Interval computations and interval-related statistical techniques: estimating uncertainty of the results of data processing and indirect measurements

V. Kreinovich

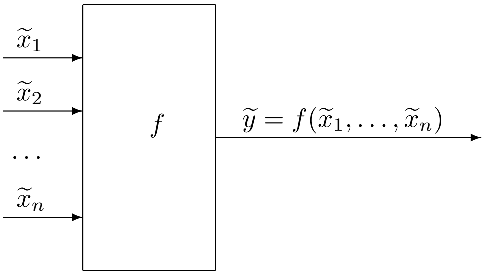
Computer Science Department, University of Texas at El Paso,
El Paso, Texas 79968, USA
E-mail: vladik@utep.edu
http://www.cs.utep.edu/vladik

In many practical situations, we only know the upper bound Δ on the measurement error: $|\Delta x| \le \Delta$. In other words, we only know that the measurement error is located on the interval $[-\Delta, \Delta]$. The traditional approach is to assume that Δx is uniformly distributed on $[-\Delta, \Delta]$. In some situations, however, this approach underestimates the error of indirect measurements. It is therefore desirable to directly process this interval uncertainty. Such "interval computations" methods have been developed since the 1950s. In this paper, we provide a brief overview of related algorithms and results.

Keywords: interval uncertainty, interval computations, interval-related statistical techniques

1. Need for Interval Computations

Data processing and indirect measurements. We are often interested in a physical quantity y that is difficult (or impossible) to measure directly: distance to a star, amount of oil in a well. A natural idea is to measure y *indirectly*: we find easier-to-measure quantities $x_1, \ldots, x_n$ related to y by a known relation $y = f(x_1, \ldots, x_n)$, and then use the results $\widetilde{x}_i$ of measuring x_i to estimate $\widetilde{y}$:



This is known as *data processing.*

Estimating uncertainty of the results of indirect measurements: a problem. Measurements are never 100% accurate. The actual value x_i of i-th measured quantity can differ from the measurement result $\widetilde{x}_i$; in other words, there are *measurement errors* $\Delta x_i \stackrel{\rm def}{=} \widetilde{x}_i - x_i$. Because of that, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of data processing is, in general, different from the actual value y: $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) \neq f(x_1, \ldots, x_n) = y$. It is desirable to describe the error $\Delta y \stackrel{\rm def}{=} \widetilde{y} - y$ of the result of data processing. For this, we must have information about the errors of direct measurements.

Uncertainty of direct measurements: need for overall error bounds (i.e., interval uncertainty). Manufacturers of a measuring instrument (MI) usually provide an upper bound Δ_i for the measurement error: $|\Delta x_i| \le \Delta_i$. (If no such bound is provided, then $\widetilde{x}_i$ is not a measurement, it is a wild guess.)

Once we get the measured value $\widetilde{x}_i$, we can thus guarantee that the actual (unknown) value of x_i is in the interval $\mathbf{x}_i \stackrel{\rm def}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. For example, if $\widetilde{x}_i = 1.0$ and $\Delta_i = 0.1$, then $x_i \in [0.9, 1.1]$.

In many practical situations, we also know the *probabilities* of different values Δx_i within this interval. It is usually assumed that Δx_i is normally distributed with 0 mean and known standard deviation.

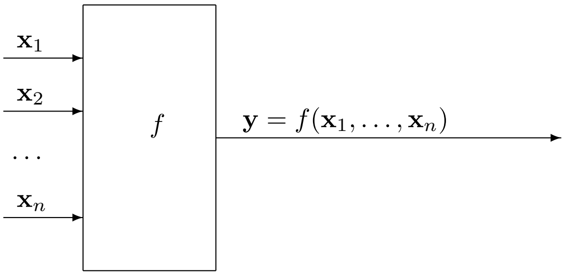
In practice, we can determine the desired probabilities by *calibration*, i.e., by comparing the results $\widetilde{x}_i$ of our MI with the results $\widetilde{x}_i^{\rm st}$ of measuring the same quantity by a standard (much more accurate) MI. However, there are two cases when calibration is not done: (1) cutting-edge measurements (e.g., in fundamental science), when our MI is the best we have, and (2) measurements on the shop floor, when calibration of MI is too expensive.

In both cases, the only information we have is the upper bound on the measurement error. In such cases, we have interval uncertainty about the actual values x_i; see, e.g.,[11].

Interval computations: a problem. When the inputs x_i of the data processing algorithms are known with interval uncertainty, we face the following problem:

- *Given:* an algorithm $y = f(x_1, \ldots, x_n)$ and n intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- *Compute:* the corresponding range of y:

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



It is known that this problem is NP-hard even for quadratic f; see, e.g.,[8]. In other words, unless P=NP (which most computer scientists believe to be impossible), no feasible algorithm is possible that would always compute the exact range $\mathbf{y}$. We thus face two major challenges: (1) find situations feasible algorithms are possible, and (2) in situations when the exact computation of $\mathbf{y}$ is not feasibly possible, find feasible algorithms for computing a good approximation $\mathbf{Y} \supseteq \mathbf{y}$.

2. Alternative Approach: Maximum Entropy (MaxEnt)

Idea: a brief reminder. Traditional engineering approach to uncertainty is to use probablistic techniques, based on probability density functions (pdf) $\rho(x)$ and cumulative distribution functions (cdf) $F(x) \stackrel{\text{def}}{=} P(X \le x)$. As we have mentioned, in many practical applications, it is very difficult to come up with the probabilities. In such applications, many different probability distributions are consistent with the same observations. In such situations, a natural idea is to select one of these distributions – e.g., the one with the largest entropy $S \stackrel{\text{def}}{=} -\int \rho(x) \cdot \ln(\rho(x))\,dx$; see, e.g.,[5].

Often, this idea works. This approach often leads to reasonable results. For example, for the case of a single variable x, if all we know is that $x \in [\underline{x}, \overline{x}]$, then MaxEnt leads to a uniform distribution on $[\underline{x}, \overline{x}]$. For several variables, if we have no information about their dependence, MaxEnt implies that different variables are independently distributed.

Sometimes, this idea does not work. Sometimes, the results of MaxEnt are misleading. As an example, let us consider the simplest algorithm $y = x_1 + \ldots + x_n$, with $\Delta x_i \in [-\Delta, \Delta]$. In this case, $\Delta y = \Delta x_1 + \ldots + \Delta x_n$. The worst case is when $\Delta_i = \Delta$ for all i, then $\Delta y = n \cdot \Delta$.

What will MaxEnt return here? If all Δx_i are uniformly distributed, then for large n, due to the Central Limit Theorem, Δy is approximately normal, with $\sigma = \Delta \cdot \dfrac{\sqrt{n}}{\sqrt{3}}$.

With confidence 99.9%, we can thus conclude that $|\Delta y| \le 3\sigma$; so, we get $\Delta \sim \sqrt{n}$, but, as we mentioned. it is possible that $\Delta = n \cdot \Delta \sim n$ which, for large n, is much larger than $\sqrt{n}$.

The conclusion from this example is that using a single distribution can be very misleading, especially if we want guaranteed results – and we do want guaranteed results in high-risk application areas such as space exploration or nuclear engineering.

3. Possibility of Linearization

Linearization is usually possible. Each interval has the form $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i - \Delta_i]$, where $\widetilde{x}_i$ is a midpoint and Δ_i is half-width. Possible values x_i are $x_i = \widetilde{x}_i + \Delta x_i$, with $|\Delta x_i| \le \Delta_i$, so $f(x_1, \ldots, x_n) = f(\widetilde{x}_1 + \Delta x_1, \ldots, \widetilde{x}_n + \Delta x_n)$. The values Δ_i are usually reasonable small, hence the values Δx_i are also small. Thus, we can expand f into Taylor series and keep only linear terms in this expansion:

$$f(\widetilde{x}_1 + \Delta x_1, \ldots) = \widetilde{y} + \sum_{i=1}^{n} c_i \cdot \Delta x_i, \text{ where } \widetilde{y} \stackrel{\text{def}}{=} f(\widetilde{x}_1, \ldots) \text{ and } c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}.$$

Here, $\max(c_i \cdot \Delta x_i) = |c_i| \cdot \Delta_i$, so the range of f is $[\widetilde{y} - \Delta, \widetilde{y} + \Delta]$, where $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.

Towards an algorithm. To compute $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$, we need to find c_i. If we replace one of $\widetilde{x}_i$ with $\widetilde{x}_i + \Delta_i$, then, due to linearization, we get

$$y_i \stackrel{\text{def}}{=} f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) = \widetilde{y} + c_i \cdot \Delta_i.$$

Thus, $|c_i| \cdot \Delta_i = |y_i - \widetilde{y}|$ and hence $\Delta = \sum\limits_{i=1}^{n} |y_i - \widetilde{y}|$.

Resulting algorithm. Compute $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, compute n values $y_i = P_f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n)$, then compute $\Delta = \sum\limits_{i=1}^{n} |y_i - \widetilde{y}|$ and $\left[\widetilde{P} - \Delta, \widetilde{P} + \Delta\right]$.

This algorithm requires $n+1$ calls to f: to compute $\widetilde{y}$ and n values y_i.

Towards a faster algorithm. When the number of inputs n is large, $n+1$ calls may be too long. To speed up computations, we can use the following property of Cauchy distribution, with density $\$\rho_\delta(x) = \dfrac{\delta}{\pi} \cdot \dfrac{1}{1 + \dfrac{x^2}{\delta^2}}$: if η_i are independently Cauchy-distributed with parameters Δ_i, then $\eta \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} c_i \cdot \eta_i$ is Cauchy-distributed with parameter $\Delta = \sum\limits_{i=1}^{c} |c_i| \cdot \Delta_i$.

Once we get simulated Cauchy-distributed values η, we can estimate Δ by the Maximum Likelihood method. We also need to scale η_i to the interval $[-\Delta_i, \Delta_i]$ on which the linear approximation is applicable.

Resulting faster algorithm.[7] First, we compute $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$. For some N (e.g., 200), for $k = 1, \ldots, N$, we repeatedly:

- use the random number generator to compute $r_i^{(k)}$, $i = 1, 2, \ldots, n$, uniformly distributed on $[0, 1]$;
- compute Cauchy distributed values as $c_i^{(k)} = \tan(\pi \cdot (r_i^{(k)} - 0.5))$;
- compute the largest value K of the values $\left|c_i^{(k)}\right|$;
- compute simulated "actual values" $x_i^{(k)} = \widetilde{x}_i + \dfrac{\Delta_i \cdot c_i^{(k)}}{K}$;
- apply f and compute $\Delta y^{(k)} = K \cdot \left(f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right) - \widetilde{y}\right)$.

Then, we compute $\Delta \in \left[0, \max\limits_k \left|\Delta y^{(k)}\right|\right]$ by applying the bisection method to the equation $\dfrac{1}{1 + \left(\dfrac{\Delta y^{(1)}}{\Delta}\right)^2} + \ldots + \dfrac{1}{1 + \left(\dfrac{\Delta y^{(N)}}{\Delta}\right)^2} = \dfrac{N}{2}$. We stop when we get Δ with accuracy $\approx 20\%$ (accuracy 1% and 1.2% is approximately the same).

The Cauchy-variate algorithm requires $N \approx 200$ calls to f. So, when $n \gg 200$, it is much faster than the above linearization-based algorithm.

4. Beyond Linearization, Towards Interval Computations

Linearizaion is sometimes not sufficient. In many application areas, it is sufficient to have an *approximate* estimate of y. However, sometimes, we need to *guarantee* that y does not exceed a certain threshold y_0: in *nuclear engineering*, the temperatures and the neutron flows should not exceed the critical values; a spaceship should land on the planet and does not fly past it, etc.

The only way to guarantee this is to have an interval $\mathbf{Y} = \left[\underline{Y}, \overline{Y}\,\right]$ for which $\mathbf{y} \subseteq \mathbf{Y}$ and $\overline{Y} \le y_0$. Such an interval is called an *enclosure*. Computing such an enclosure is one of the main tasks of interval computations.

Interval computations: a brief history. The origins of interval computations can be traced to the work of Archimedes from Ancient Greece who used intervals to bound values like π; see, e.g.,[1]. Its modern revival was boosted by three pioneers: Mieczyslaw Warmus (Poland), Teruo Sunaga (Japan), and Ramon Moore (USA) in 1956–59. The first successful application was taking interval uncertainty into account when planning spaceflights to the Moon. Since then, there were many successful applications: to design of elementary particle colliders (Martin Berz, Kyoko Makino, USA), to checking whether an asteroid will hit the Earth (M. Berz, R. Moore, USA), to robotics (L. Jaulin, France; A. Neumaier, Austria), to chemical engineering (Marc Stadtherr, USA), etc.[4,9].

Interval arithmetic: foundations of interval techniques. The problem is to compute the range

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

For arithmetic operations $f(x_1, x_2)$ (and for elementary functions), we have explicit formulas for the range. For example, when $x_1 \in \mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$ and $x_2 \in \mathbf{x}_2 = [\underline{x}_2, \overline{x}_2]$, then:

- The range $\mathbf{x}_1 + \mathbf{x}_2$ for $x_1 + x_2$ is $[\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$.
- The range $\mathbf{x}_1 - \mathbf{x}_2$ for $x_1 - x_2$ is $[\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2]$.
- The range $\mathbf{x}_1 \cdot \mathbf{x}_2$ for $x_1 \cdot x_2$ is

$$[\min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2), \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2)].$$

The range $1/\mathbf{x}_1$ for $1/x_1$ is $[1/\overline{x}_1, 1/\underline{x}_1]$ (if $0 \notin \mathbf{x}_1$).

Straightforward interval computations. In general, we can parse an algorithm (i.e., represent it as a sequence of elementary operations) and then perform the same operations, but with *intervals* instead of *numbers*.

For example, to compute $f(x) = (x - 2) \cdot (x + 2)$, the computer first computes $r_1 := x - 2$, then $r_2 := x + 2$, and $r_3 := r_1 \cdot r_2$. So, for estimating the range of $f(x)$ for $x \in [1, 2]$, we compute $\mathbf{r}_1 := [1, 2] - [2, 2] = [-1, 0]$, $\mathbf{r}_2 := [1, 2] + [2, 2] = [3, 4]$, and $\mathbf{r}_3 := [-1, 0] \cdot [3, 4] = [-4, 0]$.

Here, the actual range is $f(\mathbf{x}) = [-3, 0]$. This example shows that we need more efficient ways of computing an enclosure $\mathbf{Y} \supseteq \mathbf{y}$.

First idea: use of monotonicity. For arithmetic, we had exact ranges, because $+$, $-$, $\cdot$ are monotonic in each variable, and monotonicity helps: if $f(x_1,\ldots,x_n)$ is (non-strictly) increasing ($f\uparrow$) in each x_i, then $f(\mathbf{x}_1,\ldots,\mathbf{x}_n)=[f(\underline{x}_1,\ldots,\underline{x}_n),f(\overline{x}_1,\ldots,\overline{x}_n)]$. Similarly, if $f\uparrow$ for some x_i and $f\downarrow$ for other x_j.

It is known that $f\uparrow$ in x_i if $\dfrac{\partial f}{\partial x_i}\ge 0$. So, to check monotonicity, we can check that the range $[\underline{r}_i,\overline{r}_i]$ of $\dfrac{\partial f}{\partial x_i}$ on $\mathbf{x}_i$ has $\underline{r}_i\ge 0$. Here, differentiation can be performed by available Automatic Differentiation (AD) tools, an estimating ranges of $\dfrac{\partial f}{\partial x_i}$ can be done by using straightforward interval computations.

For example, for $f(x)=(x-2)\cdot(x+2)$, the derivatives is $2x$, so its range on $\mathbf{x}=[1,2]$ is $[2,4]$, with $2\ge 0$. Thus, we get the exact range $f([1,2])=[f(1),f(2)]=[-3,0]$.

Second idea: centered form. In the general non-monotonic case, we can use the general version of linearization – the Intermediate Value Theorem, according to which

$$f(x_1,\ldots,x_n)=f(\widetilde{x}_1,\ldots,\widetilde{x}_n)+\sum_{i=1}^n\frac{\partial f}{\partial x_i}(\chi)\cdot(x_i-\widetilde{x}_i)$$

for some $\chi_i\in\mathbf{x}_i$. Because of this theorem, we can conclude that $f(x_1,\ldots,x_n)\in\mathbf{Y}$, where

$$\mathbf{Y}=\widetilde{y}+\sum_{i=1}^n\frac{\partial f}{\partial x_i}(\mathbf{x}_1,\ldots,\mathbf{x}_n)\cdot[-\Delta_i,\Delta_i].$$

Here also, differentiation can be done by Automatic Differentiation (AD) tools, and estimating the ranges of derivatives can be done, if appropriate, by monotonicity, or else by straightforward interval computations, or also by centered form (this will take more time but lead to more accurate results).

Third idea: bisection. It is known that the inaccuracy of the first order approximation (like the ones we used) is $O(\Delta_i^2)$. So, when Δ_i is too large and the accuracy is low, we can split the corresponding interval in half (reducing the inaccuracy from Δ_i^2 to $\Delta_i^2/4$), and then take the union of the resulting ranges.

For example, the function $f(x)=x\cdot(1-x)$ is not monotonic for $x\in$

$\mathbf{x} = [0, 1]$. So, we take $\mathbf{x}' = [0, 0.5]$ and $\mathbf{x}'' = [0.5, 1]$; on the 1st subinterval, the range of the derivative is $1 - 2 \cdot \mathbf{x} = 1 - 2 \cdot [0, 0.5] = [0, 1]$, so $f \uparrow$ and $f(\mathbf{x}') = [f(0), f(0.5)] = [0, 0.25]$. On the 2nd subinterval, we have $1-2\cdot\mathbf{x} = 1-2\cdot[0.5, 1] = [-1, 0]$, so $f \downarrow$ and $f(\mathbf{x}'') = [f(1), f(0.5)] = [0, 0.25]$. The resulting estimate is $f(\mathbf{x}') \cup f(\mathbf{x}'') = [0, 0.25]$, which is the exact range.

These ideas underlie efficient interval computations algorithms and software packages[3,4,6,9].

5. Partial Information about Probabilities

Formulation of the problem. In the ideal case, we know the probability distributions. In this case, in principle, we can find the distribution for $y = f(x_1, \ldots, x_n)$ by using Monte-Carlo simulations.

In the previous section, we considered situations when we only know an interval of possible values. In practice, in addition to the intervals, we sometimes also have *partial* information about the probabilities. How can we take this information into account?

How to represent partial information about probabilities. In general, there are many ways to represent a probability distribution; it is desirable to select a representation which is the most appropriate for the corresponding practical problem. In most practical problems, the ultimate objective is to make decisions. According to decision theory, a decision maker should look for an alternative a that maximizes the expected utility $E_x[u(x, a)] \to \max\limits_a$.

When the utility function $u(x)$ is smooth, we can expand it in Taylor series $u(x) = u(x_0) + (x - x_0) \cdot u'(x_0) + \ldots$; this shows that, to estimate $E[u]$, we must know moments. In this case, partial information means that we only have interval bounds on moments. There are known algorithms for processing such bounds; see, e.g.,[10].

Another case is when we have a threshold-type utility function $u(x)$: e.g., for a chemical plant, drastic penalties start if the pollution level exceeds a certain threshold x_0. In this case, to find the expected utility, we need the know the values of the cdf $F(x) = P(\xi \le x)$. Partial information means that, for every x, we only have interval bounds $[\underline{F}(x), \overline{F}(x)]$ on the actual (unknown) cdf; such bounds are known as a *p-box.* There are also known algorithms for processing such boxes; see, e.g.,[2,10].

Example of processing p-boxes. Suppose that we know p-boxes $[\underline{F}_1(x_1), \overline{F}_1(x_1)]$ and $[\underline{F}_2(x_2), \overline{F}_2(x_2)]$ for quantities x_1 and x_2, we do not have any information about the relation between x_1 and x_2, and we want to find the p-box corresponding $\underline{F}(y), \overline{F}(y)]$ corresponding to $y = x_1 + x_2$.

It is known that for every two events A and B,

$$P(A \vee B) = P(A) + P(B) - P(A \,\&\, B) \le P(A) + P(B).$$

In particular, $P(\neg A \vee \neg B) \le P(\neg A) + P(\neg B)$. Here, $P(\neg A) = 1 - P(A)$, $P(\neg B) = 1 - P(B)$, and $P(\neg A \vee \neg B) = 1 - P(A \,\&\, B)$, thus, $1 - P(A \,\&\, B) \le (1 - P(A)) + (1 - P(B))$ and so, $P(A \,\&\, B) \ge P(A) + P(B) - 1$. We also know that $P(A \,\&\, B) \ge 0$, hence $P(A \,\&\, B) \ge \max(P(A) + P(B) - 1, 0)$. Let us use this inequality to get the desired bounds for $F(y)$.

If $\xi_1 \le x_1$ and $\xi_2 \le x_2$, then $\xi \stackrel{\text{def}}{=} \xi_1 + \xi_2 \le x_1 + x_2$. Thus, if $x_1 + x_2 = y$, then $F(y) = P(\xi \le y) \ge P(\xi_1 \le x_1 \,\&\, \xi_2 \le x_2)$. Due to the above inequality, $P(\xi_1 \le x_1 \,\&\, \xi_2 \le x_2) \ge P(\xi \le x_1) + P(\xi_2 \le x_2) - 1$. Here, $P(\xi_i \le x_i \ge \underline{F}_i(x_i)$, so $F(y) \ge \underline{F}_1(x_1) + \underline{F}_2(x_2) - 1$. Thus, as the desired lower bound $\underline{F}(y)$, we can take the largest of the corresponding right-hand sides: $\underline{F}(y) = \max\left(\max\limits_{x_1, x_2 : x_1 + x_2 = y} (\underline{F}_1(x_1) + \underline{F}_2(x_2) - 1), 0\right)$, i.e.,

$$\underline{F}(y) = \max\left(\max_{x_1} (\underline{F}_1(x_1) + \underline{F}_2(y - x_1) - 1), 0\right).$$

To find the upper bound for $F(y)$, let us find a similar lower bound for $1 - F(y) = P(\xi > y)$. If $x_1 + x_2 = y$, $\xi_1 > x_1$, and $\xi_2 > x_2$, then $\xi = \xi_1 + \xi_2 > y$. Here, $P(\xi_i > x_i) = 1 - P(\xi_i \le x_i) = 1 - F_i(x_i)$. Thus,

$$1 - F(y) = P(\xi > y) \ge P(\xi_1 > x_1 \,\&\, \xi_2 > x_2) \ge P(\xi_1 > x_1) + P(\xi_2 > x_2) - 1$$

$$= (1 - F_1(x_1)) + (1 - F_2(x_2)) - 1 = 1 - F_1(x_1) - F_2(x_2),$$

hence $F(y) \le F_1(x_1) + F_2(x_2)$. Since $F_i(x_i) \le \overline{F}_i(x_i)$, we have $F(y) \le \overline{F}_1(x_1) + \overline{F}_2(x_2)$. Thus, as the desired upper bound $\overline{F}(y)$, we can take the smallest of the corresponding right-hand sides:

$$\overline{F}(y) = \min\left(\min_{x_1, x_2 : x_1 + x_2 = y} (\overline{F}_1(x_1) + \overline{F}_2(x_2)), 1\right), \text{ i.e.,}$$

$$\overline{F}(y) = \min\left(\min_{x_1} (\overline{F}_1(x_1) + \overline{F}_2(y - x_1)), 1\right).$$

Similar formulas can be derived for other elementary operations.

How to represent p-boxes. Representing a p-box means representing two cdfs $\underline{F}(x)$ and $\overline{F}(x)$. For each cdf $F(x)$, to represent all its values with accuracy $\dfrac{1}{n}$, it is sufficient to store $n-1$ *quantiles* $x_1 < \ldots < x_{n-1}$, i.e., values x_i for which $F(x_i) = \dfrac{i}{n}$. These values divide the real line into segments $[x_i, x_{i+1}]$, where $x_0 \stackrel{\rm def}{=} -\infty$ and $x_{n+1} \stackrel{\rm def}{=} +\infty$.

Each real value x belongs to one of these segments $[x_i, x_{i+1}]$, in which case, due to monotonicity of $F(x)$, we have $F(x_i) = \dfrac{i}{n} \le F(x) \le \dfrac{i+1}{n} = F(x_{i+1})$, hence $\left|F(x) - \dfrac{i}{n}\right| \le \dfrac{1}{n}$.

Need to go beyond p-boxes. In many practical situations, we need to maintain the value within a certain interval: e.g., the air conditioning must maintain the temperature within certain bounds, a spaceship must land within a certain region, etc. In such cases, the utility drastically drops if we are outside the interval; thus, the expected utility is proportional to the probability $F(a,b) = P(\xi \in (a,b])$ to be within the corresponding interval $(a,b]$. In such situations, partial information about probabilities means that for a and b, we only know the interval $[\underline{F}(a,b), \overline{F}(a,b)]$ containing the actual (unknown) values $F(a,b)$.

When we know the exact cdf $F(x)$, then we can compute $F(a,b)$ as $F(a) - F(b)$. However, in case of partial information, it is not sufficient to only know the p-box. For example, let us assume that x is uniformly distributed on some interval of known width $\varepsilon > 0$, but we do not know on which. In this case, as one can easily see, for every x, $\underline{F}(x) = 0$ and $\overline{F}(x) = 1$ – irrespective on ε. On the other hand, for any interval $[a,b]$, we have $\overline{F}(a,b) = \min\left(\dfrac{b-a}{\varepsilon}, 1\right)$. This bound clearly depends on ε and thus, cannot be uniquely determined by the p-box values.

How to process this more general information. Good news is that we process this more general information similarly to how we process p-boxes.

Specifically, when $\xi_1 \in \mathbf{x}_1 = (\underline{x}_1, \overline{x}_1]$ and $\xi_2 \in \mathbf{x}_2 = (\underline{x}_2, \overline{x}_2]$, then $\xi = \xi_1 + \xi_2 \in \mathbf{x}_1 + \mathbf{x}_2 = (\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$. Thus, if $\mathbf{x}_1 + \mathbf{x}_2 \subseteq \mathbf{y} = [\underline{y}, \overline{y}]$, we have

$$F(\underline{y}, \overline{y}) \ge P(\xi_1 \in \mathbf{x}_1 \,\&\, \xi_2 \in \mathbf{x}_2) \ge P(\xi_1 \in \mathbf{x}_1) + P(\xi_2 \in \mathbf{x}_2) - 1 \ge$$

$$\underline{F}_1(\mathbf{x}_1) + \underline{F}_2(\mathbf{x}_2) = 1.$$

So, as the desired lower bound $\underline{F}(\underline{y}, \overline{y})$, we can take the largest of the corresponding right-hand sides:

$$\underline{F}(\underline{y}, \overline{y}) = \max\left(\max_{\mathbf{x}_1, \mathbf{x}_2 : \mathbf{x}_1 + \mathbf{x}_2 \subseteq \mathbf{y}} (\underline{F}_1(\mathbf{x}_1) + \underline{F}_2(\mathbf{x}_2) - 1), 0\right).$$

This formula is very similar to the formula for p-boxes. The formula for the upper bound comes from the fact that $F(\underline{y}, \overline{y}) = F(\overline{y}) - F(\underline{y})$, and thus, $F(\underline{y}, \overline{y}) \leq \overline{F}(\overline{y}) - \underline{F}(\underline{y})$. We already know the values $\overline{F}(\overline{y}) - \underline{F}(\underline{y})$, thus we can take their difference as the desired upper bound $\overline{F}(\underline{y}, \overline{y})$:

$$\overline{F}(\underline{y}, \overline{y}) = \min\left(\min_{x_1} (\overline{F}_1(x_1) + \overline{F}_2(\overline{y} - x_1)), 1\right) -$$

$$\max\left(\max_{x_1} (\underline{F}_1(x_1) + \underline{F}_2(y - x_1) - 1), 0\right).$$

Similar formulas can be obtained for other elementary operations.

How to represent this more general information. Not so good news is that representing such a more general information is much more difficult than representing p-boxes.

Indeed, similarly to p-boxes, we would like to represent all the values $\underline{F}(a, b)$ and $\overline{F}(a, b)$ with a given accuracy $\frac{1}{n}$, i.e., we would like to find the values $x_1 < \ldots < x_N$ for which $x_i \leq a \leq x_{i+1}$ and $x_j \leq b \leq x_{j+1}$ implies $|\underline{F}(a, b) - \underline{F}(x_i, x_j) \leq \frac{1}{n}$ and $|\overline{F}(a, b) - \overline{F}(x_i, x_j) \leq \frac{1}{n}$.

For p-boxes, we could use $N = n$ values x_i. Let us show that for the bounds on $P(a, b)$, there is no upper bound on the number of values needed. Namely, we will show that in the above example, when $\varepsilon \to 0$, the corresponding number of points N grows indefinitely: $N \to \infty$. Indeed, when $j = i$, $a = x_i$, and $b = x_{i+1}$, then, due to $\overline{F}(x_i, x_i) = 0$, the above condition means $\overline{F}(x_i, x_{i+1}) \leq \frac{1}{n}$. Thus, we must have $\frac{x_{i+1} - x_i}{\varepsilon} \leq \frac{1}{n}$, i.e., $x_{i+1} - x_i \leq \frac{\varepsilon}{n}$. The next point x_{i+1} is this close to the previous one, so, e.g., on the unit interval $[0, 1]$, we need at least $N \geq \frac{n}{\varepsilon}$ such points. When $\varepsilon \to 0$, the number of such points indeed tends to infinity.

It is worth mentioning that we *can* have an upper bound on N if we know an upper bound d on the probability density $\rho(x)$: in this case, $F(a, b) \leq (b - a) \cdot d$ and thus, to get the desired accuracy $\frac{1}{n}$, it is sufficient to have $x_{i+1} - x_i = \frac{1}{n \cdot d}$. On an interval of width W, we thus need

$N = Wx_{i+1} - x_i = W \cdot n \cdot d$ points.

Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721. The author is greatly thankful to Scott Ferson, to Franco Pavese, and to all the participants of the International Conference on Advanced Mathematical and Computational Tools in Metrology and Testing AMTCM'2014 (St. Petersburg, Russia, September 9–12, 2014) for valuable discussions.

References

1. Archimedes, On the measurement of the circle, In: T. L. Heath (ed.), *The Works of Archimedes* (Dover, New York, 1953).
2. S. Ferson et al., *Constructing Probability Boxes and Dempster-Shafer Structures* (Sandia Nat'l Labs, Report SAND2002-4015, 2003).
3. Interval computations website http://www.cs.utep.edu/interval-comp
4. L. Jaulin et al., *Applied Interval Analysis* (Springer, London, 2001).
5. E. T. Jaynes and G. L. Bretthorst, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, UK, 2003).
6. V. Kreinovich, Interval computations and interval-related statistical techniques, In: F. Pavese and A. B. Forbes (eds.), *Data Modeling for Metrology and Testing in Measurement Science* (Birkhauser-Springer, Boston, 2009), pp. 117–145.
7. V. Kreinovich and S. Ferson, A new Cauchy-Based black-box technique for uncertainty in risk analysis, *Reliability Engineering and Systems Safety* **85**(1–3), 267–279 (2004).
8. V. Kreinovich et al., *Computational Complexity and Feasibility of Data Processing and Interval Computations* (Kluwer, Dordrecht, 1997).
9. R. E. Moore, R. B. Kreinovich, and M. J. Cloud, *Introduction to Interval Analysis* (SIAM Press, Philadelphia, Pennsylvania, 2009).
10. H. T. Nguyen et al., *Computing Statistics under Interval and Fuzzy Uncertainty* (Springer, Berlin, Heidelberg, 2012).
11. S. G. Rabinovich, *Measurement Errors and Uncertainty:Theory and Practice* (Springer, Berlin, 2005).