Uncertainty propagation: how to best deal with p-box uncertainty

Yi Luo, Olga Kosheleva, Vladik Kreinovich, Kittawit Autchariyapanikul

Abstract Many traditional statistical methods implicitly assume that we know the actual probability distributions. However, based on a finite sample, we can only provide approximate values of the actual probabilities – and, with a certain confidence level, bounds of these probabilities. In particular, instead of the actual cumulative distribution function, we only have bounds on its values. This is known as a p-box. Another important case when p-boxes appear is the case of chaotic dynamics, when we cannot exactly predict the probabilities of different future states, we can only predict ranges of these probabilities. In this paper, we show how to propagate such p-box uncertainty throught data processing algorithms.

1 What are p-boxes, and why we need to study them

How to make decisions: decision theory recommendations. According to decision theory, decisions of a rational decision agent should be determined by a special *utility function* $u(x)$ that describes the decision maker's preferences. Specifically, out of all possible actions, the rational decision maker should select the alternative

Yi Luo
Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34,
30167 Hannover, Germany, e-mail: yi.luo@irz.uni-hannover.de

Olga Kosheleva
Department of Teacher Education, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: olgak@utep.edu

Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

Kittawit Autchariyapanikul
Faculty of Economics, Maejo University, Chiang Mai, Thailand, e-mail: kittawit_a@mju.ac.th

for which the expected value of $E[u(x)]$ of the utility is the largest possible; see, e.g., [2, 3, 5, 6, 8, 9, 11].

What information about the probability distribution is needed to make a decision. To estimate the expected value of utility, we need to have some information about the corresponding probability distribution.

In many cases, the utility $u(x)$ smoothly depends on the variable x. In this case, we can expand the dependence of utility on x in Taylor series and keep only a few first terms in this expansion. In other words, for some typical value x_0, we replace the original utility function with a linear combination of the terms $(x-x_0)^k$ for integer $k \geq 0$. In such case, the expected value of utility function becomes a linear combination of the expected values $E[(x-x_0)^k]$ of these terms, i.e., as a linear combination of the moments of the probability distribution. So, to make reasonable decisions, there is no need to know the exact form of the probability distribution – it is sufficient to know its first few moments.

In such situations, if we use a value x which is slightly larger or slightly smaller than the optimal value x_0, the result will be slightly suboptimal, but it will be usually not a big deal.

There are, however, another decision situations where there is a critical threshold. For example, chemical plants, no matter how good they are, emit a small amount of potentially dangerous substances into the atmosphere. In allowed very small concentrations, these substances are harmless, but if their concentration exceeded the allowed threshold, this is somewhat bad new for the environment, but a really bad news for the company – it will face a huge fine and maybe a closure of the offending plant – that would cause even bigger losses. In such situations, as the quantity x increases, the utility $u(x)$ first changes slightly until we reach the threshold value x_0, and then changes abruptly from a positive to a zero (or even negative) value – and changes very slightly after that.

In such situations, the utility function can be well approximated by a piecewise-constant function that takes the value $u_0 > 0$ fo $x \leq x_0$ and the value 0 for $x > x_0$. The expected value of this utility function is equal to u_0 times the probability $\text{Prob}(x \leq x_0)$. A function that assigns, to each threshold x_0, the probability $\text{Prob}(x \leq x_0)$, is known as the *cumulative distribution function*, or cdf, for short; see, e.g., [12].

Summarizing: to make decisions, we need to know the moments and the cdf of the probability distribution. Out of these two characteristics, knowing the cdf is the most critical part of information.

From cdf to a p-box. All probabilities come from observations. In general, how can we empirically determine probabilitiea? A usual way is to consider several repetitions of the situation and compute the frequency of the desired outcome – e.g., the frequency with which the coin falls heads. When the sample size increases, this frequency becomes closer and closer to the actual probability. For large sample sizes, the distribution of the difference $f-p$ between the frequency f and the probability p is close to Gaussian, with mean 0 and a known standard deviation. Based on this information, we can find the interval that contain the actual value of the probability p with a desired confidence; e.g.:

- the 3σ interval $[f-3\sigma, f+3\sigma]$ contain p with confidence 99.9%,
- the 6σ interval $[f-6\sigma, f+3\sigma]$ contain p with confidence $1-10^{-8}$, etc.

So, at each moment of time, we only know the frequency – which is only approximately equal to the probability. We do not know the exact values of the probability, we know the interval $[\underline{p}, \overline{p}] = [f-k_0\cdot\sigma, f+k_0\cdot\sigma]$ (for some k_0) containing p.

In particular, for each x, instead of the actual value of the probability $F(x)$, we only know the interval $[\underline{F}(x), \overline{F}(x)]$ that contains the actual (unknown) values of $F(x)$. A function that transforms a value x into such an interval is known as the *probabiliyt box*, or p-box, for short; see, e.g., [1]. From this decision-making viewpoint, p-boxes are ubiquitous in representing uncertainty.

Chaotic dynamics: another reason why p-boxes are ubiquitous. There is another reason why p-boxes are ubiquitous. This is related to the fact that for many real-life processes, even when we know the differential equations that determine their dynamics, we still cannot accuratey predict neither the exact values of the future quantities, nor even the probability of different future values. This happens because in such equations, initial minuscule non-detectable deviations can lead to drastic differences in the future state.

In such situations, even if we can safely ignore the measurement uncertainty in measuring the original state, about the future state, we can only know the interval of possible values – and for each value x from this interval, instead of the exact value of the probability $F(x)$, we only know the interval $[\underline{F}(x), \overline{F}(x)]$, i.e., we know the p-box.

2 Uncertainty propagation under p-box uncertainty: an important task

Need for uncertainty propagation. We want to know the current and future values of different quantities. Some of the current quantities we can directly measure. In such cases, from the description of the corresponding measurement process, we get the description of the uncertainty of the resulting estimate.

However, many quantities are difficult (or even impossible) to measure directly, and it is definitely not possible to directly measure future values of different quantities. To estimate such not-easy-to-directly-measure quantities, we need:

- to 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 then
- to use the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring x_i to compute the y-estimate

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

In such situations, based on the known uncertainty of the measurement results $\widetilde{x}_i$, we need to determine the uncertainty of the estimate $\widetilde{y}$. Estimating this uncertainty is known as *uncertainty propagation.*

Uncertainty propagation under probabilistic uncertainty: a brief reminder. In some cases, we have explicit analytical formulas formulas that describe the probabilistic characteristics of $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ (such as mean, standard deviation, etc.) based on the characteristics of $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$; see, e.g., [10]. However, in many other cases, there are no such analytical formulas. In such cases, we can use *Monte-Carlo simulations*; see, e.g., [12]. Namely:

- for some large N, for each $k = 1, \ldots, N$, we simulate the values $\Delta x_i^{(k)}$ distributed according to the known probability distribution for Δx_i;
- compute the modified values $x_i^{(k)} = \widetilde{x}_i - \Delta x_i^{(k)}$, then $y^{(k)} = f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$, and $\Delta y^{(k)} = \widetilde{y} - y^{(k)}$;
- The resulting values $\Delta y^{(1)}, \ldots, \Delta y^{(N)}$ form a sample distributed according to the desired distrubution for Δy. Based on this sample, we can estimate the desired statistical characteristics of the probability distribution of Δy.

Uncertainty propagation under p-box uncertainty: challenge and idea. When we have p-box uncertainty about x_i, we would like to find the p-box for Δy. Similarly to the probabilistic case, sometimes, we have explicit analytical expressions for the resulting p-box; see, e.g., [1] and references therein. However, also similarly to the probabilistic case, in many situations, we do not have such expressions.

In the p-box cases, for each quantity x_i, we know the p-box $[\underline{F}_i(x_i), \overline{F}_i(x_i)]$. Here, in contrast to the probabilistic case, we do not have a single probability distribution for Δx_i, we have many possible probability distributions – namely, all distributions $F_i(x_i)$ for which $F_i(x_i) \in [\underline{F}_i(x_i), \overline{F}_i(x_i)]$. So, a natural idea (see, e g., [7]) is:

- to take several possible distributions from the p-box for x_i,
- for each combination c of these distributions, to run Monte-Carlo simulations and to estimate the resulting cdf $F_{yc}(y)$.

Then, for each y, we can estimate the desired range $[\underline{F}_y(y), \overline{F}_y(y)]$ as

$$\left[\min_c F_{yc}(y), \max_c F_{yc}(y)\right].$$

Remaining question – and what we do in this paper. A natural remaining question is: how to select possible distrbutions from the given p-box.

In this paper, we provide a natural way to do it.

3 How to select distributions from a p-box: our idea and the resulting algorithm

Need for discretization. Of course, in reality, we can only use finitely many values $F(x)$, i.e., only values $f(v_1), \ldots, F(v_m)$ corresponding to a finite number of points

$v_1 < v_2 < \ldots < v_m$. For example, we can take uniformly distributed values

$$v_1, \;\; v_2 = v_1 + h, v_3 = v_1 + 2h, \ldots, v_t = v_1 + (t-1) \cdot h, \ldots, v_m = v_1 + (m-1) \cdot h.$$

For each of these selected values, we know the interval $[\underline{F}(v_i), \overline{F}(v_i)]$. For these intervals, we have $\underline{F}(v_i) \le \underline{F}(v_{i+1})$ and $\overline{F}(v_i) \le \overline{F}(v_{i+1})$ for all i.

We need to select the values $F_i = F(v_i) \in [0,1]$ for which $F_i \in [\underline{F}(v_i), \overline{F}(v_i)]$ and $F_i \le F_{i+1}$ for all i.

Comment. For simulation purposes, we may need to know the values $F(x)$ for values $x \ne v_i$. These values can be found, e.g., by linear interpolation: for each $x \in [v_i, v_{i+1}$, we take

$$F(x) = \frac{v_{i+1} - x}{v_{i+1} - v_i} \cdot F_i + \frac{x - v_i}{v_{i+1} - v_i} \cdot F_{i+1}.$$

Natural idea. We have the set S of all possible tuples $(F_1, \ldots, F_m)$ for which, for all i, we have $F_i \in [\underline{F}(v_i), \overline{F}(v_i)]$ and $F_i \le F_{i+1}$. A natural idea is to use Monte-Carlo simulation to select each tuple. For this purpose, we need to find an appropriate probability distribution on the set S.

A priori, we have to reason to believe that some tuples from the set S are more probable than others. It is therefore reasonable to assume that all tuples from the set S are equally probable, i.e., that we use a uniform distribution on this set. This argument is known as the *Laplace Indeterminacy Principle*; it is a specific case of a more general *Maximum Entropy* approach to selecting a probability distribution; see, e.g., [4].

How to implement this idea. To find a single tuple, we select, for each i, values F_i that are uniformly distributed on the interval $[\underline{F}_i, \overline{F}_i]$. For this purpose, we can select values r_i uniformly distributed on the interval $[0,1]$ and take $F_i = \underline{F}_i + r_i \cdot (\overline{F}_i - \underline{F}_i)$. Then:

- If the resulting values satisfy the monotonicity condition $F_1 \le F_2 \ldots \le F_m$, then we return the resulting tuple.
- On the other hand, if some monotonicity conditions are not satisfied, we ignore this tuple and repeat the above process – until we get the tuple that satisfies the monotonicity condition.

How feasible is the proposed implementation? What is the probability that for two randomly selected paitr (F_i, F_{i+1}), we get $F_i \le F_{i+1}$? For each pair, the worst case – with probability 1/2 – is when the intervals $[\underline{F}(v_i), \overline{F}(v_i)]$ and $[\underline{F}(v_{i+1}), \overline{F}(v_{i+1})]$ coincide. In all other cases, as one can easily check, this probability is larger than 1/2. We need to satisfy $m-1$ such inequalities: from $F_1 \le F_2$ to $F_{m-1} \le F_m$. So, the probability that our process will lead to a tuple is larger than or equal to $(1/2)^{m-1}$.

For a typical case $m = 11$, when we divide the range of x_i into 10 subranges, this probability is at least $(1/2)^{10} = 1/1024$. So, on average, after at most 1000 iterations of the above process, we will get the desired tuple. So, if we want to come up with a 1000 tuples, we need to run this process million times. This may sound

large, but for a computer, this is practically nothing – the most time in Monte-Carlo simulations is usually spent not on generating random inputs, but on running the algorithm $y = f(x_1, \ldots, x_n)$ for different inputs. In this sense, the above procedure is quite feasible.

Comment. This completes the main part of this paper. In the following section, we answer an auxiliary question: can we have a faster algorithm for selecting tuples? Specifically, in this section, we describe two seemingly reasonable faster approaches, and we explain that these approaches do not always lead to a uniform distribution on the set S.

4 Can we do it faster – not with the following two seemingly reasonable approaches

First seemingly reasonable approach: idea. We select F_1 uniformly from the interval $[\underline{F}(v_1), \overline{F}(v_1)]$.

Once we have selected F_1, we have two restrictions on F_2: that $F_1 \le F_2$ and that $\underline{F}(v_2) \le F_2 \le \overline{F}(v_2)$. These two restrictions can be combined into a single one: $\max(F_1, \underline{F}(v_2) \le F_2 \le \overline{F}(v_2)$. Thus, it seems reasonable to select F_2 uniformly from the interval $[\max(F_1, \underline{F}(v_2), \overline{F}(v_2)]$.

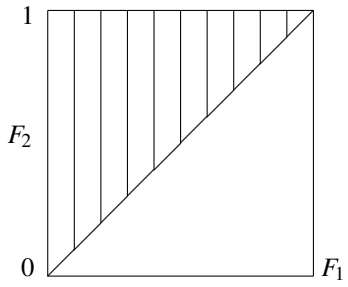
Similarly, once we have selected the value F_i, then we have two restrictions on F_{i+1}: that $F_i \le F_{i+1}$ and that $\underline{F}(v_{i+1}) \le F_{i+1} \le \overline{F}(v_{i+1})$. These two restrictions can be combined into a single one: $\max(F_i, \underline{F}(v_{i+1})) \le F_{i+1} \le \overline{F}(v_{i+1})$. Thus, it seems reasonable to select F_{i+1} uniformly from the interval $[\max(F_i, \underline{F}(v_{i+1})), \overline{F}(v_{i+1})]$. So, we arrive at the following algorithm.

First seemingly reasonable approach: resulting algorithm. First, we select F_1 uniformly from the interval $[\underline{F}(v_1), \overline{F}(v_1)]$. Then, we sequentially select F_2, F_3, …, F_m as follows: once we have selected F_i, we select F_{i+1} uniformly from the interval

$$[\max(F_i, \underline{F}(v_{i+1})), \overline{F}(v_{i+1})].$$

First seemingly reasonable approach: an example where this approach does not lead to the desired uniform distribution. To show that the above algorithm does not lead to a uniform distribution, let us consider a simple case when $m = 2$ and $[\underline{F}(v_1), \overline{F}(v_1)] = [\underline{F}(v_2), \overline{F}(v_2)] = [0, 1]$. According to the above algorithm, in this case, F_1 is selected uniformly from the interval $[0, 1]$, so its mean value is 1/2.

On the other hand, here, the set S is a subset of the unit square determined by the inequality $F_1 \le F_2$, see below.



In this case, the area of S is $|S| = 1/2$, and so, the mean value of F_1 is

$$\frac{1}{|S|} \cdot \int_0^1 (1 - F_1)\, dF_1 = \frac{1}{1/2} \cdot \int_0^1 (F_1 - F_1^2)\, dF_1 =$$

$$2 \cdot \left(\frac{F_1^2}{2} - \frac{F_1^3}{3} \right) \Bigg|_0^1 = 2 \cdot \left(\frac{1}{2} - \frac{1}{3} \right) = 2 \cdot \frac{1}{6} = \frac{1}{3}.$$

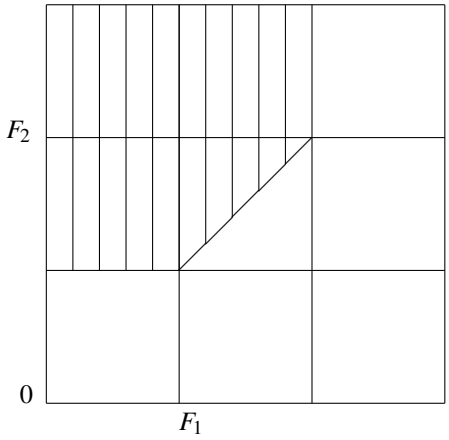
So, the mean value 1/2 produced by the first faster algorithm is different from the mean value 1/3 corresponding to the uniform distribution. Thus, the first faster algorithm does not always return tuples uniformly distributed on the set S.

Second seemingly reasonable approach: idea. In the above example, instead of limiting F_2 to the interval $[F_1, 1]$, we could simply generate random values G_1 and G_2 from the corresponding intervals, and then take the smallest as F_1 and the largest as F_2. One can check that in this example, this idea indeed leads to a uniform distribution on the set S. This leads to the following seemingly rational algorithm.

Second seemingly reasonable approach: resulting algorithm. For each i, we uniformly select some value G_i from the interval $[\underline{F}(v_i), \overline{F}(v_i)]$. Then, as F_1, we select the smallest of these values, as F_2, the smallest of the remaining values, etc. In other words, we:

- sort the values G_i into an increasing sequence $G_{(1)} \le G_{(2)} \le \ldots \le G_{(m)}$ and then
- take $F_i = G_{(i)}$ for all i.

Second seemingly reasonable approach: an example where this approach does not lead to the desired uniform distribution. Let us consider an example with $m = 2$, $[\underline{F}(v_1), \overline{F}(v_1)] = [0, 2/3]$, and $[\underline{F}(v_2), \overline{F}(v_2)] = [1/3, 1]$. In this case, the set S takes the following form:



In this case, the set S consists of three equal-size small squares

$$\left[0,\frac{1}{3}\right]\times\left[\frac{1}{3},\frac{2}{3}\right],\ \left[0,\frac{1}{3}\right]\times\left[\frac{2}{3},1\right],\ \text{and}\ \left[\frac{1}{3},\frac{2}{3}\right]\times\left[\frac{2}{3},1\right],$$

and a half of the fourth (central) square

$$\left[\frac{1}{3},\frac{2}{3}\right]\times\left[\frac{1}{3},\frac{2}{3}\right].$$

For the uniform distribution, the probability for a tuple to be in a subset s of the set S is equal to the ratio of their areas. The are of the set S is equal to 3.5 times the area of each small square, while the area of the subset s of all the tuples for which both F_1 and F_2 are in the interval

$$\left[\frac{1}{3},\frac{2}{3}\right]$$

is equal to half of the small square. Thus, for the uniform distribution, the probability to be in s is equal to $0.5/3.5 = 1/7$.

On the other hand, in our second fast algorithm, the values F_i are simply sorted G_i's. Thus, both values F_i belongs to the interval $[1/3,2/3]$ if and only if both G_i belong to this interval. The probability of each G_i belonging to this interval is 1/2, and the values G_i are independent. So, the probability that both G_i's belong to this interval is equal to $(1/2)\cdot(1/2) = 1/4$.

So, the value 1/4 produced by the second faster algorithm is different from the value 1/7 corresponding to the uniform distribution. Thus, the second faster algorithm also does not always return tuples uniformly distributed on the set S.

Acknowledgments

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

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. P. C. Fishburn, *Utility Theory for Decision Making*, John Wiley & Sons Inc., New York, 1969.
3. P. C. Fishburn, *Nonlinear Preference and Utility Theory*, The John Hopkins Press, Baltimore, Maryland, 1988.
4. E. T. Jaynes and G. L. Bretthorst, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK, 2003.
5. V. Kreinovich, "Decision making under interval uncertainty (and beyond)", In: P. Guo and W. Pedrycz (eds.), *Human-Centric Decision-Making Models for Social Sciences*, Springer Verlag, 2014, pp. 163–193.
6. R. D. Luce and R. Raiffa, *Games and Decisions: Introduction and Critical Survey*, Dover, New York, 1989.
7. Y. Luo, M.-Z. Lyu, M. Broggi, M. Thiele, V. C. Fragkoulis, and M. Beer, "Unified framework for hybrid aleatory and epistemic uncertainty propagation via decoupled multi-probability density evolution method", *Reliability Engineering and System Safety*, 2026, Vol. 270, Paper 112171
8. H. T. Nguyen, O. Kosheleva, and V. Kreinovich, "Decision making beyond Arrow's 'impossibility theorem', with the analysis of effects of collusion and mutual attraction", *International Journal of Intelligent Systems*, 2009, Vol. 24, No. 1, pp. 27–47.
9. H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, Berlin, Heidelberg, 2012.
10. S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, New York, 2005.
11. H. Raiffa, *Decision Analysis*, McGraw-Hill, Columbus, Ohio, 1997.
12. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.