
- We want to know the future state of the world.
- And we want to know how to make the future state of the world better.
- To fully describe the state of the world means to describe the values of all physical quantities.
- Some of these we can simply measure: e.g., distance from Maastricht to Düsseldorf.
- However, many other values y we cannot (yes) measure directly: e.g., distance from the Earth to a nearby galaxy.

6. Why data processing (cont-d)

- And definitely we cannot directly measure the future values.
- To estimate such values y, we:
  - find easier-to-measure quantities $x_1, \ldots, x_n$ that are related to y by a known dependence $y = f(x_1, \ldots, x_n)$, and
  - use the measurement results $\widetilde{x}_i$ to estimate y as $\widetilde{y} \stackrel{\text{def}}{=} f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

7. Need for uncertainty propagation

- Measurements are never 100% accurate.
- Expert estimates are even less accurate.
- So, estimation errors are non-zero: $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$.
- As a result, $\widetilde{y} \stackrel{\text{def}}{=} f(\widetilde{x}_1, \ldots, \widetilde{x}_n) \neq f(x_1, \ldots, x_n) = y$.
- In practice, it is important to know how big is the resulting error $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$.
- For example, suppose that our estimate for the amount of oil in an oilfield is 150 million tons.
- If this is 150 ± 50 this is good news, we can start digging.
- But if it is 150 ± 200, maybe there is not oil at all.
- In this case, we should perform additional measurements before starting expensive digging.

8. Ideal case: probabilistic uncertainty

- Ideally, we should know:
  - which values of estimation error Δx are possible, and
  - with what frequency each possible value occurs.
- This is known as *probabilistic uncertainty*.
- Ideally, major sources of error have been eliminated.
- What remains is the joint effect of many small sources.
- In this case, according to Central Limit Theorem, the distribution of Δx is close to Gaussian.
- Gaussian distribution is uniquely determined by its means and standard deviation.
- If there is a non-zero mean, we can simply subtract it from measurement results (example: home scales).
- So, what is left is standard deviation σ.

9. Interval uncertainty

- How can we find the characteristics of probabilistic uncertainty?
- For that, we need to know the actual value.
- If we have a much more accurate measuring instruments, its result is approximately the actual value.
- So, by comparing the result, we can estimate σ.
- But there are two cases when this is not done:
  - for state-of-the-art measurements, there are no more accurate instruments;
  - for measurements on the shop floor, sensors are cheap, but calibrating the is too expensive.
- In this case, manufacturers of the measuring instrument have to provide an upper bound Δ on the measurement error $|\Delta x| \le \Delta$.
- Without such bound, this is not a measuring instrument.

10. Interval uncertainty (cont-d)

- In this case, based on the measurement result $\widetilde{x}$, all we know about the actual value x is that it belongs to the interval $[\underline{x}, \overline{x}] \stackrel{\text{def}}{=} [\widetilde{x} - \Delta, \widetilde{x} + \Delta]$.
- This is known as *interval uncertainty*.

11. Fuzzy uncertainty: general idea

- Experts often describe to what extend some values as possible by using imprecise (”fuzzy”) words from natural language like “small”.
- We want to use this knowledge in a computer, and for computers, a natural language is numbers.
- So, we need to translate such words into numbers.
- In a computer, “definitely true” is 1, and “definitely false” is 0.
- So, it is natural to use values between 0 and 1 to described intermediate degrees of certainty.
- This is known as fuzzy uncertainty.
- For each statement, we elicit its confidence degree from the expert.

12. Fuzzy uncertainty: need for "and"- and "or"-operations

- However, this is not enough.
- We need to know to what extent specific value of Δx_1 is possible and specific value of Δx_2 is possible, etc.
- We cannot elicit all this from the experts: there are astronomically many such combined statements.
- So, we need to estimate the degree of a combination $A \,\&\, B$ based on the known degrees a and b of A and B.
- Such an estimation algorithm $f_\&(a, b)$ is known as "and"-operation or, for historical reasons, t-norm.
- Empirically, most widely used t-norms are min and $a \cdot b$.
- Similarly, we need an "or"-operation, aka t-conorm.
- The most widely used t-conorms are max and $a + b - a \cdot b$.

13. Interval-valued and type-2 fuzzy uncertainty

- The whole motivation for fuzzy uncertainty is that experts cannot indicate precise values of the quantity.
- But similarly, they cannot describe their degrees of certainty by precise numbers.
- At best, they can describe an interval of possible degrees – e.g., $[0.7, 0.8]$.
- This is known as interval-valued fuzzy uncertainty.
- Even more realistically, they can use fuzzy words to describe their certainty – i.e., use fuzzy values.
- This is known as type-2 fuzzy uncertainty.
- In principle, the degree can also be type-2 – then we have type-3 fuzzy uncertainty, etc.

14. Possibility of linearization

- We want to estimate characteristics of Δy based on known characteristics of Δx_i.
- This dependence may be complex.
- How do computers compute complex functions?
- In a computer, only arithmetic operations are hardware supported.
- To compute other functions, such as $\exp(x)$ and $\sin(x)$, we expand them in Taylor series and compute the sum of several first terms.
- Estimation errors are usually relatively small.
- Even for a crude 10% error, its square is 1% – much less than 10% and can be thus safely ignored.

15. Possibility of linearization (cont-d)

- If we keep only linear terms, we get

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i, \text{ where } c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}_{|x_1=\widetilde{x}_1,\ldots,x_n=\widetilde{x}_n}.$$

- This is known as linearization.

16. Resulting formulas for probabilistic uncertainty

- Errors of difference measurements are usually independent, so

$$\sigma^2 = \sum_{i=1}^{n} c_i^2 \cdot \sigma_i^2.$$

- In principle, we can estimate c_i by numerical differentiation

$$c_i \approx \frac{f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}}{h_i}.$$

- However, this means n calls to f.
- In many computations, each call can take hours on high performance computer, and n is in thousands.
- This makes the above formula not practical.

17. Probabilistic uncertainty: Monte-Carlo techniques

- To speed up computations, we can use Monte-Carlo simulations:
  - several (N) times, we generate random values $r_i^{(k)}$, $k = 1, \ldots, N$, normally distributed with standard deviation σ_i;
  - compute $r^{(k)} = f(\widetilde{x}_1 + r_1^{(k)}, \ldots, \widetilde{x}_n + r_n^{(k)}) - \widetilde{y}$;
  - compute

$$\sigma = \frac{1}{N} \cdot \sum_{k=1}^{N} \left(r^{(k)}\right)^2.$$

- This estimates σ with relative accuracy $1/\sqrt{N}$.
- So, to get σ with 20% accuracy, it is enough to have 25 calls to f – even if n is in thousands.

18. What if we cannot afford more than one call to f?

- What can we do if we cannot afford even 25 calls to f?
- One possibility is to replace individualized estimates with statistical ones.
- Based on the previous computations, we can find the mean M_i and standard deviation Σ_i of each value c_i^2.
- Then, $E[\sigma^2] = \sum\limits_{i=1}^{n} M_i \cdot \sigma_i^2$ and $V[\sigma^2] = \sum\limits_{i=1}^{n} \Sigma_i^2 \cdot \sigma_i^4$.
- So, with confidence level depending on k_0, all the values σ^2 are contained in the k_0-sigma interval $[E - k_0 \cdot \sqrt{V}, E + k_0 \cdot \sqrt{V}]$.

19. What if we cannot afford more than one call to f (cont-d)

- For $k_0 = 2$, we have confidence 95%; for $k_0 = 3$, confidence 99.9%; for $k_0 = 6$, confidence $1 - 10^{-8}$.
- With this confidence, we conclude that

$$\sigma^2 \le \sum_{i=1}^{n} M_i \cdot \sigma_i^2 + k_0 \cdot \sqrt{\sum_{i=1}^{n} \Sigma_i^2 \cdot \sigma_i^4}.$$

20. Resulting formulas for interval uncertainty

- For interval uncertainty, we get
$$\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i.$$
- This formula is also not practical for large n.
- Good news is that there is a faster Monte-Carlo method here:
  - several (N) times, we generate random values $r_i^{(k)}$, $k = 1, \ldots, N$, distributed by Cauchy distribution with density
$$F(x) \sim \frac{1}{1 + \left(\dfrac{x}{\Delta_i}\right)^2};$$
  - compute $r^{(k)} = f(\widetilde{x}_1 + r_1^{(k)}, \ldots, \widetilde{x}_n + r_n^{(k)}) - \widetilde{y}$;
  - one can prove that $r^{(k)}$ are also Cauchy distributed, with the desired parameter Δ;
  - use Maximum Likelihood method to compute Δ from $r^{(k)}$.

21. What if we cannot afford more than one call to f?

- Similarly to the probabilistic case, we can replace individualized estimates with statistical ones.
- Based on the previous computations, we can find the mean M_i and standard deviation Σ_i of each value $|c_i|$.
- Then, $E[\Delta] = \sum\limits_{i=1}^{n} M_i \cdot \Delta_i$ and $V[\sigma^2] = \sum\limits_{i=1}^{n} \Sigma_i^2 \cdot \Delta_i^2$.
- So, with confidence level depending on k_0, all the values σ^2 are contained in the k_0-sigma interval $[E - k_0 \cdot \sqrt{V}, E + k_0 \cdot \sqrt{V}]$.
- For $k_0 = 2$, we have confidence 95%; for $k_0 = 3$, confidence 99.9%; for $k_0 = 6$, confidence $1 - 10^{-8}$.
- With this confidence, we conclude that

$$\Delta \le \sum_{i=1}^{n} M_i \cdot \Delta_i + k_0 \cdot \sqrt{\sum_{i=1}^{n} \Sigma_i^2 \cdot \Delta_i^2}.$$

22. Resulting formulas for fuzzy uncertainty

- The value y is possible if for some possible values x_i, we have $y = f(x_1, \ldots, x_n)$.
- So, y is possible if:
  - either $x_1^{(1)}$ is possible *and* $x_2^{(1)}$ is possible, ..., and
    $$y^{(1)} = f\left(x_1^{(1)}, \ldots, x_n^{(1)}\right),$$
  - or $x_1^{(1)}$ is possible *and* $x_2^{(2)}$ is possible, ..., and
    $$y^{(2)} = f\left(x_1^{(2)}, \ldots, x_n^{(2)}\right), \text{ etc.}$$
- We have infinitely many such tuples $\left(x_1^{(k)}, x_2^{(k)}, \ldots\right)$.
- The only "or"-operation for which applying it infinitely many times does not lead to 1 is max, so
  $$m(y) = \max(f_\&(m_1(x_1), \ldots, m_n(x_n)) : f(x_1, \ldots, x_n) = y).$$

23. Resulting formulas for fuzzy uncertainty (cont-d)

- This is known as *Zadeh's extension principle*.
- For $f_\& = \min$, this is equivalent to interval computations for all α:

$$y(\alpha) = \{f(x_1, \ldots, x_n) : x_i \in x_i(\alpha)\}, \text{ where } x(\alpha) \stackrel{\text{def}}{=} \{x : m(x) \geq \alpha\}.$$

- The sets $x(\alpha)$ are known as α-*cuts*.
- We can thus use interval methods for fuzzy uncertainty as well.
- Similar reduction is known for type-2 and other fuzzy techniques.
- For other $f_\&$, there are also efficient techniques.

24. What if different inputs have different types of uncertainty: how uncertainty is usually estimated

- The usual; way to deal with such cases is to reduce all uncertainty to a single type.
- For example, suppose that all we know is an interval, and we want to describe it in probabilistic terms.
- We have no reason to believe that some values from this interval are more probable than others.
- Thus, it makes sense to assign equal probability to all these values.
- This argument is known as *Laplace Interdeterminacy Principle*.
- The resulting distribution is known as *uniform*.

25. What if different inputs have different types of uncertainty: how uncertainty is usually estimated (cont-d)

- Similarly, all possible tuples are equally probable, so we have a uniform distribution on a box

$$[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n].$$

- This means that all random variables x_i are independent.
- This sounds reasonable, but it drastically underestimates uncertainty.
- Let us show this on a simple example.
- Suppose that $y = x_1 + \ldots + x_n$ and that each x_i is in the interval $[-1, 1]$.
- Then, the largest possible value of y is n, and the smallest is $-n$.
- But for independent uniform distributions, for large n, the sum is Gaussian, with $\sigma \sim \sqrt{n}$.
- For normal distribution, deviations larger than 6σ has probability 10^{-8} – practically impossible.

26. What if different inputs have different types of uncertainty: how uncertainty is usually estimated (cont-d)

- So, by using this reduction, we conclude that y cannot exceed $C \cdot \sqrt{n}$.
- But for large n, this is much smaller than n.
- Similarly, for each random approximation error, we can reduce it to interval uncertainty $[-k_0 \cdot \sigma, k_0 \cdot \sigma]$, for $k_0 = 2, 3$, or 6.
- But this drastically overestimates uncertainty: from $\sim \sqrt{n}$, we get a much larger estimate $\sim n$.

27. What if different inputs have different types of uncertainty: what we propose

- Due to linearization, we have

$$\Delta y = \Delta y_{\rm prob} + \Delta y_{\rm int} + \Delta y_{\rm fuz} + \ldots.$$

- Here $\Delta y_T = \sum\limits_{i \in T} c_i \cdot \Delta x_i$ for all inputs i of type T.
- What we propose is to estimate each component Δy_T separately.
- Then, if needed, we can combine them by reducing them to a single uncertainty type.

28. What if data processing is done by an AI tool

- Traditional data processing algorithms are very time-consuming.
- In the last decades, in many cases, machine learning was used to speed up computations:
  - we train a neural network (NN), on many examples, to produce $y = f(x_1, \ldots, x_n)$ based on the inputs x_i, and
  - after that, we use this neural network instead of the original algorithm.
- This is faster, since neurons work in parallel, and each neuron computes a very simple function $\max(0, w_1 \cdot x_1 + \ldots + w_n \cdot x_n + w_0)$.
- This is similar to a human brain:
  - biological neurons are million times slower that computer cells, but
  - the brain often makes decision at about the same speed.

29. What if data processing is done by an AI tool: good news

- Good news is that for NNs, we do not need to call f to compute c_i.
- Indeed, the whole training is done by backpropagation, i.e., by gradient descent.
- In this process, we compute derivatives with respect to weights w_i.
- It turns out that, based on these derivatives, we can easily compute the desired derivatives c_i.

30. What if we need to make more accurate estimations

- In some practical applications, we need more accurate uncertainty estimations.
- For example, in space exploration
- NNs can also speed up computations:
  - when we need more accurate uncertainty estimation
  - and thus, we need to take into account terms quadratic (and even higher order) in Δx_i.
- In general, if all we know are intervals, we need to find

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

- In this case, $\overline{y}$ is the maximum of the function $f(x_1, \ldots, x_n)$ on the box

$$[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n].$$

- Similarly, $\underline{y}$ is the minimum of the function $f(x_1, \ldots, x_n)$ on this box.

31. What if we need to make more accurate estimations (cont-d)

- In general, already for quadratic functions, computing max and min is, in general, NP-hard.
- This means that, if, as most computer scientists believe, P $\neq$ NP, no feasible algorithm can always compute min and max.
- Our idea: use gradient descent for find min and max.
- Yes, we sometimes end up in a local max or local min.
- But this is already true for NN, where, because of gradient descent, instead of best fit we may end up in a local min.
- In short, we already sacrificed global min by using NN.
- So, it makes sense to make the same sacrifice when estimating $\underline{y}$ and $\overline{y}$.

32. What if we need to make more accurate estimations: technical details

- We have a technical problem:
  - gradient descent works for unconstrained optimization;
  - while here, we optimize under constraints $\underline{x}_i \le x_i \le \overline{x}_i$.
- This is easy to repair: we introduce new unknowns t_i for which
$$x_i = \underline{x}_i + \frac{2t_i^2}{1+t_i^2} \cdot \Delta_i.$$
- Then, when $t_i \in (-\infty, +\infty)$, the expression x_i takes all the values from $\underline{x}_i$ to $\overline{x}_i = \underline{x}_i + 2\Delta_i$.
- So, we indeed reduce the original constrained optimization to the unconstrained one.

33. What if we need to make more accurate estimations: technical details (cont-d)

- One can prove that this is the fastest-to-compute reduction:
- Indeed, we need at least one division:
  - otherwise we have a polynomial $x_i(t_i)$, and
  - polynomials are not bounded.
- One can check that we cannot have just one more arithmetic operation.
- And the only way to get two more operations is to have $t_i \cdot t_i$ and $+1$ (which is $++$, a fast version of addition).

34. References

- V. Kreinovich and S. Ferson, "A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis", *Reliability Engineering and Systems Safety*, 2004, Vol. 85, No. 1–3, pp. 267–279, https://www.cs.utep.edu/vladik/2002/tr02-28a.pdf

- V. Kreinovich, "From Processing Interval-Valued Fuzzy Data to General Type-2: Towards Fast Algorithms", *Proceedings of the IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems T2FUZZ'2011*, part of the IEEE Symposium Series on Computational Intelligence, Paris, France, April 11–15, 2011, pp. ix–xii, https://www.cs.utep.edu/vladik/2011/tr11-10.pdf

- A. Pownuk, V. Kreinovich, and S. Sriboonchitta, "Fuzzy Data Processing Beyond Min t-Norm", In: C. Berger-Vachon, A. M. Gil Lafuente, J. Kacprzyk, Y. Kondratenko, J. M. Merigo Lindahl, and C. Morabito (eds.), *Complex Systems: Solutions and Challenges in Economics, Management, and Engineering*, Springer Verlag, 2018, pp. 237–250, https://www.cs.utep.edu/vladik/2016/tr16-101.pdf

35. References (cont-d)

- O. Kosheleva and V. Kreinovich, "How to Propagate Uncertainty via AI Algorithms", *Proceedings of the 10th International Workshop on Reliable Engineering Computing REC'2024*, Beijing, China, October 26–27, 2024, pp. 5–12, https://www.cs.utep.edu/vladik/2024/tr24-31.pdf
- C. Q. Lauter, M. Ceberio, V. Kreinovich, and O. M. Kosheleva, "Uncertainty Quantification for Results of AI-Based Data Processing: Towards More Feasible Algorithms", In: F. Pavese, A. Forbes, A. Bosnjakovic, S. Eichstaedt, and J. Sousa (eds.), Advanced Mathematical and Computational Tools for Metrology and Testing XIII, World Scientific, Singapore, 2025, pp. 95–108, https://www.cs.utep.edu/vladik/2023/tr23-66a.pdf

36. Acknowledgments

This work was supported by:

- the AT&T Fellowship in Information Technology,
- the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany,
- the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329,
- the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590),
- the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand,
- the Ho Chi Minh City University of Banking, Vietnam, and
- Thang Long University, Hanoi, Vietnam.