How to Efficiently Propagate P-Box Uncertainty

O. M. Kosheleva

Department of Teacher Education, University of Texas at El Paso
El Paso, Texas 79968, USA
E-mail: olgak@utep.edu

V. Kreinovich

Department of Computer Science, University of Texas at El Paso
El Paso, Texas 79968, USA
**E-mail: vladik@utep.edu, www.cs.utep.edu/vladik*

In many practical situations, to get the desired estimate or prediction, we need to process existing data. This data usually comes from measurements, and measurements are never 100% accurate. Because we only know the input values with uncertainty, the results of processing this data also comes with uncertainty. To make an appropriate decision, we need to know how accurate is the resulting estimate, i.e., how the input uncertainty "propagates" through the data processing algorithm. In the ideal case, when we know the probability distribution of each measurement error, we can, in principle, use Monte-Carlo simulations to describe the uncertainty of the data processing result. In practice, however, we often only have partial information about the measurement uncertainty: for example, instead of the exact values of the cumulative distribution function $F(x)$, we only know bounds on $F(x)$. Such information is known as the probability box (p-box, for short). In this paper, we provide feasible algorithms for propagating p-box uncertainty.

1. Formulation of the problem

What are the main objectives of science and engineering. The main objectives of science and engineering can be roughly summarized as follows: to understand the current state of the world, to predict the future state of the world, and to come up with a proper strategy to make this future state as good for us as possible.

How can we describe this in precise terms. To describe the current or future state of the world, we need to list the values of all the numerical characteristics of this state – i.e., in other words, the values of the corresponding physical quantities. For example, to describe the current state of the weather in a given area, we need to list the temperature, wind speed,

wind direction, atmospheric pressure, humidity, etc. The state of a mechanical system can be characterized by the locations and velocities of all the particles that form this system, etc.

Similarly, to describe the strategy for changing the state, we need to list the numerical characteristics of the corresponding control. For example, if we want to drive from Point A to Point B, we need to know the exact path, i.e., distances, angles, speeds, etc.

Information about the world comes from measurements. Our information about the values of different physical quantities comes from measurements – this is, in effect, a definition of measurement: a process to supply us with the values of different physical quantities.

Need for data processing. Some quantities we can measure directly. For example, we can directly measure the width of an office desk, we can directly measure the weight of a person or even of a truck. However, for some physical quantities, it is very difficult – or even impossible – to measure them directly. For example, we cannot directly measure the distance to a faraway star or the overall amount of oil in an oil field. Since we cannot measure the corresponding quantity y directly, the only way to measure them is to measure them *indirectly*, i.e.:

- to measure some easier-to-measure quantities $x_1, \ldots, x_n$ that are related to the desired quantity y by a known dependence $y = f(x_1, \ldots, x_n)$, and then
- apply the algorithm f to the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of these measurements and thus get an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for the desired quantity y.

This is an important particular case of what is called *data processing*.

Data processing is also needed to predict the future value y of each quantity of interest based on the current values $x_1, \ldots, x_n$ of this and related quantities. Data processing is needed to determine the appropriate control values based on the current state. In all these cases, we apply an appropriate algorithm $y = f(x_1, \ldots, x_n)$ to the results $\widetilde{x}_i$ of measuring some quantities x_i.

Need to take uncertainty into account. Measurement are never 100% accurate; see, e.g., [5]. Each measurement result $\widetilde{x}_i$ is, in general, different from the actual (unknown) value x_i of the corresponding quantity. The difference $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ – known as the *measurement error* – is, in general, different from 0.

Because $\widetilde{x}_i \neq x_i$, the result $\widetilde{y}$ of applying the algorithm f to measurement results $\widetilde{x}_i$ is, in general, different from the value $y = f(x_1, \ldots, x_n)$ that we would have gotten if we knew the exact values of the measured quantities $x_1, \ldots, x_n$.

In practice, it is important to know how big the difference $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ can be. For example, if we estimated the amount of oil in a given field as 150 million tons, then, if it is 150 ± 20, this is good news, we can start exploring the field. However, if it is 150 ± 200, then maybe there is no oil in this field at all, so we need to perform additional measurements before investing millions of dollars into this exploitation.

In general, we need to know how uncertainty in x_i affects the uncertainty of the result of applying the algorithm f to the inputs x_i, i.e., how uncertainty *propagates* from the inputs x_i to the output y. In other words, we need to solve the following problem.

Uncertainty propagation: a general formulation of the problem. We *know*:

- the data processing algorithm $y = f(x_1, \ldots, x_n)$,
- the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$, and
- some information about the measurement errors $\Delta x_i = \widetilde{x}_i - x_i$.

Based on this information, we *need to get* the information about the estimation error $\Delta y = \widetilde{y} - y$, where $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, and $y = f(x_1, \ldots, x_n)$.

How can we describe uncertainty: ideal case. To analyze how uncertainty propagates, we need to analyze how this uncertainty is described. To fully describe the uncertainty of each measurement, we need to know what values of measurement error are possible, and how frequent are different possible values. In other words, we need to know the probability distribution of each measurement error Δx_i.

Measurement errors corresponding to different measurements are usually caused by different factors and are, thus, independent. Our objective is to use this information about Δx_i to provide the probability distribution for the desired estimation error Δy.

How to propagate uncertainty: ideal case. In this ideal case, we can use Monte-Carlo simulations to produce the desired distribution for Δy. Namely, several times $k = 1, \ldots, K$, we:

- simulate the measurement errors $\Delta x_i^{(k)}$ according to the known

distribution for each such error,
- compute the simulated values $x_i^{(k)} = \widetilde{x}_i - \Delta x_i^{(k)}$, and
- compute $y^{(k)} = f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$ and $\Delta y^{(k)} = y^{(k)} - \widetilde{y}$.

One can check that the resulting values $y^{(k)}$ are distributed according to the desired distribution of Δy.

How can we describe the corresponding probability distributions. There are many ways to represent a probability distribution: by the probability density function (pdf), by the cumulative distribution function, by moments, etc. The problem is that most of these representations are not universal:

- some distributions do not have finite moments – e.g., Cauchy distribution;
- some distributions do not have the probability density function – e.g., distribution located at a single value with probability 1.

The only universal representation is by using a cumulative distribution function (cdf) $F_i(X) \stackrel{\text{def}}{=} \text{Prob}(\Delta x_i \le X)$.

Comment. There are infinitely many real numbers X, and thus, infinitely many values $F_i(X)$. Of course, we cannot store infimitely many values. The usual way to describe a general function $F_i(X)$ is to store its values for some inputs $X_{i,1} < X_{i,2} < \ldots < X_{i,N}$ that form a dense grid on the real line.

Thus, in this paper, we will be using this way to represent uncertainty – each probability distribution will be described by the values $F_i(X_{i,1}), \ldots, F_i(X_{i,N})$ of the corresponding cdf at a dense grid of values $X_{i,j}$.

Often, we only have partial information about the probabilities. In many real-life cases, we often only have partial information about the probabilities. This means that for each possible value X_i of the measurement error Δx_i:

- instead of knowing the exact value $F_i(X_{i,j})$,
- we only have partial information about $F_i(X_{i,j})$ – i.e., we have *bounds* on this value.

Usually, possible values of $F_i(X_{i,j})$ form an interval $[\underline{F}_i(X_{i,j}), \overline{F}_i(X_{i,j})]$. So, a natural way to describe such cases is to have a function that assigns

such interval to each value $X_{i,j}$. This function is known as a *probability box*, or *p-box*, for short; see, e.g., [1].

Comment. Usually, when we estimate a characteristic c of a probability distribution – be it mean, standard deviation, or cdf – from a sample:

- we have an estimate $\widetilde{c}$ based on the this sample – e.g., the sample mean, the sample standard deviation, etc. – and
- we also have bounds Δ_c (with some sufficient certainty) on the absolute value of the difference $\Delta c = \widetilde{c} - c$ between the sample-based estimate $\widetilde{c}$ and the actual (unknown) value c of this characteristic.

In this case, we can conclude that the actual value c lies somewhere in the interval $[\widetilde{c} - \Delta_c, \widetilde{c} + \Delta_c]$.

Some methods for estimating the value of a statistical characteristic directly produce the interval of possible values $[\underline{c}, \overline{c}]$. This interval can also be represented in the form $[\widetilde{c} - \Delta_c, \widetilde{c} + \Delta_c]$ if we take

$$\widetilde{c} = \frac{\underline{c} + \overline{c}}{2} \text{ and } \Delta_c = \frac{\overline{c} - \underline{c}}{2}.$$

So, in the following text, we will inter-changingly use both ways of representing the interval of possible values of each statistical characteristic.

Finally, the formulation of the specific problem: of propagating p-box uncertainty. We *know*:

- the data processing algorithm $y = f(x_1, \ldots, x_n)$,
- the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$, and
- n p-boxes $[\underline{F}_i(X_{i,j}), \overline{F}_i(X_{i,j})] = [\widetilde{F}_i(X_{i,j}) - \Delta_{i,j}, \widetilde{F}_i(X_{i,j}) + \Delta_{i,j}]$ that describe possible probability distributions for the measurement errors $\Delta x_i = \widetilde{x}_i - x_i$.

We are interested in the information about a statistical characteristic c describing the difference $\Delta y = \widetilde{y} - y$, where $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ and $y = f(x_1, \ldots, x_n)$. This characteristic can be the mean, the standard deviation, or the value $F(Y_j)$ of the cdf of Δy.

Based on the known information, we *need to estimate* the range

$$[\widetilde{c} - \Delta_c, \widetilde{c} + \Delta_c]$$

of possible values of c for all possible distributions $F_i(X)$ of Δx_i for Δx_i which $F_i(X_{i,j}) \in [\underline{F}_i(X_{i,j}), \overline{F}(X_{i,j})]$ for all i and j.

Comment. By using the Monte-Carlo techniques, we can estimate the value $\widetilde{c}$ corresponding to the sample-based distributions $\widetilde{F}_i(X)$. So, the main task to estimate the value Δ_c – that described possible deviations from this value.

What is known and what is not known. There exist feasible algorithms for propagating p-box uncertainty for many important cases; see, e.g., [1]. However, there is no general efficient algorithm for such propagation.

What we do in this paper. In this paper, we provide a feasible algorithm for solving this problem.

2. Analysis of the problem

Cannot we just use Monte-Carlo simulations? The main difference between the p-box case and the ideal case – when we know all the probability distributions – is that for each input x_i, we may have several different probability distributions. So, a natural idea is to try many of them.

A simple analysis shows that this is not feasible. Even if we consider 2 values for each of N points $X_1, \ldots, X_N$, this means 2^N options. Trying 2^N options is not feasible: already for reasonable $N \approx 300$, this requires more computational steps than the lifetime of the Universe; see, e.g., [3,4]. It is usually assumed that only algorithm that require no more than polynomial number of $O(n^k)$ computational steps are feasible.

So, we cannot use this straightforward approach, we need to come up with new ideas.

Possibility of linearization. For different possible distributions of Δx_i, we have different values of the cdf $F_i(X)$, and thus, different values of $\Delta F_{i,j} \stackrel{\text{def}}{=} \widetilde{F}_i(X_{i,j}) - F_i(X_{i,j})$. As a result, in general, we get different values of the desired characteristic c, and thus, different values of the difference $\Delta c = \widetilde{c} - c$.

The value Δc depends on all the differences $\Delta F_{i,j}$:

$$\Delta c = D(\Delta F_{1,1}, \ldots, \Delta F_{1,N}, \ldots, \Delta F_{i,j}, \ldots, \Delta F_{n,N}),$$

for some function D. Our objective is to find this function D.

When the sample is reasonable large, sample-based estimates of all the characteristics – including cdf – are reasonably accurate. This means that the differences $\widetilde{F}_i(X_{i,j}) - F_i(X_{i,j})$ are reasonably small. In general, this means that terms which are quadratic in terms of $\Delta F_{i,j}$ are much smaller

than linear terms and can, thus, be safely ignored. For example if $\Delta F_{i,j} \approx 10\%$, then $(\Delta F_{i,j})^2 \approx 1\% \ll 10\%$.

Thus, we can use the usual linearization idea (frequent in physics [2,7] and in processing measurement results in general [5]):

- we expand the dependence D of Δc on $\Delta F_{i,j}$ in Taylor series, and
- we ignore terms which are quadratic (and higher order) in $\Delta F_{i,j}$ and keep only linear terms.

Thus, we arrive at the linear dependence:

$$\Delta c = \sum_{i,j} a_{i,j} \cdot \Delta F_{i,j}, \tag{1}$$

for some coefficients $a_{i,j}$.

So, the question of funding the function D is reduced to the question of finding the appropriate coefficients $a_{i,j}$.

How can we find the coefficients $a_{i,j}$. For each $i = 0, 1, \ldots, n$, and $j = 1, \ldots, N$, we form a cdf $F_i^{(j)}(X)$ for which we have $F_{i'}^{(j)}(X_{i',j}) = \widetilde{F}_{i'}(X_{i,j})$ for all $i' \neq i$, while for $i' = i$:

- we have $F_i^{(j)}(X_{i,k}) = \underline{F}_i(X_{i,k})$ for $k \leq j$, and
- we have $F_i^{(j)}(X_{i,k}) = \overline{F}_i(X_{i,k})$ for $k > i$.

We use Monte-Carlo (or any other) method to find the value $\Delta c_{i,j}$ corresponding to the cdf's $F_i^{(j)}(X)$.

Because of linearity (1), we have

$$\Delta c_{i,j} - \Delta c_{i,j-1} = a_{i,j} \cdot (\overline{F}_i(X_j) - \underline{F}_i(X_{i,j})) = 2a_{i,j} \cdot \Delta_{i,j}.$$

So, we can estimate $a_{i,j}$ as

$$a_{i,j} = \frac{\Delta c_{i,j} - \Delta c_{i,j-1}}{2\Delta_{i,j}}.$$

Once we know $a_{i,j}$, how can we estimate Δ_c? We know that each value $\Delta F_{i,j}$ lies somewhere in the interval $[-\Delta_{i,j}, \Delta_{i,j}]$. We also an additional constraint on these values – that the resulting values $F_i(X_{i,j}) = \widetilde{F}_i(X_{i,j}) + \Delta F_{i,j}$ must be non-strictly increasing in j;

$$\widetilde{F}_i(X_{i,j}) + \Delta F_{i,j} \leq \widetilde{F}_i(X_{i,j+1}) + \Delta F_{i,j+1}.$$

Thus, we arrive at the following algorithm for computing Δ_c.

3. Algorithm for propagating p-box uncertainty: description

- First, we use the Monte-Carlo – or any other method – to find the value $\widetilde{c}$ of the characteristic c when each Δx_i is distributed according to the cdf $\widetilde{F}_i(X)$.
- Then, for each $i = 0, 1, \ldots, n$, and $j = 1, \ldots, N$, we form a cdf $F_i^{(j)}(X)$ for which we have $F_{i'}^{(j)}(X_{i',j}) = \widetilde{F}_{i'}(X_{i,j})$ for all $i' \neq i$, while for $i' = i$:
  - we have $F_i^{(j)}(X_{i,k}) = \underline{F}_i(X_{i,k})$ for $k \leq j$, and
  - we have $F_i^{(j)}(X_{i,k}) = \overline{F}_i(X_{i,k})$ for $k > i$.
- We use Monte-Carlo (or any other) method to find the value $\Delta c_{i,j}$ corresponding to the cdf's $F_i^{(j)}(X)$.
- Then, we estimate

$$a_{i,j} = \frac{\Delta c_{i,j} - \Delta c_{i,j-1}}{2\Delta_{i,j}}.$$

- Finally, we find the desired value Δ_c by solving the following optimization problem:

$$\Delta c = \sum_{i,j} a_{i,j} \cdot \Delta F_{i,j} \to \max$$

under the the following constraints:

$$-\Delta_{i,j} \leq \Delta F_{i,j} \leq \Delta_{i,j} \text{ for all } i \text{ and } j, \text{ and}$$

$$\widetilde{F}_i(X_{i,j}) + \Delta F_{i,j} \leq \widetilde{F}_i(X_{i,j+1}) + \Delta F_{i,j+1} \text{ for all } i \text{ and } j.$$

Comments.

- In the last optimization problem, we maximize a linear expression under linear constraints – which means that this problem is a particular case of linear programming. There are many feasible algorithms for solving linear programming problems; see, e.g., [8]. Specifically, solving a linear programming problem with u unknowns requires $O(u^{2+1/18})$ computational steps. In our problem, we have $u = N \cdot n$ unknowns, so this is indeed feasible.
- Our algorithm requires $N \cdot n + 1$ calls to Monte-Carlo simulation procedure, which is also feasible.

4. Discussion: how accurate are the results of the proposed algorithm?

Need to estimate accuracy. Our algorithms uses Monte-Carlo simulations, and Monte-Carlo simulations only provide an estimate for a statistical characteristic. It is therefore desirable to know how accurate are the resulting estimates for Δ_c – and what can we do to get the estimate of Δ_c with given relative accuracy ε.

Accuracy of the result of Monte-Carlo simulations: reminder. It is known that if we run the simulations T times, the resulting uncertainty decreases as $1/\sqrt{T}$; see, e.g.,[6].

Analysis of the problem and the resulting estimate. Let Δ denote the size of the differences $\Delta F_{i,j} = \overline{F}_i(X_{i,j}) - \underline{F}_i(X_{i,j})$. So, in linear approximation, the value Δ_c is proportional to Δ.

Let ε be the relative accuracy with which we want to estimate the value Δ_c. For example, we can take $\varepsilon = 20\%$:

- remember, this is accuracy with which we determine accuracy;
- measuring instrument can have accuracy 10%, but 11.6% accuracy does not make too much practical sense.

This means that we need absolute accuracy $\varepsilon \cdot \Delta$. In general, if we use values at N points, a monotonic function is represented with accuracy $\sim 1/N$. Thus, we need to have $N \sim 1/(\varepsilon \cdot \Delta)$.

Let δ be the accuracy with which we determine each value $\Delta c_{i,j}$. Each term in the linear dependence (1) is close to the difference $\Delta c_{i,j} - \Delta c_{i,j-1}$. Thus, the value of each term is of order δ.

The standard deviation of the sum of $N \cdot n$ independent terms grows as $\sqrt{N \cdot n}$. So, the accuracy with which we determine Δc is $\delta \cdot \sqrt{N \cdot n}$. Thus, to reach accuracy $\varepsilon \cdot \Delta$, we need to select $\delta = \varepsilon \cdot \Delta / \sqrt{N \cdot n}$.

Let M denote the number of calls to f that we use to estimate each $\Delta c_{i,j}$. In general, M iterations provide relative accuracy $\sim 1/\sqrt{M}$. To get $1/\sqrt{M} \sim \delta = \varepsilon \cdot \Delta / \sqrt{N \cdot n}$, we thus need:

$$M \sim \varepsilon^{-2} \cdot \Delta^{-2} \cdot N \cdot n \sim \varepsilon^{-3} \cdot \Delta^{-2} \text{ calls to } f.$$

We need to compute $N \cdot n \sim n \cdot \varepsilon^{-1} \cdot \Delta^{-1}$ values $\Delta c_{i,j}$. Thus, overall, we need

$$N \cdot n \cdot M \sim (\varepsilon^{-1} \cdot \Delta^{-1}) \cdot (\varepsilon^{-3} \cdot \Delta^{-2}) = \varepsilon^{-4} \cdot \Delta^{-3} \text{ calls to } f.$$

This is polynomial in ε^{-1} and Δ^{-1} and thus, still feasible.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

The authors are greatly thankful to Franco Pavese for his encouragement and to all the participants of the International Conference on Advanced Mathematical Tools in Metrology and Testing AMCTM 2023 (September 26–28, 2023) for valuable discussions.

References

1. S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz, *Constructing Probability Boxes and Dempster-Shafer Structures*, Sandia National Laboratories, Report SAND2002-4015, January 2003.
2. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
3. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.
4. C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.
5. S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, New York, 2005.
6. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.
7. K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2021.
8. R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Springer, New York, 2014.