

How Interval Measurement Uncertainty Affects the Results of Data Processing: A Calculus-Based Approach to Computing the Range of a Box

Andrew Pownuk and Vladik Kreinovich
Computational Science Program
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
ampownuk@utep.edu, vladik@utep.edu

Abstract

In many practical applications, we are interested in the values of the quantities y_1, \dots, y_m which are difficult (or even impossible) to measure directly. A natural idea to estimate these values is to find easier-to-measure related quantities x_1, \dots, x_n and to use the known relation to estimate the desired values y_j . Measurements come with uncertainty, and often, the only thing we know about the actual value of each auxiliary quantity x_i is that it belongs to the interval $[\underline{x}_i, \bar{x}_i] = [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$, where \tilde{x}_i is the measurement result, and Δ_i is the upper bound on the absolute value of the measurement error $\tilde{x}_i - x_i$. In such situations, instead of a single value of a tuple $y = (y_1, \dots, y_m)$, we have a range of possible values. In this paper, we provide calculus-based algorithms for computing this range.

1 Formulation of the Problem

Need for indirect measurements. In many practical situations, we are interested in the values of the quantities y_1, \dots, y_m which are difficult – or even impossible – to measure directly. Since we cannot measure these quantities directly, a natural idea is to measure them *indirectly* (see, e.g., [6]), i.e.:

- to measure related quantities x_1, \dots, x_n which are related to the desired quantities y_j by known relations, and
- to use appropriate algorithms to find the values of the desired quantities:

$$\begin{aligned}y_1 &= f_1(x_1, \dots, x_n); \\y_2 &= f_2(x_1, \dots, x_n); \end{aligned} \tag{1}$$

$$\dots$$

$$y_m = f_m(x_1, \dots, x_n).$$

Comment. In the real world, the relations are usually smooth; see, e.g., [1, 7].

Need to take into account measurement uncertainty. If we knew the exact values x_1, \dots, x_n of all the auxiliary quantities, then, by using the relations (1), we would be able to find the exact values of all the desired quantities y_1, \dots, y_m .

In practice, however, measurements are never absolutely precise. The measurement result \tilde{x}_i is, in general, different from the actual (unknown) values of the corresponding quantity. When we plug in values $\tilde{x}_i \neq x_i$ into the formula (1), we, in general, get the values $\tilde{y}_j = f_j(\tilde{x}_1, \dots, \tilde{x}_n)$ which are, in general, different from the desired values y_j . How can we gauge the resulting uncertainty in y_j ?

Case of interval measurement uncertainty. In many practical situations, the only information that we have about the measurement error $\Delta x_i \stackrel{\text{def}}{=} \tilde{x}_i - x_i$ is the upper bound Δ_i provided by the manufacturer of the corresponding measuring instrument. (If the manufacturer provide no such bound, then it is not a measuring instrument, it is a device for producing wild guesses.)

In this case, once we know the measurement result \tilde{x}_i , the only information we have about the actual value x_i is that it is somewhere on the interval $[\underline{x}_i, \bar{x}_i]$, where $\underline{x}_i \stackrel{\text{def}}{=} \tilde{x}_i - \Delta_i$ and $\bar{x}_i \stackrel{\text{def}}{=} \tilde{x}_i + \Delta_i$; see, e.g., [2, 4, 5, 6].

There is no a priori known relation between the values x_i , so the set of all possible values of x_i should not depend on the values of all other quantities x_j , $j \neq i$. Thus, the set of all possible values of the tuple $x = (x_1, \dots, x_n)$ is the box

$$[\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_n, \bar{x}_n]. \quad (2)$$

Resulting problem. Once we know that x belongs to the box (2), what are the possible values of the tuple $y = (y_1, \dots, y_m)$? In mathematical terms, what is the range of the box (2) under the mapping (1)?

In this paper, we describe calculus-based techniques for solving this problem.

2 Analysis of the Problem and the Resulting Algorithms

Simplest case when we have only one desired quantity y_1 : analysis of the problem. Let us start with the simplest case, when we have only one desired quantity y_1 . In this case, we are interested in the range of the function $f_1(x_1, \dots, x_n)$ when each x_i is in the corresponding interval $[\underline{x}_i, \bar{x}_i]$. For smooth (even for continuous) functions, this range is connected and is, thus, an interval $[\underline{y}_1, \bar{y}_1]$, where:

- \underline{y}_1 is the smallest possible value of the function $f_1(x_1, \dots, x_n)$ on the given box, and
- \bar{y}_1 is the largest possible value of the function $f_1(x_1, \dots, x_n)$ on the given box.

For each variable x_i , the maximum (or minimum) of the expression $y_1 = f_1(x_1, \dots, x_n)$ is attained:

- either at one of the endpoints of this interval, i.e., for $x_i = \underline{x}_i$ or $x_i = \bar{x}_i$,
- or inside the corresponding interval $(\underline{x}_i, \bar{x}_i)$.

According to calculus, if the maximum or minimum is attained inside an interval, then the corresponding derivative $\frac{\partial f_1}{\partial x_i}$ is equal to 0. So, for each i , it is sufficient to consider three possible cases:

- the case when $x_i = \underline{x}_i$;
- the case when $x_i = \bar{x}_i$, and
- the case when $\frac{\partial f_1}{\partial x_i} = 0$.

Thus, to find the minimum \underline{y}_1 and the maximum \bar{y}_1 of the function $y_1 = f_1(x_1, \dots, x_n)$ over the box, it is sufficient to consider all possible combinations of these 3 cases.

In other words, we arrive at the following algorithm.

Case when we have only one desired quantity y_1 : algorithm. Consider all systems of equations, in which, for each i , we have one of the three alternatives: $x_i = \underline{x}_i$, $x_i = \bar{x}_i$, and $\frac{\partial f_1}{\partial x_i} = 0$. There are 3^n such systems.

For each of these systems, we find the corresponding values $x = (x_1, \dots, x_n)$ and compute the corresponding value $y_1 = f_1(x_1, \dots, x_n)$. The largest of thus computed values is \bar{y}_1 , the smallest is \underline{y}_1 .

Comment. This algorithm requires solving an exponential number of systems and thus takes exponential time. This is, however, unavoidable, since it is known that already for quadratic functions $f_1(x_1, \dots, x_n)$, the problem of computing the bounds \underline{y} and \bar{y} is NP-hard; see, e.g., [3]. This means that, unless P=NP (which most computer scientists believe to be impossible), super-polynomial (e.g., exponential) computation time is unavoidable – at least for some inputs.

Exponential time does not mean that the algorithm is not practical – for reasonably small n , solving 3^n system is quite reasonable. For example, for $n = 10$, we need to solve less than 60,000 systems, it is a large number, but it is quite doable. For $n = 15$, we need to solve about 5 million systems – still possible.

What we plan to do next. In the following subsections, we show how we can extend this calculus-based approach to the general case, and thus reduce the

difficult-to-solve problem of finding the range to more well-studied problems of solving systems of equations.

Case when the number m of desired quantities is equal to the number n of auxiliary ones: analysis of the problem. To find the range means to find its border. At almost all points on the border, there is – locally – at least one tangent plane. A plane in an m -dimensional space has the form

$$\sum_{j=1}^m c_j \cdot y_j = c_0.$$

Thus, at this border point $y = (y_1, \dots, y_m)$, the linear expression

$$y = \sum_{j=1}^m c_j \cdot y_j = f(x_1, \dots, x_n) \stackrel{\text{def}}{=} \sum_{j=1}^m c_j \cdot f_j(x_1, \dots, x_n)$$

attains its local maximum or local minimum.

Similarly to the previous case, this may mean that one of the inputs x_i attains its largest possible value \bar{x}_i or its smallest possible value $x_i = \underline{x}_i$. In this case, the corresponding condition $x_i = \underline{x}_i$ or $x_i = \bar{x}_i$ determines the $(n - 1)$ -dimensional set – which could be part of the border.

It may also mean that the maximum or minimum of the linear function is attained when all the values x_i are inside the corresponding intervals. In this case, we get

$$\frac{\partial f}{\partial x_i} = 0$$

for all i , i.e., we get

$$\sum_{j=1}^m c_j \cdot \frac{\partial f_j}{\partial x_i} = 0$$

for all i .

In algebraic terms, the existence of non-always-zero values c_j that satisfy the above equality for all i means that $m = n$ gradient vectors

$$\left(\frac{\partial f_j}{\partial x_1}, \dots, \frac{\partial f_j}{\partial x_n} \right)$$

that correspond to different j are linearly dependent. According to linear algebra, this is equivalent to requiring that the determinant of the Jacobian matrix

$\left\| \frac{\partial f_j}{\partial x_i} \right\|$ is equal to 0:

$$\det \left\| \frac{\partial f_j}{\partial x_i} \right\| = 0. \tag{3}$$

So, we arrive at the following algorithm.

Case when the number m of desired quantities is equal to the number n of auxiliary ones: algorithm. To find the border of the desired range, for each i from 1 to $m = n$, we form two systems of equations:

- the system (1) in which we substitute $x_i = \underline{x}_i$, and
- the system (1) in which we substitute $x_i = \bar{x}_i$.

Each of these systems provides a set of co-dimension 1 that could potentially serve as part of the border of the desired set.

To these possible border sets, we add the set corresponding to the equation (3). This equation defined a set of co-dimension 1, and plugging this set into (1), we can a y -set of co-dimension one – which can also be part of the border.

We know that the actual border can contain only segments of the above type, so once we have computed all these segments, we can reconstruct the border.

General case: analysis of the problem. We have already considered the case when $m = n$. There are two remaining cases: when $n < m$ and when $m < n$.

When $n < m$, the set of all possible values of the tuple y is of smaller dimension than the m , so this set is its own boundary.

Let us now consider the case when $m < n$. In this case, also, some linear combination

$$f(x_1, \dots, x_n) = \sum_{j=1}^m c_j \cdot f_j(x_1, \dots, x_n)$$

attains its maximum or its minimum. Let v denote the number of inputs x_i for which at this maximum-or-minimum point, we have $x_i = \underline{x}_i$ or $x_i = \bar{x}_i$. For each of the remaining $n - v$ variables x_i , we then have the equation

$$\sum_{j=1}^m c_j \cdot \frac{\partial f_j}{\partial x_i} = 0. \quad (4)$$

This equality (4) must hold for all $(n - v)$ values of i , so we must have $(n - v)$ equations.

We can select one of the values c_j equal to 1, then the other $m - 1$ values of c_j can be determined if we consider the first $m - 1$ conditions (4) as a system of linear equations with $m - 1$ unknowns. Substituting these values for c_j into the remaining $n - v - (m - 1)$ equalities (4), we thus get $n - v - (m - 1)$ equalities that relate $n - v$ unknowns.

In general, each additional equality imposed on elements of a set decreases its dimension by 1. For example, in the 3-D space:

- the set of all the points that satisfy a certain equality is usually a 2-D surface,
- the set of points that satisfy two independent equalities in a 3-D space is a 1-D line, etc.

In our case, the dimension of the set of all the $(n - v)$ -dimensional tuples x that satisfy all $n - v - (m - 1)$ equalities is equal to the difference

$$(n - v) - (n - v - (m - 1)) = m - 1.$$

The image of this $(m - 1)$ -dimensional set under the transformation (1) is also $(m - 1)$ -dimensional, so it forms a surface in the m -dimensional space of all possible tuples $y = (y_1, \dots, y_m)$.

As a result, we get the following algorithm.

General case: algorithm. We consider all possible subsets I of the set $\{1, \dots, n\}$ of all indices of the inputs x_i . For each such subset I of size v , we consider all 2^v possible combinations of values \underline{x}_i and \bar{x}_i .

For each such combination, we consider the system of equations (4) for all $i \notin I$. We can set up one of the values c_j to 1 and the first $m - 1$ equations (4) to describe c_j as a function of x_1, \dots, x_m . Substituting the resulting expressions for c_j in terms of x_i into the remaining $n - v - (m - 1)$ equalities (4), we get a $(m - 1)$ -dimensional set of tuples x . Substituting this set of tuples into the formula (1), we get a $(m - 1)$ -dimensional set of y -tuples.

We thus get several $(m - 1)$ -dimensional sets, and we know that the actual bored can only consist of the above fragments.

Acknowledgments

This work was supported in part by the National Science Foundation grant HRD-1242122 (Cyber-ShARE Center of Excellence).

The authors are thankful to all the participants of the 2017 UTEP/NMSU Workshop on Mathematics, Computer Science, and Computation Science (El Paso, Texas, November 4, 2017) for valuable discussions.

References

- [1] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
- [2] L. Jaulin, M. Kiefer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics*, Springer, London, 2001.
- [3] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.
- [4] G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.
- [5] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.
- [6] S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, Berlin, 2005.

- [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, 2017